# CONIPUTVR-AIDOD ANALISSS ${ }^{0}$ MDCHANICAL SYSTVNUS 

Parviz E.Nikravesh


## NOMENCLATURE

Matrices are in boldface upper-case characters.
Column matrices, algebraic vectors, and arrays are in boldface lower-case characters.
Scalars are in lightface characters.

| $\mathbf{a}$ | Column vector (array) |
| :--- | :--- |
| $\mathbf{a}^{T}$ | Row vector (array) |
| $\mathbf{A}$ | Matrix |
| $a_{i j}$ | Element of matrix A in $i$ ith row and $j$ th column |
| 0 | Zero vector |
| $\mathbf{0}$ | Zcro (null) matrix |

## OVERSCORES

Geometric vector
$3 \times 3$ skew-symmetric matrix

- $\quad 4 \times 4$ skew-symmetric matrix containing a negative $3 \times 3$ skew-symmetric matrix
$+\quad 4 \times 4$ skew-symmetric matrix containing a positive $3 \times 3$ skew-symmetric matrix
First derivative with respect to time
Second derivative with respect to time


## SUPERSCRIPTS

-1 Matrix inverse
$i$ th time step
Matrix or vector transpose
Type of constraint or force
Components of a vector in a body-fixed coordinate system
Components of a vector or matrix in Euler-parameter space

## SUBSCRIPTS

$i$ th body in a system
Projection of a vector along a known axis

| $\gamma$ | Vector of right-hand side of acceleration equations |
| :---: | :---: |
| $\theta$ | Angle between two vectors |
| $\lambda$ | Vector of Lagrange multipliers |
| $\mu_{i}$ | Polar moment of inertia for body $i$ |
| $\xi_{i} \eta_{i} \zeta_{i}$ | Local (body-fixed) Cartesian coordinate system |
| $\rho$ | Radius of a circle |
| $\sigma_{i}$ | Lagrange multiplier associated with the constraint on $\mathbf{p}_{i}$ |
| $\phi$ | Angle of rotation |
| $\phi_{1}, \phi_{2}, \phi_{3}$ | Bryant angles |
| $\psi, \theta, \sigma$ | Euler angles |
| $\vec{\omega}_{i}$ | Angular velocity vector for body $i$ |
| $\omega_{i}$ | Global components of $\vec{\omega}_{i}$ |
| $\omega_{i}^{\prime}$ | Local components of $\vec{\omega}_{i}$ |
| $\Phi ; \boldsymbol{\Phi}$ | One constraint; vector of constraints |
| $\Phi_{4}$ | Jacobian matrix of constraints |
| $b$ | Number of bodies |
| $\mathbf{b}_{i}$ | Vector containing quadratic velocity terms for body $i$ |
| b | Vector of quadratic velocity terms |
| $\vec{d}$ | Vector with its ends on two different bodies |
| d | Global components of $\vec{d}$ |
| $e_{0}, e_{1}, e_{2}, e_{3}$ | Euler parameters |
| $\mathrm{e}_{i}$ | Vector of three Euler parameters $e_{1}, e_{2}, e_{3}$ for body $i$ |
| $\vec{f}_{i}$ | Force acting on body $i$ |
| $\mathbf{f}_{i}$ | Global components of $\vec{f}_{i}$ |
| $\mathrm{g}_{i}$ | Vcctor of forces for body $i$ containing $\mathbf{f}_{i}$ and $\mathbf{n}_{i}^{\prime}$ |
| g | Vector of forces for a system |
| $\mathrm{g}^{(c)}$ | Vector of constraint reaction forces |
| $\mathbf{h}_{i}$ | Velocity vector for body $i$ containing $\dot{\mathbf{r}}_{i}$ and $\boldsymbol{\omega}_{i}^{\prime}$ |
| h | Vector of velocities for a system |
| $k$ | Number of degrees of freedom (DOF) |
| l | Vector with its ends on two different bodies |
| $l$ | Global components of $\vec{l}$ |
| $m$ | Number of constraint equations |

$m^{(p)} \quad$ Mass of a particle
$m_{i} \quad$ Mass of body $i$
$n \quad$ Number of coordinates
$\vec{n}_{i} \quad$ Moment acting on body $i$
$\mathbf{n}_{i} \quad$ Global components of $\vec{n}_{i}$
$\mathbf{n}_{i}^{\prime} \quad$ Local components of $\vec{n}_{i}$
$\mathbf{n}_{i}^{*} \quad$ Components of $\vec{n}_{i}$ in four-dimensional space
$\mathbf{p}_{i} \quad$ Vector of four Euler parameters $e_{0}, e_{1}, e_{2}, e_{3}$ for body $i$
$\mathbf{q}_{i} \quad$ Vector of coordinates for body $i$
q Vector of coordinates for a system
$\vec{r}_{i} \quad$ Translational position vector for body $i$
$\mathbf{r}_{i} \quad$ Global coordinates of $\vec{r}_{i}$
$\vec{s}_{i} \quad$ Vector with both ends on body $i$ (constant magnitude)
$\mathbf{s}_{i} \quad$ Global components of $\vec{s}_{i}$
$\mathbf{s}_{i}^{\prime} \quad$ Local components of $\vec{s}_{i}$
$t$ Time
$t^{0} \quad$ Initial time
$t^{e} \quad$ Final (end) time
$\vec{u} \quad$ Unit vector
$u \quad$ Global components of $\vec{u}$; vector of dependent coordinates
v Vector of independent coordinates
$x y z \quad$ Global Cartesian coordinate system
$\mathbf{y} \quad$ Vector of integration variables
$\mathbf{A}_{i} \quad$ Rotational transformation matrix for body $i$
$\mathbf{G}_{i} \quad 3 \times 4$ transformation matrix for body $i$
I $3 \times 3$ or general identity matrix
I* $4 \times 4$ identity matrix
$\mathbf{J}_{i} \quad$ Global inertia tensor for body $i$
$\mathbf{J}_{i}^{\prime} \quad$ Local (constant) inertia tensor for body $i$
$\mathbf{J}^{*} \quad 4 \times 4$ inertia tensor
L Lower triangular matrix
$\mathbf{L}_{i} \quad 3 \times 4$ transformation matrix for body $i$
$\mathbf{M}_{i} \quad 6 \times 6$ mass matrix for body $i$ containing $\mathbf{N}_{i}$ and $\mathbf{J}_{i}{ }^{\prime}$
M Mass matrix for a system
$\mathbf{N}_{i} \quad 3 \times 3$ diagonal mass matrix for body $i$
U Upper triangular matrix
$V \quad$ Potential energy

# Computer-Aided Analysis of <br> Mechanical Systems 

# PARVIZ E. NIKRAVESH 

Aerospace and Mechanical
Engineering Department
University of Arizona

Nikravesh．
p．cm．
Bibliography：p．
Includes index ISBN 0－13－164220－0
1．Machinery，Kinematics of－－Data processing．2．Machinery， Dynamics of－Data processing．I．Titc．
TJ175．N52 1987 87－22908
621 ． 11 －dcle CIP

Editorial／production supervision and interior design：Elena Le Pera
Cover design：Photo Plus Art
Manufacturing buyer：Cindy Grant

## To the memory of my sister，Henriette．



A Division of Simon \＆Schuster
Englewood Cliffs，New Jersey 07632

All rights reserved．No part of this book may be reproduced，in any form or by any means， without permission in writing from the publisher．

Printed in the United States of America
$\begin{array}{llllllllll}10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1\end{array}$

ISBN ローコヨーコロ4ここローロ 025

Prentice－Hall International（UK）Limited，London
Prentice－Hall of Australia Pty．Limited，Sydney
Prentice－Hall Canada Inc．，Toronto
Prentice－Hall Hispanoamericana，S．A．，Mexico
Prentice－Hall of India Private Limited，New Delhi
Prentice－Hall of Japan，Inc．，Tokyo
Simon \＆Schuster Asia Pte．Ltd．，Singapore
Editora Prentice－Hall do Brasil，Ltda．，Rio de Janeiro

## Contents

Preface ..... ix
Note on Unit System ..... xiii
1 INTRODUCTION ..... 1
1.1 Computers in Design and Manufacturing ..... 1
1.1.1 Computer-Aided Analysis ..... 2
1.2 Multibody Mechanical Systems ..... 3
1.3 Branches of Mechanics ..... 6
1.3.1 Methods of Analysis ..... 6
1.4 Computational Methods ..... 9
1.4.1 Efficiency versus Simplicity ..... 10
1.4.2 A General-Purpose Program ..... 14
2 VECTORS AND MATRICES ..... 19
2.1 Geometric Vectors ..... 19
2.2 Matrix and Algebraic Vectors ..... 21
2.2.1 Matrix Operations ..... 21
2.2.2 Algebraic Vector Operations ..... 24
2.3 Vector and Matrix Differentiation ..... 28
2.3.1 Time Derivatives ..... 28
2.3.2 Partial Derivatives ..... 29
Problems ..... 33
3 BASIC CONCEPTS AND NUMERICAL METHODS in Kinematics ..... 35
3.1 Definitions ..... 35
3.1.1 Classification of Kinematic Pairs ..... 37
3.1.2 Vector of Coordinates ..... 38
3.1.3 Degrees of Freedom ..... 40
3.1.4 Constraint Equations ..... 41
3.I.5 Redundant Constraints ..... 41
3.2 Kinematic Analysis ..... 42
3.2.1 Coordinate Partitioning Method ..... 43
3.2.2 Method of Appended Driving Constraints ..... 48
3.3 Linear Algebraic Equations ..... 50
3.3.1 Gaussian Methods ..... 51
3.3.2 Pivoting ..... 53
3.3.3 L-U Factorization ..... 56
3.3.4 L-U Factorization with Pivoting ..... 61
3.3.5 Subroutines for Linear Algebraic Equations ..... 63
3.4 Nonlinear Algebraic Equations ..... 66
3.4.1 Newton-Raphson Method for One Equation in One Unknown ..... 66
3.4.2 Newton-Raphson Method for n Equations in n Unknowns ..... 67
3.4.3 A Subroutine for Nonlinear Algebraic Equations ..... 70
Problems ..... 72
4 PLANAR KINEMATICS ..... 77
4.1 Cartesian Coordinates ..... 77
4.2 Kinematic Constraints ..... 80
4.2.1 Revolute and Translational Joints (LP) ..... 81
4.2.2 Composite Joints (LP) ..... 84
4.2.3 Spur Gears and Rack and Pinion (HP) ..... 86
4.2.4 Curve Representation ..... 89
4.2.5 Cam-Followers (HP) ..... 93
4.2.6 Point-Follower (HP) ..... 97
4.2.7 Simplified Constraints ..... 98
4.2.8 Driving Links ..... 100
4.3 Position, Velocity, and Acceleration Analysis ..... 101
4.3.1 Systematic Generation of Some Basic Elements ..... 103
4.4 Kinematic Modeling ..... 105
4.4.1 Slider-Crank Mechanism ..... 105
4.4.2 Quick-Return Mechanism ..... 110
Problems ..... 115
5 A FORTRAN PROGRAM FOR ANALYSIS OF PLANAR KINEMATICS ..... 119
5.1 Kinematic Analysis Program (KAP) ..... 119
5.1.1 Model-Description Subroutines ..... 123
5.1.2 Kinematic Analysis ..... 127
5.1.3 Function Evaluation ..... 130
5.1.4 Input Prompts ..... 134
5.2 Simple Examples ..... 134
5.2.1 Four-Bar Linkage ..... 135
5.2.2 Slider-Crank Mechanism ..... 137
5.2.3 Quick-Return Mechanism ..... 139
5.3 Program Expansion ..... 140
Problems ..... 140
6 EULER PARAMETERS ..... 153
6.1 Coordinates of A Body ..... 153
6.1.1 Euler's Theorem on the Motion of a Body ..... 157
6.1.2 Active and Passive Points of View ..... 157
6.1.3 Euler Parameters ..... 158
6.1.4 Determination of Euler Parameters ..... 160
6.1.5 Determination of the Direction Cosines ..... 164
6.2 Identities with Euler Parameters ..... 166
6.2.1 Identities with Arbitrary Vectors ..... 170
6.3 The Concept of Angular Velocity ..... 172
6.3.1 Time Derivatives of Euler Parameters ..... 174
6.4 Semirotating Coordinate Systems ..... 176
6.5 Relative Axis of Rotation ..... 177
6.5.1 Intermediate Axis of Rotation ..... 180
6.6 Finite Rotation ..... 180
Problems ..... 181
7 SPATIAL KINEMATICS ..... 186
7.1 Relative Constraints between Two Vectors ..... 186
7.1.1 Two Perpendicular Vectors ..... 188
7.1.2 Two Parallel Vectors ..... 188
7.2 Relative Constraints between Two Bodies ..... 189
7.2.1 Spherical, Universal, and Revolute Joints (LP) ..... 190
7.2.2 Cylindrical, Translational, and Screw Joints(LP) 192
7.2.3 Composite Joints ..... 196
7.2.4 Simplified Constraints ..... 199
7.3 Position, Velocity, and Acceleration Analysis ..... 200
7.3.1 Modified Jacobian Matrix and Modified Vector $\boldsymbol{\gamma}$ ..... 201
Problems ..... 204
8 BASIC CONCEPTS IN DYNAMICS ..... 208
8.1 Dynamics of a Particle ..... 208
8.2 Dynamics of a System of Particles ..... 209
8.3 Dynamics of a Body ..... 211
8.3.1 Moments and Couples ..... 212
8.3.2 Rotational Equations of Motion ..... 215
8.3.3 The Inertia Tensor ..... 217
8.3.4 An Unconstrained Body ..... 219
8.4 Dynamics of a System of Bodies ..... 221
8.4.1 A System of Unconstrained Bodies ..... 221
8.4.2 A System of Constrained Bodies ..... 222
8.4.3 Constraint Reaction Forces ..... 223
8.5 Conditions for Planar Motion ..... 224
9 PLANAR DYNAMICS ..... 227
9.1 Equations of Motion ..... 227
9.2 Vector of Forces ..... 229
9.2.1 Gravitational Force ..... 229
9.2.2 Single Force or Moment ..... 229
9.2.3 Translational Actuators ..... 231
9.2.4 Translational Springs ..... 232
9.2.5 Translational Dampers ..... 234
9.2.6 Rotational Springs ..... 236
9.2.7 Rotational Dampers ..... 237
9.3 Constraint Reaction Forces ..... 237
9.3.1 Revolute Joint ..... 237
9.3.2 Revolute-Revolute Joint ..... 240
9.3.3 Translational Joint ..... 242
9.4 System of Planar Equations of Motion ..... 242
9.5 Static Forces ..... 244
9.6 Static Balance Forces ..... 245
9.7 Kinetostatic Analysis ..... 247
Problems ..... 248
10 A FORTRAN PROGRAM FOR ANALYSIS OF PLANAR DYNAMICS ..... 253
10.1 Solving the Equations of Motion ..... 253
10.2 Dynamic Analysis Program (DAP) ..... 254
10.2.1 Model-Description Subroutines ..... 258
10.2.2 Dynamic Analysis ..... 260
10.2.3 Function Evaluation ..... 263
10.2.4 Force Evaluation ..... 263
10.2.5 Reporting ..... 265
10.2.6 Static Analysis ..... 266
10.2.7 Input Prompts ..... 267
10.3 Simple Examples ..... 268
10.3.1 Four-Bar Linkage ..... 268
10.3.2 Horizontal Platform ..... 269
10.3.3 Dump Truck ..... 273
10.4 Time Step Selection ..... 277
Problems ..... 281
11 SPATIAL DYNAMICS ..... 289
11.1 Vector of Forces ..... 289
11.1.1 Conversion of Moments ..... 289
11.2 Equations of Motion for an Unconstrained Body ..... 291
11.3 Equations of Motion for a Constrained Body ..... 292
11.4 System of Equations ..... 293
11.4.1 Unconstrained Bodies ..... 294
11.4.2 Constrained Bodies ..... 296
11.5 Conversion of Kinematic Equations ..... 297
Problems ..... 299
12 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS ..... 301
12.1 Initial-Value Problems ..... 301
12.2 Taylor Series Algorithms ..... 302
12.2.1 Runge-Kutta Algorithms ..... 303
12.2.2 A Subroutine for a Runge-Kutta Algorithm ..... 304
12.3 Polynomial Approximation ..... 307
12.3.1 Explicit Multistep Algorithms ..... 308
12.3.2 Implicit Multistep Algorithms ..... 308
12.3.3 Predictor-Corrector Algorithms ..... 309
12.3.4 Methods for Starting Multistep Algorithms ..... 309
12.4 Algorithms for Stiff Systems ..... 310
12.5 Algorithms for Variable Order and Step Size ..... 311
Problems ..... 311
13 NUMERICAL METHODS IN DYNAMICS ..... 313
13.1 Integration Arrays ..... 313
13.2 Kinematically Unconstrained Systems ..... 314
13.2.1 Mathematical Constraints ..... 315
13.2.2 Using Angular Velocities ..... 317
13.3 Kinematically Constrained Systems ..... 318
13.3.1 Constraint Violation Stabilization Method ..... 319
13.3.2 Coordinate Partitioning Method ..... 321
13.3.3 Automatic Partitioning of the Coordinates ..... 324
13.3.4 Stiff Differential Equation Method ..... 327
13.4 Joint Coordinate Method ..... 330
13.4.1 Open-Chain Systems ..... 331
13.4.2 Closed-Loop Systems ..... 334
Problems ..... 335
14 STATIC EQUILIBRIUM ANALYSIS ..... 339
14.1 An Iterative Method ..... 339
14.1.1 Coordinate Partitioning ..... 340
14.2 Potential Energy Function ..... 341
14.2.1 Minimization of Potential Energy ..... 342
14.3 Fictitious Damping Method ..... 344
14.4 Joint Coordinates Method ..... 345
Appendix A. EULER ANGLES AND BRYANT ANGLES ..... 347
A. 1 Euler Angles ..... 347
A.1.1 Time Derivatives of Euler Angles ..... 349
A. 2 Bryant Angles ..... 351
A.2.1 Time Derivatives of Bryant Angles ..... 352
Appendix B. RELATIONSHIP BETWEEN EULER PARAMETERS AND EULER ANGLES ..... 353
B. 1 Euler Parameters in Terms of Euler Angles ..... 353
B. 2 Euler Angles in Terms of Euler Parameters ..... 354
Appendix C. COORDINATE PARTITIONING WITH L-U FACTORIZATION ..... 355
REFERENCES ..... 357
BIBLIOGRAPHY ..... 359
INDEX ..... 363

## Preface

This book is designed to introduce fundamental theories and numerical methods for use in computational mechanics. These theories and methods can be used to develop computer programs for analyzing the response of simple and complex mechanical systems. In such programs the equations of motion are formulated systematically, and then solved numerically. Because they are relatively easy to use, the book focuses on Cartesian coordinates for formulating the equations of motion. After the reader has become familiar with this method of formulation, it can serve as a stepping stone to formulating the equations of motion in other sets of coordinates. The numerical algorithms that are discussed in this book can be applied to the equations of motion when formulated in any coordinate system.

## Organization of the Book

The text is organized in such a way that it can be used for teaching or for selfstudy. The concepts and numerical methods used in kinematics are systematically treated before the concepts and numerical methods used in dynamics are introduced. Separate chapters on each of these topics allow the text to be used for the study of each topic separately or for some desired combination of topics. Furthermore, the text first treats the less complex problems of planar kinematic and dynamic analysis before it discusses spatial kinematic and dynamic analysis.

With the exception of the first two chapters and the last chapter, the text can be divided into two subjects - kinematics and dynamics. Chapter 1 gives an introduction to the subject of computational methods in kinematics and dynamics. Simple examples illustrate how a problem can be formulated using different coordinate systems. Chapter 1 also explains why Cartesian coordinates provide a simple tool, if not necessarily the most computationally efficient one. Chapter 2 presents a review of vector and matrix
algebra, with an emphasis on the kind of formulation that lends itself to implementation in computer programs.

Chapters 3 through 7 deal with kinematics. Chapter 3 introduces the basic concepts in kinematics that are applicable to both planar and spatial systems. Algebraic constraint equations, the various coordinate systems, and the idea of degrees of freedom are presented as a foundation for both the analytical and the numerical aspects of kinematic analysis. Position, velocity, and acceleration analysis techniques are presented and illustrated through the solution of simple mechanisms. Numerical methods for solving the associated kinematic equations are presented and illustrated. These include methods for solving sets of linear and nonlinear algebraic equations. A comprehensive treatment of planar kinematics using Cartesian coordinates is presented in Chapter 4. In that chapter, a library of kinematic constraints is defined and the governing algebraic constraint equations are derived.

Chapter 5 contains a FORTRAN program for planar kinematic analysis. The program is developed and explained as a collection of subroutines that carry out the functions of kinematic analysis. The problems at the end of Chapter 5 provide guidelines for the extensions that allow for the expansion of the program to treat broader classes of planar kinematic systems.

Chapter 6 presents a set of spatial rotational coordinates known as Euler parameters. The physical properties of Euler parameters and the development of their algebraic properties are introduced to allow the reader to become comfortable with and confident in their use. Also, velocity relationships - including the definition of angular velocity - and other identities are developed that are necessary for the formulation of spatial kinematic and dynamic analysis.

Chapter 7 presents a unified formulation of spatial kinematics using Cartesian coordinates and Euler parameters. Vector relationships that are required for the definition of kinematic joints are first presented and then applied to derive the governing equations for a library of spatial kinematic joints. Although this book does not provide a source listing for a spatial kinematic analysis program, the computer program in Chapter 5 and the constraint formulations in Chapter 7 provide all the information that the reader needs to develop a computer program.

Chapters 8 through 13 deal with dynamics. Basic concepts in dynamics are presented in Chapter 8. Discussion begins with familiar concepts of the dynamics of a particle and progresses to the dynamics of systems of particles and, finally, to the dynamics of rigid bodies. By means of a building block formulation, the complete theory of the dynamics of systems of rigid bodies is developed in a systematic and understandable way. The Newton-Euler equations of motion are derived and used as a fundamental tool in the dynamic analysis of systems of rigid bodies that are connected by kinematic joints. The Lagrange multiplier formulation for constrained systems is developed, and the reaction forces between the joints are derived in terms of the Lagrange multipliers.

Chapter 9 discusses the planar dynamics of systems of constrained rigid bodies, drawing upon the kinematics theory discussed in Chapter 4 and the basic dynamics theory discussed in Chapter 8. Even though the numerical methods for solving the differential equations of motion are discussed in detail in Chapters 12 and 13, a FORTRAN program for planar dynamic analysis is presented in Chapter 10. This program, which is
a collection of subroutines used to implement a variety of computations required in the formulation and solution of equations of motion, builds upon the kinematic analysis program in Chapter 5. The computer program is demonstrated through the solution of simple examples, and extensions to the program are included as problems at the end of the chapter.

Chapter 11 presents the formulation of spatial system dynamics using Cartesian coordinates and Euler parameters. The equations of motion of kinematically constrained systems of rigid bodies are derived and developed in a form suitable for computational implementation. Chapter 12 presents a brief overview of numerical methods for solving ordinary differential equations. A FORTRAN listing of a fourth-order Runge-Kutta algorithm illustrates the implementation of these numerical methods along with some examples. Chapter 13 presents a number of advanced numerical methods for multibody dynamics. Alternate techniques and algorithms for the solution of mixed systems of differential and algebraic equations that arise in system dynamics are presented.

In the analysis of multibody mechanical systems, it may be necessary to go beyond kinematics and dynamics and find the static equilibrium state of a system. Chapter 14 discusses several computation-based methods for static equilibrium analysis.

## Level of Courses

The book can be covered in two successive courses. The student is required to know the fundamentals of kinematics and dynamics, to have a basic knowledge of numerical methods, and to know computer programming, preferably FORTRAN.

The first course - a senior undergraduate or a first-year graduate course - could cover Chapters 1 through 5, 9 , and 10 , on planar motion; if students do not have the proper background in numerical methods in ordinary differential equations, Chapter 12 should also be covered to the extent necessary. The course could be project-oriented: students could be assigned to find existing medium- to large-scale mechanical systems and analyze them using the computer programs that are provided in the book. The second course would then cover Chapters 6 through 8 and 11 through 14, on spatial motion; this would be quite suitable as a graduate-level course. Students, divided into groups, should be able to develop a spatial-motion dynamic analysis program.

Another possibility would be one course, covering Chapters 1 through 7, on the subject of kinematics, and a second course, covering Chapters 8 through 14, on the subject of dynamics.

## Exercises

Problem assignments can be found at the end of most chapters. The problems are designed to clarify certain points and to provide ideas for program development and analysis techniques. However, by no means do these problems represent the ultimate flexibility and power of the formulations and algorithms that are stated in the book. Most realistic multibody problems that arise in engineering practice can be treated by employing similar techniques and ideas.

## Computer Programs

Two FORTRAN programs called KAP and DAP, for planar kinematic and dynamic analysis, respectively, are developed and listed in the book. Other programs, for static equilibrium analysis, or for spatial kinematic and dynamic analysis, can be developed by the reader by following the formulations and algorithms that are discussed in various chapters. Source codes for KAP, DAP, and other complementary programs can be obtained on a floppy disk from the publisher.

## ACKNOWLEDGMENTS

I began working on the first version of this manuscript in 1980 at the University of Iowa while I was teaching two newly developed courses on this subject. Some of the material that was taught in these courses grew out of research collaboration with two of my colleagues, Dr. Edward J. Haug and Dr. Roger A. Wehage. I would like to express my appreciation to Ed for his encouragement and comments on the earlier versions of the manuscript. I am deeply grateful to Roger for many stimulating discussions over the years. Without his initiative, creativity, and support this book would not have been possible.

I would like to thank all of the graduate students who assisted me in many ways. They checked many of the formulas, found many errors, and generated many ideas. In particular, I would like to express my thanks to Mr. Hamid M. Lankarani for his assistance during the past two years and to Mr. Jorge A. C. Ambrosio for his effort in generating the computer graphic images.

Finally, I would like to thank my wife, Agnes, for her support and patience, and for not getting upset about the many evenings, weekends, and holidays that I spent working on this book.

## NOTE ON UNIT SYSTEM

The system of units adopted in this book is, unless otherwise stated, the international system of units (SI). In most examples and problems, the variables are organized as the elements of arrays suitable for programming purposes. These variables usually represent various different quantities and therefore have different units. If the unit of each element of an array were to be stated, it would cause notational confusion. Therefore, in order to eliminate this problem, the units of the variables are not stated in most parts of the text. The reader must assign the correct unit to each variable. The unit of degree or radian alone is stated for variables representing angular quantities.

SI Units Used in This Book

| Quantity | Unit SI | Symbol |
| :--- | :--- | :--- |
| (Base Units) |  |  |
| Length | meter | m |
| Mass | kilogram | kg |
| Time | second | s |
|  |  |  |
| (Derived Units) |  |  |
| Acceleration, translational | meter $/ \mathrm{second}^{2}$ | $\mathrm{~m} / \mathrm{s}^{2}$ |
| Acceleration, angular | radian ${ }^{\dagger} / \mathrm{second}^{2}$ | $\mathrm{rad} / \mathrm{s}^{2}$ |
| Damping coefficient | newton-second $/ \mathrm{meter}$ | $\mathrm{N} . \mathrm{s} / \mathrm{m}$ |
| Force | newton | $\mathrm{N}\left(=\mathrm{kg} \cdot \mathrm{m} / \mathrm{s}^{2}\right)$ |
| Moment of force | newton-meter | $\mathrm{N} . \mathrm{m}$ |
| Moment of Inertia, mass | kilogram-meter ${ }^{2}$ | $\mathrm{~kg} \cdot \mathrm{~m}^{2}$ |
| Pressure | pascal | $\mathrm{Pa}\left(=\mathrm{N} / \mathrm{m}^{2}\right)$ |
| Spring constant | newton $/ \mathrm{meter}^{2}$ | $\mathrm{~N} / \mathrm{m}$ |
| Velocity, translational | meter $/ \mathrm{second}$ | $\mathrm{m} / \mathrm{s}$ |
| Velocity, angular | radian ${ }^{\dagger} /$ second | $\mathrm{rad} / \mathrm{s}$ |

[^0]
## 1

## Introduction

The major goal of the engineering profession is to design and manufacture marketable products of high quality. Today's industries are utilizing computers in every phase of the design, management, manufacture, and storage of their products. The process of design and manufacture, beginning with an idea and ending with a final product, is a closed-loop process. Almost every link in the loop can benefit from the power of digital computers.

### 1.1 COIMPUTERS IN DESIGN AND MANUFACTURING

Factory automation is one of the major objectives of modern industry. Although there is no one plan for factory automation, a general configuration is presented in Fig. 1.1. In this configuration, all branches of the factory communicate and exchange information through a central data base. Various parts of the product are designed in the computeraided engineering (CAE) branch, and then the design is sent to the computer-aided manufacturing (CAM) branch for parts manufacturing and final assembly. Two of the major subbranches of CAE are computer-aided product design and computer-aided manufacturing design.

The computer-aided product design branch, better known as computer-aided design (CAD), ${ }^{\dagger}$ may consider the design of single parts or it may concern itself with the final product as an assembly of those parts. Computerized product design requires such capabilities as computer-aided analysis, computer-aided drafting, design sensitivity analysis, or optimization. The computer-aided analysis capability serves as part of the design proc-

[^1]

Figure 1.1 Automated design and manufacturing.
ess and is also used as a model simulator for the finished manufactured product. Analysis may be considered especially appropriate for a product whose initial design has to be modified several times during the manufacturing process. Thus computer-aided analysis can be used as a substitute for laboratory or field tests in order to reduce the cost.

The computer-aided manufacturing design branch is concerned with the design of the manufacturing process. This branch considers the manufacturability of newly designed parts and employs techniques to improve the manufacturing process, in addition to on-line control of the manufacturing process.

### 1.1.1 Computer-Aided Analysis

The computer-aided analysis process (CAA) allows the engineer to simulate the behavior of a product and modify its design prior to actual production. In contrast, prior to the introduction of CAA, the manufacturer had to construct and test a series of prototypes, a process which was not only time-consuming but also costly. Most optimal design techniques require repetitive analysis processes. Although one of the major goals of an automated factory is computer-aided design, computer-aided analysis techniques must be developed first.

Computer-aided analysis techniques may be applied to the study of electrical and electronic circuits, structures, or mechanical systems. The development of algorithms for analyzing electrical circuits began in the early days of electronic computers. Similar techniques were also employed to develop computer programs for structural analysis. Today, these programs, known as finite-element techniques, have become highly advanced and are used widely in various fields of engineering.

It was not until the early 1970s that computational techniques found their way into the field of mechanical engineering. One of the areas of mechanical engineering where computational techniques can be employed is the analysis of multibody mechanical systems.

### 1.2 MULTIBODY MECHANICAL SYSTEMS

A mechanical system is defined as a collection of bodies (or links) in which some or all of the bodies can move relative to one another. Mechanical systems may range from the very simple to the very complex. An example of a simple mechanical system is the single pendulum, shown in Fig. 1.2(a). This system contains two bodies - the pendulum and the ground. Examples of more complex mechanical systems are the four-bar linkage and the slider-crank mechanism, shown in Fig. 1.2(b) and (c), respectively. The four-bar linkage is the most commonly used mechanism for motion transmission. The slider-crank mechanism finds its greatest application in the internal-combustion engine.

While the motion of the systems in Fig. 1.2 is planar (two-dimensional), other mechanical systems may experience spatial (three-dimensional) motion. For example, the suspension and the steering system of an automobile, shown in Fig. 1.3, contain several spatial mechanisms. This system as a whole has several degrees of freedom. While the kinematics of the individual linkages in this vehicle are more complicated than those of the mechanisms shown in Fig. 1.2, the concept remains the same.

A cascade of simple planar linkage systems can be put together to perform rather complex tasks. The deployable satellite antenna shown in Fig. 1.4 contains such a cascade of six four-bar linkages. ${ }^{6}$ Before deployment, the panels of the antenna are folded in order to occupy the minimum space. Once the satellite is in orbit the panels are unfolded in a predefined sequence, as shown in Fig. 1.5. When the unfolding process is completed, the four-bar linkages become a truss structure to support the panels.


Figure 1.2 Examples of simple mechanical systems: (a) a single pendulum, (b) a four-bar mechanism, and (c) a slider-crank mechanism.


Figure 1.3 The suspension system and the steering mechanism of an automobile.


Figure 1.4 A deployed satellite antenna.

(a)

(b)

(c)

Figure 1.5 Unfolding process of the antenna in orbit: (a) folded panels; (b-e) unfold-
ing process.


Figure 1.5 (continued)

Another example of a mechanical system is a robotic device. A robot can be fixed to a stationary base or to a movable base, as shown in Fig. 1.6. The motion and the position of the end effector of a robot are controlled through force actuators located about each joint connecting the bodies that make up the robot.


Figure 1.6 Examples of robots with (a) stationary base and (b) movable base.

Any mechanical system can be represented schematically as a multibody system in the manner shown in Fig. 1.7. The actual shape or outline of a body may not be of immediate concern in the process of analysis. Of primary importance is the connectivity of the bodies, the inertial characteristics of the bodies, the type and the location of the joints, and the physical characteristics of the springs, dampers, and other elements in the system.


Figure 1.7 Schematic representation of a multibody system.

### 1.3 BRANCHES OF MECHANICS

There are two different aspects to the study of a mechanical system: analysis and design. When a mechanical system is acted on by a given excitation, for example, an external force, the system exhibits a certain response. The process which allows an engineer to study the response of an already existing system to a known excitation is called analysis. This requires a complete knowledge of the physical characteristics of the mechanical system, such as material composition, shape, and arrangement of parts. The process of determining which physical characteristics are necessary for a mechanical system to perform a prescribed task is called design or synthesis. The design process requires the application of scientific techniques along with the engineer's judgment. The scientific techniques in the design process are merely tools to be used by the engineer. These are mainly analysis techniques and optimization methods. Although these techniques can be employed in a systematic manner in the design process, the overall process hinges on the judgment of the design engineer. Since the scientific aspect of the design process requires analysis techniques as a tool, it is important to learn about methods of analysis prior to design.

The branch of analysis which studies motion, time, and forces is called mechanics. It consists of two parts - statics and dynamics. Statics considers the analysis of stationary systems - systems in which time is not a factor. Dynamics, on the other hand, deals with systems that are nonstationary - systems that change their response with respect to time. Dynamics is divided into two disciplines - kinematics and kinetics. Kinematics is the study of motion regardless of the forces that produce the motion. More explicitly, kinematics is the study of displacement, velocity, and acceleration. Kinetics, on the other hand, is the study of motion and its relationship with the forces that produce that motion.

The focus of this book is on the dynamics of mechanical systems, with an emphasis on computational methods. In addition, one chapter is devoted to computational methods in static equilibrium analysis, since this may be needed prior to dynamic analysis for certain mechanical systems.

### 1.3.1 Methods of Analysis

Before we analyze the motion of any mechanical system, we must make some simplifying assumptions. For example, if the overall acceleration of a vehicle under the applied
load of the engine is to be determined, then the vibrational motions of certain parts of the vehicle are of no significance. If one decides to consider the vibration and local deformation of every part of the vehicle, then determining the response of the system becomes highly complicated, if not impossible. Therefore, these simplifying assumptions serve two purposes: to make the problem solvable and to eliminate the expenditure of effort on unnecessary or insignificant responses.

Classical methods of analysis in mechanics have relied upon graphical and often quite complex techniques. These techniques are based on geometrical interpretations of the system under consideration. As an example, consider the slider-crank mechanism shown in Fig. 1.8. The crank is rotating with a constant angular velocity. The objective is to find the velocity of the slider. A graphical solution to this problem can be achieved rather easily. The velocity of point $A, \vec{v}^{A}$, has a magnitude of $v^{A}=(1.0)(0.1)=0.1 \mathrm{~m} / \mathrm{s}$ and is perpendicular to the crank $O A$, as shown in Figure 1.9(a). The velocity of point $B, \vec{v}^{B}$, is in the direction of the motion of the slider, and the velocity of point $B$ relative to point $A$, denoted by vector $\vec{v}^{B A}$, is perpendicular to the connecting rod $A B$. A vector expression relating these velocities is given as

$$
\begin{equation*}
\vec{v}^{B}=\vec{v}^{A}+\vec{v}^{D A} \tag{1.1}
\end{equation*}
$$

A vector diagram (velocity polygon) corresponding to this expression is shown in Fig. 1.9(b). From this diagram the magnitude and the direction of $\vec{v}^{B}$ can be found.


Figure 1.8 A slider-crank mechanism.


Figure 1.9 Graphical solution.

Although a graphical solution to this problem is rather simple, its accuracy is limited. The graphical approach can yield more accurate results if some trigonometric formulas and geometric relations are introduced into the process. For example, for the slider-crank mechanism, since the angle $\phi$ and the lengths of the crank and the connecting rod are known, other geometric information for this system can be found easily, as depicted in Fig. 1.10(a). Then a vector diagram can be constructed with complete details as shown in Fig. 1.10(b). From this diagram, $v^{B}$ can be calculated from the elementary

(a)

(b)

Figure 1.10 Geometric approach with detailed information.
relationship between the sides and the angles of a triangle:

$$
v^{B}=(0.1) \frac{\sin 44.48}{\sin 75.52}=0.072 \mathrm{~m} / \mathrm{s}
$$

and of course the direction of $\vec{v}^{B}$ is known.
An alternative method to the graphical approach is the solution by vector algebra. In this method, all the vectors are expressed in terms of their components in a common coordinate system. For example, the components of the velocity vectors of the slidercrank mechanism in the $x y$ coordinate system are

$$
\mathbf{v}^{A}=\left[\begin{array}{c}
0.1 \cos 60 \\
-0.1 \sin 60
\end{array}\right] \quad \mathbf{v}^{B}=\left[\begin{array}{c}
v^{B} \\
0
\end{array}\right] \quad \mathbf{v}^{B A}=\left[\begin{array}{c}
v^{B A} \cos 75.52 \\
v^{B A} \sin 75.52
\end{array}\right]
$$

Substitution of these components in Eq. 1.1 yields

$$
\begin{aligned}
v^{B} & =0.1 \cos 60+v^{B A} \cos 75.52 \\
0 & =-0.1 \sin 60+v^{B A} \sin 75.52
\end{aligned}
$$

which results in $v^{B A}=0.089 \mathrm{~m} / \mathrm{s}$ and $v^{B}=0.072 \mathrm{~m} / \mathrm{s}$.
Kinematic analysis with vector algebra may lead to solving linear or nonlinear simultaneous algebraic equations. For example, the geometric relations between the sides and the angles of the triangle made by the slider-crank mechanism can be expressed as

$$
\begin{align*}
a \cos \phi+b \cos \theta-d & =0  \tag{1.2}\\
a \sin \phi-b \sin \theta & =0
\end{align*}
$$

where $a=O A, b=A B$, and $d=O B$. Since $a$ and $b$ have known values, Eq. 1.2 can be written as

$$
\begin{array}{r}
0.1 \cos \phi+0.2 \cos \theta-d=0 \\
0.1 \sin \phi-0.2 \sin \theta=0 \tag{1.3}
\end{array}
$$

For any given value of the crank angle $\phi$, the solution to Eq .1 .3 yields values for $\theta$ and $d$. For example for $\phi=30^{\circ}$, it is found that $\theta=14.48^{\circ}$ and $d=0.28 \mathrm{~m}$. The time derivative of Eq. 1.3 yields the velocity equations,

$$
\begin{align*}
-0.1 \dot{\phi} \sin \phi-0.2 \dot{\theta} \sin \theta-\dot{d} & =0  \tag{1.4}\\
0.1 \dot{\phi} \cos \phi-0.2 \dot{\theta} \cos \theta & =0
\end{align*}
$$

The angular velocity of the crank is $\omega=\dot{\phi}=-1 \mathrm{rad} / \mathrm{s}$. (Since the direction of the angular velocity of the crank is opposite to the defined positive direction of $\phi$, a negative sign is given to $\phi$.) Substituting this in Eq. 1.4 yields $\theta=-0.45 \mathrm{rad} / \mathrm{s}$ and $d=$ $0.072 \mathrm{~m} / \mathrm{s}$, where $d$ represents the velocity of point $B$.

The method of solution with vector algebra is an analytic approach. This approach is more systematic when compared with the graphical method. A problem formulated analytically can be solved repeatedly for different values of input. For example, if the angle $\phi$ and the angular velocity $\omega$ of the crank are varied as a function of time, Eqs. 1.3 and 1.4 can be solved repeatedly to obtain the solution. Although this process can be performed with pencil and paper, a computer program can do the job more efficiently.

The usefulness of writing a computer program becomes even more apparent when the mechanical system under consideration is more complex than a planar slider-crank mechanism. For example, the spatial five-bar linkage shown in Fig. I.11 has two input angles, $\phi_{1}$ and $\phi_{2}$. If this system is considered for kinematic analysis, a graphical approach would be very tedious as well as inaccurate. In contrast, if the problem is solved analytically, the solution is accurate and is found efficiently.


Figure 1.11 A spatial five-bar mechanism.
An analytical approach using a computer program manifests itself when a mechanical system is considered for dynamic (kinetic) analysis. Equations to represent the motion of a system that contains the applied loads and other characteristics of the system are either differential equations or mixed algebraic-differential equations. An exact (closed-form) solution to these equations cannot be found except for highly simplified cases. Regardless of the complexity of the equations of motion, it is always possible to solve them numerically.

### 1.4 COMPUTATIONAL METHODS

The purpose of computer-aided analysis of mechanical systems is to develop basic methods for computer formulation and solution of the equations of motion. This requires systematic techniques for formulating the equations and numerical methods for solving them. A computer program for the analysis of mechanical systems can be either a special-purpose program or a general-purpose program.

A special-purpose program is a rigidly structured computer code that deals with only one type of application. The equations of motion for that particular application are
derived a priori and then formulated into the program. As input to the program, the user can provide information such as the dimensions and physical characteristics of each part. Such a program can be made computationally efficient and its storage requirement can be minimized, with the result that it will be suitable for implementation on small personal computers. The major drawback of a special-purpose program is its lack of flexibility for handling other types of applications.

A general-purpose program can be employed to analyze a variety of mechanical systems. For example, the planar motion of a four-bar linkage under applied loads and the spatial motion of a vehicle driven over a rough terrain can be simulated with the same general-purpose program. The input data to such a program are provided by the user and must completely describe the mechanical system under consideration. The input must contain such information as number of bodies, connectivity between the bodies, joint types, force elements, and geometric and physical characteristics. The program then generates all of the governing equations of motion and solves them numerically. A general-purpose program, compared with a special-purpose program, is not computationally as efficient and requires more memory space, but it is flexible in use.

The computational efficiency of a general-purpose program depends upon several factors, two of which are the choice of coordinates and the method of numerical solution. The choice of coordinates directly influences both the number of the equations of motion and their order of nonlinearity. Furthermore, depending upon the form of these equations, one method of numerical solution may be preferable to another in terms of efficiency and accuracy.

### 1.4.1 Efficiency versus Simplicity

The governing equations of motion for a mechanical system can be derived and expressed in a variety of forms, dependent mainly upon the type of coordinates being employed. A set of coordinates $\mathbf{q}$ selected for a system can describe the position of the elements in the system either with respect to each other or with respect to a common reference frame. In order to show how different sets of coordinates can lead to different formulations describing the same system, a simple example is given here. In this example, a four-bar linkage is considered for kinematic analysis. Therefore, all of the governing equations of motion are algebraic equations: i.e., no differential equations are involved.

The first formulation shown here considers only one coordinate to describe the configuration of the system, since a four-bar linkage has only one degree of freedom. This is referred to as the generalized coordinate of the system. In a system of $n$ degrees of freedom, there will be $n$ generalized coordinates. As shown in Fig. 1.12, the angle $\phi$, describing the orientation of the crank with respect to the ground, can be selected as the generalized coordinate; i.e.,

$$
\begin{equation*}
\mathbf{q}=[\phi] \tag{1.5}
\end{equation*}
$$

For any given configuration, i.e., known $\phi$, any other information on the position of any point in the system can be calculated. For example, the angles $\theta_{1}, \theta_{2}$, and $\theta_{3}$ can be found from the following formulas:

$$
\begin{gather*}
\left(r^{2}+l^{2}+s^{2}-d^{2}\right)-2 r l \cos \phi+2 l s \cos \theta_{1}-2 r s \cos \left(\phi-\theta_{1}\right)=0  \tag{1.6}\\
\left(r^{2}+l^{2}+s^{2}-d^{2}\right)-2 r l \cos \phi+2 d s \cos \theta_{2}=0 \tag{1.7}
\end{gather*}
$$



Figure 1.12 A four-bar mechanism with generalized coordinate $\phi$.

$$
\begin{equation*}
\phi+\theta_{1}+\theta_{2}+\theta_{3}-2 \pi=0 \tag{1.8}
\end{equation*}
$$

In these equations $r, d$, and $s$ represent the lengths of the links, and $l$ represents the distance between points $A$ and $D$. These formulas are derived from simple geometric realizations. It is clear that for a given $\phi$, Eq. 1.6 yields $\theta_{1}$, then Eq. 1.7 yields $\theta_{2}$, and finally Eq. 1.8 yields $\theta_{3}$. The solution of these equations requires direct substitution there is no need to solve a set of simultaneous algebraic equations. Now, it should be clear that the coordinates of a typical point attached to one of the links, in this case the point $F$ on link $B C$, can be found easily.

The second way of formulating the kinematic equations for the four-bar linkage considers three coordinates. For this or any other mechanism, the selected coordinates may define the orientation of each moving body with respect to a nonmoving body or with respect to another moving body. Therefore, in this book these coordinates are referred to as relative coordinates. As shown in Fig. 1.13, angles $\phi_{1}, \phi_{2}$, and $\phi_{3}$ are selected as a set of coordinates

$$
\mathbf{q}=\left[\begin{array}{l}
\phi_{1}  \tag{1.9}\\
\phi_{2} \\
\phi_{3}
\end{array}\right]
$$

These angles are measured between the positive $x$ axis and the positive vectors representing the links. Since the four-bar linkage has only one degree of freedom, the three


Figure 1.13 Relative coordinates describing the configuration of a four-bar system.
coordinates are not independent. Two loop equations relating these coordinates can be written as follows:

$$
\begin{array}{r}
r \cos \phi_{1}+d \cos \phi_{2}+s \cos \phi_{3}-l=0  \tag{1.10}\\
r \sin \phi_{1}+d \sin \phi_{2}+s \sin \phi_{3}=0
\end{array}
$$

For any given configuration, i.e., known $\phi_{1}$, the set of two simultaneous algebraic equations must be solved for $\phi_{2}$ and $\phi_{3}$. After Eq. 1.10 is solved, other information such as the $x y$ coordinates of a point $F$ can be calculated.

The third formulation uses three Cartesian coordinates per link - the $x$ and $y$ coordinates of the center point of each link and the angle of the link which is measured with respect to the $x$ axis, as shown in Fig. 1.14. Thus, the set of coordinates describing the configuration of the four-bar linkage is

$$
\mathbf{q}=\left[\begin{array}{lllllllll}
x_{1} & y_{1} & \phi_{1} & x_{2} & y_{2} & \phi_{2} & x_{3} & y_{3} & \phi_{3} \tag{1.11}
\end{array}\right]^{T}
$$

These nine coordinates are dependent upon each other through eight equations:

$$
\begin{align*}
x_{1}-\frac{r}{2} \cos \phi_{1} & =0 \\
y_{1}-\frac{r}{2} \sin \phi_{1} & =0 \\
x_{1}+\frac{r}{2} \cos \phi_{1}-x_{2}+\frac{d}{2} \cos \phi_{2} & =0 \\
y_{1}+\frac{r}{2} \sin \phi_{1}-y_{2}-\frac{d}{2} \sin \phi_{2} & =0  \tag{1.12}\\
x_{2}+\frac{d}{2} \cos \phi_{2}-x_{3}-\frac{s}{2} \cos \phi_{3} & =0 \\
y_{2}+\frac{d}{2} \sin \phi_{2}-y_{3}-\frac{s}{2} \sin \phi_{3} & =0 \\
x_{3}-\frac{s}{2} \cos \phi_{3}-l & =0 \\
y_{3}-\frac{s}{2} \sin \phi_{3} & =0
\end{align*}
$$

For any known configuration, any of the nine variables can be specified, and then the remaining eight variables can be found by solving the set of eight nonlinear algebraic equations in eight unknowns.


Figure 1.14 Cartesian coordinates describing the configuration of a four-bar linkage.

The three preceding forms of formulation with generalized coordinates, relative coordinates, and Cartesian coordinates describe the kinematics of a four-bar mechanism. For dynamic analysis, the differential equations of motion for the four-bar linkage, or for any other mechanical system, can also be derived in terms of any of these sets of coordinates. For the four-bar linkage, formulation with generalized coordinates yields one second-order differential equation in terms of $\phi, \dot{\phi}$, and $\ddot{\phi}$. This equation is highly nonlinear and complex in terms of $\phi$ and $\phi$. The equations of motion, for the four-bar linkage in terms of the relative..coordinates, consist of three second-order differential equations in terms of $\phi_{i}, \phi_{i}$ and $\phi_{i}$ for $i=1,2$, and 3 . The order of nonlinearity of these equations is not as high or as complex as in the first case. However, with these three differential equations, the two algebraic constraint equations of Eq. 1.10 must be considered. Therefore, the governing equations of motion for this system in terms of relative coordinates are a mixed set of algebraic-differential equations. Similarly, in terms of the Cartesian coordinates, nine second-order differential equations can be derived. Those, in conjunction with the eight algebraic constraint equations of Eq. 1.12 , would define the governing equations of motion for the four-bar linkage. These algebraic-differential equations are loosely coupled and have a relatively low order of nonlinearity when compared with the previous sets.

A crude but general comparison between these three sets of coordinates, with regard to several crucial and important aspects, is summarized in Table 1.1. A general conclusion that can be made from this table is that the smaller the number of coordinates and equations, the higher the order of nonlinearity and complexity of the governing equations of motion, and vice versa. Other aspects for comparison, not listed in this table, are the numerical solution of the governing equations of motion and the numerical error encountered in the solution for different formulations.

TABLE 1.1

|  | Generalized <br> coordinates | Relative <br> coordinates | Cartesian <br> Number of coordinates |
| :--- | :--- | :--- | :--- |
| Number of second-order <br> differential equations | Minimum $^{\dagger}$ | Minimum ${ }^{\dagger}$ | Moderate <br> Number of algebraic <br> constraint equations |
| Order of nonlinearity <br> Derivation of the <br> equations of motion | High | Moderate | Large |
| Computational <br> efficiency | Hard | Large |  |
| Development of a <br> general-purpose <br> computer program | Difficult | Moderate <br> Moderately <br> hard | Low $^{\dagger}$ |

${ }^{\dagger}$ An advantage over the other two sets of coordinates
${ }^{\text {F }}$ Computational efficiency in solving the governing equations of motion is dependent on the form and the number of equations, and the method of numerical solution. Thercfore, this is a very general remark.

Numerical methods for solving ordinary differential equations have been well known for decades. Well-developed algorithms with reliable error control mechanisms have been used extensively in every area of science and engineering. This can be considered another advantage in formulating the equations of motion in terms of a set of generalized coordinates. On the other hand, the governing equations of motion in terms of a set of relative coordinates or Cartesian coordinates are mixed algebraic-differential equations. Methods of numerical solution for such equations, compared with those for the ordinary differential equations, are still in their infancy (this subject is discussed in detail in Chap. 13).

Numerical solutions for differential equations provide only an approximation to the actual (exact) solution. The deviation between the numerical solution and the actual response is the numerical error inherent in the solution. One of the main factors influencing the amount of error in the solution is the number of equations. Generally, the larger the number of equations, the greater the chances for accumulation of numerical error. This can be considered one more advantage in using a minimum number of coordinates.

At this point, it can be concluded that the points in favor of using a set of generalized coordinates for formulating the governing equations of motion outnumber those favoring the other coordinates. However, the disadvantages must not be overlooked. The complexity in deriving the equations of motion and the difficulty in developing a versatile computer program for general usage require an advanced knowledge of dynamics and prior experience in developing large-scale codes. In contrast to the generalized coordinates, the derivation of the equations of motion with Cartesian coordinates is simple. The resulting equations can easily be put into general usage and into a versatile computer program. If computational efficiency is not the decisive factor, then a set of Cartesian coordinates can be an attractive candidate.

It can be concluded that a set of relative coordinates falls in the middle of the "comparison scale." Therefore, selection of a set of relative coordinates might be a good compromise between the generalized and Cartesian coordinates in formulating the governing equations of motion. In Chap. 13, it will be shown how a dynamic analysis algorithm can be developed to take advantage of the simplicity of the Cartesian coordinates for formulating the equations of motion and the efficiency of the generalized or relative coordinates for the numerical solution.

### 1.4.2 A General-Purpose Program

A general-purpose computer program for the dynamic analysis of mechanical systems must perform four basic functions:

1. Accepting data from the user
2. Generating the governing equations of motion
3. Solving the equations
4. Communicating the result to the user

The first step is referred to as the input phase, the second and third steps are the analysis phase, and the fourth step is the output phase.

Input. The user must furnish for the program a description of the system under consideration through a set of engineering data. As an example, assume that the doublewishbone suspension system with steering shown in Fig. 1.15(a) is considered for dynamic analysis. A schematic representation of this system is shown in Fig. 1.15(b). The system consists of six moving bodies, a nonmoving body (the ground), a spring and a damper, four spherical joints, and four revolute joints. The input data describing this system must contain such information as:

1. Number of bodies, number and types of joints
2. Mass and moments of inertia of each moving body
3. Connectivity information between the bodies
4. Connectivity information and characteristics of the spring and damper
5. Tire characteristics (if its deformation is to be considered)
6. Direction of gravity
7. Initial conditions on the position and velocity of each body
8. Steering input (from the driver) and applied forces to the wheel (from the road)

Note that in rigid-body dynamic analysis, the shape of bodies need not be described this information is needed only if a graphical display of the system is required.

The minimum requirement for generating and communicating a set of input data to the analysis program is an alphanumeric computer terminal. The input can be entered manually via the keyboard and transmitted to the analysis program, which may reside on a mainframe computer, a minicomputer, or a personal microcomputer. The process of generating the input data can be facilitated by developing a preprocessing program and employing a digitizer tablet, a graphics terminal, or a CAD system.


Figure 1.15 (a) A double-wishbone suspension system, and (b) its corresponding schematic representation.

Analysis. On the basis of the input data, the analysis program generates all of the necessary equations describing the system. These equations are then solved numerically in order to obtain the response of the system under the specified loads. The numeri-


Figure 1.16 Graphic display of the response for the double-wishbone suspension system (a) to a steering command, and (b) to an obstacle on the road.
cal algorithms provide the solution to the equations at discretized points in time. The response at each time point is communicated to the user; it contains such information as the position, velocity, and acceleration of each moving body and the reaction forces at the joints.

Output. The minimum output-device requirement is either a terminal screen or a printer. The numerical result of the dynamic simulation for systems undergoing planar motion and having only a few moving bodies may not be too extensive. When it is not, one can interpret and understand the dynamic response for such systems from a printed output. But the task can become extremely time-consuming and tedious when the number of moving bodies is large, particularly when the system undergoes spatial motion. The difficulty of interpreting the dynamic response can be resolved by developing a postprocessor program capable of communicating the result to the user through various forms of output device, e.g., a printer, a plotter, or a graphic display unit.

Possibly one of the most expressive forms of communication is computer graphics. For this purpose, the user must provide for the graphics package the exact or an approximate shape of each body. This can be done by defining a set of points on each body and specifying the connectivity between these points. The lines produced from the connectivity information, when displayed, represent surfaces of the outline of each body. The outlines can be positioned in their proper orientation according to the position data provided by the output at any required point in time.

Figure 1.16 is a graphical display of the result of a dynamic simulation of the suspension system of Fig. 1.15. Figure 1.16(a) shows the response of the wheel to a steering command, and Fig. 1.16(b) shows the response of the system when it encounters an


Figure 1.17 Dynamic response of the double-wishbone suspension system presented as a series of graphic displays to form an animation.
obstacle on the road. Figure 1.17 shows the graphic presentation of the system at several instants of time. When a sequence of graphic displays at small and successive increments of time is generated and displayed at a rate of at least 30 frames per second, an animated picture of the motion is created. This requires a high-speed graphic display device that is capable of displaying several thousand lines or polygons, flicker-free, in one second.

## Vectors

## and

## Matrices

Vector and matrix algebra form the mathematical foundation for kinematics and dynamics. Geometry of motion is at the heart of both the kinematics and the dynamics of mechanical systems. Vector analysis is the time-honored tool for describing geometry. In its geometric form, however, vector algebra is not well suited to computer implementation.

In this chapter, a systematic matrix formulation of vector algebra, referred to as algebraic vector representation, is presented for use throughout the text. This form of vector representation, in contrast to the more traditional geometric form of vector representation, is easier to use for either formula manipulation or computer implementation. Elementary properties of vector and matrix algebra are stated in this chapter without proof.

### 2.1 GEOMETRIC VECTOR

When we write a vector in the form $\vec{a}$, it is understood from the arrow notation that we are referring to the vector in its geometric sense: it begins at a point $A$ and ends at a point $B$. The magnitude of vector $\vec{a}$ is denoted by $a$. A unit vector in the direction of $\vec{a}$ is shown as $\vec{u}_{(a)}$.

Vectors lying in the same plane are called coplanar vectors. Collinear vectors have the same direction and the same line of action. Equal vectors have the same magnitude and direction. A zero or null vector has zero magnitude and therefore no specified direction.

Multiplication of a vector $\vec{a}$ by a scalar $\alpha$ is defined as a vector in the same direction as $\vec{a}$ that has a magnitude $\alpha a$. The negative of a vector is obtained by multiplying the vector by -1 ; it changes the direction of the vector.

The vector sum of two vectors $\vec{a}$ and $\vec{b}$ is written as

$$
\begin{equation*}
\vec{c}=\vec{a}+\vec{b} \tag{2.1}
\end{equation*}
$$

The product of a sum of two scalars $\alpha+\beta$ and a vector $\vec{a}$ is expanded as

$$
\begin{equation*}
(\alpha+\beta) \vec{a}=\alpha \vec{a}+\beta \vec{a} \tag{2.2}
\end{equation*}
$$

A vector $\vec{a}$ can be resolved into its Cartesian components $a_{(x)}, a_{(y)}$, and $a_{(z)}$ along the $x, y$, and $z$ axes of a Cartesian system. Here, the unit vectors $\vec{u}_{(x)}, \vec{u}_{(y)}$, and $\vec{u}_{(z)}$ are directed along the coordinate axes $x, y$, and $z .{ }^{\dagger}$ In vector notation, the resolution of the vector into its components is expressed as

$$
\begin{equation*}
\vec{a}=a_{(x)} \vec{u}_{(x)}+a_{(y)} \vec{u}_{(y)}+a_{(z)} \vec{u}_{(z)} \tag{2.3}
\end{equation*}
$$

If the angles between the vector $\vec{a}$ and the positive $x, y$, and $z$ axes are denoted by $\theta_{(x)}$, $\theta_{(y)}$, and $\theta_{(z)}$, the components of vector $\vec{a}$ are given as

$$
\begin{align*}
a_{(x)} & =a \cos \theta_{(x)} \\
a_{(y)} & =a \cos \theta_{(y)}  \tag{2.4}\\
a_{(z)} & =a \cos \theta_{(z)}
\end{align*}
$$

The quantities $\cos \theta_{(x)}, \cos \theta_{(y)}$, and $\cos \theta_{(z)}$ are the direction cosines of vector $\vec{a}$.
The scalar (or dot) product of two vectors $\vec{a}$ and $\vec{b}$ is defined as the product of the magnitudes of the two vectors and the cosine of the angle between them; i.e.,

$$
\begin{align*}
\vec{a} \cdot \vec{b} & =a b \cos \theta  \tag{2.5}\\
& =a_{(x)} b_{(x)}+a_{(y)} b_{(y)}+a_{(z)} b_{(z)}  \tag{2.6}\\
& =\vec{b} \cdot \vec{a} \tag{2.7}
\end{align*}
$$

where the angle $\theta$ between the vectors is measured in the plane of intersection of the vectors. If the two vectors are nonzero, i.e., if $a \neq 0$ and $b \neq 0$, then their scalar product is zero only if $\cos \theta=0$. Two nonzero vectors are thus said to be orthogonal (perpendicular) if their scalar product is zero. For any vector $\vec{a}$,

$$
\begin{equation*}
\vec{a} \cdot \vec{a}=a^{2} \tag{2.8}
\end{equation*}
$$

The vector (or cross) product of two vectors $\vec{a}$ and $\vec{b}$ is defined as the vector

$$
\begin{align*}
\vec{c} & =\vec{a} \times \vec{b}  \tag{2.9}\\
& =a b \sin \theta \vec{u}  \tag{2.10}\\
& =\left(a_{(y)} b_{(z)}-a_{(z)} b_{(y)}\right) \vec{u}_{(x)}+\left(a_{(z)} b_{(x)}-a_{(x)} b_{(z)}\right) \vec{u}_{(y)}+\left(a_{(x)} b_{(y)}-a_{(y)} b_{(x)}\right) \vec{u}_{(z)} \tag{2.11}
\end{align*}
$$

where $\vec{u}$ is a unit vector that is orthogonal to the plane of intersection of the two vectors $\vec{a}$ and $\vec{b}$, taken in the positive right-hand coordinate direction, and $\theta$ is the angle between vectors $a$ and $b$. Since reversal of the order of the vectors $\vec{a}$ and $\vec{b}$ in Eq. 2.9 would yield an opposite direction for the unit vector, it is clear that

$$
\begin{equation*}
\vec{b} \times \vec{a}=-\vec{a} \times \vec{b} \tag{2.12}
\end{equation*}
$$

While not obvious on geometrical grounds, the following identities are valid:

$$
\begin{align*}
(\vec{a}+\vec{b}) \cdot \vec{c} & =\vec{a} \cdot \vec{c}+\vec{b} \cdot \vec{c}  \tag{2.13}\\
(\vec{a}+\vec{b}) \times \vec{c} & =\vec{a} \times \vec{c}+\vec{b} \times \vec{c} \tag{2.14}
\end{align*}
$$

[^2]From the definition of scalar product, vector product, and unit coordinate vectors, the following identities are valid:

$$
\begin{align*}
\vec{u}_{(x)} \cdot \vec{u}_{(y)} & =\vec{u}_{(y)} \cdot \vec{u}_{(z)}=\vec{u}_{(z)} \cdot \vec{u}_{(x)}=0  \tag{2.15}\\
\vec{u}_{(x)} \cdot \vec{u}_{(x)} & =\vec{u}_{(y)} \cdot \vec{u}_{(y)}=\vec{u}_{(z)} \cdot \vec{u}_{(z)}=1
\end{align*}
$$

and

$$
\begin{align*}
& \vec{u}_{(x)} \times \vec{u}_{(x)}=\vec{u}_{(y)} \times \vec{u}_{(y)}=\vec{u}_{(z)} \times \vec{u}_{(z)}=\overrightarrow{0} \\
& \vec{u}_{(x)} \times \vec{u}_{(y)}=\vec{u}_{(z)} \\
& \vec{u}_{(y)} \times \vec{u}_{(z)}=\vec{u}_{(x)}  \tag{2.16}\\
& \vec{u}_{(z)} \times \vec{u}_{(x)}=\vec{u}_{(y)}
\end{align*}
$$

### 2.2 MATRIX AIND ALGEBRAIC VECTORS

Compact matrix notation often allows one to concentrate on the form of a system of equations and what it means, rather than on the minute details of its construction. Matrix manipulation also allows for the organized development and simplification of systems of equations.

A matrix with $m$ rows and $n$ columns is said to be of dimension $m \times n$ and is denoted by a boldface capital letter; it is written in the form

$$
\mathbf{A} \equiv\left[a_{i j}\right] \equiv\left[\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1 n} \\
a_{21} & a_{22} & \cdots & a_{2 n} \\
\vdots & \vdots & & \vdots \\
a_{m 1} & a_{m 2} & \cdots & a_{m n}
\end{array}\right]_{(m \times n)}
$$

where a typical element $a_{i j}$ is located at the intersection of the $i$ th row and $j$ th column. The transpose of a matrix is formed by interchanging rows and columns and is designated by the superscript $T$. Thus, if $a_{i j}$ is the $i j$ element of matrix $\mathbf{A}, a_{j i}$ is the $i j$ element of its transpose $\mathbf{A}^{T}$.

A matrix with only one column is called a column matrix and is denoted by a boldface lower-case letter; e.g., a. A matrix with only one row is called a row matrix and is denoted as $\mathbf{a}^{T}$; i.e., as the transpose of a column matrix. An $m \times n$ matrix can be considered to be constructed of $n$ column matrices $\mathbf{a}_{j}$, where $j=1, \ldots, n$, or $m$ row matrices $\mathbf{a}_{i}^{T}$; where $i=1, \ldots, m$.

The vector $\vec{a}$ in Eq. 2.3 is uniquely defined by its Cartesian components and can be written in matrix notation as follows:

$$
\mathbf{a}=\left[\begin{array}{l}
a_{(x)}  \tag{2.17}\\
a_{(y)} \\
a_{(z)}
\end{array}\right] \equiv\left[a_{(x)}, a_{(y)}, a_{(z)}\right]^{T}
$$

This is the algebraic (or component) representation of a vector.

### 2.2.1 Matrix Operations

In this section, the terminology of matrix algebra is briefly reviewed. Several useful identities are stated that are used extensively throughout this text.

A square matrix has an equal number of rows and columns. A diagonal matrix is a square matrix with $a_{i j}=0$ for $i \neq j$ and at least one nonzero diagonal term. An $n \times n$ diagonal matrix is denoted by

$$
\mathbf{A} \equiv \operatorname{diag}\left[a_{11}, a_{22}, \ldots, a_{m n}\right]
$$

If square matrices $\mathbf{B}_{i}, i=1, \ldots, k$, are arranged along the diagonal of a matrix $\mathbf{D}$ to give

$$
\mathbf{D} \equiv\left[\begin{array}{llllll}
\mathbf{B}_{1} & & & & \\
& \mathbf{B}_{2} & & & \mathbf{0} & \\
& & \cdot & & & \\
& \mathbf{0} & & & \cdot & \mathbf{B}_{k}
\end{array}\right]
$$

then the matrix is called a quasi-diagonal matrix and is denoted by

$$
\mathbf{D} \equiv \operatorname{diag}\left[\mathbf{B}_{1}, \mathbf{B}_{2}, \ldots, \mathbf{B}_{k}\right]
$$

even though $\mathbf{D}$ is not a true diagonal matrix. An $n \times n$ unit or identity matrix, denoted normally as $\mathbf{I}$, is a diagonal matrix with $a_{i i}=1, i=1, \ldots, n$. A null matrix or zero matrix, designated as $\mathbf{0}$, has $a_{i j}=0$ for all $i$ and $j$.

If two matrices $\mathbf{A}$ and $\mathbf{B}$ are of the same dimension, they are defined to be equal matrices if $a_{i j}=b_{i j}$ for all $i$ and $j$. The sum of two equidimensional matrices $\mathbf{A}$ and $\mathbf{B}$ is a matrix with the same dimension, defined as

$$
\begin{equation*}
\mathbf{C}=\mathbf{A}+\mathbf{B} \tag{2.18}
\end{equation*}
$$

where $c_{i j}=a_{i j}+b_{i j}$ for all $i$ and $j$. The difference between two matrices $\mathbf{A}$ and $\mathbf{B}$ of the same dimension is defined as the matrix

$$
\begin{equation*}
\mathbf{C}=\mathbf{A}-\mathbf{B} \tag{2.19}
\end{equation*}
$$

where $c_{i j}=a_{i j}-b_{i j}$ for all $i$ and $j$. For matrices having the same dimension, the following identities are valid:

$$
\begin{align*}
(\mathbf{A}+\mathbf{B})+\mathbf{C} & =\mathbf{A}+(\mathbf{B}+\mathbf{C})=\mathbf{A}+\mathbf{B}+\mathbf{C}  \tag{2.20}\\
\mathbf{A}+\mathbf{B} & =\mathbf{B}+\mathbf{A} \tag{2.21}
\end{align*}
$$

Multiplication of a matrix by a scalar is defined as

$$
\begin{equation*}
\alpha \mathbf{A}=\mathbf{C} \tag{2.22}
\end{equation*}
$$

where $c_{i j}=\alpha a_{i j}$.
Let $\mathbf{A}$ be an $m \times p$ matrix and let $\mathbf{B}$ be a $p \times n$ matrix, written in the form

$$
\mathbf{A} \equiv\left[\begin{array}{c}
\mathbf{a}_{1}^{T} \\
\mathbf{a}_{2}^{T} \\
\vdots \\
\vdots \\
\mathbf{a}_{m}^{T}
\end{array}\right] \quad \mathbf{B} \equiv\left[\mathbf{b}_{1}, \mathbf{b}_{2}, \ldots, \mathbf{b}_{n}\right]
$$

where the $\mathbf{a}_{i}^{T}, i=1, \ldots, m$, are row vectors with $p$ elements and the $\mathbf{b}_{i}, i=1, \ldots, n$, are column vectors with $p$ elements. Then the matrix product of $\mathbf{A}$ and $\mathbf{B}$ is defined as the $m \times n$ matrix

$$
\begin{equation*}
\mathbf{C}=\mathbf{A B} \tag{2.23}
\end{equation*}
$$

where

$$
\mathbf{C}=\left[\begin{array}{cccc}
\mathbf{a}_{1}^{T} \mathbf{b}_{1} & \mathbf{a}_{1}^{T} \mathbf{b}_{2} & \cdots & \mathbf{a}_{1}^{T} \mathbf{b}_{n}  \tag{2.24}\\
\mathbf{a}_{2}^{T} \mathbf{b}_{1} & \mathbf{a}_{2}^{T} \mathbf{b}_{2} & \cdots & \mathbf{a}_{2}^{T} \mathbf{b}_{n} \\
\vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots \\
\mathbf{a}_{m}^{T} \mathbf{b}_{1} & \mathbf{a}_{m}^{T} \mathbf{b}_{2} & \cdots & \mathbf{a}_{m}^{T} \mathbf{b}_{n}
\end{array}\right]
$$

or $c_{i j}=\mathbf{a}_{i}^{T} \mathbf{b}_{j}$. The scalar product $\mathbf{a}^{T} \mathbf{b}$ for two vectors $\mathbf{a}=\left[a_{1}, a_{2}, \ldots, a_{p}\right]^{T}$ and $\mathbf{b}=\left[b_{1}, b_{2}, \ldots, b_{p}\right]^{T}$ is defined as

$$
\begin{equation*}
\mathbf{a}^{T} \mathbf{b}=a_{1} b_{1}+a_{2} b_{2}+\cdots+a_{p} b_{p} \tag{2.25}
\end{equation*}
$$

It is important to note that the product of two matrices is defined only if the number of columns in the first matrix equals the number of rows in the second matrix. It is clear from the definition that, in general,

$$
\begin{equation*}
\mathbf{A B} \neq \mathbf{B} \mathbf{A} \tag{2.26}
\end{equation*}
$$

In fact, the products $\mathbf{A B}$ and $\mathbf{B A}$ are defined only if both $\mathbf{A}$ and $\mathbf{B}$ are square and of equal dimension.

The following identities are valid, assuming that the matrices have proper dimensions:

$$
\begin{align*}
(\mathbf{A}+\mathbf{B}) \mathbf{C} & =\mathbf{A C}+\mathbf{B C}  \tag{2.27}\\
(\mathbf{A B}) \mathbf{C} & =\mathbf{A}(\mathbf{B C})=\mathbf{A B C}  \tag{2.28}\\
(\mathbf{A}+\mathbf{B})^{T} & =\mathbf{A}^{T}+\mathbf{B}^{T}  \tag{2.29}\\
(\mathbf{A B})^{T} & =\mathbf{B}^{T} \mathbf{A}^{T} \tag{2.30}
\end{align*}
$$

If $a_{i j}=a_{j i}$ for all $i$ and $j$, the matrix $\mathbf{A}$ is called symmetric; i.e., $\mathbf{A}=\mathbf{A}^{T}$. If $a_{i j}=-a_{j j}$ for all $i$ and $j$, the matrix $\mathbf{A}$ is called skew-symmetric; i.e., $\mathbf{A}=-\mathbf{A}^{T}$. Note that in this case, $a_{i j}=0$, for all $i$.

Consider an $m \times p$ matrix $\mathbf{A}$. If linear combinations of the rows of the matrix are nonzero; i.e., if

$$
\begin{equation*}
\mathbf{A}^{T} \boldsymbol{\alpha} \neq 0 \tag{2.31}
\end{equation*}
$$

for all $\boldsymbol{\alpha}=\left[\alpha_{1}, \alpha_{2}, \ldots, \alpha_{m}\right]^{T} \neq 0$, then the rows of $\mathbf{A}$ are said to be linearly independent. Otherwise, if

$$
\begin{equation*}
\mathbf{A}^{T} \boldsymbol{\alpha}=\boldsymbol{0} \tag{2.32}
\end{equation*}
$$

for at least one $\boldsymbol{\alpha} \neq \boldsymbol{0}$, then the rows of $\mathbf{A}$ are said to be linearly dependent and at least one of them can be written as a linear combination of the others.

The row rank (column rank) of a matrix $\mathbf{A}$ is defined as the largest number of linearly independent rows (columns) in the matrix. The row and the column ranks of any matrix are equal. Each of them can thus be called the rank of the matrix. A square matrix with linearly independent rows (columns) is said to have full rank. When a square matrix does not have full rank, it is called singular. For a nonsingular matrix there is an inverse, denoted by $\mathbf{A}^{-1}$, such that

$$
\begin{equation*}
\mathbf{A} \mathbf{A}^{-1}=\mathbf{A}^{-1} \mathbf{A}=\mathbf{I} \tag{2.33}
\end{equation*}
$$

The following identities are valid:

$$
\begin{align*}
\left(\mathbf{A}^{-1}\right)^{T} & =\left(\mathbf{A}^{T}\right)^{-1}  \tag{2.34}\\
(\mathbf{A B})^{-1} & =\mathbf{B}^{-1} \mathbf{A}^{-1} \tag{2.35}
\end{align*}
$$

A special nonsingular matrix that arises often in kinematics is called an orthogonal matrix, with the property that ${ }^{+}$

$$
\begin{equation*}
\mathbf{A}^{-1}=\mathbf{A}^{T} \tag{2.36}
\end{equation*}
$$

Therefore, for an orthogonal matrix,

$$
\begin{equation*}
\mathbf{A}^{T} \mathbf{A}=\mathbf{I} \tag{2.37}
\end{equation*}
$$

Since constructing the inverse of a nonsingular matrix is generally time-consuming, it is important to know when a matrix is orthogonal. In this special case, the inverse is trivially constructed by using Eq. 2.36.

### 2.2.2 Algebraic Vector Operations

The algebraic representation of vectors provides a powerful tool for vector algebra. A reader who is not familiar with this notation and arithmetic may not realize at first its ease of use and flexibility. However, after learning how to operate with algebraic vectors, the reader will find that the traditional geometrical vector operation is rigid and limited for formula manipulation.

An algebraic vector is defined as a column matrix. When an algebraic vector represents a geometric vector in three-dimensional space, the algebraic vector has three components and is called a 3-vector. However, algebraic vectors with more than three components will also be defined and employed in this text.

A 3-vector a was shown in Eq. 2.17 in terms of its $x y z$ components. The components of a vector can be specified in terms of the other coordinate systems besides the $x y z$ coordinate system, such as the $x^{\prime} y^{\prime} z^{\prime}$ or $\xi \eta \zeta$ system. In order not to restrict the following notation to the $x y z$ components of a vector, we show the components of vectors a and $\mathbf{b}$ as

$$
\begin{align*}
& \mathbf{a}=\left[\begin{array}{l}
a_{1} \\
a_{2} \\
a_{3}
\end{array}\right]  \tag{2.38}\\
& \mathbf{b}=\left[\begin{array}{l}
b_{1} \\
b_{2} \\
b_{3}
\end{array}\right] \tag{2.39}
\end{align*}
$$

and thus the vector sum of Eq. 2.1 becomes, in algebraic notation,

$$
\begin{equation*}
\mathbf{c}=\mathbf{a}+\mathbf{b} \tag{2.40}
\end{equation*}
$$

It is also true that $\vec{a}=\vec{b}$ if the components of the vectors are equal; i.e., if $\mathbf{a}=\mathbf{b}$. Multiplication of a vector $\vec{a}$ by a scalar $\alpha$ occurs component by component, so the vector $\alpha \vec{a}$ is described by the column vector $\alpha \mathbf{a}$. A null or zero vector, denoted by 0 , has all of its components equal to zero.

The scalar product of two vectors may be expressed in algebraic form as

$$
\begin{equation*}
\mathbf{a}^{T} \mathbf{b}=\mathbf{b}^{T} \mathbf{a}=a_{1} b_{1}+a_{2} b_{2}+a_{3} b_{3} \tag{2.41}
\end{equation*}
$$

${ }^{\dagger}$ The correct terminology would have been orthonormal instead of orthogonal.

Note that two vectors $\mathbf{a}$ and $\mathbf{b}$ are orthogonal if

$$
\begin{equation*}
\mathbf{a}^{T} \mathbf{b}=0 \tag{2.42}
\end{equation*}
$$

A skew-symmetric matrix associated with a vector a is defined as

$$
\tilde{\mathbf{a}}=\left[\begin{array}{ccc}
0 & -a_{3} & a_{2}  \tag{2.43}\\
a_{3} & 0 & -a_{1} \\
-a_{2} & a_{1} & 0
\end{array}\right]
$$

Note that the tilde placed over a vector indicates that the components of the vector are used to generate a skew-symmetric matrix. Now the vector product $\vec{a} \times \vec{b}$ in Eq. 2.9 can be written in component form as

$$
\mathbf{c}=\overline{\mathbf{a}} \mathbf{b}=\left[\begin{array}{c}
a_{2} b_{3}-a_{3} b_{2}  \tag{2.44}\\
a_{3} b_{1}-a_{1} b_{3} \\
a_{1} b_{2}-a_{2} b_{1}
\end{array}\right]
$$

For later use, it is helpful to develop some standard properties of the tilde operation. First note that

$$
\tilde{\mathbf{a}}^{T}=\left[\begin{array}{ccc}
0 & a_{3} & -a_{2}  \tag{2.45}\\
-a_{3} & 0 & a_{1} \\
a_{2} & -a_{1} & 0
\end{array}\right]
$$

Also, for a scalar $\alpha$,

$$
\begin{equation*}
\alpha \tilde{\mathbf{a}}=(\widetilde{\alpha \mathbf{a}}) \tag{2.46}
\end{equation*}
$$

For any vectors $\mathbf{a}$ and $\mathbf{b}$, a direct calculation shows that

$$
\begin{equation*}
\tilde{\mathbf{a}} \mathbf{b}=-\tilde{\mathbf{b}} \mathbf{a} \tag{2.47}
\end{equation*}
$$

Direct calculation may also be done to show that

$$
\begin{equation*}
\overline{\mathbf{a}} \mathbf{a}=0 \tag{2.48}
\end{equation*}
$$

Hence, by Eq. 2.45,

$$
\begin{equation*}
\mathbf{a}^{T} \tilde{\mathbf{a}}^{T}=-\mathbf{a}^{T} \tilde{\mathbf{a}}=\boldsymbol{0}^{T} \tag{2.49}
\end{equation*}
$$

It can also be verified by direct calculation that

$$
\begin{equation*}
\tilde{\mathbf{a}} \tilde{\mathbf{b}}=\mathbf{b a}^{T}-\mathbf{a}^{T} \mathbf{b} \mathbf{I} \tag{2.50}
\end{equation*}
$$

where $I$ is a $3 \times 3$ identity matrix. Also,

$$
\begin{align*}
(\widetilde{\tilde{\mathbf{a}}}) & =\mathbf{b} \mathbf{a}^{T}-\mathbf{a b}^{T}  \tag{2.51}\\
& =\tilde{\mathbf{a}} \tilde{\mathbf{b}}-\tilde{\mathbf{b}} \tilde{\mathbf{a}}  \tag{2.52}\\
\tilde{\mathbf{a}} \tilde{\mathbf{b}}+\mathbf{a b}^{T} & =\tilde{\mathbf{b}} \mathbf{a}+\mathbf{b a}^{T} \tag{2.53}
\end{align*}
$$

It can also be verified by direct calculation that

$$
\begin{equation*}
(\widetilde{\mathbf{a}+\mathbf{b}})=\tilde{\mathbf{a}}+\tilde{\mathbf{b}} \tag{2.54}
\end{equation*}
$$

Table 2.1 should assist the reader in becoming familiar with the algebraic notation.

## Example 2.1

Test the validity of Eq. 2.47 with two vectors $\mathbf{a}=[-2,1,-3]^{T}$ and $\mathbf{b}=$ $[1,-2,4]^{T}$.

Solution The product ãb is computed as follows:

$$
\text { äb }=\left[\begin{array}{rrr}
0 & 3 & 1 \\
-3 & 0 & 2 \\
-1 & -2 & 0
\end{array}\right]\left[\begin{array}{r}
1 \\
-2 \\
4
\end{array}\right]=\left[\begin{array}{r}
-2 \\
5 \\
3
\end{array}\right]
$$

The product $\tilde{\mathbf{b}} \mathbf{a}$ is computed similarly:

$$
\tilde{\mathbf{b}} \mathbf{a}=\left[\begin{array}{rrr}
0 & -4 & -2 \\
4 & 0 & -1 \\
2 & 1 & 0
\end{array}\right]\left[\begin{array}{r}
-2 \\
1 \\
-3
\end{array}\right]=\left[\begin{array}{r}
2 \\
-5 \\
-3
\end{array}\right]
$$

It can be seen that $\tilde{\mathbf{a}} \mathbf{b}=-$ - $\mathbf{b} \mathbf{a}$.

## Example 2.1a

For vectors $\mathbf{a}$ and $\mathbf{b}$, verify Eq. 2.51 .
Solution The product ãb was found to be $[-2,5,3]^{T}$. Therefore,

$$
\widetilde{\widetilde{\mathbf{a}} \mathbf{b}}=\left[\begin{array}{rrr}
0 & -3 & 5 \\
3 & 0 & 2 \\
-5 & -2 & 0
\end{array}\right]
$$

The right-hand side of Eq. 2.51 is computed as follows:

$$
\begin{aligned}
\mathbf{b a}^{T}-\mathbf{a b}^{T} & =\left[\begin{array}{r}
1 \\
-2 \\
4
\end{array}\right][-2,1,-3]-\left[\begin{array}{r}
-2 \\
1 \\
-3
\end{array}\right][1,-2,4] \\
& =\left[\begin{array}{rrr}
-2 & 1 & -3 \\
4 & -2 & 6 \\
-8 & 4 & -12
\end{array}\right]-\left[\begin{array}{rrr}
-2 & 4 & -8 \\
1 & -2 & 4 \\
-3 & 6 & -12
\end{array}\right] \\
& =\left[\begin{array}{rrr}
0 & -3 & 5 \\
3 & 0 & 2 \\
-5 & -2 & 0
\end{array}\right]
\end{aligned}
$$

which verifies the validity of Eq. 2.51 .

## Example 2.2

Show that

$$
\vec{a} \times(\vec{b} \times \vec{c})+\vec{b} \times(\vec{c} \times \vec{a})+\vec{c} \times(\vec{a} \times \vec{b})=\overrightarrow{0}
$$

Solution Using algebraic vector notation, we write the left-hand side of this expression as

$$
\tilde{\mathbf{a}} \tilde{\mathbf{b}} \mathbf{c}+\tilde{\mathbf{b}} \mathbf{c} \mathbf{a}+\tilde{\mathbf{c}} \mathbf{a} \mathbf{b}
$$

Employing Eq. 2.50 for $\tilde{\mathbf{a}} \tilde{\mathbf{b}}, \tilde{\mathbf{b}} \tilde{\mathbf{c}}$, and $\tilde{\mathbf{c}} \tilde{\mathbf{a}}$, the above terms become

$$
\left(\mathbf{b a} \mathbf{a}^{T}-\mathbf{a}^{T} \mathbf{b I}\right) \mathbf{c}+\left(\mathbf{c} \mathbf{b}^{T}-\mathbf{b}^{T} \mathbf{c I}\right) \mathbf{a}+\left(\mathbf{a c}^{T}-\mathbf{c}^{T} \mathbf{a} \mathbf{I}\right) \mathbf{b}
$$

TABLE 2.1 Vector Terms in Geometric and Algebraic Forms

| Geometric | Algebraic |
| :--- | :--- |
| $\vec{a}$ | $\mathbf{a}$ |
| $\vec{a}+\vec{b}$ | $\mathbf{a}+\mathbf{b}$ |
| $\alpha \vec{a}$ | $\alpha \mathbf{a}$ |
| $\vec{a} \cdot \vec{b}$ | $\mathbf{a}^{T} \mathbf{b}$ |
| $\vec{a} \times \vec{b}$ | $\tilde{\mathbf{a}} \mathbf{b}$ |
| $\vec{a} \cdot(\vec{b} \times \vec{c})$ | $\mathbf{a}^{T} \tilde{\mathbf{b}} \mathbf{c}$ |
| $(\vec{b} \times \vec{c}) \cdot \vec{a}$ | $(\tilde{\mathbf{b}} \mathbf{c})^{T} \mathbf{a}\left(=-\mathbf{c}^{T} \overline{\mathbf{b}} \mathbf{a}\right)$ |
| $\vec{a} \times(\vec{b} \times \vec{c})$ | $\tilde{\mathbf{a}} \mathbf{b}$ |
| $(\vec{a} \times \vec{b}) \times \vec{c}$ | $\widetilde{\tilde{a}} \mathbf{b} \mathbf{c}$ |

or

$$
\mathbf{b a}^{T} \mathbf{c}-\mathbf{a}^{T} \mathbf{b} \mathbf{c}+\mathbf{c b}^{T} \mathbf{a}-\mathbf{b}^{T} \mathbf{c a}+\mathbf{a c}^{T} \mathbf{b}-\mathbf{c}^{T} \mathbf{a} \mathbf{b}
$$

Observe that $\mathbf{b a}^{T} \mathbf{c}=\mathbf{a}^{T} \mathbf{c b}$, since $\mathbf{a}^{T} \mathbf{c}$ is a scalar and can be placed to the left or to the right of vector $\mathbf{b}$. Since $\mathbf{a}^{T} \mathbf{c}=\mathbf{c}^{T} \mathbf{a}$, then $\mathbf{b a}^{T} \mathbf{c}=\mathbf{c}^{T} \mathbf{a b}$. Similarly, it can be shown that $\mathbf{c b}^{T} \mathbf{a}=\mathbf{a}^{T} \mathbf{b c}, \mathbf{a c}^{T} \mathbf{b}=\mathbf{b}^{T} \mathbf{c a}$, and the identity is proved to be zero.

Consider three vectors $\mathbf{a}=\left[a_{1}, a_{2}, a_{3}\right]^{T}, \mathbf{b}=\left[b_{1}, b_{2}, b_{3}\right]^{T}$, and $\mathbf{c}=\left[c_{1}, c_{2}, c_{3}\right]^{T}$. For these vectors, the following representations in matix form will be used in this text:

$$
\begin{align*}
& \mathbf{A}=[\mathbf{a}, \mathbf{b}, \mathbf{c}]=\left[\begin{array}{lll}
a_{1} & b_{1} & c_{1} \\
a_{2} & b_{2} & c_{2} \\
a_{3} & b_{3} & c_{3}
\end{array}\right]  \tag{2.55}\\
& \mathbf{A}^{T}=[\mathbf{a}, \mathbf{b}, \mathbf{c}]^{T}=\left[\begin{array}{lll}
a_{1} & a_{2} & a_{3} \\
b_{1} & b_{2} & b_{3} \\
c_{1} & c_{2} & c_{3}
\end{array}\right] \tag{2.56}
\end{align*}
$$

The algebraic representation of vectors allows one to define vectors with more than three components; i.e., vectors with higher dimension than 3 . A vector with $n$ components is called an $n$-vector. For example, the vector $\mathbf{a}=\left[a_{1}, a_{2}, a_{3}\right]^{T}$ is a 3-vector, and

$$
\begin{equation*}
\mathbf{d}=\left[a_{1}, a_{2}, a_{3}, b_{1}, b_{2}, b_{3}, c_{1}, c_{2}, c_{3}\right]^{T}=\left[\mathbf{a}^{T}, \mathbf{b}^{T}, \mathbf{c}^{T}\right]^{T} \tag{2.57}
\end{equation*}
$$

is a 9 -vector. Note that the right side of Eq. 2.57 is a column vector. In this text, for clarification purposes in particular cases, the dimension of a vector is shown as a superscript; e.g., vector $\mathbf{d}$ for Eq. 2.57 can be shown as $\mathbf{d}^{(9)}$.

A matrix can be represented in terms of its subvectors and submatrices. For example, the $3 \times 4$ matrix $C$ can be represented as

$$
\begin{equation*}
\mathbf{C}=[\mathbf{a}, \mathbf{A}] \tag{2.58}
\end{equation*}
$$

where a is a 3 -vector and $\mathbf{A}$ is a $3 \times 3$ matrix. Vector a represents the elements of the first column of $\mathbf{C}$, and the matrix $\mathbf{A}$ represents the elements in columns 2, 3, and 4 of $\mathbf{C}$.

## Example 2.3

If $\mathbf{C}=[\mathbf{a}, \mathbf{A}]$ and $\mathbf{D}=[\mathbf{b}, \mathbf{B}]$ are two $3 \times 4$ matrices, express $\mathbf{C D}^{T}$ and $\mathbf{C}^{T} \mathbf{D}$ in terms of $\mathbf{a}, \mathbf{A}, \mathbf{b}$, and $\mathbf{B}$.

Solution The product CD $^{T}$ yields a $3 \times 3$ matrix:

$$
\mathbf{C D}^{T}=[\mathbf{a}, \mathbf{A}]\left[\begin{array}{l}
\mathbf{b}^{T} \\
\mathbf{B}^{T}
\end{array}\right]=\mathbf{a b}^{T}+\mathbf{A} \mathbf{B}^{T}
$$

and the product $\mathbf{C}^{T} \mathbf{D}$ yields a $4 \times 4$ matrix:

$$
\mathbf{C}^{T} \mathbf{D}=\left[\begin{array}{l}
\mathbf{a}^{T} \\
\mathbf{A}^{T}
\end{array}\right][\mathbf{b}, \mathbf{B}]=\left[\begin{array}{cc}
\mathbf{a}^{T} \mathbf{b} & \mathbf{a}^{T} \mathbf{B} \\
\mathbf{A}^{T} \mathbf{b} & \mathbf{A}^{T} \mathbf{B}
\end{array}\right]
$$

### 2.3 VECTOR AND MATRIX DIFFERENTIATION

In the kinematics and dynamics of mechanical systems, vectors representing the positions of points on bodies, or equations describing the geometry or the dynamics of the motion, are often functions of time or some other variables. In analyzing these equations, time derivatives or partial derivatives with respect to some variables of the vectors and equations are needed. In this section, these derivatives are defined and the notation used in the text is explained.

### 2.3.1 Time Derivatives

In analyzing velocities and accelerations, time derivatives of vectors that locate points or bodies or equations that describe the geometry of motion must often be calculated. Consider a vector $\mathbf{a} \equiv \mathbf{a}(t)=\left[a_{1}(t), a_{2}(t), a_{3}(t)\right]^{T}$, where $t$ is a scalar parameter that may play the role of time or some other variable. The time derivative of $a$ vector $\mathbf{a}$ is denoted by

$$
\begin{equation*}
\frac{d}{d t} \mathbf{a}(t)=\left[\frac{d}{d t} a_{1}(t), \frac{d}{d t} a_{2}(t), \frac{d}{d t} a_{3}(t)\right]^{T} \equiv \dot{\mathbf{a}} \tag{2.59}
\end{equation*}
$$

Thus, for vectors that are written in terms of their components in a fixed Cartesian coordinate system, the derivative of a vector is obtained by differentiating its components. The derivative of the sum of two vectors gives

$$
\begin{equation*}
\frac{d}{d t}(\mathbf{a}(t)+\mathbf{b}(t))=\dot{\mathbf{a}}+\dot{\mathbf{b}} \tag{2.60}
\end{equation*}
$$

which is completely analogous to the ordinary differentiation rule that the derivative of a sum is the sum of the derivatives. The following vector forms of the product rule of differentiation can also be verified:

$$
\begin{align*}
\frac{d}{d t}(\alpha \mathbf{a}) & =\dot{\alpha} \mathbf{a}+\alpha \dot{\mathbf{a}}  \tag{2.61}\\
\frac{d}{d t}\left(\mathbf{a}^{T} \mathbf{b}\right) & =\dot{\mathbf{a}}^{T} \mathbf{b}+\mathbf{a}^{T} \dot{\mathbf{b}}  \tag{2.62}\\
\frac{d}{d t}(\tilde{\mathbf{a}} \mathbf{b}) & =\dot{\tilde{\mathbf{a}}} \mathbf{b}+\tilde{\mathbf{a}} \dot{\mathbf{b}} \tag{2.63}
\end{align*}
$$

where $\alpha(t)$ is a scalar function of time. Note also that

$$
\begin{equation*}
\dot{\tilde{\mathbf{a}}}=\tilde{\tilde{\mathbf{a}}} \tag{2.64}
\end{equation*}
$$

Many uses may be made of these derivative formulas. For example, if the length of a vector $\mathbf{a}(t)$ is fixed, i.e., if $\mathbf{a}(t)^{T} \mathbf{a}(t)=\mathbf{c}$, then

$$
\begin{equation*}
\dot{\mathbf{a}}^{T} \mathbf{a}=0 \tag{2.65}
\end{equation*}
$$

If $\mathbf{a}$ is a position vector that locates a given point, then $\dot{\mathbf{a}}$ is the velocity of that point. Hence Eq. 2.65 indicates that the velocity is orthogonal to the position vector when the position vector has a constant magnitude.

The second time derivative of a vector $\mathbf{a} \equiv \mathbf{a}(t)$ is denoted as

$$
\begin{equation*}
\frac{d}{d t}\left(\frac{d}{d t} \mathbf{a}(t)\right) \equiv \frac{d}{d t}(\dot{\mathbf{a}}) \equiv \ddot{\mathbf{a}} \tag{2.66}
\end{equation*}
$$

Thus, for vectors that are written in terms of their components in a fixed Cartesian coordinate system, the second time derivative may be calculated in terms of the second time derivatives of the components of the vector.

Just as in the differentiation of a vector function, the derivative of a matrix whose components depend on a variable $t$ may be defined. Consider a matrix $\mathbf{A}(t)=\left[a_{i j}(t)\right]$. The derivative of $\mathbf{A}(t)$ is defined as

$$
\begin{equation*}
\frac{d}{d t} \mathbf{A}(t)=\left[\frac{d}{d t} a_{i j}(t)\right] \equiv \dot{\mathbf{A}} \tag{2.67}
\end{equation*}
$$

With this definition, it can be verified that

$$
\begin{align*}
\frac{d}{d t}(\mathbf{A}(t)+\mathbf{B}(t)) & =\dot{\mathbf{A}}+\dot{\mathbf{B}}  \tag{2.68}\\
\frac{d}{d t}(\mathbf{A}(t) \mathbf{B}(t)) & =\dot{\mathbf{A} \mathbf{B}}+\mathbf{A} \dot{\mathbf{B}}  \tag{2.69}\\
\frac{d}{d t}(\alpha(t) \mathbf{A}(t)) & =\dot{\alpha} \mathbf{A}+\alpha \dot{\mathbf{A}}  \tag{2.70}\\
\frac{d}{d t}(\mathbf{A}(t) \mathbf{a}(t)) & =\dot{\mathbf{A}} \mathbf{a}+\mathbf{A} \dot{\mathbf{a}} \tag{2.71}
\end{align*}
$$

## Example 2.4

If $\mathbf{a}$ is a nonzero time-dependent 3 -vector, $\mathbf{A}=[\mathbf{a}, \tilde{\mathbf{a}}]$ is a $3 \times 4$ matrix, and $\mathbf{C}=\mathbf{A} \mathbf{A}^{T}$, what is the condition on a for which $\dot{\mathbf{C}}$ will be a null matrix?

Solution Matrix $\mathbf{C}$ is found to be

$$
\mathbf{C}=[\mathbf{a}, \tilde{\mathbf{a}}]\left[\begin{array}{c}
\mathbf{a}^{T} \\
-\tilde{\mathbf{a}}
\end{array}\right]=\mathbf{a} \mathbf{a}^{T}-\tilde{\mathbf{a}} \tilde{\mathbf{a}}
$$

The time derivative of $\mathbf{C}$ is

$$
\dot{\mathbf{C}}=\dot{\mathbf{a}} \mathbf{a}^{T}+\mathbf{a} \dot{\mathbf{a}}^{T}-\dot{\overline{\mathbf{a}}} \tilde{\mathbf{a}}-\tilde{\mathbf{a}} \dot{\tilde{a}}=\dot{\mathbf{a}} \mathbf{a}^{T}+\mathbf{\mathbf { a }} \dot{\mathbf{a}}^{T}-\mathbf{a} \dot{\mathbf{a}}^{T}+\dot{\mathbf{a}}^{T} \mathbf{a I}-\dot{\mathbf{a}}^{T}+\mathbf{a}^{T} \dot{\mathbf{a}} \mathbf{I}
$$

where Eq. 2.50 is employed. Since $\dot{\mathbf{a}}^{T} \mathbf{a}=\mathbf{a}^{\mathrm{T}} \mathbf{a}$, after simplification it is found that $\dot{\mathbf{C}}=2 \dot{\mathbf{a}}^{T} \mathbf{a}$. Therefore, $\dot{\mathbf{C}}=\mathbf{0}$ if a has a constant magnitude; i.e., $\dot{\mathbf{a}}^{T} \mathbf{a}=0$.

### 2.3.2 Partial Derivatives

In dealing with systems of nonlinear differential and algebraic equations in many variables, it is essential that a matrix calculus notation be employed. To introduce the nota-
tion used here, let $\mathbf{q}$ be a $k$-vector of real variables and $\Phi$ be a scalar differentiable function of $\mathbf{q}$. Using $j$ as column index, the following notation is defined:

$$
\begin{equation*}
\Phi_{\mathrm{q}} \equiv \frac{\partial \Phi}{\partial \mathbf{q}} \equiv\left[\frac{\partial \Phi}{\partial q_{j}}\right]_{(1 \times k)} \tag{2.72}
\end{equation*}
$$

Equation 2.72 indicates that the partial derivative of a scalar function with respect to a variable vector is a row vector.

## Example 2.5

Vector $\mathbf{q}$ designates four variables as $\mathbf{q}=\left[x_{1}, x_{2}, x_{3}, x_{4}\right]^{T}$. Find the partial derivative of a scalar function $\Phi$ with respect to $\mathbf{q}$ where $\Phi=-x_{1}+3 x_{2} x_{4}^{2}$, ,

Solution Since $\partial \Phi / \partial x_{1}=-1, \partial \Phi / \partial x_{2}=3 x_{4}^{2}, \partial \Phi / \partial x_{3}=0$, and $\partial \Phi / \partial x_{4}=$ $6 x_{2} x_{4}^{2}$; then, using Eq. $2.72, \Phi_{\mathrm{q}}$ is written as follows:

$$
\Phi_{\mathbf{q}}=\left[\begin{array}{llll}
-1 & 3 x_{i 4}^{2} & 0 & 6 x_{2} x_{4}
\end{array}\right]
$$

If $\boldsymbol{\Phi}(\mathbf{q})=\left[\Phi_{1}(\mathbf{q}), \Phi_{2}(\mathbf{q}), \ldots, \Phi_{m}(\mathbf{q})\right]^{T}$ is an $m$-vector of differentiable functions of $\mathbf{q}$, using $i$ as row index and $j$ as column index, the following notation is defined:

$$
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{q}} \equiv \frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{q}} \equiv\left[\frac{\partial \Phi_{i}}{\partial q_{j}}\right]_{(m \times k)} \tag{2.73}
\end{equation*}
$$

Equation 2.73 indicates that the partial derivative of $m$ functions of a $k$-vector of variables with respect to that vector is defined as an $m \times k$ matrix.
Example 2.6
Vector $\mathbf{q}$ containing six variables is given as $\mathbf{q}=\left[x_{1}, y_{1}, x_{2}, y_{2}, x_{3}, y_{3}\right]^{T}$. Determine the partial derivative of two functions $\boldsymbol{\Phi}=\left[\Phi_{1}, \Phi_{2}\right]^{T}$ with respect to $\mathbf{q}$ where

$$
\begin{gathered}
\Phi_{1}=x_{1}+3 y_{1}-x_{2}+2 x_{3}-y_{3} \\
\Phi_{2}=x_{1} y_{1}+y_{2}+2 y_{3}
\end{gathered}
$$

Solution The partial derivative of $\boldsymbol{\Phi}$ with respect to $\boldsymbol{q}$ is a $2 \times 6$ matrix:

$$
\mathbf{\Phi}_{\mathbf{q}}=\left[\begin{array}{rrrrrr}
1 & 3 & -1 & 0 & 2 & -1 \\
y_{1} & x_{1} & 0 & 1 & 0 & 2
\end{array}\right]
$$

Note that the first and the second rows of this matrix contain the partial derivatives of $\Phi_{1}$ and $\Phi_{2}$ with respect to $\mathbf{q}$ respectively.

The partial derivative of the scalar product of two $n$-vector functions $\mathbf{a}(\mathbf{q})$ and $\mathbf{b}(\mathbf{q})$, by careful manipulation, is found to be a row vector:

$$
\begin{equation*}
\frac{\partial}{\partial \mathbf{q}}\left(\mathbf{a}^{T} \mathbf{b}\right)=\mathbf{b}^{T} \mathbf{a}_{\mathbf{q}}+\mathbf{a}^{T} \mathbf{b}_{\mathbf{q}} \tag{2.74}
\end{equation*}
$$

where the dimension of the resultant row vector is the same as the dimension of vector $\mathbf{q}$.

## Example 2.7

Vectors $\mathbf{a}$ and $\mathbf{b}$ are functions of a single variable $\alpha$. Determine the partial derivative of $\mathbf{a}^{T} \mathbf{b}$ with respect to $\alpha$ if

$$
\mathbf{a}=\left[\begin{array}{c}
2 \alpha \\
-1 \\
\alpha
\end{array}\right] \quad \mathbf{b}=\left[\begin{array}{r}
-3 \\
\alpha \\
-\alpha
\end{array}\right]
$$

Solution The derivatives of $\mathbf{a}$ and $\mathbf{b}$ with respect to $\alpha$ are

$$
\mathbf{a}_{\alpha}=\left[\begin{array}{l}
2 \\
0 \\
1
\end{array}\right] \quad \mathbf{b}_{\alpha}=\left[\begin{array}{r}
0 \\
1 \\
-1
\end{array}\right]
$$

Using Eq. 2.74, it is found that

$$
\frac{\partial}{\partial \alpha}\left(\mathbf{a}^{T} \mathbf{b}\right)=[-3, \alpha,-\alpha]\left[\begin{array}{l}
2 \\
0 \\
1
\end{array}\right]+[2 \alpha,-1, \alpha]\left[\begin{array}{r}
0 \\
1 \\
-1
\end{array}\right]=-2 \alpha-7
$$

This result can be obtained directly, in order to verify Eq. 2.74, by determining the scalar product $\mathbf{a}^{T} \mathbf{b}$ :

$$
\mathbf{a}^{T} \mathbf{b}=[2 \alpha,-1, \alpha]\left[\begin{array}{r}
-3 \\
\alpha \\
-\alpha
\end{array}\right]=-\alpha^{2}-7 \alpha
$$

Then the partial derivative of $\mathbf{a}^{T} \mathbf{b}$ with respect to $\alpha$ is found to be $-2 \alpha-7$.

## Example 2.8

Vector $\mathbf{q}$ contains two variables $\alpha$ and $\beta$; i.e., $\mathbf{q}=[\alpha, \beta]^{T}$. Vectors $\mathbf{a}$ and $\mathbf{b}$ are functions of $\mathbf{q}$, as follows:

$$
\mathbf{a}=\left[\begin{array}{c}
\alpha-\beta \\
\alpha+\beta^{2} \\
\beta-1
\end{array}\right] \quad \mathbf{b}=\left[\begin{array}{r}
-\alpha^{2}+\beta \\
\alpha+2 \\
-\alpha-\beta^{2}
\end{array}\right]
$$

Determine the partial derivative of $\mathbf{a}^{T} \mathbf{b}$ with respect to $\mathbf{q}$.
Solution The derivatives of $\mathbf{a}$ and $\mathbf{b}$ with respect to $\mathbf{q}$ are:

$$
\mathbf{a}_{\mathbf{q}}=\left[\begin{array}{rr}
1 & -1 \\
1 & 2 \beta \\
0 & 1
\end{array}\right] \quad \mathbf{b}_{\mathbf{q}}=\left[\begin{array}{rr}
-2 \alpha & 1 \\
1 & 0 \\
-1 & -2 \beta
\end{array}\right]
$$

Using Eq. 2.74, it is found that

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{q}}\left(\mathbf{a}^{T} \mathbf{b}\right)= & {\left[-\alpha^{2}+\beta, \alpha+2,-\alpha-\beta^{2}\right]\left[\begin{array}{cc}
1 & -1 \\
1 & 2 \beta \\
0 & 1
\end{array}\right] } \\
& +\left[\alpha-\beta, \alpha+\beta^{2}, \beta-1\right]\left[\begin{array}{rc}
-2 \alpha & 1 \\
1 & 0 \\
-1 & -2 \beta
\end{array}\right] \\
= & {\left[-\alpha^{2}+\beta+\alpha+2, \alpha^{2}-\beta+2 \alpha \beta+4 \beta-\alpha-\beta^{2}\right] } \\
+ & {\left[-2 \alpha^{2}+2 \alpha \beta+\alpha+\beta^{2}-\beta+1, \alpha-\beta-2 \beta^{2}+2 \beta\right] } \\
= & {\left[-3 \alpha^{2}+2 \alpha \beta+\beta^{2}+2 \alpha+3, \alpha^{2}+2 \alpha \beta-3 \beta^{2}+4 \beta\right] }
\end{aligned}
$$

The partial derivative of the vector product of two $n$-vector functions $\mathbf{a}(\mathbf{q})$ and $\mathbf{b}(\mathbf{q})$ is found to be

$$
\begin{equation*}
\frac{\partial}{\partial \mathbf{q}}(\tilde{\mathbf{a}} \mathbf{b})=\tilde{\mathbf{a}} \mathbf{b}_{\mathbf{q}}-\tilde{\mathbf{b}} \mathbf{a}_{\mathbf{q}} \tag{2.75}
\end{equation*}
$$

The resultant matrix of Eq. 2.75 is an $n \times m$ matrix, where $m$ is the dimension of $\mathbf{q}$.

## Example 2.7a

Evaluate Eq. 2.75 for vectors $\mathbf{a}$ and $\mathbf{b}$.
Solution The derivatives of $\mathbf{a}$ and $\mathbf{b}$ with respect to $\alpha$ are already available; therefore

$$
\frac{\partial}{\partial \alpha}(\tilde{\mathbf{a}} \mathbf{b})=\left[\begin{array}{rrr}
0 & -\alpha & -1 \\
\alpha & 0 & -2 \alpha \\
1 & 2 \alpha & 0
\end{array}\right]\left[\begin{array}{r}
0 \\
1 \\
-1
\end{array}\right]-\left[\begin{array}{rrr}
0 & \alpha & \alpha \\
-\alpha & 0 & 3 \\
-\alpha & -3 & 0
\end{array}\right]\left[\begin{array}{l}
2 \\
0 \\
1
\end{array}\right]=\left[\begin{array}{c}
-2 \alpha+1 \\
-3+4 \alpha \\
4 \alpha
\end{array}\right]
$$

## Example 2.9

Vectors $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$ are functions of vector $\mathbf{q}$. Find $\mathbf{d}_{\mathbf{q}}$ where $\mathbf{d}=\tilde{\mathbf{a}} \tilde{\mathbf{b}} \mathbf{c}$.
Solution In expressions such as ãbe where several functions appear in a nonlinear form, it is helpful to find equivalent forms of the expression. Each equivalent expression should have a different vector appearing at its extreme right. For instance, d can be presented in three forms:

$$
\mathbf{d}=\tilde{\mathbf{a}} \tilde{\mathbf{b}} \mathbf{c}=-\tilde{\mathbf{a}} \tilde{\mathbf{c}} \mathbf{b}=-(\widetilde{\mathbf{b} \mathbf{c}}) \mathbf{a}
$$

where Eq. 2.47 has been employed. Now, these identities easily yield the partial derivative of $\mathbf{d}$ with respect to $\mathbf{q}$ :

$$
\mathbf{d}_{\mathbf{q}}=(\tilde{\mathbf{a}} \tilde{\mathbf{b}}) \mathbf{c}_{\mathbf{q}}-(\tilde{\mathbf{a}} \tilde{\mathbf{c}}) \mathbf{b}_{\mathbf{q}}-(\tilde{\mathbf{b}}) \mathbf{a}_{\mathbf{q}}
$$

This approach can be used to verify Eq. 2.75.

## PROBLEMS

2.1 Let $\vec{a}=\vec{u}_{(x)}+2 \vec{u}_{(y)}-\vec{u}_{(z)}$ and $\vec{b}=2 \vec{u}_{(x)}-\vec{u}_{(y)}+\vec{u}_{(z)}$. Use the algebraic vector approach to calculate the following:
(a) $\vec{a}+\vec{b}$
(b) $\vec{a} \cdot \vec{b}$
(c) $\alpha \vec{a}$
(d) $\vec{a} \times \vec{b}$
(e) $(\vec{a}-\vec{b}) \times \vec{a}$
2.2 If $\vec{a}$ and $\vec{b}$ are arbitrary 3-vectors, verify the following identities by direct calculation:
(a) Eq. 2.50
(b) Eq. 2.51
(c) Eq. 2.52
(d) Eq. 2.53
2.3 If $\vec{a}, \vec{b}, \vec{c}$, and $\vec{d}$ are 3-vectors, use the algebraic vector approach to show that the following identities are valid:
(a) $\vec{a} \cdot(\vec{a} \times \vec{b})=0$
(b) $(\vec{a} \times \vec{b}) \cdot(\vec{c} \times \vec{d})+(\vec{b} \times \vec{c}) \cdot(\vec{a} \times \vec{d})+(\vec{c} \times \vec{a}) \cdot(\vec{b} \times \vec{d})=0$
2.4 Show that if $\mathbf{A}$ is a square matrix, the matrices $\mathbf{B}=\frac{1}{2}\left(\mathbf{A}+\mathbf{A}^{T}\right)$ and $\mathbf{C}=\frac{1}{2}\left(\mathbf{A}-\mathbf{A}^{T}\right)$ are symmetric and skew-symmetric, respectively.
2.5 Show that any square matrix $\mathbf{A}$ can be uniquely expressed as $\mathbf{A}=\mathbf{B}+\mathbf{C}$, where $\mathbf{B}$ and $\mathbf{C}$ are symmetric and skew-symmetric, respectively.
2.6 Show that for an arbitrary angle $\phi$, the matrix $\mathbf{A}=\left[\begin{array}{rr}\cos \phi & -\sin \phi \\ \sin \phi & \cos \phi\end{array}\right]$ is orthogonal.
2.7 Show that for arbitrary angles, $\phi, \psi, \theta$, and $\sigma$, the following matrices are orthogonal $\mathrm{c} \equiv \cos$ and $\mathrm{s} \equiv \sin )$ :
(a) $\mathbf{A}=\left[\begin{array}{ccc}\mathrm{c} \phi & 0 & \mathrm{~s} \phi \\ 0 & 1 & 0 \\ -\mathrm{s} \phi & 0 & \mathrm{c} \phi\end{array}\right]$
(b) $\mathbf{A}=\left[\begin{array}{ccc}c \psi & -\mathrm{s} \psi \mathrm{c} \theta & \mathrm{s} \psi \mathrm{s} \theta \\ \mathrm{s} \psi & \mathrm{c} \psi \mathrm{c} \theta & -\mathrm{c} \psi \mathrm{s} \theta \\ 0 & \mathrm{~s} \theta & \mathrm{c} \theta\end{array}\right]$
(c) $\mathbf{A}=\left[\begin{array}{ccc}\mathrm{c} \psi \mathrm{c} \sigma-\mathrm{s} \psi \mathrm{c} \theta \mathrm{s} \sigma & -\mathrm{c} \psi \mathrm{s} \sigma-\mathrm{s} \psi \mathrm{c} \theta \mathrm{c} \sigma & \mathrm{s} \psi \mathrm{s} \theta \\ \mathrm{s} \psi \mathrm{c} \sigma+\mathrm{c} \psi \mathrm{c} \theta \mathrm{s} \sigma & -\mathrm{s} \psi \mathrm{s} \sigma+\mathrm{c} \psi \mathrm{c} \theta \mathrm{c} \sigma & -\mathrm{c} \psi \mathrm{s} \theta \\ \mathrm{s} \theta \mathrm{s} \sigma & \mathrm{s} \theta \mathrm{c} \sigma & \mathrm{c} \theta\end{array}\right]$
2.8 If $\mathbf{e}$ is a 3-vector and $e_{0}$ is a scalar, show that

$$
\mathbf{A}=\left(2 e_{0}^{2}-1\right) \mathbf{I}+2\left(\mathbf{e e}^{T}+e_{0} \tilde{\mathbf{e}}\right)
$$

is a $3 \times 3$ orthogonal matrix, knowing that $e_{0}^{2}+\mathbf{e}^{T} \mathbf{e}=1$.
2.9 Vector $\mathbf{a}$ is a 3 -vector and $\mathbf{B}$ is a $3 \times 4$ matrix defined as $\mathbf{B}=[\mathbf{a}, \tilde{\mathbf{a}}]$. What is the condition for $\mathbf{A}=\mathbf{B B}^{T}$ to be an orthogonal matrix?
2.10 In Prob. 2.9, show that under no condition can matrix $\mathbf{C}=\mathbf{B}^{T} \mathbf{B}$ be orthogonal.
2.11 Let $\mathbf{B}=[\mathbf{a}, \tilde{\mathbf{a}}]$ and $\mathbf{C}=[\mathbf{a},-\tilde{\mathbf{a}}]$. Show that $\dot{\mathbf{B}} \mathbf{C}^{T}=\mathbf{B} \dot{\mathbf{C}}^{T}$.
2.12 Vector $\boldsymbol{\Phi}$ contains two functions as follows:

$$
\mathbf{\Phi}=\left[\begin{array}{c}
2 x-3 x y+y^{2}-x z+y z^{2}-4 x y z \\
-x^{2}+x y^{2}-2 y+5 y z-x z^{2}
\end{array}\right]
$$

If vector $\mathbf{q}$ is defined as $\mathbf{q}=[x, y, z]^{T}$, find:
(a) $\Phi_{q}$
(b) $\dot{\boldsymbol{\Phi}}$

Show that $\dot{\boldsymbol{\Phi}}=\boldsymbol{\Phi}_{\mathrm{q}} \dot{\mathbf{q}}$.
2.13 Use vectors a, b, and $\mathbf{q}$ from Example 2.8 and evaluate Eq. 2.75.
2.14 For two 3 -vectors $\mathbf{s}$ and $\boldsymbol{\omega}$, vector $\dot{\mathbf{s}}$ is defined as $\dot{\mathbf{s}}=\tilde{\boldsymbol{\omega}} \mathbf{s}$. Show that

$$
\ddot{\mathbf{s}}=-\tilde{\mathbf{s}} \dot{\omega}+\tilde{\boldsymbol{\omega}} \tilde{\omega} \mathbf{s}
$$

2.15 For two 3 -vectors $\mathbf{s}$ and $\boldsymbol{\omega}$ show that

$$
\tilde{\mathbf{s}} \tilde{\omega} \tilde{\mathbf{s}} \omega=\tilde{\omega} \tilde{\mathbf{s}} \tilde{\omega} \omega
$$

2.16 Vectors $\mathbf{a}$ and $\mathbf{b}$ are defined as $\mathbf{a}=\mathbf{A}_{\mathbf{1}} \mathbf{c}_{1}$ and $\mathbf{b}=\mathbf{A}_{2} \mathbf{c}_{2}$, where

$$
\mathbf{A}_{i}=\left[\begin{array}{rr}
\cos \phi_{i} & -\sin \phi_{i} \\
\sin \phi_{i} & \cos \phi_{i}
\end{array}\right] \quad i=1,2 \quad \mathbf{c}_{1}=\left[\begin{array}{r}
1.2 \\
-0.5
\end{array}\right] \quad \mathbf{c}_{2}=\left[\begin{array}{r}
-0.3 \\
0.8
\end{array}\right]
$$

(a) Let $\Phi=\mathbf{a}^{T} \mathbf{b}$ and $\mathbf{q}=\left[x_{1}, y_{1}, \phi_{1}, x_{2}, y_{2}, \phi_{2}\right]^{T}$. Evaluate $\Phi_{\mathbf{q}}$ for $\phi_{1}=30^{\circ}$ and $\phi_{2}=45^{\circ}$.
(b) Let $\mathbf{d}=\left[\begin{array}{l}x_{2}-x_{1} \\ y_{2}-y_{1}\end{array}\right]$ and $\boldsymbol{\Phi}=$ ãd. Evaluate $\boldsymbol{\Phi}_{\mathrm{q}}$ for $\phi_{1}=30^{\circ}, \phi_{2}=45^{\circ}, x_{1}=6.2$,

$$
y_{1}=1.0, x_{2}=-1.9, \text { and } y_{2}=2.3 .
$$

2.17 Let $\mathbf{a}$ and $\mathbf{b}$ be two 3 -vectors, $\mathbf{B}=[\mathbf{b}, \tilde{\mathbf{b}}]$, and $\mathbf{C}=\mathbf{B}^{T} \mathbf{a}$. Find $\mathbf{C}_{\mathbf{a}}$ and $\mathbf{C}_{\mathbf{b}}$.
2.18 Let $\mathbf{x}$ be an $n$-vector of real variables and $\mathbf{A}$ be a real $n \times n$ natrix. Show that

$$
\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^{T} \mathbf{A} \mathbf{x}\right)=\mathbf{x}^{T} \mathbf{A}^{T}+\mathbf{x}^{T} \mathbf{A}=\mathbf{x}^{T}\left(\mathbf{A}^{T}+\mathbf{A}\right)
$$

2.19 If the matrix A in Prob. 2.18 is symmetric, show that

$$
\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^{T} \mathbf{A} \mathbf{x}\right)=2 \mathbf{x}^{T} \mathbf{A}
$$

## Basic Concepts and Numerical Methods

 in KinematicsKinematics, which is the study of the motion of rigid bodies, is useful in two important ways. First, it is frequently necessary to generate, transmit, or control motion by the use of cams, gears, and linkages. An analysis of the displacement, velocity and acceleration of the system is necessary to determine the design geometry of the mechanical parts. Furthermore, as a result of the generated motion, forces are frequently developed that must be accounted for in the design of parts. Second, it is often necessary to determine the motion of a system of rigid bodies that results from applied forces. In both types of problems, one must first have command of the principles of rigid-body kinematics.

Kinematics analysis requires, in general, solution of nonlinear algebraic equations. For small problems with only a few variables and a few equations, it might be possible to write and solve these equations by hand. However, for large problems with many variables and even for accurate analysis of smaller problems, hand calculation, if not impossible, is tedious and unlikely to succeed. Therefore, numerical methods and computer programs are the obvious choice for fast and accurate solution of kinematic equations.

This chapter presents some of the definitions used in kinematics. The general forms of the kinematic equations are presented. Although systematic methods of deriving these equations are not discussed until Chaps. 4, 5, 6, and 7, numerical methods for solving such equations are discussed in this chapter. Several efficient methods for solving linear algebraic equations and nonlinear algebraic equations are reviewed. Algorithms and listings of computer programs for some of these methods are also presented.

### 3.1 DEFINITIONS

A rigid body is defined as a system of particles for which distances between particles remain unchanged. If a particle on such a body is located by a position vector fixed to
the body, the vector never changes its position relative to the body, even when the body is in motion. In reality, all solid materials change shape to some extent when forces are applied to them. Nevertheless, if movement associated with the changes in shape is small compared with the overall movement of the body, then the concept of rigidity is acceptable. For example, displacements due to elastic vibration of the connecting rod of an engine may be of no consequence in the description of engine dynamics as a whole, so the rigid-body assumption is clearly in order. On the other hand, if the problem is one of describing stress in the connecting rod due to vibration, then the deformation of the connecting rod becomes of prime importance and cannot be neglected.

In this text, essentially all analysis is based on the assumption of rigidity. A mechanism is a set of rigid elements that are arranged to produce a specified motion. This definition of a mechanism includes classical linkages, as well as interconnected bodies that make up a vehicle, a vending machine, aircraft landing gear, an engine, and many other mechanical systems. While one can study the kinematics of a deformable body by defining the position of every point in the body in its deformed state, this introduces considerable complexity that is not needed in many applications. This text is concerned with rigid (nondeforming) bodies. The term bodies, therefore, will be used instead of rigid bodies.

Kinematics is the study of motion, quite apart from the forces that produce the motion. More particularly, kinematics is the study of position, velocity, and acceleration in a system of bodies that make up a mechanism.

Kinematic synthesis is the process of finding the geometry of a mechanism that will yield a desired set of motion characteristics. Kinematic analysis, on the other hand, is the process of predicting position, velocity, and acceleration once a mechanism is specified. The processes of kinematic synthesis and kinematic analysis are normally intertwined. In order to do a synthesis, the engineer needs to be able to do an analysis to evaluate designs under consideration. Thus, kinematic analysis may be viewed as a tool that is needed to support the kinematic synthesis process.

The individual bodies that collectively form a mechanism are said to be links. The combination of two links in contact constitutes a kinematic pair, or joint. An assemblage of interconnected links is called a kinematic chain. A mechanism is formed when at least one of the links of the kinematic chain is held fixed and any of its other links can move. The fixed link(s) is (are) called the ground or frame.

If all the links of a mechanism move in a plane or in parallel planes, the mechanism is called a planar mechanism. If some links undergo motion in three-dimensional space, the mechanism is called a spatial mechanism.

A mechanism that is formed from a collection of links or bodies that are kinematically connected to one another but for which it is not possible to move to successive links across kinematic joints and return to the starting link is called an open-loop or open-chain mechanism. An open-loop mechanism may contain links with single joints. An example of this kind of mechanism is the double pendulum shown in Fig. 3.1(a). A closed-loop mechanism is formed from a closed chain, wherein each link is connected to at least two other links of the mechanism and it is possible to traverse a closed loop. Figure 3.1(b) shows a four-bar linkage, which is a closed-loop mechanism. Kinematic analysis considers systems containing only closed loops.


Figure 3.1 (a) Open-loop mechanism double pendulum, (b) Closed-loop mecha-nism-four-bar linkage.

A closed-loop mechanism may contain one or more loops (or closed paths) in its kinematic structure. If the number of loops in a closed-loop mechanism is 1 , then the mechanism is called a single-loop mechanism. If the closed-loop mechanism contains more than one loop, then the mechanism is called a multiloop mechanism. Figure 3.2(a) is an example of a single-loop mechanism, and Fig. 3.2(b) shows a multiloop mechanism.

### 3.1.1 Classification of Kinematic Pairs

Mechanisms and kinematic pairs can be classified in a number of different ways. One method is purely descriptive; e.g., gear pairs, cams, sliding pairs, and so on. Such a division is convenient, and some of these pairs will be studied in Chaps. 4 and 7 under such headings. However, a broader view of the grouping of kinematic pairs is presented here.

Kinematic pairs may be classified generally into two groups. ${ }^{17}$ Joints with surface contact are referred to as lower pairs, and those with point or line contact are referred to as higher pairs. Figure 3.3 gives a number of examples of kinematic pairs. The pairs (a), (b), (e), and (f) in Fig. 3.3 are examples of lower-pair joints, and pairs (c) and (d) are examples of higher-pair joints.

The constraint formulation for most lower-pair joints is generally simpler to derive than that for higher-pair joints.

Relative motion between two bodies of a kinematic pair may be planar or spatial. For example, pairs (a), (b), (c), and (d) in Fig. 3.3 display relative motion between bodies in a manner that can be considered either for planar or spatial kinematic analysis. In contrast, pairs (e) and (f) can be studied only in a spatial kinematic sense.

(a)

(b)

Figure 3.2 (a) Single-loop mechanism, (b) Multiloop mechanism.


Figure 3.3 Examples of kinematic pairs: (a) revolute joint, (b) translational joint, (c) gear set, (d) cam follower, (e) screw joint, (f) spherical (ball) joint.

### 3.1.2 Vector of Coordinates

Any set of parameters that uniquely specifies the position (configuration) of all bodies of a mechanism is called a set of coordinates. For systems in motion, these parameters vary with time. The term coordinates can refer to any of the commonly used coordinate systems, but it can also refer to any of an infinite variety of other sets of parameters that serve to specify the configuration of a system. Vectors of coordinates are designated in this text by column vectors $\mathbf{q} \equiv\left[q_{1}, q_{2}, \ldots, q_{n}\right]^{T}$, where $n$ is the total number of coordinates used in describing the system. Examples of commonly used coordinates are Lagrangian and Cartesian coordinates. In this text, Cartesian coordinates are used almost exclusively. The general distinction between the Lagrangian and Cartesian coordinate systems is that the former allows definition of the position of a body relative to a moving coordinate system, whereas the latter normally requires that the position of each body in space be defined relative to a fixed global coordinate system. Thus the Cartesian coordinate system requires that a large number of coordinates be defined to specify the position of each body of the system.

In order to specify the configuration of a planar system, a body-fixed $\xi \eta$ coordinate system is embedded in each body of the system, as shown in Fig. 3.4(a). Body $i$ ( $i$ is an identifying number assigned to each body) can be located by specifying the global


Figure 3.4 Global and body-fixed coordinate systems: (a) planar motion, (b) spatial motion.
translational coordinates $\mathbf{r}_{i}=[x, y]_{i}^{T}$ of the origin of the body-fixed $\xi_{i} \eta_{i}$ reference system and the angle $\phi_{i}$ of rotation of this system relative to the global $x y$ axes. The column vector $\mathbf{q}_{i} \equiv \equiv[x, y, \phi]_{i}^{T}$ is the vector of coordinates for body $i$ in a plane. ${ }^{\dagger}$

For spatial systems, six coordinates are required to define the configuration of each body; e.g., body $i$ shown in Fig. 3.4(b). The three components of the vector $\vec{r}_{i}$ i.e., the global translational coordinates $\mathbf{r}_{i} \equiv[x, y, z]_{i}^{T}$ - locate the origin of the bodyfixed $\xi_{i} \eta_{i} \zeta_{i}$ reference system relative to the global $x y z$ axes, and the three rotational coordinates $\phi_{1 i}, \phi_{2 i}$, and $\phi_{3 i}$ specify the angular orientation of the body. Therefore, column vector $\mathbf{q}_{i} \equiv\left[x, y, z, \phi_{1}, \phi_{2}, \phi_{3}\right]_{i}^{T}$ is the vector of coordinates for body $i$ in threedimensional space.

The concept of angular orientation of a body in three-dimensional space is discussed in detail in Chap. 6. It will be shown that instead of three rotational coordinates, four rotational coordinates with one equation relating these four coordinates can be used to avoid singularity problems. In this case, the coordinates for body $i$ become $\mathbf{q}_{i} \equiv$ $\left[x, y, z, e_{0}, e_{1}, e_{2}, e_{3}\right]_{i}^{T}$. The advantage of presenting the angular orientation of a body with four coordinates instead of three is also discussed in Chap. 6.

If a mechanism with $b$ bodies is considered, the number of coordinates required is $n=3 \times b$ if the system is planar, and $n=6 \times b$ (or $7 \times b$ ) if the system is spatial. The overall vector of coordinates for the system is denoted by $\mathbf{q}=\left[\mathbf{q}_{1}^{T}, \mathbf{q}_{2}^{T}, \ldots, \mathbf{q}_{b}^{T}\right]^{T}$. Since bodies making up a mechanism are interconnected by joints, all of the coordinates are not independent - there are equations of constraint relating the coordinates.

Lagrangian coordinates, unlike Cartesian coordinates, do not necessarily assign the same number of coordinates to each body of the system. Some of the coordinates may be measured relative to a global coordinate system while others are measured relative to moving coordinate systems. An example of a set of Lagrangian coordinates is shown in Fig. 3.5. The variables $\phi_{1}, \phi_{2}$, and $d$ define the configuration of the slidercrank mechanism at every instant. The vector of coordinates for the system can thus be
${ }^{\dagger}$ For notational simplification, the body index is moved outside the bracket; e.g., $\mathbf{q}_{i}=[x, y, \phi]_{i}^{T}$ is used instead of $\mathbf{q}_{i}=\left[x_{i}, y_{i}, \boldsymbol{\phi}_{i}\right]^{T}$.


Figure 3.5 Slider-crank mechanism with Lagrangian coordinates.
defined as $\mathbf{q} \equiv\left[\phi_{1}, \boldsymbol{\phi}_{2}, d\right]^{T}$. It must also be noted that these coordinates are not independent. If the lengths of links 1 and 2 are given as $l_{1}$ and $l_{2}$, once $\phi_{1}$ is specified, $\phi_{2}$ and $d$ may be defined by simple trigonometry.

When a system is in motion, some or all of the coordinates describing the configuration of the system vary with time. In this text $t$ denotes time and is considered to be an independent variable. In kinematics and dynamics, the motion of a body or a mechanism is analyzed for an interval of time from $t^{6}$ (initial time) to $t^{c}$ (final time). In most problems, it is convenient to let $t^{0}=0$.

### 3.1.3 Degrees of Freedom

The minimum number of coordinates required to fully describe the configuration of a system is called the number of degrees of freedom (DOF) of the system. Consider the triple pendulum shown in Fig. 3.6. Here, no fewer than three angles, $\phi_{1}, \phi_{2}$, and $\phi_{3}$, can uniquely determine the configuration of the system. Therefore, the triple pendulum has 3 degrees of freedom. Similarly, for the four-bar mechanism shown in Fig. 3.7, three variables, $\phi_{1}, \phi_{2}$, and $\phi_{3}$, define the configuration of the system. However, the angles are not independent. There exist two algebraic constraint equations,

$$
\begin{array}{r}
l_{1} \cos \phi_{1}+l_{2} \cos \phi_{2}-l_{3} \cos \phi_{3}-d_{1}=0 \\
l_{1} \sin \phi_{1}+l_{2} \sin \phi_{2}-l_{3} \sin \phi_{3}-d_{2}=0 \tag{3.1}
\end{array}
$$

which define loop closure of the mechanism. The two equations can be solved for $\phi_{2}$ and $\phi_{3}$ as a function of $\phi_{1}$. Therefore, $\phi_{1}$ is the only variable needed to define the configuration of the system, and so there is only 1 degree of freedom for the four-bar mechanism.


Figure 3.6
Triple pendulum.


Figure 3.7 Four-bar mechanism.

In a mechanical system, if $k$ is the number of degrees of freedom of the system, then $k$ independent coordinates are required to completely describe the system. These $k$ quantities need not all have the dimensions of length. Depending on the problem at hand, it may be convenient to choose some coordinates with dimensions of length and some that are dimensionless, such as angles or direction cosines. Any set of coordinates that are independent and are equal in number to the number of degrees of freedom of the system is called a set of independent coordinates. Any remaining coordinates, which may be determined as a function of the independent coordinates, are called dependent coordinates.

### 3.1.4 Constraint Equations

A kinematic pair imposes certain conditions on the relative motion between the two bodies it comprises. When these conditions are expressed in analytical form, they are called equations of constraint. Since in a kinematic pair the motion of one body fully or partially defines the motion of the other, it becomes clear that the number of degrees of freedom of a kinematic pair is less than the total number of degrees of freedom of two free rigid bodies. Therefore, a constraint is any condition that reduces the number of degrees of freedom in a system.

A constraint equation describing a condition on the vector of coordinates of a system can be expressed as follows:

$$
\begin{equation*}
\Phi \equiv \Phi(\mathbf{q})=0^{-} \tag{3.2}
\end{equation*}
$$

In some constraint equations, the variable time may appear explicitly;

$$
\begin{equation*}
\Phi \equiv \Phi(\mathbf{q}, t)=0 \tag{3.3}
\end{equation*}
$$

mom

For example, Eq. 3.1 describes two constraint equations for the four-bar mechanism of Fig. 3.7, which has a vector of coordinates $\mathbf{q}=\left[\phi_{1}, \phi_{2}, \phi_{3}\right]^{T}$. These equations are of the form stated by Eq. 3.2.

Algebraic equality constraints in terms of the coordinates, and perhaps time, are said to be holonomic constraints. In general, if constraint equations contain inequalities or relations between velocity components that are not integrable in closed form, they are said to be nonholonomic constraints. In this text, for brevity, the term constraint will refer to a holonomic constraint, unless specified otherwise.

### 3.1.5 Redundant Constraints

A brief study of a mechanism is essential prior to actual kinematic or dynamic analysis. Knowledge of the number of degrees of freedom of the mechanism can be useful when constraint equations are being formulated. Often, the pictorial description of a mechanism can be misleading. Several joints may restrict the same degree of freedom and may therefore be equivalent or redundant.

As an example, consider the double parallel-crank mechanism shown in Fig. 3.8(a). This system has 1 degree of freedom. If this system is modeled for kinematic analysis as four moving bodies and six revolute joints, the set of constraint equations will contain redundant equations. The reason for redundancy becomes clear when one of the coupler links is removed to obtain the mechanism shown in Fig. 3.8(b). The two mechanisms are kinematically equivalent.


Figure 3.8 (a) A double parallel-crank mechanism, and (b) its kinematically equivalent system.

For a system having $m$ independent constraint equations and $n$ coordinates, the number of degrees of freedom is determined as follows:

$$
\begin{equation*}
k=n-m \tag{3.4}
\end{equation*}
$$

In planar motion, a moving body can have three coordinates, and a revolute joint introduces two constraint equations. For the mechanism of Fig. 3.8(b), there are three moving bodies $(n=3 \times 3=9)$ and four revolute joints ( $m=4 \times 2=8$ ). Therefore, $k=9-8=1$ DOF. However, for the mechanism of Fig 3.8(a), $n=4 \times 3=12$ and $m=4 \times 3=12$, which yields $k=12-12=0 \mathrm{DOF}$, which is obviously incorrect. Therefore, Eq. 3.4 yields a correct answer only when the $m$ constraint equations are independent.

## Example 3.1

Five coordinates $\mathbf{q}=\left[l_{1}, \phi_{1}, l_{2}, \phi_{2}, l_{3}\right]^{T}$ are used to describe the configuration of bodies in a mechanism. Determine the number of degrees of freedom of the system if the coordinates are dependent according to the following six constraint equations:

$$
\begin{aligned}
& \Phi_{1}=6 \cos \phi_{1}-l_{2}=0 \\
& \Phi_{2}=6 \sin \phi_{1}-l_{3}=0 \\
& \Phi_{3}=l_{1} \cos \phi_{1}-2 \cos \phi_{2}=0 \\
& \Phi_{4}=l_{1} \sin \phi_{1}-2 \sin \phi_{2}-l_{3}+3=0 \\
& \Phi_{5}=2 \cos \phi_{2}+\left(6-l_{1}\right) \cos \phi_{1}-l_{2}=0 \\
& \Phi_{6}=2 \sin \phi_{2}+\left(6-l_{1}\right) \sin \phi_{1}-3=0
\end{aligned}
$$

Solution An investigation of the six equations reveals that $\Phi_{2}=\Phi_{4}+\Phi_{6}$ and $\Phi_{1}=\Phi_{3}+\Phi_{5}{ }^{\dagger}$ Therefore, two of the equations are redundant, and hence, $m=$ $6-2=4$. Since $n=5$, then $k=5-4=1 \mathrm{DOF}$.

### 3.2 KINEMATIC ANALYSIS

Kinematics is the study of the position, velocity, and acceleration of mechanisms. In kinematic analysis, only constraint equations are considered. The first and second time derivatives of the constraint equations yield the kinematic velocity and acceleration equations.

[^3]For position analysis, at any given instant, the value of $k$ coordinates must be known (where $k$ is equal to the number of degrees of freedom). Hence, the constraint equations can be solved for the other $m=n-k$ coordinates. Similarly, for velocity and acceleration analysis, the value of $k$ velocities and $k$ accelerations must be known in order to solve the kinematic velocity and acceleration equations for the other, unknown velocities and accelerations.

The process of kinematic analysis is presented in two slightly different forms in the next two sections. Each method has a computational advantage and disadvantage in relation to the other.

### 3.2.1 Coordinate Partitioning Method

The fundamentals of kinematic analysis with the coordinate partitioning method can be best understood by following the process in a simple example.

## Example 3.2

The four-bar mechanism of Fig. 3.7 is considered for kinematic analysis and is shown again in Fig. 3.9. All of the lengths are known, and it is given that the crank is rotating counterclockwise (CCW) with a constant angular velocity of $2 \pi \mathrm{rad} / \mathrm{s}$ from the initial orientation of $\phi_{1}^{0}=2.36 \mathrm{rad}$. The constraint equations of Eq. 3.1 are written as follows:

$$
\begin{array}{r}
0.2 \cos \phi_{1}+0.4 \cos \phi_{2}-0.3 \cos \phi_{3}-0.35=0 \\
0.2 \sin \phi_{1}+0.4 \sin \phi_{2}-0.3 \sin \phi_{3}-0.1=0 \tag{1}
\end{array}
$$

For position analysis the substitution $\phi_{1}=2.36$ is made in Eq. 1 to get

$$
\begin{align*}
0.4 \cos \phi_{2}-0.3 \cos \phi_{3} & =0.49 \\
0.4 \sin \phi_{2}-0.3 \sin \phi_{3} & =-0.04 \tag{2}
\end{align*}
$$

These equations are solved to find $\phi_{2}=0.57 \mathrm{rad}$ and $\phi_{3}=2.11 \mathrm{rad}$. For velocity analysis, the first time derivative of Eq. 1 is written as

$$
\begin{align*}
-0.2 \sin \phi_{1} \dot{\phi}_{1}-0.4 \sin \phi_{2} \dot{\phi}_{2}+0.3 \sin \phi_{3} \dot{\phi}_{3} & =0  \tag{3}\\
0.2 \cos \phi_{1} \dot{\phi}_{1}+0.4 \cos \phi_{2} \dot{\phi}_{2}-0.3 \cos \phi_{3} \dot{\phi}_{3} & =0
\end{align*}
$$

For $\phi_{1}=2.36, \phi_{2}=0.57, \phi_{3}=2.11$, and known angular velocity of the crank, i.e., $\phi_{1}=6.28 \mathrm{rad} / \mathrm{s}$, Eq. 3 becomes

$$
\begin{align*}
-0.22 \dot{\phi}_{2}+0.26 \dot{\phi}_{3} & =0.89  \tag{4}\\
0.34 \dot{\phi}_{2}+0.16 \dot{\phi}_{3} & =0.89
\end{align*}
$$

The solution of Eq. 4 yields $\dot{\phi}_{2}=0.76 \mathrm{rad} / \mathrm{s}$ and $\dot{\phi}_{3}=4.09 \mathrm{rad} / \mathrm{s}$. Similarly, for acceleration analysis, the time derivatives of Eq. 3 is

$$
\begin{array}{r}
-0.2 \sin \phi_{1} \ddot{\phi}_{1}-0.2 \cos \phi_{1} \dot{\phi}_{1}^{2}-0.4 \sin \phi_{2} \ddot{\phi}_{2}-0.4 \cos \phi_{2} \dot{\phi}_{2}^{2}+ \\
0.3 \sin \phi_{3} \ddot{\phi}_{3}+0.3 \cos \phi_{3} \dot{\phi}_{3}^{2}=0
\end{array}
$$

$$
\begin{align*}
& 0.2 \cos \phi_{1} \ddot{\phi}_{1}-0.2 \sin \phi_{1} \dot{\phi}_{1}^{2}+0.4 \cos \phi_{2} \ddot{\phi}_{2}-0.4 \sin \phi_{2} \dot{\phi}_{2}^{2}-  \tag{5}\\
& 0.3 \cos \phi_{3} \dot{\phi}_{3}+0.3 \sin \phi_{3} \dot{\phi}_{3}^{2}=0
\end{align*}
$$

Since $\phi_{1}, \phi_{2}, \phi_{3}, \dot{\phi}_{1}, \dot{\phi}_{2}, \dot{\phi}_{3}$ are known and $\ddot{\phi}_{1}=0$ (indicating constant angular velocity), Eq. 5 becomes

$$
\begin{align*}
-0.22 \ddot{\phi}_{2}+0.26 \ddot{\phi}_{3} & =-2.86 \\
0.34 \ddot{\phi}_{2}+0.16 \ddot{\phi}_{3} & =1.39 \tag{6}
\end{align*}
$$

This yields $\ddot{\phi}_{2}=6.62 \mathrm{rad} / \mathrm{s}^{2}$ and $\ddot{\phi}_{3}=-5.39 \mathrm{rad} / \mathrm{s}^{2}$.
The process of position, velocity, and acceleration analysis can be repeated for different positions of the crank. If $\phi_{1}$ is varied from its initial value through a complete revolution of the crank, then at every step the position, velocity, and acceleration analysis yield the results shown in the following table:

| $\phi_{1}$ | $\phi_{2}$ | $\phi_{3}$ | $\dot{\phi}_{2}$ | $\dot{\phi}_{3}$ | $\ddot{\phi}_{2}$ | $\ddot{\phi}_{3}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2.36 | 0.57 | 2.11 | 0.76 | 4.09 | 6.62 | $-5.39$ |
| 2.52 | 0.59 | 2.21 | 0.94 | 3.93 | 7.21 | -7.17 |
| 2.67 | 0.62 | 2.31 | 1.13 | 3.73 | 7.91 | -8.97 |
| 2.83 | 0.65 | 2.40 | 1.33 | 3.48 | 8.66 | -10.74 |
| . | . |  | . | . | . | . |
| - | - | . | . | . | - | . |
| - | . | , | - | - | - | - |
| 8.49 | 0.55 | 2.01 | 0.60 | 4.20 | 6.21 | -3.61 |
| 8.64 | 0.57 | 2.11 | 0.76 | 4.09 | 6.62 | -5.39 |

$\dot{\phi}_{1}=6.28$ and $\ddot{\phi}_{I}=0$.

The result of the position analysis for one complete revolution of the crank is shown in Fig. 3.9.

In the preceding example, the angle $\phi_{1}$, which has a known value at every instant, is called the independent coordinate or the driving coordinate. The remaining coordinates, such as $\phi_{2}$ and $\phi_{3}$, are called the dependent coordinates or the driven coordinates. The number of independent coordinates is equal to the number of degrees of freedom of the system; therefore the number of dependent coordinates is equal to the number of independent constraint equations in the system.

Kinematic analysis with coordinate partitioning considers the partitioned form of the coordinate vector $\mathbf{q}=\left[\mathbf{u}^{T}, \mathbf{v}^{T}\right]^{T}$, where $\mathbf{u}$ and $\mathbf{v}$ are the dependent and independent coordinates, respectively. The $m$ constraint equations

$$
\begin{equation*}
\Phi \equiv \Phi(\mathbf{q})=0 \tag{3.5}
\end{equation*}
$$

may be expressed as

$$
\begin{equation*}
\Phi \equiv \Phi(\mathbf{u}, \mathbf{v})=0 \tag{3.6}
\end{equation*}
$$

Constraint equations as presented by Eq. 3.6 are, in general, nonlinear. For position analysis, iterative numerical methods may be used to solve the set of nonlinear algebraic equations. One such method is discussed in Section 3.4. The $k$ independent coordinates $\mathbf{v}$ are specified at each instant of time $t$. Then, Eq. 3.6 becomes a set of $m$ equations in


Figure 3.9 The result of the position analysis for the four-bar mechanism of Figure 3.7.
$m$ unknowns, which may be solved for the $m$ dependent coordinates $\mathbf{u}$. If the constraints of Eq. 3.5 are independent, then the existence of a solution to $\mathbf{u}$ for a given $\mathbf{v}$ is asserted by the implicit function theorem ${ }^{\dagger}$ of calculus.

Differentiation of Eq. 3.5 yields velocity equations

$$
\begin{equation*}
\mathbf{\Phi}_{\mathbf{q}} \dot{\mathbf{q}}=0 \tag{3.7}
\end{equation*}
$$

where $\dot{\mathbf{q}}=\left[\dot{q}_{1}, \dot{q}_{2}, \ldots, \dot{q}_{n}\right]^{T}$ is the vector of velocities. The matrix $\mathbf{\Phi}_{\mathbf{q}} \equiv[\partial \boldsymbol{\Phi} / \partial \mathbf{q}]$, which contains partial derivatives of the constraint equations with respect to the coordinates, is called the constraint Jacobian matrix. Let $\dot{\mathbf{v}}=\left[\dot{v}_{1}, \dot{v}_{2}, \ldots, \dot{v}_{k}\right]^{T}$ represent the independent velocities with known values, and let $\dot{\mathbf{u}}=\left[\dot{u}_{1}, \dot{u}_{2}, \ldots, \dot{u}_{m}\right]^{T}$ represent the $m$ dependent velocities. Equation 3.7 may be rewritten in partitioned form as

$$
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{u}} \dot{\mathbf{u}}=-\boldsymbol{\Phi}_{\mathbf{y}} \dot{\mathbf{v}} \tag{3.8}
\end{equation*}
$$

${ }^{\dagger}$ Implicit Function Theorem: ${ }^{3}$ Consider a point $\mathbf{q}^{i}=\left[\mathbf{u}^{i}, \mathbf{v}^{i}\right]^{T}$ at time $t^{i}$ for which the constraint equations are satisfied; i.e., for which

$$
\begin{equation*}
\Phi\left(\mathbf{u}^{i}, v^{i}\right)=0 \tag{a}
\end{equation*}
$$

If the partitioning of $\mathbf{q}$ into $\mathbf{u}$ and $\mathbf{v}$ has been selected so that the matrix $\Phi_{\mathbf{u}} \equiv[\partial \Phi / \partial \mathbf{u}]$ at $\left(\mathbf{u}^{i}, \mathbf{v}^{i}\right)$ is nonsingular, then in some neighborhood of $\mathbf{v}^{i}$ Eq. $a$ has a unique solution $\mathbf{u}=\boldsymbol{\Psi}(\mathbf{v})$; i.e., $\boldsymbol{\Phi}(\boldsymbol{\Psi}(\mathbf{v}), \mathbf{v})=\mathbf{0}$. Furthermore, if $\boldsymbol{\Phi}(\mathbf{u}, \mathbf{v})$ is $j$ times continuously differentiable in its arguments, so is $\boldsymbol{\Psi}(\mathbf{v})$.
where $\Phi_{u}$ and $\boldsymbol{\Phi}_{\mathrm{v}}$ are two submatrices of $\boldsymbol{\Phi}_{\mathbf{q}}$ that contain the columns of $\boldsymbol{\Phi}_{\mathbf{q}}$ associated with $\mathbf{u}$ and $\mathbf{v}$, respectively. The term on the right side of Eq. 3.8 is denoted by

$$
\begin{equation*}
\nu=-\boldsymbol{\Phi}_{\mathbf{v}} \dot{\mathbf{v}} \tag{3.9}
\end{equation*}
$$

Since the constraint equations of Eq. 3.5 are assumed to be independent, then $\boldsymbol{\Phi}_{\mathrm{u}}$ is an $m \times m$ nonsingular matrix, and so Eq. 3.8 may be solved directly for $\dot{\mathbf{u}}$, once $\dot{\mathbf{v}}$ is given.

Differentiating the velocity equations of Eq. 3.7 yields the acceleration equations

$$
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{q}} \ddot{\mathbf{q}}+\left(\boldsymbol{\Phi}_{\mathbf{q}} \dot{\mathbf{q}}\right)_{\mathbf{q}} \dot{\mathbf{q}}=\mathbf{0} \tag{3.10}
\end{equation*}
$$

where $\ddot{\mathbf{q}}=\left[\ddot{q}_{1}, \ddot{q}_{2}, \ldots, \ddot{q}_{n}\right]^{T}$ is the vector of accelerations. Let $\ddot{\mathbf{v}}=\left[\ddot{v}_{1}, \ddot{v}_{2}, \ldots, \ddot{v}_{k}\right]^{T}$ represent the independent accelerations, and $\ddot{\mathbf{u}}=\left[\ddot{u}_{1}, \ddot{u}_{2}, \ldots, \ddot{u}_{m}\right]^{T}$ represent the dependent accelerations. Equation 3.10 can be written in partitioned form as

$$
\begin{equation*}
\boldsymbol{\Phi}_{\mathrm{u}} \ddot{\mathbf{u}}=-\boldsymbol{\Phi}_{\mathrm{v}} \ddot{\mathbf{v}}-\left(\boldsymbol{\Phi}_{\mathrm{q}} \dot{\mathbf{q}}\right)_{\mathrm{q}} \dot{\mathbf{q}} \tag{3.11}
\end{equation*}
$$

Since $\boldsymbol{\Phi}_{\mathbf{u}}$ is nonsingular, Eq. 3.11 can be solved for $\ddot{\mathbf{u}}$, once $\ddot{\mathbf{v}}$ is given. Note that the velocity and acceleration equations of Eqs. 3.8 and 3.11 are sets of linear algebraic equations in $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, respectively, whereas the constraint equations of Eq. 3.6 are nonlinear algebraic equations.

The general procedure for kinematic analysis, using the coordinate partitioning method, may be summarized in the following algorithm:

## ALGORITHM K-I

(a) Set a time step counter $i$ to $i=0$ and initialize $t^{i}=t^{0}$ (initial time).
(b) Partition $\mathbf{q}$ into dependent and independent sets $\mathbf{u}$ and $\mathbf{v}$.
(c) Specify independent coordinates $\mathbf{v}^{i}$ and solve Eq. 3.6 iteratively for $\mathbf{u}^{i}$.
(d) Specify independent velocities $\dot{\mathbf{v}}^{i}$ and solve Eq. 3.8 for $\dot{\mathbf{u}}^{i}$.
(e) Specify independent accelerations $\ddot{\mathbf{v}}^{i}$ and solve Eq. 3.11 for $\ddot{\mathbf{u}}^{i}$.
(f) If the final time has been reached, then terminate; otherwise increment $t^{\prime}$ to a new time $t^{i+1}$, let $i \rightarrow i+1$, and go to (c).

The simple form of the constraint equations of Example 3.2 may give the impression that the constraint equations can always be explicitly partitioned into terms containing the independent coordinates and terms containing the dependent coordinates. In general, this partitioning is not possible for highly nonlinear equations. However, regardless of the order of nonlinearity of the constraint equations, the velocity and acceleration equations can be partitioned according to Eqs. 3.8 and 3.11, since they are linear in terms of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, respectively.

## Example 3.3

Derive the velocity and acceleration equations for the constraint equations

$$
\begin{array}{r}
2 \cos \phi_{1}+3 \cos \left(\phi_{1}-\phi_{2}\right)-2 \cos \left(\phi_{3}+\phi_{4}\right)-4 \cos \phi_{4}-5=0 \\
2 \sin \phi_{1}+3 \sin \left(\phi_{1}-\phi_{2}\right)-2 \sin \left(\phi_{3}+\phi_{4}\right)-4 \sin \phi_{4}-1=0 \tag{1}
\end{array}
$$

Then express these equations in partitioned form if $\phi_{1}$ and $\phi_{3}$ are assumed to be the independent coordinates.

Solution The vector of coordinates is $\mathbf{q}=\left[\phi_{1}, \phi_{2}, \phi_{3}, \phi_{4}\right]^{T}$, thus $\mathbf{v}=\left[\phi_{1}, \phi_{3}\right]^{T}$ and $\mathbf{u}=\left[\phi_{2}, \phi_{4}\right]^{\top}$. Since the constraint equations are nonlinear, they cannot be
partitioned explicitly in terms of $\mathbf{u}$ and $\mathbf{v}$, and therefore they are left as they are in Eq. 1 .

The kinematic velocity equations can be found either by direct differentiation of Eq. 1 or by using Eq. 3.7, as follows:

$$
\left[\begin{array}{cc}
-2 \sin \phi_{1}-3 \sin \left(\phi_{1}-\phi_{2}\right) & 3 \sin \left(\phi_{1}-\phi_{2}\right) \\
2 \cos \phi_{1}+3 \cos \left(\phi_{1}-\phi_{2}\right) & -3 \cos \left(\phi_{1}-\phi_{2}\right)  \tag{2}\\
2 \sin \left(\phi_{3}+\phi_{4}\right) & 2 \sin \left(\phi_{3}+\phi_{4}\right)+4 \sin \phi_{4} \\
-2 \cos \left(\phi_{3}+\phi_{4}\right) & -2 \cos \left(\phi_{3}+\phi_{4}\right)-4 \cos \phi_{4}
\end{array}\right]\left[\begin{array}{l}
\dot{\phi}_{1} \\
\dot{\phi}_{2} \\
\dot{\phi}_{3} \\
\dot{\phi}_{4}
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
$$

where the $2 \times 4$ matrix at the left of Eq. 2 is the Jacobian matrix. Partitioning of Eq. 2 yields the velocity equations in the form of Eq. 3.8:

$$
\begin{align*}
& {\left[\begin{array}{cc}
3 \sin \left(\phi_{1}-\phi_{2}\right) & 2 \sin \left(\phi_{3}+\phi_{4}\right)+4 \sin \phi_{4} \\
-3 \cos \left(\phi_{1}-\phi_{2}\right) & -2 \cos \left(\phi_{3}+\phi_{4}\right)-4 \cos \phi_{4}
\end{array}\right]\left[\begin{array}{l}
\dot{\phi}_{2} \\
\dot{\phi}_{4}
\end{array}\right]=} \\
& \quad-\left[\begin{array}{cc}
-2 \sin \phi_{1}-3 \sin \left(\phi_{1}-\phi_{2}\right) & 2 \sin \left(\phi_{3}+\phi_{4}\right) \\
2 \cos \phi_{1}+\cos \left(\phi_{1}-\phi_{2}\right) & -2 \cos \left(\phi_{3}+\phi_{4}\right)
\end{array}\right]\left[\begin{array}{l}
\dot{\phi}_{1} \\
\dot{\phi}_{3}
\end{array}\right] \tag{3}
\end{align*}
$$

The kinematic acceleration equations can be found either by direct differentiation of Eq. 2 or by using Eq. 3.10, as follows:

$$
\left[\begin{array}{rc}
-2 \sin \phi_{1}-3 \sin \left(\phi_{1}-\phi_{2}\right) & 3 \sin \left(\phi_{1}-\phi_{2}\right) \\
2 \cos \phi_{1}+3 \cos \left(\phi_{1}-\phi_{2}\right) & -3 \cos \left(\phi_{1}-\phi_{2}\right)
\end{array}\right.
$$

$$
\left.\begin{array}{cc}
2 \sin \left(\phi_{3}+\phi_{4}\right) & 2 \sin \left(\phi_{3}+\phi_{4}\right)+4 \sin \phi_{4} \\
-2 \cos \left(\phi_{3}+\phi_{4}\right) & -2 \cos \left(\phi_{3}+\phi_{4}\right)-4 \cos \phi_{4}
\end{array}\right]\left[\begin{array}{l}
\phi_{1} \\
\ddot{\phi}_{2} \\
\ddot{\phi}_{3} \\
\ddot{\phi}_{4}
\end{array}\right]+
$$

$$
\left[\begin{array}{l}
-2 \cos \phi_{1} \dot{\phi}_{1}^{2}-3 \cos \left(\phi_{1}-\phi_{2}\right)\left(\dot{\phi}_{1}-\dot{\phi}_{2}\right)^{2}+ \\
-2 \sin \phi_{1} \dot{\phi}_{1}^{2}-3 \sin \left(\phi_{1}-\phi_{2}\right)\left(\dot{\phi}_{1}-\dot{\phi}_{2}\right)^{2}+  \tag{4}\\
\\
2 \cos \left(\phi_{3}+\phi_{4}\right)\left(\dot{\phi}_{3}+\dot{\phi}_{4}\right)^{2}+4 \cos \phi_{4} \dot{\phi}_{4}^{2} \\
2 \sin \left(\phi_{3}+\phi_{4}\right)\left(\dot{\phi}_{3}+\dot{\phi}_{4}\right)^{2}+4 \sin \phi_{4} \dot{\phi}_{4}^{2}
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
$$

Partitioning of Eq. 4 yields the acceleration equations, in the form of Eq. 3.11:

$$
\begin{aligned}
& {\left[\begin{array}{rr}
3 \sin \left(\phi_{1}-\phi_{2}\right) & 2 \sin \left(\phi_{3}+\phi_{4}\right)+4 \sin \phi_{4} \\
-3 \cos \left(\phi_{1}-\phi_{2}\right) & -2 \cos \left(\phi_{3}+\phi_{4}\right)-4 \cos \phi_{4}
\end{array}\right]\left[\begin{array}{l}
\ddot{\phi}_{2} \\
\ddot{\phi}_{4}
\end{array}\right]=} \\
&-\left[\begin{array}{cr}
-2 \sin \phi_{1}-3 \sin \left(\phi_{1}-\phi_{2}\right) & 2 \sin \left(\phi_{3}+\phi_{4}\right) \\
2 \cos \phi_{1}+\cos \left(\phi_{1}-\phi_{2}\right) & -2 \cos \left(\phi_{3}+\phi_{4}\right)
\end{array}\right]\left[\begin{array}{l}
\ddot{\phi}_{1} \\
\ddot{\phi}_{3}
\end{array}\right]-
\end{aligned}
$$

$$
\left[\begin{array}{l}
-2 \cos \phi_{1} \dot{\phi}_{1}^{2}-3 \cos \left(\phi_{1}-\phi_{2}\right)\left(\dot{\phi}_{1}-\dot{\phi}_{2}\right)^{2}+ \\
-2 \sin \phi_{1} \dot{\phi}_{1}^{2}-3 \sin \left(\phi_{1}-\phi_{2}\right)\left(\dot{\phi}_{1}-\dot{\phi}_{2}\right)^{2}+
\end{array}\right.
$$

$$
\left.\begin{array}{l}
2 \cos \left(\phi_{3}+\phi_{4}\right)\left(\dot{\phi}_{3}+\dot{\phi}_{4}\right)^{2}+4 \cos \phi_{4} \dot{\phi}_{4}^{2}  \tag{5}\\
2 \sin \left(\phi_{3}+\phi_{4}\right)\left(\dot{\phi}_{3}+\dot{\phi}_{4}\right)^{2}+4 \sin \phi_{4} \dot{\phi}_{4}^{2}
\end{array}\right]
$$

### 3.2.2 Method of Appended Driving Constraints

This method, unlike the coordinate partitioning method, does not partition the coordinates into independent and dependent sets. Additional constraint equations, called the driving constraints, equal in number to the number of degrees of freedom of the system, are appended to the original kinematic constraints. The driving constraints are equations representing each independent coordinate as a function of time. This method is best illustrated by an example.

## Example 3.4

The four-bar mechanism of Example 3.2 is considered here again. The kinematic constraints are

$$
\begin{align*}
& \Phi_{1} \equiv 0.2 \cos \phi_{1}+0.4 \cos \phi_{2}-0.3 \cos \phi_{3}-0.35=0  \tag{1}\\
& \Phi_{2} \equiv 0.2 \sin \phi_{1}+0.4 \sin \phi_{2}-0.3 \sin \phi_{3}-0.1=0
\end{align*}
$$

Since the crank angle $\phi_{1}$ is the independent variable, a driving constraint can be written in terms of $\phi_{1}$. The initial value at $t=0$ for $\phi_{1}$ is $\phi_{1}^{0}=2.36 \mathrm{rad}$, and the constant angular velocity of the crank is $2 \pi \mathrm{rad} / \mathrm{s}$. Therefore $\phi_{1}=2.36+6.28 t$ can be used to represent $\phi_{1}$ as a function of time. This driving equation can be rewritten as

$$
\begin{equation*}
\Phi^{(d)} \equiv \phi_{1}-2.36-6.28 t=0 \tag{2}
\end{equation*}
$$

At any instant of time, i.e., known $t$, Eqs. 1 and 2 represent three equations in three unknowns $\phi_{1}, \phi_{2}$, and $\phi_{3}$. If these equations are solved for $t=0$, it is found that $\phi_{1}=2.36, \phi_{2}=0.57$, and $\phi_{3}=2.11$.

For velocity analysis, the first time derivatives of Eqs. 1 and 2 are found and written as

$$
\left[\begin{array}{ccc}
-0.2 \sin \phi_{1} & -0.4 \sin \phi_{2} & 0.3 \sin \phi_{3}  \tag{3}\\
0.2 \cos \phi_{1} & 0.4 \cos \phi_{2} & -0.3 \cos \phi_{3} \\
1 & 0 & 0
\end{array}\right]\left[\begin{array}{l}
\dot{\phi}_{1} \\
\dot{\phi}_{2} \\
\dot{\phi}_{3}
\end{array}\right]=\left[\begin{array}{c}
0 \\
0 \\
6.28
\end{array}\right]
$$

For known values of $\phi_{1}, \phi_{2}$, and $\phi_{3}$, Eq. 3 yields $\dot{\phi}_{1}=6.28, \dot{\phi}_{2}=0.76$, and $\phi_{3}=4.09$.

For acceleration analysis, the time derivative of Eq. 3 is found to be
$\left[\begin{array}{ccc}-0.2 \sin \phi_{1} & -0.4 \sin \phi_{2} & 0.3 \sin \phi_{3} \\ 0.2 \cos \phi_{1} & 0.4 \cos \phi_{2} & -0.3 \cos \phi_{3} \\ 1 & 0 & 0\end{array}\right]\left[\begin{array}{l}\ddot{\phi}_{1} \\ \ddot{\phi}_{2} \\ \ddot{\phi}_{3}\end{array}\right]=$

$$
\left[\begin{array}{c}
0.2 \cos \phi_{1} \dot{\phi}_{1}^{2}+0.4 \cos \phi_{2} \dot{\phi}_{2}^{2}-0.3 \cos \phi_{3} \dot{\phi}_{3}^{2}  \tag{4}\\
0.2 \sin \phi_{1} \dot{\phi}_{1}^{2}+0.4 \sin \phi_{2} \dot{\phi}_{2}^{2}-0.3 \sin \phi_{3} \dot{\phi}_{3}^{2} \\
0
\end{array}\right]
$$

Since $\phi_{1}, \phi_{2}, \phi_{3}, \dot{\phi}_{1}, \dot{\phi}_{2}$, and $\dot{\phi}_{3}$ are known, Eq. 4 yields $\ddot{\phi}_{1}=0, \ddot{\phi}_{2}=6.62$, and $\phi_{3}=5.39$.

This process can be repeated at different instants of time. If $t$ is incremented by $\Delta t=0.025 \mathrm{~s}$, the same table as shown in Example 3.2 is obtained.

The method of appended driving constraints can now be stated in its most general form. If there are $m$ kinematic constraints, then $k$ driving constraints must be appended to the kinematic constraints to obtain $n=m+k$ equations:

$$
\begin{align*}
\boldsymbol{\Phi} & \equiv \boldsymbol{\Phi}(\mathbf{q})=\mathbf{0} \\
\boldsymbol{\Phi}^{(d)} & \equiv \boldsymbol{\Phi}(\mathbf{q}, t)=\mathbf{0} \tag{3.12}
\end{align*}
$$

where superscript ( $d$ ) denotes the driving constraints. Equation 3.12 represents $n$ equations in $n$ unknowns $\mathbf{q}$ which can be solved at any specified time $t$.

The velocity equations are obtained by taking the time derivative of Eq. 3.12:

$$
\begin{align*}
\boldsymbol{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} & =0  \tag{3.13}\\
\boldsymbol{\Phi}_{\mathbf{q}}^{(d)} \dot{\mathbf{q}}+\boldsymbol{\Phi}_{t}^{(d)} & =0
\end{align*}
$$

or

$$
\left[\begin{array}{c}
\boldsymbol{\Phi}_{\mathbf{q}}  \tag{3.14}\\
\boldsymbol{\Phi}_{\mathbf{q}}^{(d)}
\end{array}\right] \dot{\mathbf{q}}=\left[\begin{array}{c}
0 \\
-\boldsymbol{\Phi}_{t}^{(d)}
\end{array}\right]
$$

which represents $n$ algebraic equations, linear in terms of $\dot{\mathbf{q}}$.
Similarly, the time derivative of Eq. 3.13 yields the acceleration equations:

$$
\begin{gather*}
\boldsymbol{\Phi}_{\mathbf{q}} \ddot{\mathbf{q}}+\left(\boldsymbol{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} \dot{\mathbf{q}}_{\mathbf{q}}=\mathbf{0}\right. \\
\boldsymbol{\Phi}_{\mathbf{q}}^{(d)} \ddot{\mathbf{q}}+\left(\boldsymbol{\Phi}_{\mathbf{q}}^{(d)} \dot{\mathbf{q}}\right)_{\mathbf{q}} \dot{q}+2 \boldsymbol{\Phi}_{\mathbf{q} \boldsymbol{q}}^{(d)} \dot{\mathbf{q}}+\boldsymbol{\Phi}_{t}^{(d)}=0 \tag{3.15}
\end{gather*}
$$

or

$$
\left[\begin{array}{c}
\boldsymbol{\Phi}_{\mathbf{q}}  \tag{3.16}\\
\boldsymbol{\Phi}_{\mathbf{q}}^{(d)}
\end{array}\right] \ddot{\mathbf{q}}=\left[\begin{array}{c}
-\left(\boldsymbol{\Phi}_{\mathbf{q}} \dot{\mathbf{q}}\right)_{\mathbf{q}} \dot{\mathbf{q}} \\
-\left(\boldsymbol{\Phi}_{\mathbf{q}}^{(d)} \dot{\mathbf{q}}\right)_{\mathbf{q}} \dot{\mathbf{q}}-2 \boldsymbol{\Phi}_{\mathbf{q} \mathbf{r}}^{(d)} \dot{\mathbf{q}}-\boldsymbol{\Phi}_{t}^{(d)}
\end{array}\right]
$$

which represents $n$ algebraic equations linear in terms of $\ddot{\mathbf{q}}$. The term $-\left(\boldsymbol{\Phi}_{\mathbf{q}} \dot{\mathbf{q}}\right)_{\mathbf{q}} \dot{\mathbf{q}}$ in Eq. 3.16 is referred to as the right side of the kinematic acceleration equations, and is represented as

$$
\begin{equation*}
\gamma=-\left(\Phi_{\mathbf{q}} \dot{\mathbf{q}}\right)_{\mathbf{q}} \dot{\mathbf{q}} \tag{3.17}
\end{equation*}
$$

In the above formulations, the driving constraints are assumed to have the general form $\boldsymbol{\Phi}(\mathbf{q}, t)=\mathbf{0}$. However, as shown in Example 3.4, the driving constraint can have a very simple form, such as $v_{j}-c(t)=0$, where $v_{j}$ is the $j$ th independent variable and $c(t)$ is a known function of time. If there are $k$ independent variables in the system and the $k$ driving constraints have the form

$$
\begin{equation*}
\boldsymbol{\Phi}^{(d)} \equiv \mathbf{v}-\mathbf{c}(t)=0 \tag{3.18}
\end{equation*}
$$

then Eqs. 3.12 through 3.16 can be simplified. In this case the Jacobian matrix becomes

$$
\left[\begin{array}{c}
\boldsymbol{\Phi}_{\mathbf{q}}  \tag{3.19}\\
\boldsymbol{\Phi}_{\mathbf{q}}^{(d)}
\end{array}\right] \equiv\left[\begin{array}{c}
\boldsymbol{\Phi}_{\mathbf{q}} \\
\hat{\mathbf{I}}
\end{array}\right]
$$

where $\hat{\mathbf{I}}$ is a permuted nonsquare identity matrix (permuted means the columns are reordered). The 1 s of $\hat{\mathbf{I}}$ are in the columns associated with the independent variables $\mathbf{v}$. Therefore, Eq. 3.19 can be expressed as

$$
\left[\begin{array}{c}
\Phi_{\mathrm{q}}  \tag{3.20}\\
\hat{I}^{\prime}
\end{array}\right] \equiv\left[\begin{array}{cc}
\Phi_{\mathrm{u}} & \Phi_{\mathrm{v}} \\
0 & \mathbf{I}
\end{array}\right]
$$

## Example 3.5

Four coordinates $\mathbf{q}=\left[x_{1}, x_{2}, x_{3}, x_{4}\right]^{T}$ are subject to the kinematic constraints

$$
\begin{array}{r}
x_{1}^{2}+2 x_{2} x_{3}-x_{1} x_{4}=0 \\
3 x_{1} x_{2}-x_{2}^{2}+x_{3} x_{4}-x_{3}=0
\end{array}
$$

The coordinates $x_{2}$ and $x_{4}$ are expressed in terms of $t$ as follows:

$$
\begin{array}{r}
x_{2}-0.2 t=0 \\
x_{4}+0.5 t-0.03 t^{2}=0
\end{array}
$$

The system Jacobian is

$$
\left[\begin{array}{c}
\boldsymbol{\Phi}_{q} \\
\hat{\mathbf{I}}
\end{array}\right]=\left[\begin{array}{cccc}
2 x_{1}-x_{4} & 2 x_{3} & 2 x_{2} & -x_{1} \\
3 x_{2} & 3 x_{1}-2 x_{2} & x_{4}-1 & x_{3} \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{array}\right]
$$

In terms of $\boldsymbol{\Phi}_{\mathbf{u}}$ and $\boldsymbol{\Phi}_{\mathbf{v}}$ the Jacobian is permuted as follows:

$$
\left[\begin{array}{cc}
\boldsymbol{\Phi}_{\mathbf{u}} & \boldsymbol{\Phi}_{\mathbf{v}} \\
\mathbf{0} & \mathbf{I}
\end{array}\right]=\left[\begin{array}{cccc}
2 x_{1}-x_{4} & 2 x_{2} & 2 x_{3} & -x_{1} \\
3 x_{2} & x_{4}-1 & 3 x_{1}-2 x_{2} & x_{3} \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}\right]
$$

The general procedure for kinematic analysis using the method of appended driving constraints may be summarized in the following algorithm:

## ALGORITHM K-II

(a) Set a time step counter $i$ to $i=0$ and initialize $t^{i}=t^{0}$ (initial time).
(b) Append $k$ driving equations to the constraint equations.
(c) Solve Eq. 3.12 iteratively to obtain $\mathbf{q}^{i}$.
(d) Solve Eq. 3.14 to obtain $\dot{\mathbf{q}}^{i}$.
(e) Solve Eq. 3.16 to obtain $\ddot{\mathbf{q}}^{i}$.
(f) If final time is reached, then terminate; otherwise increment $t^{i}$ to $t^{i+1}$, let $i \rightarrow i+1$, and go to (c).

Kinematic analysis with the method of appended driving constraints usually requires the solution of a larger set of equations than with the coordinate partitioning method. However, this is not a major drawback since the programming effort for computer implementation of algorithm K-II is much less than of algorithm K-I. A computer program for kinematic analysis of mechanical systems based on this method is described in Chap. 5.

### 3.3 LINEAR ALGEBRAIC EQUATIONS

It was shown in Secs. 3.21 and 3.22 that the kinematic velocity and acceleration analysis of mechanical systems requires the solution of a set of linear algebraic equations. Al-
though position analysis involves the solution of a set of nonlinear algebraic equations, it will be seen that most numerical methods solve these by iteratively solving a set of linear equations. Therefore, for almost every step of kinematic analysis - position, velocity, or acceleration-sets of linear algebraic equations must be solved.

Consider a system of $n$ linear algebraic equations with real constant coefficients,

$$
\begin{gather*}
a_{11} x_{1}+a_{12} x_{2}+\cdots+a_{1 n} x_{n}=c_{1} \\
a_{21} x_{1}+a_{22} x_{2}+\cdots+a_{2 n} x_{n}=c_{2}  \tag{3.21}\\
\cdots \\
a_{n 1} x_{1}+a_{n 2} x_{2}+\cdots+a_{n n} x_{n}=c_{n}
\end{gather*}
$$

which can be written in matrix form as

$$
\begin{equation*}
\mathbf{A x}=\mathbf{c} \tag{3.22}
\end{equation*}
$$

There are many methods for solving this set of equations. Cramer's rule offers one of the best-known methods, but also the most inefficient. Among the more efficient methods are the Gaussian elimination, Gauss-Jordan reduction, and L-U factorization methods.

### 3.3.1 Gaussian Methods

The Gaussian elimination method for solving linear equations is based on the elementary idea of eliminating variables one at a time. This method consists of two major steps: (1) a forward elimination, which converts the matrix $\mathbf{A}$ into an upper-triangular matrix, and (2) a back substitution, which solves for the unknown $\mathbf{x}$. There are a variety of Gaussian elimination algorithms that are similar in principle, but slightly different in approach. The method presented here converts the matrix $\mathbf{A}$ to an upper-triangular matrix with 1s on the diagonal. The process is best illustrated by an example that can be followed easily.

## Example 3.6

Solve the set of equations

$$
\left[\begin{array}{rrr}
3 & 1 & -1 \\
-1 & 2 & 1 \\
2 & -3 & 1
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
2 \\
6 \\
-1
\end{array}\right]
$$

to find $x_{1}, x_{2}$, and $x_{3}$.

## Solution

FORWARD ELIMINATION

1. Multiply the first equation by $\frac{1}{3}$, to put a 1 in the $a_{11}$ position, as follows:

$$
\left[\begin{array}{rrr}
1 & \frac{1}{3} & -\frac{1}{3} \\
-1 & 2 & 1 \\
2 & -3 & 1
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
\frac{2}{3} \\
6 \\
-1
\end{array}\right]
$$

and add the first equation to the second and then add -2 times the first equation to the third, to put zeros in the first column below the diagonal:

$$
\left[\begin{array}{ccc}
1 & \frac{1}{3} & -\frac{1}{3} \\
0 & \frac{7}{3} & \frac{2}{3} \\
0 & -\frac{11}{3} & \frac{5}{3}
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{c}
\frac{2}{3} \\
\frac{20}{3} \\
-\frac{7}{3}
\end{array}\right]
$$

2. Multiply the second equation by $\frac{3}{7}$, to put a 1 in the $a_{22}$ position, as shown:

$$
\left[\begin{array}{rrr}
1 & \frac{1}{3} & -\frac{1}{3} \\
0 & 1 & \frac{2}{7} \\
0 & -\frac{11}{3} & \frac{5}{3}
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
\frac{2}{3} \\
\frac{20}{7} \\
-\frac{7}{3}
\end{array}\right]
$$

and add $\frac{11}{3}$ times the second equation to the third, to put a zero below the diagonal:

$$
\left[\begin{array}{rrr}
1 & \frac{1}{3} & -\frac{1}{3} \\
0 & 1 & \frac{2}{7} \\
0 & 0 & \frac{19}{7}
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{c}
\frac{2}{3} \\
\frac{20}{7} \\
\frac{57}{7}
\end{array}\right]
$$

3. Multiply the third equation by $\frac{7}{19}$, to put a 1 in the $a_{33}$ position:

$$
\left[\begin{array}{rrr}
1 & \frac{1}{3} & -\frac{1}{3} \\
0 & 1 & \frac{2}{7} \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{c}
\frac{2}{3} \\
\frac{20}{7} \\
3
\end{array}\right]
$$

## BACK SUBSTITUTION

1. The third equation yields $x_{3}=3$.
2. The second equation then yields $x_{2}+\frac{2}{7}(3)=\frac{20}{7}$, so $x_{2}=2$.
3. The first equation yields $x_{1}+\frac{1}{3}(2)-\frac{1}{3}(3)=\frac{2}{3}$, so $x_{1}=1$.

Note that the forward-elimination step requires division by $a_{j j}$ at the $j$ th step. The preceding operation is valid only if $a_{i j} \neq 0$.

The Gauss-Jordan reduction method combines the forward-elimination and backsubstitution steps of the Gaussian elimination method into one step. Matrix $\mathbf{A}$ is converted to a diagonal unit matrix, using elementary arithmetic. This method may also be illustrated with a simple example.

## Example 3.7

Solve the set of linear algebraic equations

$$
\left[\begin{array}{rrr}
2 & -1 & 1 \\
1 & 3 & -2 \\
-1 & 1 & 2
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-3 \\
3 \\
5
\end{array}\right]
$$

using the Gauss-Jordan reduction method.

## Solution

1. Multiply the first equation by $\frac{1}{2}$, to put a 1 in the $a_{11}$ position, as shown:

$$
\left[\begin{array}{rrr}
1 & -\frac{1}{2} & \frac{1}{2} \\
1 & 3 & -2 \\
-1 & 1 & 2
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-\frac{3}{2} \\
3 \\
5
\end{array}\right]
$$

and add -1 times the first equation to the second and then add the first equation to the third to put zeros below the diagonal:

$$
\left[\begin{array}{rrr}
1 & -\frac{1}{2} & \frac{1}{2} \\
0 & \frac{7}{2} & -\frac{5}{2} \\
0 & \frac{1}{2} & \frac{5}{2}
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-\frac{3}{2} \\
\frac{9}{2} \\
\frac{7}{2}
\end{array}\right]
$$

2. Multiply the second equation by $\frac{2}{7}$ to put a 1 in the $a_{22}$ position, as shown:

$$
\left[\begin{array}{rrr}
1 & -\frac{1}{2} & \frac{1}{2} \\
0 & 1 & -\frac{5}{7} \\
0 & \frac{1}{2} & \frac{5}{2}
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-\frac{3}{2} \\
\frac{9}{7} \\
\frac{7}{2}
\end{array}\right]
$$

and add $\frac{1}{2}$ times the second equation to the first and then add $-\frac{1}{2}$ times the second equation to the third, to put zeros above and below the $a_{22}$ position:

$$
\left[\begin{array}{rcr}
1 & 0 & \frac{1}{7} \\
0 & 1 & -\frac{5}{7} \\
0 & 0 & \frac{20}{7}
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-\frac{6}{7} \\
\frac{9}{7} \\
\frac{20}{7}
\end{array}\right]
$$

3. Multiply the third equation by $\frac{7}{20}$, to put a 1 in the $a_{33}$ position:

$$
\left[\begin{array}{rrr}
1 & 0 & \frac{1}{7} \\
0 & 1 & -\frac{5}{7} \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-\frac{6}{7} \\
\frac{9}{7} \\
1
\end{array}\right]
$$

and add $-\frac{1}{7}$ times the third equation to the first and then add $\frac{5}{7}$ times the third equation to the second, to put zeros above the $a_{33}$ position:

$$
\left[\begin{array}{lll}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-1 \\
2 \\
1
\end{array}\right]
$$

Since the coefficient matrix has been converted to the identity matrix it is clear that $x_{1}=-1, x_{2}=2$, and $x_{3}=1$ is the solution.

### 3.3.2 Pivoting

In the forward-elimination step of the Gaussian methods, the algorithm fails if at the $j$ th step $a_{j j}$ is zero. Also, when the pivot element $a_{j j}$ becomes too small, numerical error
may occur. Therefore, the order in which the equations are treated during the forwardelimination step significantly affects the accuracy of the algorithm. To circumvent this difficulty, the order in which the equations are used is determined by the algorithm; i.e., the order may not necessarily be the original order $1,2, \ldots, n$. The algorithm reorders the equations depending on the actual system being solved. This process is called pivoting. Two types of pivoting, partial pivoting and full pivoting, are discussed here.

In partial pivoting, during the $j$ th forward-elimination step of the Gaussian algorithm, the equation with the largest coefficient (in absolute value) of $x_{j}$ on or below the diagonal is chosen for pivoting. During the elimination step, the rows of the matrix and also the elements of vector $c$ are interchanged. The following example illustrates this procedure.

## Example 3.8

Perform Gaussian elimination with partial (row) pivoting on the following set of equations:

$$
\left[\begin{array}{rrrr}
4 & -3 & 5 & 2 \\
-3 & 1 & 1 & -6 \\
5 & -5 & 10 & 0 \\
2 & -3 & 9 & -7
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
-1.5 \\
9 \\
2.5 \\
13.5
\end{array}\right]
$$

## Solution

## FORWARD ELIMINATION WITH PARTIAL PIVOTING

1. The largest coefficient in column 1 of the matrix is 5 . Therefore interchange the third and first equations, to obtain

$$
\left(\left[\begin{array}{rrrr}
5 & -5 & 10 & 0 \\
-3 & 1 & 1 & -6 \\
4 & -3 & 5 & 2 \\
2 & -3 & 9 & -7
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
2.5 \\
9 \\
-1.5 \\
13.5
\end{array}\right]\right.
$$

Then perform forward elimination, as in Example 3.6, to obtain

$$
\left[\begin{array}{rrrr}
1 & -1 & 2 & 0 \\
0 & -2 & 7 & -6 \\
0 & 1 & -3 & 2 \\
0 & -1 & 5 & -7
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{r}
0.5 \\
10.5 \\
-3.5 \\
12.5
\end{array}\right]
$$

2. The largest coefficient in column 2 on or below the diagonal is -2 ; no interchange is necessary. Forward elimination yields

$$
\left[\begin{array}{rrrr}
1 & -1 & 2 & 0 \\
0 & 1 & -3.5 & 3 \\
0 & 0 & 0.5 & -1 \\
0 & 0 & 1.5 & -4
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
0.5 \\
-5.25 \\
1.75 \\
7.25
\end{array}\right]
$$

3. The largest coefficient in column 3 on or below the diagonal is 1.5 . Interchange the fourth and third equations to obtain

$$
\left(\left[\begin{array}{rrrr}
1 & -1 & 2 & 0 \\
0 & 1 & -3.5 & 3 \\
0 & 0 & 1.5 & -4 \\
0 & 0 & 0.5 & -1
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
0.5 \\
-5.25 \\
7.25 \\
1.75
\end{array}\right]\right.
$$

Then perform forward elimination to obtain

$$
\left[\begin{array}{cccc}
1 & -1 & 2 & 0 \\
0 & 1 & -3.5 & 3 \\
0 & 0 & 1 & -2.66 \\
0 & 0 & 0 & 0.33
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
0.5 \\
-5.25 \\
4.83 \\
-0.66
\end{array}\right]
$$

4. Multiply the fourth equation by $\frac{1}{0.33}$ to obtain

$$
\left[\begin{array}{crlc}
1 & -1 & 2 & 0 \\
& 1 & 3.5 & 3 \\
& & 1 & -2.66 \\
& & & 1
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
0.5 \\
-5.25 \\
4.83 \\
-2
\end{array}\right]
$$

Back substitution now yields $\mathbf{x}=[0.5,-1,-0.5,-2]^{T}$.

The preceding pivoting method is referred to as partial pivoting with row interchange, since the rows of the matrix are interchanged. The method can be modified for partial pivoting with column interchange.

Full or complete pivoting is the selection of the largest of all the coefficients (in absolute value) on the diagonal and to its right and below as the basis for the next stage of elimination, which operates on the corresponding variable. In full pivoting, both row interchange and column interchange are required. Note that when two columns of the coefficient matrix are interchanged, their corresponding variables in the vector $\mathbf{x}$ are interchanged.
Example 3.9
Apply the Gaussian elimination method with full pivoting to the following set of equations:

$$
\left[\begin{array}{rrr}
2 & -1 & 1 \\
-1 & 0 & 2 \\
1 & 4 & -2
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
0 \\
5 \\
-5
\end{array}\right]
$$

## Solution

1. The largest coefficient in the matrix is 4 . Interchanging columns 2 and 1 , we get

$$
\left[\begin{array}{rrr}
-1 & 2 & 1 \\
0 & -1 & 2 \\
4 & 1 & -2
\end{array}\right]\left[\begin{array}{l}
x_{2} \\
x_{1} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
0 \\
5 \\
-5
\end{array}\right]
$$

Interchanging rows 3 and 1 yields

$$
\left(\left[\begin{array}{rrr}
4 & 1 & -2 \\
0 & -1 & 2 \\
-1 & 2 & 1
\end{array}\right]\left[\begin{array}{l}
x_{2} \\
x_{1} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-5 \\
5 \\
0
\end{array}\right]\right.
$$

The elimination step gives

$$
\left[\begin{array}{ccc}
1 & 0.25 & -0.5 \\
0 & -1 & 2 \\
0 & 2.25 & 0.5
\end{array}\right]\left[\begin{array}{l}
x_{2} \\
x_{1} \\
x_{3}
\end{array}\right]=\left[\begin{array}{c}
-1.25 \\
5 \\
-1.25
\end{array}\right]
$$

2. The largest coefficient in the $2 \times 2$ lower right submatrix is 2.25 . Interchanging rows 3 and 2 , we have

$$
\left(\left[\begin{array}{ccc}
1 & 0.25 & -0.5 \\
0 & 2.25 & 0.5 \\
0 & -1 & 2
\end{array}\right]\left[\begin{array}{l}
x_{2} \\
x_{1} \\
x_{3}
\end{array}\right]=\left[\begin{array}{c}
-1.25 \\
-1.25 \\
5
\end{array}\right]\right.
$$

The elimination step gives

$$
\left[\begin{array}{llc}
1 & 0.25 & -0.5 \\
0 & 1 & 0.22 \\
0 & 0 & 2.22
\end{array}\right]\left[\begin{array}{l}
x_{2} \\
x_{1} \\
x_{3}
\end{array}\right]=\left[\begin{array}{r}
-1.25 \\
-0.55 \\
4.44
\end{array}\right]
$$

3. Multiplying the last equation by $\frac{1}{2.22}$, we get

$$
\left[\begin{array}{ccc}
1 & 0.25 & -0.5 \\
0 & 1 & 0.22 \\
0 & 0 & 1
\end{array}\right]\left[\begin{array}{l}
x_{2} \\
x_{1} \\
x_{3}
\end{array}\right]=\left[\begin{array}{c}
-1.25 \\
-0.55 \\
2
\end{array}\right]
$$

Back substitution now yields $x_{3}=2, x_{1}=-1$, and $x_{2}=0$.

In most computer programs, partial or full pivoting is carried out simply by interchanging the row (column) indices of the two rows (columns) to be interchanged. Two integer arrays hold the indices for column and row numbers of the matrix.

### 3.3.3 L-U Factorization

The $L-U$ factorization method is a compact form of the Gaussian elimination method of operating on a matrix $\mathbf{A}$. After the operation is completed, the set of linear equations $\mathbf{A x}=\mathbf{c}$ is efficiently solved for any given $\mathbf{c}$ vector.

For any nonsingular matrix $\mathbf{A}$, it can be proved that there exists an upper triangular matrix $\mathbf{U}$ with nonzero diagonal elements and a lower triangular matrix $\mathbf{L}$ with unit diagonal elements, such that

$$
\begin{equation*}
\mathbf{A}=\mathbf{L} \mathbf{U} \tag{3.23}
\end{equation*}
$$

The process of factoring $\mathbf{A}$ into the product $\mathbf{L U}$ is called $L-U$ factorization. Once the L-U factorization is obtained, by whatever method, the equation

$$
\begin{equation*}
\mathbf{A x}=\mathbf{L U x}=\mathbf{c} \tag{3.24}
\end{equation*}
$$

is solved by transforming Eq. 3.24 into

$$
\begin{equation*}
\mathbf{L y}=\mathbf{c} \tag{3.25}
\end{equation*}
$$

and

$$
\begin{equation*}
\mathbf{U x}=\mathbf{y} \tag{3.26}
\end{equation*}
$$

Equation 3.25 is first solved for $\mathbf{y}$ and Eq. 3.26 is then solved for $\mathbf{x}$. Since Eqs. 3.25 and 3.26 are both triangular systems of equations, the solutions are easily obtained by forward and backward substitution.

Crout's method calculates the elements of $\mathbf{L}$ and $\mathbf{U}$ recursively, without overwriting previous results. ${ }^{2}$ To illustrate how Crout's method generates the elements of $\mathbf{L}$ and $\mathbf{U}$, consider a matrix $\mathbf{A}$ of rank 4, requiring no row or column interchanges, i.e., no pivoting. The matrix $\mathbf{A}$ can be written as

$$
\left[\begin{array}{llll}
1 & 0 & 0 & 0  \tag{3.27}\\
l_{21} & 1 & 0 & 0 \\
l_{31} & l_{32} & 1 & 0 \\
l_{41} & l_{42} & l_{43} & 1
\end{array}\right]\left[\begin{array}{llll}
u_{11} & u_{12} & u_{13} & u_{14} \\
0 & u_{22} & u_{23} & u_{24} \\
0 & 0 & u_{33} & u_{34} \\
0 & 0 & 0 & u_{44}
\end{array}\right]=\left[\begin{array}{llll}
a_{11} & a_{12} & a_{13} & a_{14} \\
a_{21} & a_{22} & a_{23} & a_{24} \\
a_{31} & a_{32} & a_{33} & a_{34} \\
a_{41} & a_{42} & a_{43} & a_{44}
\end{array}\right]
$$

Now an auxiliary matrix $\mathbf{B}$ can be defined, consisting of elements of $\mathbf{L}$ and $\mathbf{U}$, such that

$$
\mathbf{B} \equiv\left[\begin{array}{llll}
u_{11} & u_{12} & u_{13} & u_{14}  \tag{3.28}\\
l_{21} & u_{22} & u_{23} & u_{24} \\
l_{31} & l_{32} & u_{33} & u_{34} \\
l_{41} & l_{42} & l_{43} & u_{44}
\end{array}\right]
$$

Elements of $\mathbf{B}$ are to be calculated one by one, in the order indicated as follows:

where $k$ indicates the $k$ th element to be calculated. The elements of $\mathbf{L}$ and $\mathbf{U}$ are calculated simply by equating $a_{j k}$ successively, according to the order shown in Eq. 3.29, with the product of the $j$ th row of $\mathbf{L}$ and the $k$ th column of $\mathbf{U}$. The Crout process for the
$n \times n$ matrix $\mathbf{A}$ is performed in $n-1$ iterations. After $i-1$ iterations matrix $\mathbf{A}$ finds the form

$$
\left.\left[\begin{array}{cccccc}
u_{11} & u_{12} & \cdot & \cdot & \cdot & u_{1 n}  \tag{3.30}\\
l_{21} & u_{22} & \cdot & \cdot & \cdot & u_{2 n} \\
\vdots & \vdots & \vdots & \cdot & \cdot \\
\vdots & \vdots & \cdot & \mathbf{D}_{i} & \\
l_{n 1} & l_{n 2} & \cdot & &
\end{array}\right]\right\} i-1 \text { rows }
$$

where the conversion process of the first $i-1$ rows and $i-1$ columns has been completed. The $(n-i+1) \times(n-i+1)$ matrix in the lower right corner is denoted by $\mathbf{D}_{i}$. The elements of $\mathbf{D}_{i}$ are not the same as the elements of the original matrix $\mathbf{A}$. In the $i$ th step, the Crout process converts matrix $\mathbf{D}_{i}$ to a new form:

$$
\mathbf{D}_{i} \equiv\left[\begin{array}{cc}
d_{i i} & \mathbf{r}_{i}^{T}  \tag{3.31}\\
\mathbf{s}_{i} & \mathbf{H}_{i+1}
\end{array}\right] \longrightarrow \text { Crout's process } \longrightarrow\left[\begin{array}{cc}
u_{i i} & \mathbf{u}_{i}^{T} \\
\boldsymbol{l}_{i} & \mathbf{D}_{i+1}
\end{array}\right]
$$

where matrix $\mathbf{D}_{i+1}$ is one row and one column smaller than matrix $\mathbf{D}_{i}$. Crout's process can be stated as follows:

## ALGORITHM LU-I

(a) Initially set an iteration counter $i=1$. In the first step, matrix $\mathbf{D}_{1}=\mathbf{A}$.
(b) Refer to the conversion formula of Eq. 3.31 and let

$$
\begin{align*}
u_{i i} & =d_{i i}  \tag{3.32}\\
\mathbf{u}_{i}^{T} & =\mathbf{r}_{i}^{T}  \tag{3.33}\\
l_{i} & =\frac{1}{u_{i i}} \mathbf{s}_{i}  \tag{3.34}\\
\mathbf{D}_{i+1} & =\mathbf{H}_{i+1}-\boldsymbol{l}_{i} \mathbf{u}_{i}^{T} \tag{3.35}
\end{align*}
$$

(c) Increment $i$ to $i+1$. If $i=n$, L-U factorization is completed. Otherwise go to (b).

Note that calculation of element $k$ of the auxiliary matrix $\mathbf{B}$, which is an element of either $\mathbf{L}$ or $\mathbf{U}$, involves only the element of $\mathbf{A}$ (or $\mathbf{D}_{i}$ ) in the same position and some elements of $\boldsymbol{B}$ that have already been calculated. As element $k$ is obtained, it is recorded in the $\mathbf{B}$ matrix. In fact it may be recorded in the corresponding position in the $\mathbf{A}$ matrix, if there is no need to keep matrix $\mathbf{A}$. This calculated result need never be written over, since it is already one of the elements of $\mathbf{L}$ or $\mathbf{U}$.

## Example 3.10

Apply L-U factorization to matrix $\mathbf{A}$, and then solve the set of algebraic equations $\mathbf{A x}=\mathbf{c}$ for the unknown $\mathbf{x}$ :

$$
\mathbf{A}=\left[\begin{array}{rrrr}
2 & 1 & 3 & -2 \\
-1 & 0 & -2 & -1 \\
0 & 1 & 2 & 3 \\
4 & 2 & 0 & -1
\end{array}\right] \quad \mathbf{c}=\left[\begin{array}{r}
1 \\
3 \\
0 \\
-1
\end{array}\right]
$$

Solution Following the LU-I algorithm, we have:
$i=1 \quad \mathbf{D}_{1}=\mathbf{A}$ results in

$$
\begin{aligned}
& d_{11}=2 \quad \mathbf{r}_{1}^{T}=\left[\begin{array}{lll}
1 & 3 & -2
\end{array}\right] \\
& \mathbf{s}_{1}=\left[\begin{array}{r}
-1 \\
0 \\
4
\end{array}\right] \quad \mathbf{H}_{2}=\left[\begin{array}{rrr}
0 & -2 & -1 \\
1 & 2 & 3 \\
2 & 0 & -1
\end{array}\right]
\end{aligned}
$$

Then,

$$
\begin{aligned}
& u_{11}=2 \quad \mathbf{u}_{1}^{T}=\left[\begin{array}{lll}
1 & 3 & -2
\end{array}\right] \\
& l_{1}=\left[\begin{array}{c}
-0.5 \\
0 \\
2
\end{array}\right] \\
& \mathbf{D}_{2}=\left[\begin{array}{rrr}
0 & -2 & -1 \\
1 & 2 & 3 \\
2 & 0 & -1
\end{array}\right]-\left[\begin{array}{c}
-0.5 \\
0 \\
2
\end{array}\right]\left[\begin{array}{lll}
1 & 3 & -2
\end{array}\right]=\left[\begin{array}{ccc}
0.5 & -0.5 & -2 \\
1 & 2 & 3 \\
0 & -6 & 3
\end{array}\right]
\end{aligned}
$$

After the first iteration, matrix $\mathbf{A}$ becomes,

$$
\begin{aligned}
& {\left[\begin{array}{cllr}
2 & 1 & 3 & -2 \\
-0.5 & 0.5 & -0.5 & -2 \\
0 & 1 & 2 & 3 \\
2 & 0 & -6 & 3
\end{array}\right]} \\
& i=2 \quad \mathbf{D}_{2}=\left[\begin{array}{llr}
0.5 & -0.5 & -2 \\
1 & 2 & 3 \\
0 & -6 & 3
\end{array}\right]
\end{aligned}
$$

results in

$$
\begin{array}{ll}
d_{22}=0.5 & \mathbf{r}_{2}^{T}=[-0.5-2] \\
\mathbf{s}_{2}=\left[\begin{array}{l}
1 \\
0
\end{array}\right] & \mathbf{H}_{3}=\left[\begin{array}{rr}
2 & 3 \\
-6 & 3
\end{array}\right]
\end{array}
$$

Then,

$$
\begin{array}{ll}
u_{22}=0.5 & \mathbf{u}_{2}^{T}=[-0.5-2] \\
\boldsymbol{l}_{2}=\left[\begin{array}{l}
2 \\
0
\end{array}\right] & \mathbf{D}_{3}=\left[\begin{array}{rr}
2 & 3 \\
-6 & 3
\end{array}\right]-\left[\begin{array}{l}
2 \\
0
\end{array}\right][-0.5-2]=\left[\begin{array}{rr}
3 & 7 \\
-6 & 3
\end{array}\right]
\end{array}
$$

After the second iteration, matrix $\mathbf{A}$ becomes,

$$
\left[\begin{array}{clcr}
2 & 1 & 3 & -2 \\
-0.5 & 0.5 & -0.5 & -2 \\
0 & 2 & 3 & 7 \\
2 & 0 & -6 & 3
\end{array}\right]
$$

$i=3$

$$
\mathbf{D}_{3}=\left[\begin{array}{rr}
3 & 7 \\
-6 & 3
\end{array}\right]
$$

results in,

$$
\begin{aligned}
d_{33} & =3 & \mathbf{r}_{3}^{T} & =[7] \\
\mathbf{s}_{3}^{T} & =[-6] & & \mathbf{H}_{4}
\end{aligned}=[3]
$$

Then,

$$
\left.\begin{array}{rlrl}
u_{33} & =3 & & \mathbf{u}_{3}^{T}=[7] \\
\boldsymbol{l}_{3} & =[-2] & & \mathbf{D}_{4}
\end{array}=[3]-[-2][7]=[17]\right]
$$

Hence, matrix A becomes

$$
\left[\begin{array}{clcc}
2 & 1 & 3 & -2 \\
-0.5 & 0.5 & -0.5 & -2 \\
0 & 2 & 3 & 7 \\
2 & 0 & -2 & 17
\end{array}\right]
$$

$i=4 \quad$ Since $i=4=n$, the $\mathrm{L}-\mathrm{U}$ factorization is completed. The L and U matrices are:

$$
\mathbf{L}=\left[\begin{array}{cccc}
1 & 0 & 0 & 0 \\
-0.5 & 1 & 0 & 0 \\
0 & 2 & 1 & 0 \\
2 & 0 & -2 & 1
\end{array}\right] \quad \mathbf{U}=\left[\begin{array}{cccc}
2 & 1 & 3 & -2 \\
0 & 0.5 & -0.5 & -2 \\
0 & 0 & 3 & 7 \\
0 & 0 & 0 & 17
\end{array}\right]
$$

The solution to $\mathbf{A x}=\mathbf{c}$ is obtained by first solving $\mathbf{L y}=\mathbf{c}$ :

$$
\left[\begin{array}{cccc}
1 & 0 & 0 & 0 \\
-0.5 & 1 & 0 & 0 \\
0 & 2 & 1 & 0 \\
2 & 0 & -2 & 1
\end{array}\right]\left[\begin{array}{l}
y_{1} \\
y_{2} \\
y_{3} \\
y_{4}
\end{array}\right]=\left[\begin{array}{r}
1 \\
3 \\
0 \\
-1
\end{array}\right] \longrightarrow \mathbf{y}=\left[\begin{array}{c}
1 \\
3.5 \\
-7 \\
-17
\end{array}\right]
$$

Then, solving $\mathbf{U x}=\mathbf{y}$ gives

$$
\left[\begin{array}{lllr}
2 & 1 & 3 & -2 \\
0 & 0.5 & -0.5 & -2 \\
0 & 0 & 3 & 7 \\
0 & 0 & 0 & 17
\end{array}\right]\left[\begin{array}{l}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
1 \\
3.5 \\
-7 \\
-17
\end{array}\right] \longrightarrow \mathbf{x}=\left[\begin{array}{r}
-2 \\
3 \\
0 \\
-1
\end{array}\right]
$$

### 3.3.4 L-U Factorization with Pivoting

In the preceding subsection, the situation in which $u_{i i}=0$, where Eq. 3.34 requires division by zero, is not discussed. In this case, partial or full pivoting must be applied. Since pivoting may change the order of the rows or columns of the matrix, this interchange information must be recorded in two additional permutation vectors.

## Example 3.11

Apply L-U factorization with full pivoting to matrix $\mathbf{A}$, and then solve the set of equations $\mathbf{A x}=\mathbf{c}$ :

$$
\mathbf{A}=\left[\begin{array}{rrrr}
1 & 2 & -1 & -3 \\
4 & 1 & -1 & 1 \\
2 & -3 & 0 & -1 \\
-1 & 0 & 5 & 2
\end{array}\right] \quad \mathbf{c}=\left[\begin{array}{r}
0 \\
4 \\
11 \\
1
\end{array}\right]
$$

Solution Two index vectors record the permutation of columns and rows during pivoting. These vectors are initialized to $[1,2,3,4]$ and $[1,2,3,4]^{T}$. The pivot element (the largest element in absolute value) is moved to position $d_{i i}$ at each step. The initial matrix is

$$
\left.\begin{array}{l} 
\\
1 \\
2 \\
3 \\
4
\end{array} \begin{array}{rrrr}
1 & 2 & 3 & 4 \\
1 & 2 & -1 & -3 \\
4 & 1 & -1 & 1 \\
2 & -3 & 0 & -1 \\
-1 & 0 & 5 & 2
\end{array}\right]
$$

$i=1$ The largest element (in absolute value) is moved to $d_{11}$ :

$$
\left.\left.\begin{array}{c}
4 \\
4 \\
2 \\
3 \\
1
\end{array} \begin{array}{rrrr}
1 & 2 & 3 & 4 \\
-1 & 0 & 5 & 2 \\
4 & 1 & -1 & 1 \\
2 & -3 & 0 & -1 \\
1 & 2 & -1 & -3
\end{array}\right] \rightarrow \begin{array}{l}
3 \\
2 \\
3 \\
1
\end{array} \begin{array}{rrrr}
2 \\
5 & 0 & -1 & 2 \\
-1 & 1 & 4 & 1 \\
0 & -3 & 2 & -1 \\
-1 & 2 & 1 & -3
\end{array}\right]
$$

Crout's algorithm then yields

$$
\left.\begin{array}{l} 
\\
4 \\
2 \\
3 \\
1
\end{array} \begin{array}{rrrr}
3 & 2 & 1 & 4 \\
5 & 0 & -1 & 2 \\
-\frac{1}{5} & 1 & \frac{19}{5} & \frac{7}{5} \\
0 & -3 & 2 & -1 \\
-\frac{1}{5} & 2 & \frac{4}{5} & -\frac{13}{5}
\end{array}\right]
$$

$i=2$ The largest element (in absolute value) in the $3 \times 3$ submatrix to the lower right is moved to $d_{22}$, to obtain

$$
\left.\begin{array}{l} 
\\
4 \\
2 \\
3 \\
1
\end{array} \begin{array}{rrrc}
3 & 1 & 2 & 4 \\
5 & -1 & 0 & 2 \\
-\frac{1}{5} & \frac{19}{5} & 1 & \frac{7}{5} \\
0 & 2 & -3 & -1 \\
-\frac{1}{5} & \frac{4}{5} & 2 & -\frac{13}{5}
\end{array}\right]
$$

Crout's algorithm then yields

$$
\begin{aligned}
& \\
& 4 \\
& 2 \\
& 3 \\
& 1
\end{aligned}\left[\begin{array}{rrrr}
3 & 1 & 2 & 4 \\
5 & -1 & 0 & 2 \\
-\frac{1}{5} & \frac{19}{5} & 1 & \frac{7}{5} \\
0 & \frac{10}{19} & -\frac{67}{19} & -\frac{33}{19} \\
-\frac{1}{5} & \frac{4}{19} & \frac{34}{19} & \left.-\frac{55}{19} \right\rvert\,
\end{array}\right]
$$

$i=3$ The largest element (in absolute value) in the $2 \times 2$ matrix to the lower right is in $d_{33}$, so no interchange is needed. Crout's algorithm then yields

$$
\begin{aligned}
& \\
& 4 \\
& 2 \\
& 3 \\
& 1
\end{aligned}\left[\begin{array}{rrrr}
3 & 1 & 2 & 4 \\
5 & -1 & 0 & 2 \\
-\frac{1}{5} & \frac{19}{5} & 1 & \frac{7}{5} \\
0 & \frac{10}{19} & -\frac{67}{19} & -\frac{33}{19} \\
-\frac{1}{5} & \frac{4}{19} & -\frac{34}{67} & -\frac{253}{67}
\end{array}\right]
$$

$i=4 \quad$ Since $i=n$, the L-U factorization is completed.
Now, to solve $\mathbf{A x}=\mathbf{c}$, first solve $\mathbf{L y}=\mathbf{c}$ :

$$
\left[\begin{array}{rccc}
1 & & & \\
-\frac{1}{5} & 1 & & \\
0 & \frac{10}{19} & 1 & \\
--\frac{1}{5} & \frac{4}{19} & -\frac{34}{67} & 1
\end{array}\right]\left[\begin{array}{l}
y_{1} \\
y_{2} \\
y_{3} \\
y_{4}
\end{array}\right]=\left[\begin{array}{r}
1 \\
4 \\
11 \\
0
\end{array}\right]
$$

The elements of vector $\mathbf{c}$ are interchanged according to the elements of the row index. Forward substitution yields $\mathbf{y}=\left[1, \frac{21}{5}, \frac{167}{19}, \frac{253}{67}\right]^{T}$. Then, solve $\mathbf{U x}=\mathbf{y}$ :

$$
\left[\begin{array}{rrr}
5 & -1 & 0 \\
& \frac{19}{5} & 1 \\
& & \frac{7}{5} \\
& & -\frac{67}{19} \\
& & -\frac{353}{67}
\end{array}\right]\left[\begin{array}{l}
x_{3} \\
x_{1} \\
x_{2} \\
x_{4}
\end{array}\right]=\left[\begin{array}{c}
1 \\
\frac{21}{5} \\
\frac{167}{19} \\
\frac{253}{67}
\end{array}\right]
$$

The elements of vector $\mathbf{x}$ are interchanged according to the column index vector. A back substitution yields $\mathbf{x}=[2,-2,1,-1]^{T}$.

### 3.3.5 Subroutines for Linear Algebraic Equations

Two FORTRAN subroutines for solving sets of linear algebraic equations, based on L-U factorization with partial pivoting, are presented in this section. The subroutines listed here are not unique. Other coding may be more compact, more general, or more efficient, but the subroutines show exactly what must be done. An overall understanding of these subroutines and their use makes it easier to understand other subroutines that follow in this text.

An explanation of the subroutines and a description of the variables appearing in the argument list of the subroutines are given here. To eliminate any redundancy of comments, these descriptions are not repeated in the FORTRAN listing of the subroutines.

Subroutine LU. This subroutine performs L-U factorization with partial pivoting on square matrices. The argument parameters in this subroutine are as follows:

A The given $\mathrm{N} \times \mathrm{N}$ matrix. On return $A$ contains $\mathbf{L}$ and $\mathbf{U}$ matrices.
ICOL Integer N -vector containing the column indices.
$\mathrm{N} \quad$ Number of rows (columns) of matrix $\mathbf{A}$.
EPS Test value for deviation from zero due to round-off error.
The subroutine employs Crout's method with column pivoting on matrix A. Therefore, the columns of the matrix are generally interchanged at each elimination step to bring the largest element (in absolute value) to the pivot position. The interchange information is recorded in an integer permutation vector ICOL. The Kth column of the interchange matrix corresponds to the $\mathrm{ICOL}(\mathrm{K})$ th column in the original matrix A. Initially, the subroutine sets $\operatorname{ICOL}(\mathrm{K})=\mathrm{K}, \mathrm{K}=1, \ldots, \mathrm{~N}$. During each pivot search, when the largest element (in absolute value) is found, it will be compared with a parameter EPS. This parameter, which must be assigned by the user, is used by the subroutine as the smallest (in magnitude) nonzero number in the computation. If the selected pivot element is smaller (in absolute value) than EPS, then the pivot element is considered to be zero; i.e., the matrix, within the specified error level, is singular. Therefore, the routine terminates the L-U factorization with an error message. For most practical problems on standard computers, a default value of 0.0001 is adequate for EPS. When $\mathrm{L}-\mathrm{U}$ factorization is successfully completed, the matrix A will contain matrices $\mathbf{L}$ and $\mathbf{U}$. A FORTRAN program for such a subroutine is as follows:

```
    SUBROUTINE LU (A,ICOL,N,EPS)
    DIMENSION A(N,N),ICOL(N)
    DO 10 K=1,N
10 ICOL(K)=K
NM1=N-1
DO 50 I=1,NM1
    PIV=ABS(A(I,I))
    IPIV=I
    IP1=I+1
    DO 20 K=IP1,N
        TEMP=ABS(A(I ,K))
        IF (TEMP.LE.PIV) GO TO 20
        PIV=TEMP
        IPIV=K
```

```
20
    CONTINUE
        IF (PIV.LT.EPS) GO TO 60
        IF (IPIV.EQ.I) GO TO 40
        I I=ICOL(I )
        IOOL(I )=IOOL(IPIV)
        IOOL(IPIV ) = I I
        DO 30 J=1,N
        TEMP=A(J , I )
        A(J,I )=A(J , IPIV)
        A(J,IPIV )=TEMP
        DO 50 J=IP1,N
        A(J,I )=A(J,I )/A(I ,I )
        DO 50 K=IP1,N
                A(J,K)=A(J ,K)-A(J ,I )*A(I ,K)
        OONTINUE
    RETURN
60 WRITE (1,200)
    STOP
200 FORMAT(5X,****THE MATRIX IS SINGULAR***')
    END
```

Subroutine LINEAR. This subroutine solves a set of linear equations in the form $\mathbf{A x}=\mathbf{c}$ by calling subroutine $\mathbf{L U}$ to factorize matrix $\mathbf{A}$ into $\mathbf{L}$ and $\mathbf{U}$ matrices. The argument parameters in this subroutine are as follows:
A. The given $\mathrm{N} \times \mathrm{N}$ matrix, either as the original $\mathbf{A}$ matrix (ILU $=1$ ), or in the form of $\mathrm{LU}(\mathrm{ILU}=0)$. In either case, on return, A contains the $\mathbf{L}$ and $\mathbf{U}$ matrices.
C An N -vector containing the right side of the known quantities $\mathbf{c}$, which upon return will contain the solution vector $\mathbf{x}$.
W An N -vector for work space.
ICOL Integer N -vector, which upon return will contain the column indices.
$\mathrm{N} \quad$ Number of rows (columns) of matrix $\mathbf{A}$.
ILU An index that must be set to 0 or 1 by the calling program. If EQ. 0 , $\mathbf{L}$ and $\mathbf{U}$ matrices are already available in A. If EQ. $1, \mathbf{L}$ and $\mathbf{U}$ matrices are not available; i.e., this subroutine must call subroutine LU.

EPS Test value for deviation from zero due to round-off error.
If L-U factorization must be employed on a matrix, then ILU must be set to 1 by the calling program. In this case, this subroutine will call subroutine LU to determine the $\mathbf{L}$ and $\mathbf{U}$ matrices and also the ICOL vector. However, if the $\mathbf{L}$ and $\mathbf{U}$ matrices and ICOL vector are already available, then ILU can be set to 0 by the calling program. This subroutine performs the steps of Eqs. 3.25 and 3.26 and stores the solution vector in the vector C . The dimensions of matrix A and vectors $\mathrm{C}, \mathrm{W}$, and ICOL must be set properly by the calling program. A FORTRAN program for such a subroutine is as follows:

```
SUBROUTINE LINEAR (A,C,W,IOOL,N,ILU,EPS)
DIMENSION A(N,N),C(N),W(N),ICOL(N)
IF (ILU.GT.0) CALL LU (A,IOOL,N,EPS)
DO 10 J=1,N
```

C.... Perform forward elimination step. LY=C
DO $30 \mathrm{~J}=2, \mathrm{~N}$
SUM=W(J)
$\mathrm{JM} 1=\mathrm{J}-1$
DO $20 \mathrm{~K}=1$, JM1
SUM=SUM-A(J,K) *W(K)
20
$W(J)=S U M$
C.....Perform back substitution step. UX=Y
$W(N)=W(N) / A(N, N)$
NP $1=\mathrm{N}+1$
DO $50 \mathrm{~J}=2, \mathrm{~N}$
$\mathrm{I}=\mathrm{NP} 1-\mathrm{J}$
SUM=W(I)
$I P 1=I+1$
DO $40 \mathrm{~K}=\mathrm{IP} 1, \mathrm{~N}$
SUM $=$ SUM-A $(I, K) * W(K)$

```

```

C.....Permute the solution vector to its original form
DO $60 \mathrm{~J}=1, \mathrm{~N}$
$60 \quad C(I C O L(J))=W(J)$
RETURN
END

```

\section*{Example 3.12}

Write a computer program, making use of subroutines LINEAR and LU, to solve a set of linear algebraic equations.
```

C**** EXAMPLE 3.12
DIMENSION B(120), I (10)
DATA EPS / 0.0001 /
C.....Read no. of rows (columns)
WRITE $(1,200)$
$\operatorname{READ}(1, *$ ) N
C.....Pointers for subarrays
N1=1
$\mathrm{N} 2=\mathrm{N} 1+\mathrm{N} * \mathrm{~N}$
$\mathrm{N} 3=\mathrm{N} 2+\mathrm{N}$
NUSED $=\mathrm{N} 3+\mathrm{N}-1$
C. ....Perform L-U factorization: ILU=1
ILU=1
CALL SOLVE (B(N1), B(N2), B(N3), I,N,ILU,EPS)
STOP
200 FORMAT (5X, 'ENTER N')
END
SUBROUTINE SOLVE (A,C,W, ICOL,N,ILU,EPS)
DIMENS ION A(N,N),C(N),W(N), IOOL(N)
C. ....Read the matrix A row by row
DO $10 \mathrm{~J}=1, \mathrm{~N}$
$\operatorname{WRITE}(1,200) \mathrm{J}$
$\operatorname{READ}(1, *)(A(J, K), K=1, N)$
C.....Read the right-hand-side vector $C$
$20 \operatorname{WRITE}(1,210)$
$\operatorname{READ}(1, *)(C(J), J=1, N)$
C.....Solve $A X=C$
CALL LINEAR (A,C,W, ICOL,N, ILU, EPS)
C.....Report the solution vector
$\operatorname{WRITE}(1,220)(C(J), J=1, N)$

```
```

C....Check for another vector C
WRITE(1, 230)
READ (1,* ) IC
IF (IC.EQ.0) RETURN
ILU=0
GOTO 20
200 FORMAT( 5X,'ENTER ROW', I 5)
210 FORMAT( 5X, 'ENTER VECTOR C')
220 FORMAT(5X,'THE SOLUTION IS',/,10F10.5)
230 FORMAT(5X,'IF ANOTHER VECIOR C IS GOING TO BE GIVEN',/,
+ 7X,'THEN ENTER 1, OTHERWISE ENTER 0')
END

```

Solution This program is written in a general form. It can accept up to 10 equations in 10 unknowns. If a set of more than 10 equations is to be solved, then the dimensions of the B and I arrays must be changed to 12 N and N , respectively, where N is the number of equations.

Note how the B array is split into smaller subarrays. This technique will be used frequently in all of the programs in this text. For a set of equations \(\mathbf{A x}=\mathbf{c}\), the matrix \(\mathbf{A}\) and vector \(\mathbf{c}\) are entered. After the solution is obtained, the user may enter a different vector \(\mathbf{c}\) to obtain a new solution. This process may be repeated as many times as needed.

\subsection*{3.4 NONLINEAR ALGEBRAIC EQUATIONS}

One of the most frequently occurring problems in scientific work is to find the roots of one or a set of nonlinear algebraic equations of the form
\[
\begin{equation*}
\Phi(\mathbf{x})=0 \tag{3.36}
\end{equation*}
\]
i.e., zeros of the functions \(\boldsymbol{\Phi}(\mathbf{x})\). The functions \(\boldsymbol{\Phi}(\mathbf{x})\) may be given explicitly or as transcendental functions. Kinematic analysis of mechanical systems is one example for which solution of constraint equations of the form of Eq. 3.36 is required. In this case, the explicit form of the constraint equations is available.

Numerous methods are available to find the zeros of Eq. 3.36. However, depending on the application, some methods may have better convergence properties than others, and some may be more efficient. In either case, the methods are, in general, iterative. The most common and frequently used method is known as the Newton-Raphson method.

\subsection*{3.4.1 Newton-Raphson Method for One Equation in One Unknown}

Consider the equation
\[
\begin{equation*}
\Phi(x)=0 \tag{3.37}
\end{equation*}
\]
to be nonlinear in the unknown \(x\). The Newton-Raphson iteration is stated as
\[
\begin{equation*}
x^{j+1}=x^{j}-\frac{1}{\Phi_{x}\left(x^{j}\right)} \Phi\left(x^{j}\right) \tag{3.38}
\end{equation*}
\]
where
\[
\begin{equation*}
\Phi_{x}\left(x^{\prime}\right)=\frac{d \Phi(x)}{d x} \quad \text { at } \quad x=x^{j} \tag{3.39}
\end{equation*}
\]
and the superscripts \(j\) and \(j+1\) are the iteration numbers. The Newton-Raphson algorithm produces a sequence of values, as follows:
\[
\begin{aligned}
& x^{1}=x^{0}-\frac{\Phi\left(x^{0}\right)}{\Phi_{x}\left(x^{0}\right)} \\
& x^{2}=x^{1}-\frac{\Phi\left(x^{1}\right)}{\Phi_{x}\left(x^{1}\right)}
\end{aligned}
\]
where \(x^{0}\) is the initial estimate of the solution of Eq. 3.37. The sequence of values, in many problems, will approach a root of \(\Phi(x)\).

The geometry of Newton-Raphson iteration is shown in Fig. 3.10. The NewtonRaphson method, when it works, is very efficient, but restrictions on the method are seldom discussed. The sketches in Figs. 3.11 through 3.13 show how the method may diverge or may converge to an unwanted solution. Since the Newton-Raphson method will not always converge, it is essential to terminate the process after a finite number of iterations.

\subsection*{3.4.2 Newton-Raphson Method for \(\boldsymbol{n}\) Equations in \(\boldsymbol{n}\) Unknowns}

Consider \(n\) nonlinear algebraic equations in \(n\) unknowns,
\[
\begin{equation*}
\Phi(x)=0 \tag{3.40}
\end{equation*}
\]
where a solution vector \(\mathbf{x}\) is to be found. The Newton-Raphson algorithm for \(n\) equations is stated as
\[
\begin{equation*}
\mathbf{x}^{j+1}=\mathbf{x}^{j}-\boldsymbol{\Phi}_{\mathbf{x}}^{-1}\left(\mathbf{x}^{\prime}\right) \boldsymbol{\Phi}\left(\mathbf{x}^{\prime}\right) \tag{3.41}
\end{equation*}
\]


Figure 3.10 Graphic representation of Newton-Raphson method.


Figure 3.11 Root at infection point.


Figure 3.12 Multiple roots.


Figure 3.13 Divergence near a local minimum or maximum.
where \(\boldsymbol{\Phi}_{\mathbf{x}}^{-1}\left(\mathbf{x}^{j}\right)\) is the inverse of the Jacobian matrix evaluated at \(\mathbf{x}=\mathbf{x}^{j}\). Equation 3.41 can be identified as the \(n\)-dimensional version of Eq. 3.38. The term \(\boldsymbol{\Phi}\left(\mathbf{x}^{j}\right)\) on the right side of Eq. 3.41 is known as the vector of residuals, which corresponds to the violation in the equations.

Equation 3.41 may be restated as a two-step operation:
\[
\begin{gather*}
\boldsymbol{\Phi}_{\mathbf{x}}\left(\mathbf{x}^{j}\right) \Delta \mathbf{x}^{j}=-\boldsymbol{\Phi}\left(\mathbf{x}^{j}\right)  \tag{3.42}\\
\mathbf{x}^{j+1}=\mathbf{x}^{j}+\Delta \mathbf{x}^{j} \tag{3.43}
\end{gather*}
\]
where Eq. 3.42 , which is a set of \(n\) linear equations, is solved for \(\Delta \mathbf{x}^{j}\). Then, \(\mathbf{x}^{j+1}\) is evaluated from Eq. 3.43. Gaussian elimination or L-U factorization methods are frequently employed to solve Eq. 3.42. The term \(\Delta \mathbf{x}^{j}=\mathbf{x}^{j+1}-\mathbf{x}^{j}\), known as the Newton difference, shows the amount of correction to the approximated solution in the \(j\) th iteration. The computational procedure is stated as follows:

\section*{ALGORITHM NR-I}
(a) Set the iteration counter \(j=0\).
(b) An initial estimate \(\mathbf{x}^{0}\) is made for the desired solution.
(c) The functions \(\boldsymbol{\Phi}\left(\mathbf{x}^{j}\right)\) are evaluated. If the magnitudes of all of the residuals \(\Phi_{i}\left(\mathbf{x}^{j}\right)\), \(i=1, \ldots, n\), are less than a specified tolerance \(\varepsilon\), i.e., if \(\left|\Phi_{i}\right|<\varepsilon\), \(i=1, \ldots, n\), then \(\mathbf{x}^{j}\) is the desired solution; therefore terminate. Otherwise, go to (d).
(d) Evaluate the Jacobian matrix \(\boldsymbol{\Phi}_{\mathbf{x}}\left(\mathbf{x}^{j}\right)\) and solve Eqs. 3.42 and 3.43 for \(\mathbf{x}^{j+1}\).
(e) Increment \(j\); i.e., set \(j\) to \(j+1\). If \(j\) is greater than a specified allowed number of iterations, then stop. Otherwise go to (c).

Algorithm NR-I is stated for the Newton-Raphson method in its simplest form. There are numerous techniques that can be included in the algorithm to improve its convergence. These techniques are not discussed in this text. Interested readers are referred to textbooks on numerical analysis.

\section*{Example 3.13}

Figure 3.14 shows a disk that is pressed against a plane surface that passes through point \(A\). Apply the Newton-Raphson method to find \(\Phi_{2}\) and \(d\) when \(\Phi_{1}=30^{\circ}\).

Solution Two constraint equations may be written, as follows:
\[
\begin{align*}
& \Phi_{1} \equiv b \cos \phi_{1}+a \cos \phi_{2}-d=0 \\
& \Phi_{2} \equiv b \sin \phi_{1}+a \sin \phi_{2}-r=0 \tag{1}
\end{align*}
\]

In order to analyze this system using the coordinate partitioning method of Sec. 3.2.1, the dependent and independent coordinates are taken to be \(\mathbf{u} \equiv\left[\Phi_{2}, d\right]^{T}\), and \(\mathbf{v} \equiv\left[\Phi_{1}\right]\). Hence, Eq. 1 may be rewritten as
\[
\begin{align*}
& \Phi_{1}=2 \cos \phi_{2}-d+10 \cos \phi_{1}=0 \\
& \Phi_{2}=2 \sin \phi_{2}-4+10 \sin \phi_{1}=0 \tag{2}
\end{align*}
\]

The Jacobian matrix for this system is
\[
\boldsymbol{\Phi}_{\mathbf{q}} \equiv\left[\begin{array}{rrr}
-2 \sin \phi_{2} & -1 & -10 \sin \phi_{1} \\
2 \cos \phi_{2} & 0 & 10 \cos \phi_{1}
\end{array}\right]
\]
where \(\mathbf{q}=\left[\phi_{2}, d, \phi_{1}\right]^{T}\), or
\[
\boldsymbol{\Phi}_{\mathbf{u}} \equiv\left[\begin{array}{rr}
-2 \sin \phi_{2} & -1  \tag{3}\\
2 \cos \phi_{2} & 0
\end{array}\right]
\]
and
\[
\Phi_{v} \equiv\left[\begin{array}{r}
-10 \sin \phi_{1}  \tag{4}\\
10 \cos \phi_{1}
\end{array}\right]
\]

For the Newton-Raphson algorithm, \(\Delta \mathbf{u}\) is evaluated by solving
\[
\left[\begin{array}{rr}
-2 \sin \phi_{2} & -1  \tag{5}\\
2 \cos \phi_{2} & 0
\end{array}\right]\left[\begin{array}{c}
\Delta \phi_{2} \\
\Delta d
\end{array}\right]=\left[\begin{array}{r}
\Phi_{1} \\
-\Phi_{2}
\end{array}\right]
\]

\[
\begin{aligned}
& a=2 \mathrm{~m} \\
& r=4 \mathrm{~m} \\
& b=10 \mathrm{~m}
\end{aligned}
\]

Figure 3.14 Roll with slip (Example 3.13).

Since Eq. 5 is simple to solve, there is no need to apply a numerical technique. It is found that
\[
\begin{align*}
\Delta \phi_{2} & =\frac{-\Phi_{2}}{2 \cos \phi_{2}} \\
\Delta d & =\Phi_{1}+\Phi_{2} \tan \phi_{2} \tag{6}
\end{align*}
\]

Iterative results using this formula are presented in the accompanying table, where \(\phi_{1}=30^{\circ}\). After three iterations, \(\phi_{2}=5.76 \mathrm{rad}\left(330^{\circ}\right)\) and \(d=10.3924\), and the residuals are \(\Phi_{1}=-0.0003\) and \(\Phi_{2}=0.00007\), which are small enough to terminate the process. Note that, since \(\Delta \phi_{2}\) is found in radians from Eq. 6, then either \(\phi_{2}\) must be converted to radians (from degrees), or \(\Delta \phi_{2}\) must be converted to degrees (from radians).
\begin{tabular}{ccccccc}
\hline \begin{tabular}{c} 
Iteration \\
number
\end{tabular} & \multicolumn{1}{c}{\(\phi_{2}^{\dagger}\)} & \(d\) & \(\Phi_{1}\) & \(\Phi_{2}\) & \(\Delta \phi_{2}\) & \(\Delta d\) \\
\hline 1 & \(5.59\left(326^{\circ}\right)\) & 10.0 & 0.19 & -0.29 & 0.19 & 0.43 \\
\hline 2 & 5.77 & 10.43 & 0.026 & 0.0206 & -0.0118 & -0.03759 \\
\hline 3 & 5.76 & 10.3924 & -0.0003 & 0.00007 & & \\
\hline
\end{tabular}
\({ }^{\dagger}\) All angles should be in radians.

\subsection*{3.4.3 A Subroutine for Nonlinear Algebraic Equations}

The Newton-Raphson algorithm of Section 3.4.2 is represented here in the form of a subroutine that can be embedded in programs to solve sets of nonlinear algebraic equations. This subroutine makes use of subroutines LINEAR and LU to solve Eqs. 3.42 and 3.43 iteratively. This subroutine is written in the simplest possible form, which can be modified easily.

Subroutine NEWTON. The argument parameters in this subroutine are as follows:
\begin{tabular}{ll} 
A, C, W, & \\
ICOL, & \\
N, EPS & Refer to subroutine LINEAR \\
NRMAX & Maximum number of iterations allowed \\
FEPS & Error tolerance on the Newton differences \\
X & Vector of dependent coordinates (unknowns)
\end{tabular}

This subroutine allows a maximum of NRMAX Newton iterations for finding the solution vector X to the set of nonlinear algebraic equations. If the solution is not found in NRMAX iterations, the subroutine terminates the process with a CONVERGENCE FAILED message. In each iteration, a call to a user-supplied subroutine FUNCT is made. This subroutine must provide the Jacobian matrix \(\boldsymbol{\Phi}_{\mathrm{x}}\) and the constraint violations
\(\boldsymbol{\Phi}\) (refer to Eq. 3.41), for the particular problem at hand, in matrix A and array C, respectively. Equation 3.42 is then solved by a call to subroutine LINEAR. The Newton differences \(\Delta \mathbf{x}\) are returned in array C , which is employed to correct the vector of unknown X. Note that Eqs. 3.42 and 3.43 can be written as
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{x}}\left(\mathbf{x}^{j}\right) \Delta \mathbf{x}^{j}=\boldsymbol{\Phi}\left(\mathbf{x}^{\prime}\right) \tag{3.44}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{x}^{j+1}=\mathbf{x}^{j}-\Delta \mathbf{x}^{j} \tag{3.45}
\end{equation*}
\]
which are employed in this subroutine. Finally, if all of the Newton differences (in absolute value) that are stored in \(C\) are less than the specified tolerance FEPS, the subroutine successfully terminates the process by returning to the calling routines. A FORTRAN program for such a subroutine is as follows;
```

        SUBROUTINE NEWION/(A,C,W, ICOL,N,EPS,NRMAX,FEPS,X)
    ```
        DIMENSION A(N,N)/C(N),W(N),ICOL(N),X(N)
        DO \(20 I=1\), NRMAX
            CALL FUNCT (A,C,N)
                CALL LINEAR (A,C,W, ICOL,N,1,EPS)
                IOONVR=0
                DO \(10 \mathrm{~J}=1, \mathrm{~N}\)
                IF (ABS (C(J)).GT.FEPS) ICONVR=1
                \(\mathrm{X}(\mathrm{J})=\mathrm{X}(\mathrm{J})-\mathrm{C}(\mathrm{J})\)
            IF (ICONVR) 30,30,20
                            20 CONTINUE
                        WRITE \((1,200)\)
        STOP
    30 RETURN
200 FORMAT(5X,'***CONVERGENCE FAILED***')
        END

\section*{Example 3.14}

Write a computer program, making use of subroutine NEWTON, to solve the problem of Example 3.13.

Solution The main routine presented here for this example is written in a general form. It can handle problems with up to 10 independent variables \(\mathbf{v}\) and 10 dependent variables \(\mathbf{u}\), without the need to increase the dimension of the arrays. The program asks the user for the following information:

Number of independent variables
Number of dependent variables
Known values for the independent variables
Initial estimates for the dependent variables
The main routine calls subroutine NEWTON, which in turn calls subroutines LINEAR (and LU) and FUNCT.

The constraint equations of Eq. 1 and the Jacobian matrix \(\boldsymbol{\Phi}_{u}\) of Eq. 3 are formulated in subroutine FUNCT. In this subroutine, the array F, which has two elements, contains the constraint violations, and the \(2 \times 2\) array (matrix) A contains the Jacobian entries.
```

C***** EXAMPLE 3.14
OOMMON /EXAMPL/ U(10),V(10)
DIMENSION B(120),I(10)
DATA EPS/0.0001/, FEPS/0.001/, NRMAX/25/
C.....Initialize the variables
WRITE (1,200)
READ (1,* ) NV,N
IF (NV.EQ.0) GOTO 10
WRITE(1,210)
READ (1,* ) (V(J) , J=1,NV)
10 WRITE (1,220)
READ (1,* ) (U(J), J=1,N)
C.....Pointers for the subarrays
N1=1
N2=N1+N*N
N3=N2+N
NUSED-N3+N-1
C.....Perform Newt on-Raphson iteration
CALL NEWION (B(N1),B(N2),B(N3),I,N,EPS,NRMAX, FEPS,U)
WRITE(1,230) (U(J),J=1,N)
STOP
200 FORMAT(5X,'ENTER NO. OF INDEPENDENT VARIABLES V',/,
+ 8X,'AND NO. OF DEPENDENT VARIABLES U')
210 FORMAT( 5X,'ENTER VALUES FOR THE INDEPENDENT VARIABLES')
220 FORMAT(5X,'ENTER ESTIMATES FOR THE DEPENDENT VARIABLES')
230 FORMAT('THE SOLUTION TO DEPENDENT VARIABLES IS:',/,10F10.5)
END
SUBROUTINE FUNCT (A,F,N)
OCMMON /EXAMPL/ U(10),V(10)
DIMENS ION A(N,N),F(N)
C***** EXAMPLE 3.13
C.....Constraint equations
F(1) = 2.0*COS(U(1))-U(2)+10.0*}\operatorname{COS}(V(1)
F(2) = 2.0*SIN(U(1))-4.0 +10.0*SIN(V(1))
C.....Jacobian matrix
A(1,1) =-2.0*SIN(U(1))
A(1,2) =-1.0
A(2,1) = 2.0*\operatorname{OSS}(U(1))
A(2,2) = 0.0
RETURN
END

```

\section*{PROBLEMS}
3.1 For each of the planar mechanical systems shown in Fig. P.3.1, answer the following questions:
(a) Is the system an open loop or a closed loop?
(b) If the system contains any closed loops, identify all of the closed loops.
(c) Determine the number of degrees of freedom of the system.
(d) Identify all of the kinematic joints.

Note: For mechanisms (b) and (i) consider two cases: where the wheel(s) do(es) not slip, or where the wheel(s) slip(s).
3.2 Determine which of the following constraint equations are nonholonomic:
(a) \(2 \cos \phi_{1}+3.6 d_{2} \cos \phi_{3}-3.1=0\)
(b) \(x_{4}-3 \cos \phi_{4}-x_{6}+2.5 \sin \phi_{6}=0\)

(a)

(d)

(g)

(b)

(e)

(h)

(c)

(f)

(i)

Figure P.3.1
(c) \(\dot{x}_{2}+0.6 t+0.1=0\)
(d) \(x_{2} \dot{y}_{3}+y_{3} \dot{x}_{2}=0\)
(e) \(\left(x_{1}-x_{3}\right)^{2}+\left(y_{1}-y_{3}\right)^{2}-d^{2}>0\)
(f) \(-e_{1} \dot{e}_{0}+e_{0} \dot{e}_{1}-e_{3} \dot{e}_{2}+e_{2} \dot{e}_{3}=0\)
3.3 For the slider-crank mechanism shown in Fig. P.3.3, assume \(l_{1}=1.2\) and \(l_{2}=2.6\).
(a) Write the constraint equations in terms of the coordinates \(\phi_{1}, \phi_{2}\), and \(d\). From these equations, derive the velocity and acceleration equations.
(b) If \(\phi_{1}\) is the driving coordinate, then for \(\phi_{1}=0.8 \mathrm{rad}, \dot{\phi}_{1}=0.1 \mathrm{rad} / \mathrm{s}\), and \(\ddot{\phi}_{1}=0\), find the remaining coordinates, velocities, and accelerations.
(c) If the slider is the driving link, then for \(d=2.5, \dot{d}=-0.2\), and \(\ddot{d}=-0.06\), find the remaining coordinates, velocities, and accelerations.


Figure P.3.3
3.4 Points \(P\) and \(Q\) are defined on the coupler of a four-bar mechanism as shown in Fig. P.3.4.

Assume \(l_{1}=0.5, l_{2}=1.2, l_{3}=0.8, l_{4}=0.7, l_{5}=0.15\), and \(l_{6}=0.2\).


Figure P.3.4
(a) Derive the constraint, velocity, and acceleration equations (loop \(A B C D\) ).
(b) Write expressions for \(x^{P}, y^{P}, x^{Q}\), and \(y^{Q}\) in terms of the coordinates.
(c) From (b), derive expressions for the velocity and acceleration of \(P\) and \(Q\).
(d) For \(\phi_{1}=\pi / 4, \dot{\phi}_{1}=-0.1 \pi\), and \(\ddot{\phi}_{1}=0\), solve the equations obtained in (a) to find the remaining coordinates, velocities, and accelerations.
(e) Use the results of (d) and substitute in the expressions of (b) and (c) to obtain the coordinates, velocities, and accelerations of \(P\) and \(Q\).
3.5 Select some of the mechanical systems of Prob. 3.1 having only closed loops, definc a proper set of coordinates, and then derive the constraint, velocity, and acceleration equations.
3.6 Assume that \(y\) is the independent coordinate in the constraint equations
\[
\begin{array}{r}
x^{2}+x y+y z-2 x+3=0 \\
y^{2}-3 z^{2}+x z+2 y-1=0
\end{array}
\]

Apply two different approaches to find the velocity and acceleration equations for the coordinate partitioning method:
(a) Take the first and the second time derivatives of the constraints.
(b) Employ Eqs. 3.8 and 3.11 directly.
3.7 Write the constraint, velocity, and acceleration equations for Prob. 3.6 in the form of the appended driving constraint method. Assume that the independent variable is defined as \(y=\) \(c_{1}, \dot{y}=c_{2}\), and \(\ddot{y}=c_{3}\). Express these equations in the form of Eqs, 3.12, 3.14, and 3.16.
3.8 A driver constraint equation for the slider-crank mechanism in Prob. 3.3 is stated as
\[
\phi_{1}-0.8-0.1 t=0
\]

Write the constraint, velocity, and acceleration equations in the form of the appended driving constraint formulation. Solve these equations for \(t=0\) and compare the result with that obtained in Prob. 3.3(b).
3.9 Solve the system of equations
\[
\left[\begin{array}{rrrr}
2 & -1 & 0 & 1 \\
-2 & 2 & 0 & -3 \\
4 & 1 & 3 & -1 \\
0 & -1 & -2 & 5
\end{array}\right]\left[\begin{array}{l}
w \\
x \\
y \\
z
\end{array}\right]=\left[\begin{array}{l}
2 \\
8 \\
1 \\
2
\end{array}\right]
\]
by the following methods:
(a) Gaussian elimination
(b) Gauss-Jordan reduction
(c) L-U factorization
3.10 Solve the system of equations
\[
\left[\begin{array}{rrrr}
3 & 1 & -1 & 2 \\
-6 & -2 & 4 & 3 \\
0 & 3 & 2 & -2 \\
1 & 1 & -5 & 6
\end{array}\right]\left[\begin{array}{l}
w \\
x \\
y \\
z
\end{array}\right]=\left[\begin{array}{r}
0 \\
4 \\
1 \\
-10
\end{array}\right]
\]
by the Gaussian elimination method
(a) Without pivoting
(b) With row pivoting when necessary
(c) With column pivoting when necessary
3.11 Solve the system of equations in Prob. 3.9 by L-U factorization and
(a) No pivoting
(b) Partial pivoting with row interchange
(c) Partial pivoting with column interchange
(d) Full pivoting
3.12 For the system shown in Fig. P.3.12, assume no slipping between the wheel and the contacting surface. If \(O A=1.5, A B=2, B C=1, r=1.2\), and \(O E=0.2\), write the constraint equations for this system in terms of \(\phi_{1}, \phi_{2}, \phi_{3}\), and \(d\). Note that two equations can be written for the loop closure and one equation can be written for the no-slip condition. Furthermore, it is known that the system was initially assembled for \(d=3\) and \(\phi_{3}=120^{\circ}\). For the crank angle \(\phi_{1}=60^{\circ}\), apply the Newton-Raphson algorithm to solve the constraint equations for the unknown coordinates. Show the result of each iteration in a table.


Figure P.3.12
3.13 For the quick-return mechanism shown in Fig. P.3.13, derive the constraint, velocity, and acceleration equations. Solve these equations by writing a computer program using subroutines NEWTON and LINEAR. Assume \(O B=0.7, C D=2.1, O E=0.9\), and the configuration is for \(\phi_{1}=\pi / 6, \dot{\phi}_{1}=-0.2\), and \(\ddot{\phi}_{1}=0\).


Figure P.3.13
3.14 For the mechanism of Prob, 3.13 assume a driver constraint as
\[
\phi_{1}-\frac{\pi}{6}+0.2 t=0
\]

Start from \(t=0\) and increment \(t\) gradually to simulate the motion of the system for a complete cycle.
3.15 Modify the computer program of Prob. 3.13 to solve Prob. 3.12. Assume that the crank rotates with a constant angular velocity of \(0.2 \mathrm{rad} / \mathrm{s}\).
3.16 The four-bar mechanism shown in Fig. P.3.16 is used to advance a film strip inside a movie projector. \({ }^{17}\)


Figure P.3.16
(a) Write the constraint equations for the four-bar linkage \(A B C D\).
(b) Write expressions for the coordinates of point \(P\) in terms of \(\phi_{1}, \phi_{2}\), and \(\phi_{3}\).
(c) Develop a computer program to solve the constraint equations for a complete revolution of the crank and compute the coordinates of P .
(d) Plot the path of \(P\) and show where point \(P\) becomes engaged and disengaged with the film strip.

\section*{4}

\section*{Planar Kinematics}

If all links of a mechanical system undergo motion in one plane or in parallel planes, the system is said to be experiencing planar motion. In this chapter, only planar mechanisms, in which all links experience planar motion, are considered. A more descriptive explanation of conditions for planar motion is given in Chap. 9.

In Chap. 3 several examples of kinematic analyses were given using a minimal set of Lagrangian coordinates. Kinematic analysis of mechanical systems using Cartesian coordinates is no different in principle from the method of analysis with Lagrangian coordinates. The use of Cartesian coordinates, however, results in a larger number of coordinates and constraint equations. The number of degrees of freedom of a system, however, is the same regardless of the type of coordinates used. Since the number of independent coordinates is equal to the number of degrees of freedom of a system, then the number of dependent Cartesian coordinates is generally greater than the number of dependent Lagrangian coordinates.

Cartesian coordinates are used exclusively in this chapter and the remainder of the text.

\subsection*{4.1 CARTESIAN COORDINATES}

In order to specify the configuration or state of a planar mechanical system, it is first necessary to define coordinates that specify the location of each body. Let the \(x y\) coordinate system shown in Fig. 4.1 be a global reference frame. Define a body-fixed \(\xi_{i} \eta_{i}\) coordinate system embedded in body \(i\). Body \(i\) can be located in the plane by specifying the global coordinates \(\mathbf{r}_{i}=[x, y]_{i}^{T}\) of the origin of the body-fixed coordinate system and the angle \(\phi_{i}\) of rotation of this system relative to the global coordinate system. This angle is considered positive if the rotation from positive \(x\) axis to positive \(\xi_{i}\) axis is counterclockwise.


Figure 4.1 Locating point \(P\) relative to the body-fixed and global coordinate systems.

A point \(P_{i}\) on body \(i\) can be located from the origin of the \(\xi_{i} \eta_{i}\) axes by the vector \(\vec{s}_{i}^{P}\). The coordinates of point \(P_{i}\) with respect to the \(\xi_{i} \eta_{i}\) coordinate system are \(\xi_{i}^{P}\) and \(\eta_{i}^{P}\). The local (body-fixed) components of vector \(\vec{s}_{i}^{P}\) are shown as \(\mathbf{s}_{i}^{\prime P}=\left[\xi^{P}, \eta^{P}\right]_{i}^{T}\). Since \(P_{i}\) is a fixed point on body \(i, \xi_{i}^{P}\) and \(\eta_{i}^{P}\) are constants, and therefore \(\mathbf{s}_{i}^{\prime P}\) is a constant vector. The global \(x y\) components of vector \(\vec{s}_{i}^{P}\) are shown as \(\mathbf{s}_{i}^{P}\). The elements of \(s_{i}^{P}\) vary when body \(i\) rotates. Point \(P_{i}\) may also be located by its global coordinates \(\mathbf{r}_{i}^{P}=\left[x^{P}, y^{P}\right]_{i}^{T}\). It is clear that the components of \(\vec{r}_{i}^{P}\) are not necessarily constant, since body \(i\) may be in motion. Position vectors such as \(\mathbf{r}_{i}^{P}, \mathbf{s}_{i}^{P}, \mathbf{s}_{i}^{\prime P}, \mathbf{r}_{i}\), and so forth, are 3-vectors. However, in planar motion, since the \(z\) component of these vectors remains constant, the vectors are treated as 2 -vectors.

The relation between the local and global coordinates of point \(P_{i}\) is
\[
\begin{equation*}
\mathbf{r}_{i}^{P}=\mathbf{r}_{i}+\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P} \tag{4.1}
\end{equation*}
\]
where
\[
\mathbf{A}_{i}=\left[\begin{array}{rr}
\cos \phi & -\sin \phi  \tag{4.2}\\
\sin \phi & \cos \phi
\end{array}\right]_{i}
\]
is the rotational transformation matrix for body \(i\). The transformation matrix \(\mathbf{A}_{i}\) is the simplified form of the following \(3 \times 3\) matrix:
\[
\mathbf{A}_{i}=\left[\begin{array}{ccc}
\cos \phi & -\sin \phi & 0 \\
\sin \phi & \cos \phi & 0 \\
0 & 0 & 1
\end{array}\right]
\]

Equation 4.1 in expanded form can be written as
\[
\left[\begin{array}{l}
x^{P} \\
y^{P}
\end{array}\right]_{i}=\left[\begin{array}{l}
x \\
y
\end{array}\right]_{i}+\left[\begin{array}{rr}
\cos \phi & -\sin \phi \\
\sin \phi & \cos \phi
\end{array}\right]_{i}\left[\begin{array}{c}
\xi^{P} \\
\eta^{P}
\end{array}\right]_{i}
\]
or
\[
\begin{align*}
x_{i}^{P} & =x_{i}+\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{P} \sin \phi_{i} \\
y_{i}^{P} & =y_{i}+\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i} \tag{4.3}
\end{align*}
\]

Note that
\[
\begin{equation*}
\mathbf{s}_{i}^{p}=\mathbf{A}_{i} \mathbf{s}_{i}^{p} \tag{4.4}
\end{equation*}
\]
is the relationship between the local and global components of vector \(\vec{s}_{i}^{P}\).

The vector of coordinates for body \(i\) is denoted by the vector
\[
\begin{align*}
\mathbf{q}_{i} & =\left[\mathbf{r}^{T}, \phi\right]_{i}^{T} \\
& =[x, y, \phi]_{i}^{T} \tag{4.5}
\end{align*}
\]

For a mechanical system with \(b\) bodies, the coordinate vector is the \(3 \times b\) vector
\[
\begin{align*}
\mathbf{q} & =\left[\mathbf{q}_{1}^{T}, \mathbf{q}_{2}^{T}, \ldots, \mathbf{q}_{b}^{T}\right]^{T} \\
& =\left[x_{1}, y_{1}, \phi_{1}, x_{2}, y_{2}, \phi_{2}, \ldots, x_{b}, y_{b}, \phi_{b}\right]^{T} \tag{4.6}
\end{align*}
\]
where \(\mathbf{q}\) without a subscript denotes the vector of coordinates for the entire system.
An algebraic 3-vector with a superscript prime is one that contains local components; e.g., \(\mathbf{s}_{i}^{P}, \mathbf{s}_{j}^{\prime}, \mathbf{n}_{j}^{\prime}\), or \(\boldsymbol{\omega}_{i}^{\prime}\). An algebraic 3-vector without a superscript prime is one that contains global components; e.g., \(\mathbf{s}_{i}^{P}, \mathbf{s}_{j}, \mathbf{n}_{j}, \boldsymbol{\omega}_{j}, \mathbf{r}_{i}^{P}\), or \(\mathbf{r}_{j}\).

\section*{Example 4.1}

For the mechanism shown in the illustration, body 3 has translational coordinates \(\mathbf{r}_{3}=[1.2,2.5]^{T}\) and \(\phi_{3}=5.67 \mathrm{rad}\). Point \(B\) on body 3 has local coordinates \(\xi_{3}^{B}=\) -1.8 and \(\eta_{3}^{B}=1.3\). Find the global components of vector \(\vec{s}_{3}^{B}\) and the global coordinates of \(B\).


Solution The global components of \(\vec{s}_{3}^{B}\) are found to be
\[
\mathbf{s}_{3}^{B}=\mathbf{A}_{3} \mathbf{s}_{3}^{\prime B}=\left[\begin{array}{rr}
\cos \phi & -\sin \phi \\
\sin \phi & \cos \phi
\end{array}\right]\left[\begin{array}{r}
-1.8 \\
1.3
\end{array}\right]=\left[\begin{array}{r}
-0.72 \\
2.10
\end{array}\right]
\]

Hence the global coordinates of \(B\) are
\[
\mathbf{r}_{3}^{B}=\mathbf{r}_{3}+\mathbf{s}_{3}^{B}=\left[\begin{array}{l}
1.2 \\
2.5
\end{array}\right]+\left[\begin{array}{r}
-0.72 \\
2.10
\end{array}\right]=\left[\begin{array}{l}
0.48 \\
4.60
\end{array}\right]
\]

\subsection*{4.2 KINEMATIC CONSTRAINTS}

In a mechanical system, the links and bodies may be interconnected by one or more kinematic joints. For example, the quick-return mechanism shown in Fig. 4.2(a) consists of five moving bodies interconnected by five revolute joints and two sliding joints. Since this mechanism undergoes planar motion, the motion of each moving body is described by three coordinates - two translational and one rotational. The kinematic joints in this system can be described as algebraic constraint equations.

In the following subsections, several commonly used kinematic joints are formulated. Some of these joints fall under the lower-pair category (LP) and the rest fall in the higher-pair (HP) category. The technique employed to formulate kinematic constraint equations for these joints may be applied to most other commonly used or specialpurpose joints.

In general, formulation of lower-pair joints does not require any information on the shape (outline) of the connected bodies. For example, the quick-return mechanism of Fig. 4.2(a) can also be represented as shown in Fig. 4.2(b), where the body outlines are drawn arbitrarily. To derive constraint equations describing each joint, one need know only the position of the joint with respect to the bodies to which it is connected.

In some higher-pair joints, either the entire shape or a partial shape of the body outline must be known. For example, in analyzing the motion of a cam-follower pair, the full or partial outlines of the cam and the follower must be described. In some other higher-pair joints, instead of the shape of the outline, the shape or curvature of a slot on one of the bodies must be known.

In the following subsections, the constraint equations are denoted by \(\boldsymbol{\Phi}\) with a superscript indicating the constraint type and the number of algebraic equations of that


Figure 4.2 Quick-return mechanism: (a) schematic presentation and (b) its equivalent representation without showing the actual outlines.
expression. For example, \(\boldsymbol{\Phi}^{(r, 2)}\) denotes the revolute joint constraint which contains two equations, and \(\Phi^{(r-1)}\) denotes the revolute-translational joint constraint which contains one equation.

\subsection*{4.2.1 Revolute and Translational Joints (LP)}

Revolute and translational joints are lower-pair kinematic joints. Examples of revolute joints are joints \(A, B, C, D\), and \(O\) in the quick-return mechanism of Fig. 4.2. Schematic representation of a revolute joint connecting to bodies \(i\) and \(j\) is shown in Fig. 4.3. The center of the joint is denoted by point \(P\). This point can be considered to be two coincident points; point \(P_{i}\) on body \(i\) and point \(P_{j}\) on body \(j\). Location of point \(P\) on body \(i\) and body \(j\) can be described by the two vectors \(\vec{s}_{i}^{P}\) and \(\vec{s}_{j}^{P}\), where \(\mathbf{s}_{i}^{P}=\left[\xi^{P}, \eta^{P}\right]_{i}^{T}\) and \(\mathbf{s}_{j}^{\prime P}=\left[\xi^{P}, \eta^{P}\right]_{j}^{T}\) are constants. The constraint equations for the revolute joint are obtained from the vector loop equation
\[
\vec{r}_{i}+\vec{s}_{i}^{P}-\vec{s}_{j}^{P}-\vec{r}_{j}=\overrightarrow{0}
\]
or
\[
\mathbf{r}_{i}+\mathbf{s}_{i}^{p}-\mathbf{r}_{j}-\mathbf{s}_{j}^{p}=0
\]
which is equivalent to
\[
\begin{equation*}
\boldsymbol{\Phi}^{(r, 2)} \equiv \mathbf{r}_{i}+\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P}-\mathbf{r}_{j}-\mathbf{A}_{j} \mathbf{s}_{j}^{\prime P}=\mathbf{0} \tag{4.7}
\end{equation*}
\]

More explicitly, Eq. 4.7 can be written in the form
\[
\boldsymbol{\Phi}^{(r, 2)} \equiv\left[\begin{array}{l}
x_{i}^{P}-x_{j}^{P}  \tag{4.8}\\
y_{i}^{P}-y_{j}^{P}
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
\]

Equation 4.8 can be written in expanded form, using Eq. 4.3, as
\[
\boldsymbol{\Phi}^{(r, 2)} \equiv\left[\begin{array}{l}
x_{i}+\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{P} \sin \phi_{i}-x_{j}-\xi_{j}^{P} \cos \phi_{j}+\eta_{j}^{P} \sin \phi_{j}  \tag{4.9}\\
y_{i}+\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}-y_{j}-\xi_{j}^{P} \sin \phi_{j}-\eta_{j}^{P} \cos \phi_{j}
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
\]

The two constraints of Eq. 4.7 reduce the number of degrees of freedom of the system by 2 . Therefore, if the two bodies of Fig. 4.3 are not connected to any other bodies, then they have 4 degrees of freedom.


Figure 4.3 Revolute joint \(P\) connecting bodies \(i\) and \(j\).

In the quick-return mechanism of Fig. 4.2, the two sliders ( \(T_{1}\) and \(T_{2}\) ) are good examples of translational joints. This type of joint may appear in different shapes in a mechanism. Figure 4.4 illustrates several forms and presentations of translational joints. In a translational joint, the two bodies translate with respect to each other parallel to an axis known as the line of translation; therefore, there is no relative rotation between the bodies. For a translational joint, there are an infinite number of parallel lines of translation.


Figure 4.4 Different representations of a translational joint.
A constraint equation for eliminating the relative rotation between two bodies \(i\) and \(j\) is written as
\[
\begin{equation*}
\phi_{i}-\phi_{j}-\left(\phi_{i}^{0}-\phi_{j}^{0}\right)=0 \tag{4.10}
\end{equation*}
\]
where \(\phi_{i}^{0}\) and \(\phi_{j}^{0}\) are the initial rotational angles. In order to eliminate the relative motion between the two bodies in a direction perpendicular to the line of translation, the two vectors \(\vec{s}_{i}\) and \(\vec{d}\) shown in Fig. 4.5 must remain parallel. These vectors are defined by locating three points on the line of translation - two points on body \(i\) and one point on body \(j\). This condition is enforced by letting the vector product of these two vectors be


Figure 4.5 A translational joint between bodies \(i\) and \(j\).
zero. A simple method would be to define another vector \(\vec{n}_{i}\) perpendicular to the line of translation and to require that \(\vec{d}\) remain perpendicular to \(\vec{n}_{i}\); i.e., that
\[
\mathbf{n}_{i}^{T} \mathbf{d}=0
\]
or
\[
\begin{equation*}
\left(x_{i}^{P}-x_{i}^{R}\right)\left(x_{j}^{P}-x_{i}^{P}\right)+\left(y_{i}^{P}-y_{i}^{R}\right)\left(y_{j}^{P}-y_{i}^{P}\right)=0 \tag{4.11}
\end{equation*}
\]
where
\[
\mathbf{n}_{i}=\left[\begin{array}{c}
x_{i}^{P}-x_{i}^{R} \\
y_{i}^{P}-y_{i}^{R}
\end{array}\right] \quad \mathbf{d}=\left[\begin{array}{c}
x_{j}^{P}-x_{i}^{P} \\
y_{j}^{P}-y_{i}^{P}
\end{array}\right]
\]

If \(n_{i}=s_{i}\), then
\[
\mathbf{n}_{i}=\left[\begin{array}{c}
x_{i}^{P}-x_{i}^{R} \\
y_{i}^{P}-y_{i}^{R}
\end{array}\right]=\left[\begin{array}{c}
-\left(y_{i}^{P}-y_{i}^{Q}\right) \\
x_{i}^{P}-x_{i}^{Q}
\end{array}\right]
\]

Therefore, Eqs. 4.10 and 4.11 yield the two constraint equations for a translational joint as \({ }^{\dagger}\)
\[
\boldsymbol{\Phi}^{(t, 2)}=\left[\begin{array}{c}
\left(x_{i}^{P}-x_{i}^{Q}\right)\left(y_{j}^{P}-y_{i}^{P}\right)-\left(y_{i}^{P}-y_{i}^{Q}\right)\left(x_{j}^{P}-x_{i}^{P}\right)  \tag{4.12}\\
\phi_{i}-\phi_{j}-\left(\phi_{i}^{0}-\phi_{j}^{0}\right)
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
\]

Note that a translational joint reduces the number of degrees of freedom of a system by 2 .

\section*{Example 4.2}

For the revolute and translational joints shown in the accompanying illustration, define the points needed in order to use Eqs. 4.9 and 4.12 and determine their coordinates.


Solution The body-fixed coordinates of point \(A\) are
\[
\xi_{3}^{A}=-1.6, \quad \eta_{3}^{A}=2.3, \quad \xi_{4}^{A}=0, \quad \eta_{4}^{A}=-1.5
\]
\({ }^{\dagger}\) The vector product of \(\vec{s}_{i}\) and \(\vec{d}\) gives directly the first constraint of Eq. 4.12.

These can be used for the revolute joint constraints of Eq. 4.9. For the translational joint, two points on body 2 and one point on body 3 are selected on a line parallel to the line of translation. The coordinates of these points are:
\(\xi_{2}^{B}=0, \quad \eta_{2}^{B}=0, \quad \xi_{2}^{C}=1, \quad \eta_{2}^{C}=0, \quad \xi_{3}^{E}=0, \quad \eta_{3}^{E}=-0.5\)
These coordinates can be used in the translational joint constraints of Eq. 4.11.

\subsection*{4.2.2 Composite Joints (LP)}

The total number of constraint equations and coordinates describing a mechanical system can be reduced if some of the joints and bodies are combined into composite joints. This technique only simplifies the analytical formulation without changing the physical kinematic characteristics of the system. For example consider three bodies connected by two revolute joints, as shown schematically in Fig. 4.6(a). This system requires nine coordinates (three per body) and four constraint equations (two per revolute joint). Therefore, this system has \(9-4=5\) degrees of freedom. This system may be represented by the kinematically equivalent system shown in Fig. 4.6(b). In this representation, body \(k\) and the two revolute joints are considered to be a revolute-revolute joint (rigid link) without any coordinates. This configuration requires six coordinates for bodies \(i\) and \(j\), and as will be shown presently, one constraint equation for the revolute-revolute joint. Therefore, this equivalent system has \(6-1=5\) degrees of freedom.


Figure 4.6 Two representations of the same system: (a) with three bodies and two revolute joints; (b) with two bodies and one revolute-revolute joint.

To formulate the constraint equation for a revolute-revolute joint, see Fig. 4.7, which illustrates two bodies connected by a link having length \(l\). The length of the link is the distance between the centers of two revolute joints. The joint connectivity points on bodies \(i\) and \(j\) are shown as \(P_{i}\) and \(P_{j}\). These points have fixed components \(\mathbf{s}_{i}^{\prime P}\) and \(\mathbf{s}_{j}^{\prime P}\) with respect to their corresponding body-fixed coordinate systems. The constraint equation can be derived by requiring that the magnitude of vector
\[
\mathbf{d}=\mathbf{r}_{j}+\mathbf{s}_{j}^{P}-\mathbf{r}_{i}-\mathbf{s}_{i}^{p}=\left[\begin{array}{l}
x_{j}^{P}-x_{i}^{p} \\
y_{j}^{P}-y_{i}^{p}
\end{array}\right]
\]
connecting points \(P_{i}\) and \(P_{j}\) remain constant and equal to \(l\). This constraint is written in the form
\[
\begin{equation*}
\Phi^{(r-r, 1)} \equiv \mathbf{d}^{T} \mathbf{d}-l^{2}=0 \tag{4.13}
\end{equation*}
\]


Figure 4.7 Revolute-revolute joint: a link connecting two bodies with two revolute joints.

Equation 4.13, if expanded, is written
\[
\begin{equation*}
\Phi^{(r-r, 1)} \equiv\left(x_{i}^{P}-x_{j}^{P}\right)^{2}+\left(y_{i}^{P}-y_{j}^{P}\right)^{2}-l^{2}=0 \tag{4.14}
\end{equation*}
\]

A second type of composite joint is called the revolute-translational joint. Figure 4.8 illustrates three examples of this type of joint. In Fig. 4.8(a), body \(i\) is connected to body \(k\) by a revolute joint and body \(k\) is connected to body \(j\) by a translational joint. The revolute joint, the translational joint, and body \(k\) can be combined into a composite joint in which point \(P\) on body \(i\) moves along an axis \(T\) on body \(j\). Similar composite joints can be used in the mechanism of Fig. 4.8(b) by eliminating roller \(k\) as a body. The mechanism of Fig. 4.8(c) is identical in principle to the other two mechanisms; here, point \(P\) on body \(i\) moves on an axis parallel to the line of translation.

Figure 4.9 illustrates schematically a revolute-translational joint between bodies \(i\) and \(j\). Point \(P_{i}\) on body \(i\) (the revolute joint) can move along the line of translation \(T\) (the translational joint) on body \(j\). Two arbitrary points \(P_{j}\) and \(Q_{j}\) are chosen on the line of translation. A constraint equation for this joint is found by requiring two vectors \(\vec{d}\) and \(\vec{s}_{j}\) to remain parallel. Or, if a vector \(\vec{n}_{j}\) is defined perpendicular to the line of translation, then
\[
\begin{equation*}
\mathbf{n}_{j}^{T} \mathbf{d}==0 \tag{4.15}
\end{equation*}
\]

(a)

(b)

(c)

Figure 4.8 Examples of composite revolute-translational joints.


Figure 4.9 Revolute-translational joint configuration.
provides the same condition. If \(s_{j}=n_{j}\), then
\[
\mathbf{n}_{j}=\left[\begin{array}{c}
x_{j}^{R}-x_{j}^{Q} \\
y_{j}^{R}-y_{j}^{Q}
\end{array}\right]=\left[\begin{array}{c}
-\left(y_{j}^{P}-y_{j}^{Q}\right) \\
x_{j}^{P}-x_{j}^{Q}
\end{array}\right] \quad \mathbf{d}=\left[\begin{array}{c}
x_{j}^{Q}-x_{i}^{P} \\
y_{j}^{Q}-y_{i}^{P}
\end{array}\right]
\]
and the constraint equation is written as
\[
\begin{equation*}
\Phi^{(t-r, 1)}=\left(x_{j}^{Q}-x_{i}^{P}\right)\left(y_{j}^{Q}-y_{j}^{P}\right)-\left(y_{j}^{Q}-y_{i}^{P}\right)\left(x_{j}^{Q}-x_{j}^{P}\right)=0 \tag{4.16}
\end{equation*}
\]

Figure 4.10 shows an example of a mechanism with two revolute-translational joints. The slider and the revolute joint at point \(B\) can be combined to represent a revolutetranslational joint. Similarly, the roller \(A\) and the revolute joint \(O\) can be combined and modeled as another revolute-translational joint.


Figure 4.10 A mechanism with two revo-lute-translational joints.

\subsection*{4.2.3 Spur Gears and Rack and Pinion (HP)}

Spur gears are used to transmit motion between parallel shafts. A pair of spur gears is shown in Fig. 4.11. The gears in Fig. 4.11(a) give the same motion to the shafts as the pair of equivalent rolling cylinders in Fig. 4.11(b). It is assumed that there is no slippage between the two cylinders.

The constraint formulation for a pair of spur gears as represented in Fig. 4.12(a) can be found easily. In this configuration the centers of the gears (shown here as circles)

(a)

(b)

Figure 4.11 (a) A pair of spur gears, and (b) a pair of equivalent rolling cylinders.

(a)

(b)

Figure 4.12 A pair of spur gears attached to (a) a nonmoving body and (b) a moving body.
do not move with respect to the \(x y\) coordinate system (ground or chassis). The points of contact \(R_{i}\) and \(R_{j}\), at time \(t\), have equal velocities; i.e.,
\[
\begin{equation*}
v_{i}^{R}=v_{j}^{R} \tag{a}
\end{equation*}
\]

If body-fixed \(\xi_{i} \eta_{i}\) and \(\xi_{j} \eta_{j}\) coordinate systems are defined for each gear, then Eq. \(a\) can be written as
\[
\begin{equation*}
\rho_{i} \dot{\phi}_{i}=-\rho_{j} \dot{\phi}_{j} \tag{b}
\end{equation*}
\]

Note that \(\dot{\phi}_{i}\) and \(\dot{\phi}_{j}\) always have opposite signs. Integrating Eq. \(b\) yields
\[
\begin{equation*}
\Phi^{(g-1,1)} \equiv\left(\phi_{i}-\phi_{i}^{0}\right) \rho_{i}+\left(\phi_{j}-\phi_{j}^{0}\right) \rho_{j}=0 \tag{4.17}
\end{equation*}
\]
where \(\phi_{i}^{0}\) and \(\phi_{j}^{0}\) are the initial conditions. Equation 4.17 is the constraint equation for a pair of spur gears with nonmoving centers. In this equation \(\phi_{i}\) and \(\phi_{j}\) must be assigned their accumulated angle of rotation; i.e., when bodies \(i\) and \(j\) rotate, the magnitudes of \(\phi_{i}\) and \(\phi_{j}\) may exceed \(2 \pi, 4 \pi, \ldots\).

The constraint formulation for a pair of spur gears attached to a moving body \(k\) is shown in Fig. 4.12(b). Since all three bodies, \(i, j\), and \(k\), can rotate, the no-slippage condition at the contact point between bodies \(i\) and \(j\) can be written as
\[
\begin{equation*}
\left(\dot{\phi}_{i}-\dot{\phi}_{k}\right) \rho_{i}=-\left(\dot{\phi}_{j}-\dot{\phi}_{k}\right) \rho_{j} \tag{c}
\end{equation*}
\]

Integrating Eq. \(c\) yields
\[
\begin{equation*}
\Phi^{(g-2,1)} \equiv\left[\left(\phi_{i}-\phi_{i}^{0}\right)-\left(\phi_{k}-\phi_{k}^{0}\right)\right] \rho_{i}+\left[\left(\phi_{j}-\phi_{j}^{0}\right)-\left(\phi_{k}-\phi_{k}^{0}\right)\right] \rho_{j}=0 \tag{4.18}
\end{equation*}
\]

If body \(k\) does not rotate, then \(\phi_{k}=\phi_{k}^{0}\) and Eq. 4.18 becomes identical to Eq. 4.17.
In the constraint formulation of Eqs. 4.17 and 4.18, it is assumed that bodies \(i\) and \(j\) remain in contact. This condition is provided by the revolute joints connecting the gears to a third body. Therefore, there is only one relative degree of freedom between bodies \(i\) and \(j\).

A rack and pinion can be considered a special form of a pair of spur gears in which the rack is a portion of a gear having an infinite pitch radius; thus its pitch circle is a straight line. A rack and pinion and its equivalent system, consisting of a rolling cylinder on a flat surface (a straight line), are shown in Fig. 4.13(a) and (b). It is assumed that no slippage occurs at the point of contact for the equivalent system.


Figure 4.13 (a) A rack-and-pinion mechanism, and (b) the equivalent system.

Constraint formulation for a rack and pinion is derived for the two configurations in Fig. 4.14. The first configuration of a rack and pinion is shown in Fig. 4.14(a), where the pinion (body \(i\) ) can rotate only with respect to the \(x y\) coordinate system and the rack can translate only parallel to the \(x\) axis. The condition for no slippage at the point of contact is written as
\[
\begin{equation*}
\dot{\phi}_{i} \rho_{i}=\dot{x}_{j} \tag{d}
\end{equation*}
\]

Integrating Eq. \(d\) yields
\[
\begin{equation*}
\Phi^{(r-p-1,1)} \equiv\left(\phi_{i}-\phi_{i}^{0}\right) \rho_{i}-\left(x_{j}-x_{j}^{0}\right)=0 \tag{4.19}
\end{equation*}
\]

A second configuration of a rank-and-pinion mechanism is shown in Fig. 4.14(b), where the pinion can both rotate and translate parallel to the rack and the \(x\) axis. The no-slip condition is
\[
\begin{equation*}
\dot{x}_{i}+\dot{\phi}_{i} \rho_{i}=\dot{x}_{j} \tag{e}
\end{equation*}
\]

Integrating Eq. e yields
\[
\begin{equation*}
\Phi^{(r--2,1)} \equiv\left(x_{i}-x_{i}^{0}\right)+\left(\phi_{i}-\phi_{i}^{0}\right) \rho_{i}-\left(x_{j}-\dot{x}_{j}^{0}\right)=0 \tag{4.20}
\end{equation*}
\]

In the constraint formulation of Eqs. 4.19 and 4.20, it is assumed that bodies \(i\) and \(j\) remain in contact. This condition is provided by the revolute and translational joints


Figure 4.14 Rack-and-pinion mechanism in two different configurations.
connecting bodies \(i\) and \(j\) to a third body. Therefore, there is only one relative degree of freedom between bodies \(i\) and \(j\).

In the configurations shown in Fig. 4.14, it is assumed that the line of translation remains parallel to the \(x\) axis. Constraint equations for the general case can also be derived where the third body can be in motion with respect to the \(x y\) coordinate system. Derivation of this constraint equation is left as an exercise for the reader.

\subsection*{4.2.4 Curve Representation}

In some of the higher-pair joints, the shape of the outline of contacting bodies must be described analytically or numerically for kinematic analysis. If, for example, the outline of a cam is circular or elliptical, then it can be described analytically. However, in general, the outline must be described numerically.

Consider, as an example, the cam shown in Fig. 4.15(a). The outline of this cam can be discretized and described in polar coordinates at the points shown in Fig. 4.15(b). Angle \(\theta\) is incremented counterclockwise from zero to \(2 \pi\) in equal or variable increments, and the corresponding values of \(s\) are recorded, as shown in Table 4.1. In order to describe this outline in a closed-form expression, a cubic interpolating spline function can be used.

A spline function is a function consisting of polynomial pieces on subintervals, joined together according to certain smoothness conditions. The choice of degree for the polynomial pieces most often made is 3 . In this case, the resulting splines are termed cubic splines. The cubic polynomials are joined together in such a way that the resulting spline function has continuous first and second derivatives everywhere. For the curve segments shown in Fig. 4.16, if two cubic polynomials \(s_{1}\) and \(s_{2}\) are determined, then, the continuity conditions at point \(B\), i.e., those for \(\theta=\theta^{B}\), are
\[
\begin{aligned}
s_{1} & =s_{2} \\
\frac{d s_{1}}{d \theta} & =\frac{d s_{2}}{d \theta} \\
\frac{d^{2} s_{1}}{d \theta^{2}} & =\frac{d^{2} s_{2}}{d \theta^{2}}
\end{aligned}
\]


Figure 4.15 (a) A cam follower, and (b) discretization of the cam outline.

\section*{TABLE 4.1}
\begin{tabular}{crr} 
Number & \(\theta(\mathrm{deg})\) & \(s(\mathrm{~cm})\) \\
\hline 1 & 0 & 3.70 \\
2 & 30 & 4.55 \\
3 & 60 & 6.80 \\
4 & 70 & 8.38 \\
5 & 80 & 10.65 \\
6 & 90 & 12.46 \\
7 & 100 & 11.10 \\
8 & 110 & 8.65 \\
9 & 120 & 7.04 \\
10 & 150 & 4.63 \\
11 & 180 & 3.80 \\
12 & 225 & 3.62 \\
13 & 270 & 3.80 \\
14 & 315 & 3.62 \\
15 & 360 & 3.70 \\
\hline
\end{tabular}


Figure 4.16 Two cubic polynomials describing segments of a curve.

A detailed discussion of cubic splines is outside the scope of this text. In most textbooks covering this subject, well-developed algorithms and computer programs are presented that can be applied to a discretized table of data, such as Table 4.1, to determine the corresponding cubic polynomials.

\section*{Example 4.3}

The curve shown in the illustration is discretized and a portion of the recorded data is listed. A cubic spline function algorithm finds three cubic polynomials for
three segments of the curve as follows:
\[
\begin{aligned}
& s_{1}=-6.538 \theta^{3}+3.230 \theta^{2}-2.173 \theta+4.108 \\
& s_{2}=6.538 \theta^{3}-8.538 \theta^{2}+1.358 \theta+3.755 \\
& s_{3}=-9.615 \theta^{3}+10.846 \theta^{2}-6.396 \theta+4.788
\end{aligned}
\]

Show that \(s_{1}\) and \(s_{2}\) and their first and second derivatives are continuous at point \(B\). Also, for a point \(P\) where \(\theta^{P}=0.26 \mathrm{rad}\), find \(s, d s / d \theta\), and \(d^{2} s / d \theta^{2}\).


Solution At \(B, \theta^{B}=0.3\); therefore \(s_{1}=s_{2}=3.57\). The first derivatives of \(s_{1}\) and \(s_{2}\) are
\[
\begin{aligned}
& \frac{d s_{1}}{d \theta}=-19.614 \theta^{2}+6.460 \theta-2.173 \\
& \frac{d s_{2}}{d \theta}=19.614 \theta^{2}-17.076 \theta+1.358
\end{aligned}
\]

Evaluating the first derivatives for \(\theta=0.3\) yields \(d s_{1} / d \theta=d s_{2} / d \theta=-2.000\). The second derivatives of \(s_{1}\) and \(s_{2}\) are
\[
\begin{aligned}
& \frac{d^{2} s_{1}}{d \theta^{2}}=-39.228 \theta+6.460 \\
& \frac{d^{2} s_{2}}{d \theta^{2}}=39.228 \theta-17.076
\end{aligned}
\]

Evaluating the second derivatives for \(\theta=0.3\) yields \(d^{2} s_{1} / d \theta^{2}=d^{2} s_{2} / d \theta^{2}=\) -5.308.

For point \(P\), since \(\theta^{B} \leqslant \theta^{P} \leqslant \theta^{A}\), function \(s_{1}\) yields \(s=3.646, d s / d \theta=\) -1.819 , and \(d^{2} s / d \theta^{2}=-3.739\).

For any point on a curve, the \(\xi \eta\) coordinates are found from the relations
\[
\begin{align*}
\xi & =s \cos \theta  \tag{4.21}\\
\eta & =s \sin \theta
\end{align*}
\]

Then, the first and second derivatives of \(\xi\) and \(\eta\) with respect to \(\theta\) are
\[
\begin{align*}
& \frac{d \xi}{d \theta}=-s \sin \theta+\frac{d s}{d \theta} \cos \theta \\
& \frac{d \eta}{d \theta}=s \cos \theta+\frac{d s}{d \theta} \sin \theta \tag{4.22}
\end{align*}
\]
and
\[
\begin{align*}
& \frac{d^{2} \xi}{d \theta^{2}}=-s \cos \theta-2 \frac{d s}{d \theta} \sin \theta+\frac{d^{2} s}{d \theta^{2}} \cos \theta \\
& \frac{d^{2} \eta}{d \theta^{2}}=-s \sin \theta+2 \frac{d s}{d \theta} \cos \theta+\frac{d^{2} s}{d \theta^{2}} \sin \theta \tag{4.23}
\end{align*}
\]

For a point \(P\), Eq. 4.22 represents the components of a vector tangent to the curve at \(P\), as shown in Fig. 4.17, such as
\[
\mathbf{g}^{\prime P}=\left[\begin{array}{l}
\mu  \tag{4.24}\\
\nu
\end{array}\right]
\]
where \(\mu=d \xi / d \theta\) and \(\nu=d \eta / d \theta\).


Figure 4.17 Vector tangent to a curve at point \(P\).

\section*{Example 4.3a}

Determine the \(\xi \eta\) components of \(P\), as defined at the beginning of Example 4.3, and then find the tangent vector \(\vec{g}\) at \(P\).

Solution Taking \(\theta=0.26\) and \(s=3.646\) and substituting into Eq, 4.21 yields \(\xi^{P}=3.523\) and \(\eta^{P}=0.937\). Since \(d s / d \theta=-1.819\), then Eq. 4.22 yields \(d \xi / d \theta=-2.695\) and \(d \eta / d \theta=3.056 ;\) i.e., \(\mathbf{g}^{\prime P}=[-2.695,3.056]^{T}\).

The outline (or a portion of the outline) of a body can be expressed in polar coordinates with respect to a point different from the origin of the \(\xi \eta\) axes. As shown in Fig. 4.18, the discretized portion of the outline is defined in terms of \(s\) and \(\theta\) with respect to a point \(Q\)


Figure 4.18 Polar coordinates of a point \(P\) with respect to point \(Q\) and the \(\xi\) axis.
and an axis parallel to the \(\xi\) axis. In this case, the \(\xi\) and \(\eta\) coordinates of any point on that portion of the outline are
\[
\begin{align*}
\xi & =\xi^{Q}+s \cos \theta  \tag{4.25}\\
\eta & =\eta^{Q}+s \sin \theta
\end{align*}
\]

\subsection*{4.2.5 Cam-Followers (HP)}

Cam-followers are used in a variety of designs; Fig. 4.19 shows a few of the many possible design variations. The contacting surfaces of a cam or a follower, in general, are


Figure 4.19 Several commonly used cam followers: (a) disk cam with offset point follower, (b) disk cam with radial roller follower, (c) disk cam with offset flat-faced follower, (d) translating cam with roller follower, (e) disk cam with oscillating roller follower, and (f) disk cam with oscillating flat-faced follower.
not expressible in closed-form functions. In this case, the cubic spline function method of Sec. 4.2.4 can be employed. In this section, the kinematic formulations for some of the cam-follower configurations are derived. It is assumed that the cam and the follower always remain in contact; i.e., no chattering is allowed.

The first formulation is derived for a disk cam with an offset point follower, as shown in Fig. 4.20. The cam is connected to a third body (possibly the chassis) with a revolute joint, and the follower is connected to this body with a translational joint. The third body may or may not move with respect to the global \(x y\) coordinate system. The coordinates of the contacting point \(P\) on body \(j\) (the follower) are known constant quantities \(\mathbf{s}_{j}^{\prime P}=\left[\xi^{P}, \eta^{P}\right]_{j}^{T}\). The global coordinates of \(P_{j}\) are
\[
\mathbf{r}_{j}^{P}=\mathbf{r}_{j}+\mathbf{A}_{j} \mathbf{s}_{j}^{\prime P}
\]

The contacting point on body \(i\) (the cam) does not have constant coordinates; i.e., the elements of vector \(\mathbf{s}_{i}^{\prime P}=\left[\xi^{P}, \eta^{P}\right]_{i}^{T}\) are functions of \(\theta\). The global coordinates of \(P_{i}\) are
\[
\mathbf{r}_{i}^{P}=\mathbf{r}_{i}+\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P}
\]

Then, the constraint equations for this cam-follower pair can be written as
\[
\begin{equation*}
\boldsymbol{\Phi}^{(c-f 1,2)} \equiv \mathbf{r}_{i}^{p}-\mathbf{r}_{j}^{P}=0 \tag{4.26}
\end{equation*}
\]


Figure 4.20 A disk cam and point-follower pair.

In expanded form, Eq. 4.26 is written
\[
\left[\begin{array}{l}
x  \tag{4.27}\\
y
\end{array}\right]_{i}+\left[\begin{array}{rr}
\cos \phi & -\sin \phi \\
\sin \phi & \cos \phi
\end{array}\right]_{i}\left[\begin{array}{l}
s \\
\cos \theta \\
s \sin \theta
\end{array}\right]_{i}-\left[\begin{array}{l}
x \\
y
\end{array}\right]_{j}-\left[\begin{array}{rr}
\cos \phi & -\sin \phi \\
\sin \phi & \cos \phi
\end{array}\right]_{j}\left[\begin{array}{l}
\xi^{p} \\
\eta^{p}
\end{array}\right]_{j}=\left[\begin{array}{l}
0 \\
0
\end{array}\right]^{2}
\]

Another type of cam-follower pair is the disk cam with offset roller follower shown in Fig. 4.21(a). The contact points \(P_{i}\) and \(P_{j}\) do not have known constant coordinates with respect to the bodies they are on. Vector \(\vec{d}\) connecting point \(P_{i}\) to point \(Q_{j}\) is perpendicular to the tangent vector to either of the outlines at the contact point. The components of vector \(\vec{d}\) can be found as follows:
\[
\mathbf{d}=\mathbf{r}_{j}+\mathbf{A}_{j} \mathbf{s}_{j}^{\prime \ell}-\mathbf{r}_{i}-\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P}
\]

Vector \(\vec{d}\) is perpendicular to the tangent vector to the cam outline at \(P_{i}\); therefore \(\mathbf{g}_{\mathbf{d}}^{T} \mathbf{d}=0\). This condition alone does not eliminate the possibilities of penetration and separation shown in Fig. 4.21(b) and (c). If the magnitude of vector \(\vec{d}\) is equal to the radius \(\rho\) of the roller, then there is no penetration or separation. This condition can be written as \(\mathbf{d}^{T} \mathbf{d}-\rho^{2}=0\). Hence, there are two constraint equations for this pair:
\[
\boldsymbol{\Phi}^{(c f-2,2)}=\left[\begin{array}{c}
\mathbf{g}_{l}^{T} \mathbf{d}  \tag{4.28}\\
\mathbf{d}^{T} \mathbf{d}-\rho^{2}
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
\]

This formulation is also valid for the cam-follower of Fig. 4.19(e).


Figure 4.21 (a) Vector loop closure for a disk cam with follower pair, (b) penetration, and (c) separation of the outlines.


Figure 4.22 (a) A disk cam with an offset flat-faced follower, (b) penetration, and (c) separation of the outlines.

A disk cam with an offset flat-faced follower is shown in Fig. 4.22(a). A vector \(\vec{n}_{j}\) is defined perpendicular to the contacting line (or face) of the follower, which must remain perpendicular to the tangent vector \(\vec{g}_{i}\); i.e., \(\mathbf{g}_{i}^{T} \mathbf{n}_{j}\) must be equal to 0 . This condition alone does not eliminate the possibility of penetration or separation shown in Fig. 4.22(b) and (c). A vector connecting points \(P_{i}\) and \(Q_{j}\) is defined as \(\mathbf{d}=\mathbf{r}_{i}^{P}-\mathbf{r}_{j}^{Q}\). This vector must also remain perpendicular to \(\vec{n}_{j}\); i.e., \(\mathbf{d}^{T} \mathbf{n}_{j}\) must be equal to 0 . Therefore, the two constraint equations for this pair are
\[
\boldsymbol{\Phi}^{(c-f, 2)}=\left[\begin{array}{l}
\mathbf{g}^{T} \mathbf{n}_{j}  \tag{4.29}\\
\mathbf{d}^{T} \mathbf{n}_{j}
\end{array}\right]=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
\]

This formulation is also valid for the cam-follower of Fig. 4.19(f).
In the cam-follower formulation, \(\theta\) is treated as an artificial coordinate. Hence, the vectors of coordinates for bodies \(i\) and \(j\) of a cam-follower pair are \(\mathbf{q}_{i}=[x, y, \phi, \theta]_{i}^{T}\) and \(\mathbf{q}_{j}=[x, y, \phi]_{j}^{T}\). It can be observed that for any of the cam-follower pairs shown in Fig. 4.19, there is one relative degree of freedom between the cam and its corresponding follower.

The constraint formulation has been derived in Eqs. 4.26, 4.28, and 4.29 for a general configuration applicable to various cam-follower mechanisms. For specific cases, the constraint formulation can be obtained in a much simpler form. The following example illustrates this point.

\section*{Example 4.4}

Derive the constraint equation for the disk cam-follower shown in the illustration. The cam and the follower are attached to a nonmoving body. The motion of the follower is in the \(y\) direction, and the outline of the cam is circular.


Solution If the radii of the cam and the roller are \(\rho_{i}\) and \(\rho_{j}\), then a constraint equation can be written as
\[
\begin{equation*}
\Phi \equiv y_{j}-y_{i}+\eta_{j}^{Q}-d=0 \tag{a}
\end{equation*}
\]
where \(d\) is expressed as a function of \(\phi_{i}\) :
\[
d \equiv\left(\rho_{i}+\rho_{j}\right) \cos \gamma+\xi_{i}^{Q} \sin \phi_{i}
\]

The angle \(\gamma\) is found from the equation
\[
\sin \gamma \equiv \frac{\xi_{i}^{Q} \cos \phi_{i}}{\rho_{i}+\rho_{j}}
\]

It can be observed that for this special case, only one constraint equation is needed, and there is no need to introduce an artificial coordinate.

\subsection*{4.2.6 Point-Follower (HP)}

Figure 4.23 shows part of a mechanism in which a slot is cut into one of the bodies. Pin \(A\), which is rigidly attached to the other body, can slide in the slot. This type of kinematic higher pair is called a point-follower joint.

A point-follower joint between bodies \(i\) and \(j\) is shown schematically in Fig. 4.24. Pin \(P\), which is attached to body \(i\), can slide and rotate in a slot on body \(j\). The coordinates of any point on the slot, relative to the body-fixed coordinate system on body \(j\), can be described by Eq. 4.21 ; i.e., \(\xi_{j}^{P}=s \cos \theta\) and \(\eta_{j}^{P}=s \sin \theta\). The constraint equations for this joint are similar to the constraint equations of a revolute joint, with the exception that the position of point \(P\) on body \(j\) is not constant but varies as a function


Figure 4.23 The motion relative to body 1 of point \(A\) attached to body 2 can be modeled by a point-follower joint.


Figure 4.24 Point-follower joint: Pin \(P\) is attached rigidly to body \(i\) and can move in the slot embedded in body \(j\).
of \(\theta\). Hence, the constraint equations are
\[
\mathbf{\Phi}^{(p-f, 2)} \equiv\left[\begin{array}{l}
x_{i}^{P}-\left(x_{j}+\xi_{j}^{P} \cos \phi_{j}-\eta_{j}^{P} \sin \phi_{j}\right)  \tag{4.30}\\
y_{i}^{P}-\left(y_{j}+\xi_{j}^{P} \sin \phi_{j}+\eta_{j}^{P} \cos \phi_{j}\right)
\end{array}\right]=0
\]

In this formulation \(\theta\) is treated as an artificial coordinate and is added to the vector of coordinates, and both of the constraint equations of Eq. 4.30 are employed to describe the point-follower joint. This formulation does not provide any stopping condition when the pin reaches either end of the slot.

\subsection*{4.2.7 Simplified Constraints}

The constraint equations describing certain kinematic conditions between two bodies can be simplified, in general, or replaced by other simple equations if one of the bodies is a nonmoving body; e.g., if one body is the chassis or ground. As an example, consider the quick-return mechanism shown in Fig. 4.25. Slider 3 is constrained to the ground with a translational joint and can move parallel to the \(y\) axis. Similarly, slider 4 is constrained to the ground with a translational joint and can move parallel to the \(x\) axis. The constraint equations of Eq. 4.12 can be used to formulate the kinematics of these two sliders. However, other equations can be used instead of Eq. 4.12. For example, slider 3 cannot move in the \(x\) direction or rotate; therefore, two equations of the form \(x_{3}=\) constant and \(\phi_{3}=\) constant can be used. Similarly, since slider 4 cannot move in the \(y\) direction or rotate, \(y_{4}=\) constant and \(\phi_{4}=\) constant can describe the translational con-


Figure 4.25 A quick-return mechanism.
straints for the slider. Therefore, in order to constrain translation of the origin or angular motion of a rigid body, one or more of the following equations may be used:
\[
\begin{align*}
& \Phi \equiv x_{i}-c_{1}=0  \tag{4.31}\\
& \Phi \equiv y_{i}-c_{2}=0  \tag{4.32}\\
& \Phi \equiv \phi_{i}-c_{3}=0 \tag{4.33}
\end{align*}
\]
where \(c_{1}, c_{2}\), and \(c_{3}\) are constant quantities. Figure 4.26 illustrates graphically the three constraints of Eqs. 4.31 through 4.33.


Figure 4.26 The body can move with (a) constant \(x_{i}\), (b) constant \(y_{i}\), and (c) constant \(\phi_{i}\).

To constrain the motion of point \(P_{i}\) on body \(i\) in the \(x\) or \(y\) direction, the following equations, which incorporate the results of Eq. 4.3, can be employed:
\[
\begin{align*}
\Phi & \equiv x_{i}^{P}-c_{4} \\
& =x_{i}+\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{P} \sin \phi_{i}-c_{4}=0  \tag{4.34}\\
\Phi & \equiv y_{i}^{P}-c_{5} \\
& =y_{i}+\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}-c_{5}=0 \tag{4.35}
\end{align*}
\]
where \(c_{4}\) and \(c_{5}\) are constants.

\subsection*{4.2.8 Driving Links}

In kinematically driven systems, the motion of one or more links (bodies) is usually defined. For example, in the quick-return mechanism of Fig. 4.27(a) or the slider-crank mechanism of Fig. 4.27(b), the driving link \(i\) rotates with known constant angular velocity \(\omega\). If kinematic analysis is to be performed using the appended driving constraints method of Sec. 3.2.2, then the motion of the driving link must be specified in the form of a driving constraint equation. For either of the mechanisms of Fig. 4.27, one moving constraint of the form
\[
\begin{equation*}
\Phi^{(d-1,1)} \equiv \phi_{i}-d_{1}(t)=0 \tag{4.36}
\end{equation*}
\]
can be employed, where \(d_{1}(t)=\phi_{i}^{0}+\omega t\) and \(\phi_{i}^{0}\) is the angle \(\phi_{i}\) at \(t=0\). If the driving link rotates with a constant angular acceleration \(\alpha\), then Eq. 4.36 can be used with \(d_{l}(t)=\frac{1}{2} \alpha t^{2}+\dot{\phi}^{0} t+\phi^{0}\), where \(\dot{\phi}^{0}\) is the angular velocity at \(t=0\).

Equation 4.36 is not the only type of driving equation. Depending on the mechanism and the motion of its driving link(s), other forms of driving equation may be required. Some simple driving constraints are:
\[
\begin{align*}
& \Phi^{(d-2,1)} \equiv x_{i}-d_{2}(t)=0  \tag{4.37}\\
& \Phi^{(d-3,1)} \equiv y_{i}-d_{3}(t)=0  \tag{4.38}\\
& \Phi^{(d-4,1)} \equiv x_{i}^{P}-d_{4}(t)=0  \tag{4.39}\\
& \Phi^{(d-5,1)} \equiv y_{i}^{P}-d_{5}(t)=0 \tag{4.40}
\end{align*}
\]


Figure 4.27 Driving link of (a) a quick-return mechanism and (b) a slider-crank mechanism.


Figure 4.28 (a) The motion of the slider is controlled in the \(x\) direction, and (b) the motion of point \(P\) is controlled in the \(y\) direction.

For example, if the motion of the slider shown in Fig. 4.28(a) is controlled as a function of time, then Eq. 4.37 can be used as the driving constraint. For the mechanism of Fig. 4.28(b), the motion of point \(P\) in the \(y\) direction is controlled as a function of time; therefore, Eq. 4.40 can be employed as the driving constraint.

\subsection*{4.3 POSITION, VELOCITY, AND ACCELERATIOIN ANALYSIS}

The kinematic constraint equations \(\boldsymbol{\Phi}\) derived in the preceding sections for planar kinematic pairs are, in general, nonlinear in terms of the coordinates \(\mathbf{q}\). If the number of coordinates describing the configuration of a mechanical system is \(n\) and the number of degrees of freedom of the system is \(k\), then \(m=n-k\) kinematic constraints can be defined as
\[
\begin{equation*}
\Phi \equiv \boldsymbol{\Phi}(\mathbf{q})=0 \tag{4.41}
\end{equation*}
\]

In addition, \(k\) driving constraint equations must be defined as
\[
\begin{equation*}
\boldsymbol{\Phi}^{(d)} \equiv \boldsymbol{\Phi}(\mathbf{q}, t)=\mathbf{0} \tag{4.42}
\end{equation*}
\]

Equations 4.41 and 4.42 represent a set of \(n\) nonlinear algebraic equations which can be solved for \(n\) unknowns \(\mathbf{q}\) at any given time \(t=t^{i}\).

The procedure suggested here for solving Eqs. 4.41 and 4.42 is the appended driving constraint method of Sec. 3.2.2. These equations are identical in form to Eq. 3.12. The main difference between the formulation in this chapter and that in Chap. 3 is the dimension, i.e., the number of coordinates and, therefore, the number of constraint equations.

\section*{Example 4.5}

A four-bar linkage was formulated for kinematic analysis in Example 3.2, using three coordinates, two kinematic constraint equations, and one driving constraint equation. Determine the number of coordinates and constraints if this mechanism is being analyzed using Cartesian coordinates.


Solution Since there are four bodies in the system - three moving links and one stationary link - the dimension of \(\mathbf{q}\) is \(4 \times 3=12\). In the accompanying illustration, body-fixed coordinates are attached to each link. Body 1 is assumed to be the frame, and bodies 2, 3, and 4 are the crank, the coupler, and the follower, respectively. According to Eq. 4.7, each revolute joint is represented by two algebraic equations, and therefore there are \(4 \times 2=8\) algebraic equations representing all four revolute joints. In addition, since body 1 does not move with respect to the \(x y\) coordinate system, three simple constraints in the form of Eqs. 4.31 through 4.33 are needed. Therefore, the total number of kinematic constraints is \(m=8+\) \(3=11\). For the rotation of the crank, one driving constraint makes the total number of equations equal to 12 .

The first and second time derivatives of Eqs. 4.41 and 4.42 yield velocity and acceleration equations that are identical in form to Eqs. 3.13 and 3.15. For position analysis using the Newton-Raphson algorithm, velocity analysis, and acceleration analysis, the Jacobian matrix of the kinematic constraints, \(\mathbf{\Phi}_{\mathbf{q}}\), is needed. Also, for acceleration analysis, the right side of the kinematic acceleration equations, vector \(\gamma\) as given in Eq. 3.17, is needed. For each of the kinematic pairs discussed in the preceding sections closed-form expressions can be derived for the entries of the Jacobian matrix and the right-side vector of the acceleration equations. From these expressions plus the constraint equations, all of the necessary terms for kinematic analysis can be assembled systematically. In Sec. 4.3.1, these expressions are derived in detail for some of the kinematic pairs. The method can be applied similarly to other constraint equations.

Derivation of the velocity and acceleration equations requires the time derivatives of the constraint equations. In turn, these require the derivatives of the coordinates of the points describing the kinematic joint. The \(x y\) coordinates of a point \(P\) attached to body \(i\) can be found from Eq. 4.1, where it is assumed that the position of the point on the body and the coordinates of the body are known. The velocity of the point in the \(x y\) coordinate system can be found by taking the time derivative of Eq. 4.1,
\[
\begin{align*}
\dot{\mathbf{r}}_{i}^{P} & =\dot{\mathbf{r}}_{i}+\dot{\mathbf{A}}_{i} \mathbf{s}_{i}^{P} \\
& =\dot{\mathbf{r}}_{i}+\mathbf{B}_{i} \mathbf{s}_{i}^{\prime P} \dot{\phi}_{i} \tag{4.43}
\end{align*}
\]
where
\[
\mathbf{B}_{i}=\left[\begin{array}{rr}
-\sin \phi & -\cos \phi  \tag{4.44}\\
\cos \phi & -\sin \phi
\end{array}\right]_{i}
\]

Similarly, the time derivative of Eq. 4.43 yields the acceleration of point \(P\) :
\[
\begin{align*}
\ddot{\mathbf{r}}_{i}^{P} & =\mathbf{r}_{i}^{\prime}+\ddot{\mathbf{A}}_{i} \mathbf{s}_{i}^{\prime P} \\
& =\mathbf{r}_{i}{ }_{i}^{\prime}+\mathbf{B}_{i} \mathbf{s}_{i}^{\prime P} \ddot{\phi}_{i}-\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P} \dot{\phi}_{i}^{2} \tag{4.45}
\end{align*}
\]

Knowing the velocity of body \(i\), i.e., \(\dot{\mathbf{q}}_{i}=\left[\dot{\mathbf{r}}^{T}, \dot{\phi}\right]_{i}^{T}\), we can use Eq. 4.43 to find the velocity of \(P_{i}\); and similarly, knowing the velocity and the acceleration of body \(i\); i.e., \(\dot{\mathbf{q}}_{i}\) and \(\ddot{\mathbf{q}}_{i}\), we can find the acceleration of \(P_{i}\) by using Eq. 4.45.

\subsection*{4.3.1 Systematic Generation of Some Basic Elements}

Systematic generation of the Jacobian matrix and the right side of the acceleration equations for some of the standard kinematic joints can best be illustrated by deriving these elements for a revolute joint.

Consider the revolute-joint constraint equations of Sec. 4.2.1. These equations in compact form are written as
\[
\boldsymbol{\Phi}^{(r, 2)} \equiv \mathbf{r}_{i}+\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P}-\mathbf{r}_{j}-\mathbf{A}_{j} \mathbf{s}_{j}^{\prime p}=\mathbf{0}
\]
and in expanded form they are
\[
\begin{aligned}
\Phi^{(r, 1 s t)} \equiv x_{i}+\xi_{i}^{p} \cos \phi_{i}-\eta_{i}^{p} \sin \phi_{i}-x_{j}-\xi_{j}^{P} \cos \phi_{j}+\eta_{j}^{P} \sin \phi_{j} & =0 \\
\Phi^{(r, 2 n d)} \equiv y_{i}+\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}-y_{j}-\xi_{j}^{p} \sin \phi_{j}-\eta_{j}^{p} \cos \phi_{j} & =0
\end{aligned}
\]

The partial derivative of these equations with respect to \(\mathbf{q}\), i.e., \(\boldsymbol{\Phi}_{\mathbf{q}}^{(r, 2)}\), provides two rows to the overall system Jacobian. Since \(\boldsymbol{\Phi}^{(r, 2)}\) is a function of only \(\mathbf{q}_{i}\) and \(\mathbf{q}_{j}, \boldsymbol{\Phi}_{q}^{(r, 2)}\) may have nonzero elements only in the columns associated with \(\mathbf{q}_{i}\) and \(\mathbf{q}_{j}\). These possible nonzero entries are
\[
\begin{align*}
& \text { (1) } \equiv \frac{\partial \Phi^{(r, \text {,st })}}{\partial x_{i}}=1.0 \\
& \frac{\partial \Phi^{(r, s t)}}{\partial y_{i}}=0.0 \\
& \text { (2) } \equiv \frac{\partial \Phi^{(r, \text { lst })}}{\partial \phi_{i}}=-\xi_{i}^{P} \sin \phi_{i}\left(\overline{(+)} \eta_{i}^{p} \cos \phi_{i}\right. \\
& =-y_{i}^{P}+y_{i} \\
& \text { (3) } \equiv \frac{\partial \Phi^{(r, \text { sst })}}{\partial x_{j}}=-1.0 \\
& \frac{\partial \Phi^{(r, 1 \text { st })}}{\partial y_{j}}=0.0 \\
& \text { (4) } \equiv \frac{\partial \Phi^{(r, ~ s t s t)}}{\partial \phi_{j}}=\xi_{j}^{p} \sin \phi_{j}+\eta_{j}^{p} \cos \phi_{j} \\
& =y_{j}^{P}-y_{j}  \tag{4.46}\\
& \frac{\partial \Phi^{(r, 2 \mathrm{nd})}}{\partial x_{i}}=0.0
\end{align*}
\]
\[
\begin{aligned}
& \text { (5) } \equiv \begin{aligned}
\equiv \frac{\partial \Phi^{(r, 2 \mathrm{nd})}}{\partial y_{i}} & =1.0 \\
(6) \equiv \frac{\partial \Phi^{(r, 2 \mathrm{nd})}}{\partial \phi_{i}} & =\xi_{i}^{p} \cos \phi_{i}-\eta_{i}^{p} \sin \phi_{i} \\
& =x_{i}^{p}-x_{i} \\
\frac{\partial \Phi^{(r, 2 \mathrm{~d})}}{\partial x_{j}} & =0.0 \\
(7) \equiv \frac{\partial \Phi^{(r, 2 \mathrm{nd})}}{\partial y_{j}} & =-1.0 \\
\text { (8) } \equiv \frac{\partial \Phi^{(r, 2 \mathrm{nd})}}{\partial \phi_{j}} & =-\xi_{j}^{p} \cos \phi_{j}+\eta_{j}^{p} \sin \phi_{j} \\
& =-x_{j}^{p}+x_{j}
\end{aligned}
\end{aligned}
\]
where four of the entries are identically zero. The position of these nonzero entries \({ }^{\dagger}\) in the rows of the Jacobian matrix corresponding to \(\boldsymbol{\Phi}^{(r, 2)}\) and columns corresponding to \(\mathbf{q}_{i}\) and \(\mathbf{q}_{j}\) is shown in Fig. 4.29.


Figure 4.29 Nonzero entities of the Jacobian matrix for a revolute joint between bodies \(i\) and \(j\).

The entries of the Jacobian matrix can also be found by taking the time derivative of the constraint equations. The time derivative of the constraint equations for a revolute joint is
\[
\begin{aligned}
& \dot{x}_{i}-\left(\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}\right) \dot{\phi}_{i}-\dot{x}_{j}+\left(\xi_{j}^{p} \sin \phi_{j}+\eta_{j}^{p} \cos \phi_{j}\right) \dot{\phi}_{j}=0 \\
& \dot{y}_{i}+\left(\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{p} \sin \phi_{i}\right) \dot{\phi}_{i}-\dot{y}_{j}-\left(\xi_{j}^{p} \cos \phi_{j}-\eta_{j}^{p} \sin \phi_{j}\right) \dot{\phi}_{j}=0
\end{aligned}
\]
or


This velocity equation is in the same form shown by the first equation of Eq. 3.13.
To obtain the right side of the acceleration equation for a revolute joint, either
Eq. 3.17 can be used with the Jacobian matrix of Eq. 4.46, or the velocity equations can

\footnotetext{
\({ }^{\dagger}\) In this text, the term nonzero entry must be interpreted as an entry with a possible nonzero value.
}
be differentiated with respect to time to obtain the acceleration equations:
\[
\begin{aligned}
& \ddot{x}_{i}-\left(\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}\right) \ddot{\phi}_{i}-\left(\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{P} \sin \phi_{i}\right) \dot{\phi}_{i}^{2}-\ddot{x}_{j} \\
& \quad+\left(\xi_{j}^{P} \sin \phi_{j}+\eta_{j}^{P} \cos \phi_{j}\right) \ddot{\phi}_{j}+\left(\xi_{j}^{P} \cos \phi_{j}-\eta_{j}^{P} \sin \phi_{j}\right) \dot{\phi}_{j}^{2}=0 \\
& \ddot{y}_{i}+\left(\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{P} \sin \phi_{i}\right) \ddot{\phi}_{i}-\left(\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}\right) \dot{\phi}_{i}^{2}-\ddot{y}_{j} \\
& \quad-\left(\xi_{j}^{P} \cos \phi_{j}-\eta_{j}^{P} \sin \phi_{j}\right) \ddot{\phi}_{j}+\left(\xi_{j}^{P} \sin \phi_{j}+\eta_{j}^{P} \cos \phi_{j}\right) \dot{\phi}_{j}^{2}=0
\end{aligned}
\]
or

where
\[
\begin{align*}
\gamma^{(r, 2)} & =\left[\begin{array}{l}
\left(\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{P} \sin \phi_{i}\right) \dot{\phi}_{i}^{2}-\left(\xi_{j}^{P} \cos \phi_{j}-\eta_{j}^{P} \sin \phi_{j}\right) \dot{\phi}_{j}^{2} \\
\left(\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}\right) \dot{\phi}_{i}^{2}-\left(\xi_{j}^{P} \sin \phi_{j}+\eta_{j}^{P} \cos \phi_{j}\right) \dot{\phi}_{i}^{2}
\end{array}\right] \\
& =\left[\begin{array}{l}
\left(x_{i}^{P}-x_{i}\right) \dot{\phi}_{i}^{2}-\left(x_{j}^{P}-x_{j}\right) \dot{\phi}_{j}^{2} \\
\left(y_{i}^{P}-y_{i}\right) \dot{\phi}_{i}^{2}-\left(y_{j}^{P}-y_{j}\right) \dot{\phi}_{j}^{2}
\end{array}\right] \\
& =\mathbf{s}_{i}^{P} \dot{\phi}_{i}^{2}-\mathbf{s}_{j}^{P} \dot{\phi}_{j}^{2} \tag{4.47}
\end{align*}
\]

From Eqs. 4.46 and 4.47 , it can be observed that the entries of the Jacobian matrix and the vector of the right side of the acceleration equations for a revolute joint can be generated systematically, in a computer program.

Tables 4.2 and 4.3 summarize the elements of the Jacobian matrix and the vector \(\gamma\), respectively, for some of the constraint equations of the basic joints. Similar elements can be derived for other kinematic pairs. Verification of these elements is left as an exercise for the reader.

\subsection*{4.4 KINEMATIC MODELING}

In general, there are many ways to kinematically model a particular mechanism, any one of which may be more advantageous in some circumstances. In kinematic analysis, the problem size, i.e., the number of coordinates and number of constraint equations, is an important factor to consider. A small number of coordinates yields a smaller problem to solve. In this section, several simple examples are considered to illustrate methods of kinematic modeling. The same techniques may be used to analyze larger problems.

\subsection*{4.4.1 Slider-Crank Mechanism}

The slider-crank is one of the most widely used mechanisms in practice. This mechanism finds its greatest application in the internal-combustion engine. The mechanism is

TABLE 4.2 Elements of the Jacobian Matrix for Some of the Basic Constraint Equations
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline & \(\partial \Phi / \partial x_{i}\) & \(\partial \Phi / \partial y_{i}\) & \(\partial \Phi / \partial \phi_{i}\) & \(\partial \Phi / \partial x_{j}\) & \(\partial \Phi / \partial y_{j}\) & \(\partial \Phi / \partial \phi_{j}\) \\
\hline \multirow[t]{2}{*}{\(\boldsymbol{\Phi}^{(r, 2)}\)} & 1 & 0 & \(-\left(y_{i}^{P}-y_{i}\right)\) & -1 & 0 & \(\left(y_{j}^{P}-y_{j}\right)\) \\
\hline & 0 & 1 & \(\left(x_{i}^{p}-x_{i}\right)\) & 0 & -1 & \(-\left(x_{j}^{P}-x_{j}\right)\) \\
\hline \multirow[t]{2}{*}{\(\Phi^{(t, 2)}\)} & & \(-\left(x_{i}^{P}-x_{i}^{\text {Q }}\right.\) ) & \[
\begin{aligned}
& -\left(x_{j}^{P}-x_{i}\right)\left(x_{i}^{P}-x_{i}^{Q}\right) \\
& -\left(y_{j}^{P}-y_{i}\right)\left(y_{i}^{P}-y_{i}^{Q}\right)
\end{aligned}
\] & \(-\left(y_{i}^{P}-y_{i}^{\text {P }}\right.\) ) & \(\left(x_{i}^{P}-x_{i}^{Q}\right)\) & \[
\begin{aligned}
& \left(x_{j}^{P}-x_{j}\right)\left(x_{i}^{P}-x_{i}^{Q}\right) \\
& +\left(y_{j}^{P}-y_{j}\right)\left(y_{i}^{P}-y_{i}^{Q}\right)
\end{aligned}
\] \\
\hline & 0 & 0 & 1 & 0 & 0 & -1 \\
\hline \(\Phi^{\left(r-r_{1}, 1\right)}\) & \(2\left(x_{i}^{P}-x_{j}^{P}\right)\) & \(2\left(y_{i}^{P}-y_{j}^{P}\right)\) & \[
\begin{aligned}
& -2\left(x_{i}^{P}-x_{j}^{p}\right)\left(y_{i}^{p}-y_{i}\right) \\
& +2\left(y_{i}^{P}-y_{j}^{P}\right)\left(x_{i}^{p}-x_{i}\right)
\end{aligned}
\] & \(-2\left(x_{i}^{P}-x_{j}^{P}\right)\) & \(-2\left(y_{i}^{P}-y_{j}^{P}\right)\) & \[
\begin{aligned}
& 2\left(x_{i}^{P}-x_{j}^{P}\right)\left(y_{j}^{P}-y_{j}\right) \\
& -2\left(y_{i}^{P}-y_{j}^{P}\right)\left(x_{j}^{P}-x_{j}\right)
\end{aligned}
\] \\
\hline \(\Phi^{(r-4,1)}\) & \(\left(y_{j}^{P}-y_{j}^{\ell}\right)\) & \(-\left(x_{j}^{P}-x_{j}^{Q}\right)\) & \[
\begin{aligned}
& -\left(y_{i}^{P}-y_{i}\right)\left(y_{j}^{P}-y_{j}^{Q}\right) \\
& -\left(x_{i}^{P}-x_{i}\right)\left(x_{j}^{P}-x_{j}^{Q}\right)
\end{aligned}
\] & \(-\left(y_{j}^{P}-y_{j}^{\mathcal{Q}}\right)\) & \(\left(x_{j}^{p}-x_{j}^{Q}\right)\) & \[
\begin{aligned}
& \left(y_{i}^{P}-y_{j}\right)\left(y_{j}^{P}-y_{j}^{Q}\right) \\
+ & \left(x_{i}^{P}-x_{j}\right)\left(x_{j}^{P}-x_{j}^{Q}\right)
\end{aligned}
\] \\
\hline
\end{tabular}

TABLE 4.3 Vector \(\gamma\) for Some of the Basic Constraint Equations
\begin{tabular}{|c|c|}
\hline \multicolumn{2}{|r|}{\(\gamma\)} \\
\hline \multirow{2}{*}{\(\Phi^{(r, 2)}\)} & \(\left(x_{i}^{P}-x_{i}\right) \dot{\phi}_{i}^{2}-\left(x_{j}^{P}-x_{j}\right) \dot{\phi}_{j}^{2}\) \\
\hline & \(\left(y_{i}^{P}-y_{i}\right) \dot{\phi}_{i}^{2}-\left(y_{j}^{P}-y_{j}\right) \dot{\phi}_{j}^{2}\) \\
\hline \multirow{2}{*}{\(\Phi^{(4,2)}\)} & \[
\begin{aligned}
& -2\left[\left(x_{i}^{P}-x_{i}^{Q}\right)\left(\dot{x}_{i}-\dot{x}_{j}\right)+\left(y_{i}^{P}-y_{i}^{Q}\right)\left(\dot{y}_{i}-\dot{y}_{j}\right)\right] \dot{\phi}_{i}^{\dagger} \\
& -\left[\left(x_{i}^{P}-x_{i}^{Q}\right)\left(y_{i}-y_{j}\right)-\left(y_{i}^{P}-y_{i}^{Q}\right)\left(x_{i}-x_{j}\right)\right] \dot{\phi}_{i}^{2}
\end{aligned}
\] \\
\hline & 0 \\
\hline \(\Phi^{(r-r, 1)}\) & \[
\begin{aligned}
& \quad-2\left\{\left(\dot{x}_{i}-\dot{x}_{j}\right)^{2}+\left(\dot{y}_{i}-\dot{y}_{j}\right)^{2}\right. \\
& +\left[\left(x_{i}^{p}-x_{i}\right)\left(x_{j}^{P}-x_{i}\right)+\left(y_{i}^{P}-y_{i}\right)\left(y_{j}^{P}-y_{j}\right)\right] \dot{\phi}_{i}^{2} \\
& +\left[\left(x_{j}^{p}-x_{j}\right)\left(x_{i}^{P}-x_{j}\right)+\left(y_{j}^{P}-y_{j}\right)\left(y_{i}^{P}-y_{j}\right)\right] \dot{\phi}_{j}^{2} \\
& -2\left[\left(x_{i}^{P}-x_{i}\right)\left(x_{j}^{P}-x_{j}\right)+\left(y_{i}^{P}-y_{i}\right)\left(y_{j}^{P}-y_{j}\right)\right] \dot{\phi}_{i} \dot{\phi}_{j} \\
& +2\left[\left(x_{i}^{P}-x_{i}\right)\left(\dot{y}_{i}-\dot{y}_{j}\right)-\left(y_{i}^{P}-y_{i}\right)\left(\dot{x}_{i}-{\left.\left.\dot{x} j_{j}\right)\right] \dot{\phi}_{i}}^{\left.-2\left[\left(x_{j}^{P}-x_{j}\right)\left(\dot{y}_{i}-\dot{y}_{j}\right)-\left(y_{j}^{p}-y_{j}\right)\left(\dot{x}_{i}-\dot{x}_{j}\right)\right] \dot{\phi}_{j}\right\}}\right.\right.
\end{aligned}
\] \\
\hline \(\Phi^{(r-r, 1)}\) & \[
\begin{aligned}
& {\left[\left(x_{i}^{P}-x_{i}\right)\left(y_{j}^{P}-y_{j}^{Q}\right)-\left(y_{i}^{p}-y_{i}\right)\left(x_{j}^{p}-x_{j}^{Q}\right)\left(\dot{\phi}_{i}-2 \dot{\phi}_{j}\right) \dot{\phi}_{i}\right.} \\
& \quad+\left[\left(x_{i}^{p}-x_{j}\right)\left(y_{j}^{P}-y_{j}^{Q}\right)-\left(y_{i}^{P}-y_{j}\right)\left(x_{j}^{p}-x_{j}^{Q}\right)\right] \dot{\phi}_{j}^{2} \\
& -2\left[\left(x_{j}^{P}-x_{j}^{Q}\right)\left(\dot{x}_{i}-\dot{x}_{j}\right)+\left(y_{j}^{P}-y_{j}^{Q}\right)\left(\dot{y}_{i}-\dot{y}_{j}\right)\right] \dot{\phi}_{j}
\end{aligned}
\] \\
\hline
\end{tabular}
\({ }^{\dagger}\) Note that \(\dot{\phi}_{i}=\dot{\phi}_{j}\).
made of four links or bodies, as illustrated in Fig. 4.30(a). The bodies are numbered from 1 to 4, as shown in Fig. 4.30(b). Body 1 is the fixed link (ground, chassis, or engine block), body 2 is the crank, body 3 is the connecting rod, and body 4 is the slider. Bodies 1 and 2,2 and 3 , and 3 and 4 are connected by revolute joints \(A, B\), and \(O\), respectively. Bodies 1 and 4 are connected by a translational joint \(T\). The number of degrees of freedom for this mechanism is \(k=4 \times 3-(3 \times 2+1 \times 2+3)=1\), since there are \(4 \times 3=12\) coordinates in the system, 3 revolute joints eliminate 6 DOF , 1 translational joint eliminates 2 DOF , and ground constraints on body 1 eliminate 3 DOF.

Body-fixed coordinates \(\xi \eta\) are attached to each body, including the ground, as shown in Fig. 4.30(c). Positioning of these coordinate systems is quite arbitrary for kinematic analysis. However, it is good practice to locate the origin of the coordinate system at the center of gravity of the body. Furthermore, aligning at least one of the coordinate axes with the link axis or parallel to some line of certain geometric or kinematic importance may simplify the task of collecting data for the kinematic pairs in the system.

For the three revolute joints, the following data are obtained from Fig. 4.30(c):
\[
\begin{array}{llll}
\xi_{4}^{A}=0.0, & \eta_{4}^{A}=0.0, & \xi_{3}^{A}=-200.0, & \eta_{3}^{A}=0.0 \\
\xi_{3}^{B}=300.0, & \eta_{3}^{B}=0.0, & \xi_{2}^{B}=-100.0, & \eta_{2}^{B}=0.0 \\
\xi_{2}^{0}=100.0 & \eta_{2}^{0}=0.0, & \xi_{1}^{0}=0.0, & \eta_{1}^{0}=0.0 \tag{a}
\end{array}
\]

For the translational joint, two points on body 4 and one point on body 1 are chosen. The location of these points is arbitrary, as long as they are on the same line of transla-

(a)

(b)

(c)

(d)

Figure 4.30 Kinematic modeling of a slider-crank mechanism.
tion. These points are \(A, C\), and \(O\). The distance \(A C\) is taken here as 100 mm . The data that define the translational joint constraints are as follows:
\[
\begin{array}{lll}
\xi_{4}^{A}=0.0, & \eta_{4}^{A}=0.0, & \xi_{4}^{C}=100.0, \\
\xi_{1}^{0}=0.0, & \eta_{1}^{0}=0.0 & \tag{b}
\end{array}
\]

Additional constraints are needed to ensure that body 1 is the nonmoving body. A global \(x y\) coordinate system is added to the configuration, as shown in Fig. 4.30(d). For convenience, the \(x y\) coordinate system is positioned to coincide with the \(\xi_{1} \eta_{1}\) coordinates. Therefore, the conditions
\[
\begin{equation*}
x_{1}=0.0, \quad y_{1}=0.0, \quad \phi_{1}=0.0 \tag{c}
\end{equation*}
\]
must be satisfied.
The driving constraint on \(\phi_{2}\) is written, using Eq. 4.36, as
\[
\begin{equation*}
\phi_{2}-5.76+1.2 t=0 \tag{d}
\end{equation*}
\]
where 5.76 rad is equal to \(360^{\circ}-30^{\circ}\). Since the crank rotates clockwise, a negative sign is selected for \(\omega\).

The data defined by Eqs. \(a\) through \(c\) are sufficient to describe the connectivity between different bodies in the system. If a computer program is available to generate all of the kinematic constraint equations, such as the program of Chap. 5, then the constraint equations are set up and analyzed automatically. Such a program requires an initial estimate of the coordinates of the system, which is used to start the Newton-Raphson iteration for position analysis. These estimates are found from Fig. 4.30(d) to be,
\[
\begin{array}{lll}
x_{1}=0.0, & y_{1}=0.0, & \phi_{1}=0.0 \\
x_{2}=-86.6, & y_{2}=50.0, & \phi_{2}=5.76  \tag{e}\\
x_{3}=-380.5, & y_{3}=40.0, & \phi_{3}=0.2 \\
x_{4}=-663.1, & y_{4}=0.0, & \phi_{4}=0.0
\end{array}
\]

These estimates need not be accurate. The Newton-Raphson algorithm starts the iterations using the estimated values and finds the exact values for the coordinates at \(t=0\). The estimates can be made by rough measurements from the actual system, or from an illustration by means of a ruler and protractor.

Since there are 12 coordinates in this problem, the computer program must generate 12 kinematic constraint equations. These equations can be obtained by substituting the data of Eqs. \(a\) to \(d\) in the constraint equations derived in the previous sections. Equation \(c\) yields three constraints:
\[
\begin{align*}
& \Phi_{1} \equiv x_{1}=0.0 \\
& \Phi_{2} \equiv y_{1}=0.0  \tag{f}\\
& \Phi_{3} \equiv \Phi_{1}=0.0
\end{align*}
\]

Substitution of Eq. \(a\) into the revolute-joint constraints of Eq. 4.9 yields six constraint equations for the three revolute joints:
\[
\begin{aligned}
& \Phi_{4} \equiv x_{4}-x_{3}+200 \cos \phi_{3}=0 \\
& \Phi_{5} \equiv y_{4}-y_{3}+200 \sin \phi_{3}=0 \\
& \Phi_{6} \equiv x_{3}+300 \cos \phi_{3}-x_{2}+100 \cos \phi_{2}=0 \\
& \Phi_{7} \equiv y_{3}+300 \sin \phi_{3}-y_{2}+100 \sin \phi_{2}=0
\end{aligned}
\]
\[
\begin{aligned}
& \Phi_{8} \equiv x_{2}+100 \cos \phi_{2}-x_{1}=0 \\
& \Phi_{9} \equiv y_{2}+100 \sin \phi_{2}-y_{1}=0
\end{aligned}
\]

Two constraint equations for the translational joint are obtained by substituting the data of Eq. \(b\) into Eq. 4.12 and using Eq. 4.3:
\[
\begin{aligned}
\Phi_{10} \equiv\left(-100 \cos \phi_{4}\right)\left(y_{1}-100 \sin \phi_{1}\right. & \left.-y_{4}\right) \\
& -\left(x_{1}-100 \cos \phi_{1}-x_{4}\right)\left(-100 \sin \phi_{4}\right)=0
\end{aligned}
\]
\(\Phi_{11} \equiv \phi_{4}-\phi_{1}=0\)
Equation \(d\) is the driving constraint:
\[
\Phi_{12} \equiv \phi_{2}-5.76+1.2 t=0
\]

The nonzero entries of the Jacobian matrix for these 12 constraints are found from Table 4.2. The positions of these entries are shown in Fig. 4.31. There are 36 nonzero entries in the \(12 \times 12\) matrix. The nonzero entries are:
(1), (2), (3), (6), (10), (14), (18), (22), (26), (34), (36) \(=1\)
(4), (8), (12), (24),
(5) \(=-200 \sin \phi_{3}\)
(9) \(=200 \cos \phi_{3}\)
(13), (23) \(=-100 \sin \phi_{2}\)
(13), (23) \(=-100 \sin \phi_{2}\)
(15) \(=-300 \sin \phi_{3}\)
(15) \(=-300 \sin \phi_{3}\)
(17), (27) \(=100 \cos \phi_{2}\)
(17), (27) \(=100 \cos \phi_{2}\)
(19) \(=300 \cos \phi_{3}\)
(19) \(=300 \cos \phi_{3}\)
(28) \(=100 \sin \phi_{4}\)
(28) \(=100 \sin \phi_{4}\)
\(29=-100 \cos \phi_{4}\)
\(29=-100 \cos \phi_{4}\)
\(30)=10,000\left(\cos \phi_{1} \cos \phi_{4}+\sin \phi_{1} \sin \phi_{4}\right)\)
\(30)=10,000\left(\cos \phi_{1} \cos \phi_{4}+\sin \phi_{1} \sin \phi_{4}\right)\)
(31) \(=-\) (28)
(31) \(=-\) (28)
(32) \(=-29\)
(32) \(=-29\)
\(33)=100\left[\sin \phi_{4}\left(y_{1}-100 \sin \phi_{1}-y_{4}\right)+\cos \phi_{4}\left(x_{1}-100 \cos \phi_{1}-x_{4}\right)\right]\)
\(33)=100\left[\sin \phi_{4}\left(y_{1}-100 \sin \phi_{1}-y_{4}\right)+\cos \phi_{4}\left(x_{1}-100 \cos \phi_{1}-x_{4}\right)\right]\)

Note that some of the entries at any given instant of time, depending on the value of the coordinates, may become zero. This example illustrates that about 83 percent of the entries of this Jacobian matrix are exactly zero. Therefore, the matrix is said to be sparse.

\subsection*{4.4.2 Quick-Return Mechanism}

Figure 4.32 (a) shows a commonly used quick-return mechanism that produces a slow cutting stroke of a tool (attached to slider \(D\) ) and a rapid return stroke. The driving crank \(O A\) turns at the constant rate of \(3 \mathrm{rad} / \mathrm{s}\). Several ways to model this mechanism are pre-
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline & \(x_{1}\) & \(y_{1}\) & \(\phi_{1}\) & \(x_{2}\) & \(y_{2}\) & \(\phi_{2}\) & \(x_{3}\) & \(y_{3}\) & \(\phi_{3}\) & \(x_{4}\) & \(y_{4}\) & \(\phi_{4}\) \\
\hline \(\partial \Phi_{1} / \partial \ldots\) & (1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline \(\partial \Phi_{2} / \partial \ldots\) & 0 & (2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline \(\partial \Phi_{3} / \partial \ldots\) & 0 & 0 & (3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline \(\partial \Phi_{4} / \partial \ldots\) & 0 & 0 & 0 & 0 & 0 & 0 & (4) & 0 & (5) & (6) & 0 & (7) \\
\hline \(\partial \Phi_{S} / \partial \ldots\) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (8) & (9) & 0 & (10) & (11) \\
\hline \(\partial \Phi_{6} / \partial \ldots\) & 0 & 0 & 0 & (12) & 0 & (13) & (14) & 0 & (15) & 0 & 0 & 0 \\
\hline \(\partial \Phi_{7} / \partial \ldots\) & 0 & 0 & 0 & 0 & (16) & (17) & 0 & (18) & (19) & 0 & 0 & 0 \\
\hline \(\partial \Phi_{8} / \partial \ldots\) & (20) & 0 & (21) & (22) & 0 & (23) & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline \(\partial \Phi_{9} / \partial \ldots\) & 0 & (24) & (25) & 0 & (26) & (27) & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline \(\partial \Phi_{10} / \partial \ldots\) & (28) & (29) & (30) & 0 & 0 & 0 & 0 & 0 & 0 & (31) & (32) & (33) \\
\hline \({ }_{\partial \Phi_{11} / \partial \ldots}\) & 0 & 0 & (34) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (35) \\
\hline \({ }_{\partial \Phi_{12} / \partial \ldots}\) & & 0 & 0 & 0 & 0 & (36) & 0 & 0 & 0 & 0 & 0 & \\
\hline
\end{tabular}

Figure 4.31 Position of nonzero entries of the Jacobian matrix for the slider-crank mechanism.
\(\underset{\rightharpoonup}{\boldsymbol{\rightharpoonup}}\)


Figure 4.32 Kinematic modeling of a quick-return mechanism.
sented in this section. Each model has some advantage and some disadvantage in relation to the others.

Model 1. A maximum number of six bodies is used in the first model, as shown in Fig. 4.32(b). This model requires five revolute joints and two translational joints. There are 18 coordinates, 10 revolute-joint constraint equations, 4 translationaljoint constraint equations, 3 ground constraints, and 1 driving constraint. Bodies are numbered as shown in Fig. 4.32(b), and the following data are deduced from this figure for the revolute and translational joints:
\[
\begin{aligned}
& \xi_{3}^{A}=50, \quad \eta_{3}^{A}=0, \quad \xi_{4}^{A}=0, \quad \eta_{4}^{A}=0 \\
& \xi_{2}^{B}=250, \quad \eta_{2}^{B}=0, \quad \xi_{5}^{B}=60, \quad \eta_{5}^{B}=0 \\
& \xi_{1}^{C}=0 \\
& \eta_{1}^{c}=0, \quad \xi_{2}^{C}=-250, \\
& \xi_{6}^{D}=0, \\
& \left.\eta_{2}^{c}=0\right\} \begin{array}{c}
\text { revolute } \\
\text { joints }
\end{array} \\
& \xi_{s}^{D}=-60, \\
& \eta_{5}^{D}=0 \text {, } \\
& \xi_{3}^{0}=-50, \\
& \eta_{6}^{D}=0 \\
& \xi_{1}^{0}=0 \text {, } \\
& \eta_{1}^{0}=300 \text {, } \\
& \xi_{4}^{A}=0, \quad \eta_{4}^{A}=0, \quad \xi_{4}^{L}=-10, \quad \eta_{4}^{\prime}=0 \\
& \xi_{2}^{J}=0, \\
& \xi_{4}^{A}=0, \quad \eta_{4}^{A}=0, \quad \xi_{4}^{L}=-10, \quad \eta_{4}^{\prime}=0 \\
& \begin{array}{ll}
\xi_{2}=0, & \eta_{2}=0 \\
\xi_{6}^{D}=0, & \eta_{6}^{D}=0,
\end{array} \\
& \eta_{2}^{J}=0 \\
& \xi_{1}^{F}=0, \\
& \begin{array}{ll}
\xi_{2}=0, & \eta_{2}=0 \\
\xi_{6}^{D}=0, & \eta_{6}^{D}=0,
\end{array} \\
& \xi_{6}^{E}=-10, \\
& \eta_{3}^{0}=0 \text { ) } \\
& \left.\begin{array}{l}
\eta_{4}^{\prime}=0 \\
\eta_{6}^{E}=0
\end{array}\right\} \begin{array}{c}
\text { translational } \\
\text { joints }
\end{array}
\end{aligned}
\]

To constrain body 1 to the ground, three conditions must be satisfied:
\[
x_{1}=0, \quad y_{1}=-300, \quad \phi_{1}=0
\]
and the driving constraint may be written as
\[
\phi_{3}-0.52-3 t=0
\]
where it is assumed that the initial angle \(\phi_{3}^{0}=30^{\circ}\).
Model 2. In the second model, the two sliders, which were modeled as bodies 4 and 6 in model 1, are combined with the revolute joints at \(A\) and \(D\). A total of four bodies, three revolute joints, and two revolute-translational joints are used in this model. The system has 12 coordinates, 6 revolute-joint constraints, 2 revolute-translational joint constraints, 3 ground constraints, and 1 driving constraint. This model is shown in Fig. 4.32(c), for which the following data can be obtained:
\[
\left.\left.\begin{array}{llll}
\xi_{2}^{B}=250, & \eta_{2}^{B}=0, & \xi_{4}^{B}=60, & \eta_{4}^{B}=0 \\
\xi_{1}^{C}=0, & \eta_{1}^{C}=0, & \xi_{2}^{C}=-250, & \eta_{2}^{C}=0 \\
\xi_{1}^{0}=0, & \eta_{1}^{0}=300, & \xi_{3}^{0}=-50, & \eta_{3}^{0}=0
\end{array}\right\} \begin{array}{c} 
\\
\xi_{3}^{A}=50,
\end{array} \begin{array}{llll}
\eta_{3}^{A}=0 & & \\
\xi_{2}^{C}=-10, & \eta_{2}^{G}=0, & \xi_{2}^{H}=15, & \eta_{2}^{H}=0 \\
\xi_{4}^{D}=-60, & \eta_{4}^{D}=0 & & \\
\xi_{1}^{E}=0, & \eta_{1}^{E}=500, & \xi_{1}^{F}=10, & \eta_{1}^{F}=500
\end{array}\right\} \begin{gathered}
\text { revints } \\
\begin{array}{c}
\text { revolute- } \\
\text { translational } \\
\text { joints }
\end{array}
\end{gathered}
\]

Note that points \(E, F, G\), and \(H\) are chosen on their corresponding lines of translation. The ground and driving constraints are modeled as in model 1.

One advantage of this model is that it has fewer coordinates than the first model. However, a disadvantage arises if an additional body is to be attached to the slider \(D\). Since in this model the slider \(D\) is not modeled as a body, no other bodies may be attached to it.

Model 3. In model 3 the slider \(D\) is chosen as a body that slides relative to the frame with a translational joint. The link \(B D\) is modeled as a massless revolute-revolute joint. This model has a total of four bodies, two revolute joints, one translational joint, one revolute-revolute joint, and one revolute-translational joint. The data for this model can be derived from Fig. 4.32(d), as follows:
\[
\left.\left.\begin{array}{llll}
\xi_{1}^{C}=0, & \eta_{1}^{C}=0, & \xi_{2}^{C}=-250, & \eta_{2}^{C}=0 \\
\xi_{1}^{O}=0, & \eta_{1}^{o}=300, & \xi_{3}^{O}=-50, & \eta_{3}^{o}=0
\end{array}\right\} \begin{array}{c}
\text { revolute } \\
\text { joints }
\end{array}\right] \begin{aligned}
& \xi_{4}^{D}=0,
\end{aligned} \begin{array}{lll}
\eta_{4}^{D}=0, & \xi_{4}^{E}=-10, & \eta_{4}^{E}=0 \\
\xi_{1}^{F}=0, & \eta_{1}^{F}=500 & \\
\xi_{2}^{B}=250, & \eta_{2}^{B}=0, & \xi_{4}^{D}=0,
\end{array} \begin{gathered}
\eta_{4}^{D}=0 \\
d=120
\end{gathered}
\]

The ground and driving constraints are modeled as in model 1.
\begin{tabular}{|c|c|c|c|c|}
\hline \[
\begin{aligned}
& \xi_{3}^{A}=50 \\
& \xi_{2}^{G}=-10 \\
& \xi_{4}^{D}=-60 \\
& \xi_{1}^{E}=0
\end{aligned}
\] & \[
\begin{aligned}
& \eta_{3}^{A}=0 \\
& \eta_{2}^{G}=0 \\
& \eta_{4}^{D}=0 \\
& \eta_{1}^{E}=500
\end{aligned}
\] & \[
\begin{aligned}
& \xi_{2}^{H}=15 \\
& \xi_{1}^{F}=10
\end{aligned}
\] & \[
\left.\begin{array}{c}
\eta_{2}^{H}=0 \\
\eta_{1}^{F}=500
\end{array}\right\}
\] & revolutetranslationa joints \\
\hline \[
\begin{aligned}
& \xi_{3}^{A}=50, \\
& \xi_{2}^{G}=-10, \\
& \xi_{4}^{D}=-60, \\
& \xi_{1}^{E}=0,
\end{aligned}
\] & \[
\begin{aligned}
& \eta_{3}^{A}=0 \\
& \eta_{2}^{G}=0 \\
& \eta_{4}^{D}=0 \\
& \eta_{1}^{E}=500
\end{aligned}
\] & \[
\begin{aligned}
& \xi_{2}^{H}=15 \\
& \xi_{1}^{F}=10
\end{aligned}
\] & \[
\left.\begin{array}{l}
\eta_{2}^{H}=0 \\
\eta_{1}^{F}=500
\end{array}\right\}
\] & revolutetranslationa joints \\
\hline
\end{tabular}

This model has the advantage of a smaller number of coordinates than model 1 , and since the slider at \(D\) is modeled as a body, it does not have the disadvantage of model 2. In none of these three models is the crank \(O A\) modeled by a massless revoluterevolute joint between points \(O\) and \(A\). Since \(O A\) is the driving link, it must have its own coordinates for implementing the driving constraint.

\section*{PROBLEMS}
4.1 Point \(C\) on body \(i\) has global coordinates \(\mathbf{r}_{i}^{C}=[3.3,2.3]^{T}\). (See Fig. P.4.1.) Attach a coordinate system to this body with its origin at \(C\). Determine the vector of coordinates if \(\alpha=50^{\circ}\). Also determine \(\mathbf{s}_{i}^{\prime P}\) and \(\mathbf{r}_{i}^{P}\) if \(d=1.2\).


Figure P.4. 1
4.2 If \(\mathbf{r}_{2}=[5,2]^{T}, \phi_{2}=30^{\circ}\), and \(\mathbf{r}_{2}^{Q}=[6,1.5]^{T}\), find the local coordinates of \(Q\). (See Fig. P.4.2)


Figure P.4. 2
4.3 Point \(P_{i}\) has local and global coordinates
\[
\mathbf{s}_{i}^{\prime P}=\left[\begin{array}{r}
1.3 \\
-2.2
\end{array}\right] \quad \mathbf{r}_{i}^{P}=\left[\begin{array}{r}
-1.7 \\
0.5
\end{array}\right]
\]

Find the translational coordinates of body \(i\) if \(\phi_{i}=32^{\circ}\). On graph paper, show the orientation of the local coordinate system with respect to the global coordinate system and locate point \(P\).
4.4 The coordinates of body \(i\) are \(\mathbf{r}_{i}=[3.2,2.8]^{T}\) and \(\phi_{i}=80^{\circ}\). Points \(A\) and \(B\) have local coordinates \(\mathbf{s}_{i}^{\prime A}=[-1.1,-0.4]^{T}\) and \(\mathbf{s}_{i}^{\prime B}=[1.9,2.3]^{T}\). Point \(C\) has global coordinates \(\mathbf{r}_{i}^{C}=\) \([5.3,4.0]^{T}\). Find the following:
(a) The global coordinates of \(A\)
(b) The global components \(\mathbf{s}_{i}^{B}\)
(c) The local coordinates of \(C\)
4.5 If Cartesian coordinates are used in constraint formulation, determine the following for each of the mechanisms shown in Fig. P.4.5:
(a) Number of bodies and coordinates
(b) Number of constraint equations
(c) Number of degrees of freedom
(d) Number of dependent and independent coordinates


Figure P.4.5
4.6 Determine the Jacobian matrix and write the velocity and acceleration equations for the constraint equations
\[
\begin{aligned}
& \Phi_{1} \equiv x_{4}+1.6 \cos \phi_{4}-0.3 \sin \phi_{4}-x_{1}+0.75 \sin \phi_{1}=0 \\
& \Phi_{2} \equiv y_{4}+1.6 \sin \phi_{4}+0.3 \cos \phi_{4}-y_{1}-0.75 \cos \phi_{1}=0
\end{aligned}
\]
4.7 Determine the minimum number of bodies and the types of joints required to model the mechanism shown in Fig. P.4.7. How many coordinates are involved?
4.8 Attach body-fixed coordinate systems to bodies 2 and 3 in Fig. P.4.8 with their origins at \(C_{2}\) and \(C_{3}\). Write constraint equations for this system in their simplest form.


Figure P.4.7


Figure P.4.8
4.9 Two vectors \(\vec{s}_{1}\) and \(\vec{s}_{2}\), attached to bodies 1 and 2, respectively, have local components \(\mathbf{s}_{1}^{\prime}=[1.2,-0.5]^{T}\) and \(\mathbf{s}_{2}^{\prime}=[-0.3,0.8]^{T}\). Vector \(\mathbf{q}\) is defined as \(\mathbf{q}=\left[x_{1}, y_{1}, \phi_{1}, x_{2}, y_{2}, \phi_{2}\right]^{T}\).
(a) If \(\phi_{1}=30^{\circ}\) and \(\phi_{2}=45^{\circ}\), evaluate the entries of the Jacobian matrix \(\Phi_{\mathbf{q}}\) for \(\Phi \equiv \mathbf{s}_{1}^{T} \mathbf{s}_{2}\).
(b) If \(x_{1}=6.2, y_{1}=1, \phi_{1}=30^{\circ}, x_{2}=-1.9, y_{2}=2.3\), and \(\phi_{2}=45^{\circ}\), cvaluate the entries of the Jacobian matrix for \(\boldsymbol{\Phi} \equiv \tilde{\mathbf{s}}_{1} \mathbf{d}\) where \(\mathbf{d}=\left[x_{2}-x_{1}, y_{2}-y_{1}\right]^{T}\).
4.10 Revolute joints \(A\) and \(B\) have local coordinates \(\mathbf{s}_{1}^{\prime A}=[0,1.5]^{T}, \mathbf{s}_{2}^{\prime A}=[-2.2,0]^{T}\), \(\mathbf{s}_{2}^{\prime B}=[2.2,0]^{T}\), and \(\mathbf{s}_{3}^{\prime B}=[2,0]^{T}\). (See Fig. P.4.10.) Coordinates of the moving bodies are estimated as \(\mathbf{r}_{1}=[-2,2.5]^{T}, \phi_{1}=-12^{\circ}, \mathbf{r}_{2}=[0.5,3.6]^{T}, \phi_{2}=-8^{\circ}, \mathbf{r}_{3}=[1.6,1.7]^{T}\), and \(\phi_{3}=56^{\circ}\). Are the constraint equations for revolute joints \(A\) and \(B\) violated or not?


Figure P.4.10
4.11 For the translational joint shown in Fig. 4.5, use the vector product operation to derive the first constraint of Eq. 4.12.
4.12 For the revolute-translational joint shown in Fig. 4.9, use the vector product operation to derive the constraint of Eq. 4.16.
4.13 Since two bodies connccted by a translational joint can translate relative to one another, it is possible for vector \(\vec{d}\) to become a zero vector at some instant. Does this cause any numerical difficulty in kinematic analysis?
4.14 For the mechanism shown in Fig. P.4.14, link \(A B E\) is the crank. If only revolute- and trans-lational-joint constraints are available, determine the number of bodies and the number of constraint equations needed to model this system. If, in addition, revolute-revolute and revolute-translational joint constraints are available, show different ways that the number of bodies and the number of constraints can bc minimized.


Figure P.4. 14
4.15 Derive constraint equations for a rack and pinion where
(a) The rack moves on a translational joint with respect to the ground and the line of translation is not parallel to any of the coordinate axes.
(b) The rack and the pinion can translate and rotate with respect to the ground (general case).
4.16 If \(\mathbf{g}_{i}^{\prime}=[\mu, \nu]_{i}^{T}\) is tangent to a curve at a point \(P_{i}\), show that \(\mathbf{n}_{i}^{\prime}=[-\nu, \mu]_{i}^{T}\) is normal to the curve at \(P_{i}\). Then, modify the cam-follower constraints of Eq. 4.29 by using the normal vectors and vector product instead of the tangent vector and scalar product.
4.17 Most books on numerical methods provide algorithms and program listings for cubic spline functions. Experiment with one of these algorithms and try to determine values of a function and its first and second derivatives at randomly specified points. The program can later be used to expand the capabilities of the kinematic and dynamic analysis programs listed in Chaps. 5 and 10.
4.18 Derive constraint equations for each of the cam-follower pairs shown in Fig. P.4.18.
4.19 Simplify the general formulation of constraint equations for the cam-follower pairs shown in Fig. P.4.19. For (a) and (b) the contacting surface of the cam is a sine function, and for (c), (d), and (e) the cam has a circular outline.


Figure P.4. 18


Figure P.4.19
4.20 Verify the entries of the Jacobian matrix and vector \(\boldsymbol{\gamma}\) listed in Tables 4.2 and 4.3.
4.21 Derive expressions for the entries of the Jacobian matrix and vector \(\gamma\) for the spur gear constraint of Eq. 4.18.
4.22 Derive expressions for the entries of the Jacobian matrix and vector \(\gamma\) for the cam-follower constraints of Eq. 4.27. Note that the Jacobian matrix must contain an additional column associated with the artificial coordinate \(\theta_{i}\).
4.23 What is the size of the Jacobian matrix for a four-bar mechanism modeled by four bodies and four revolute joints? Show the location of the nonzero entries in this matrix. What percentage of the elements of the matrix are nonzero elements?
4.24 Derive constraint equations to keep the translational speed of a body constant along a known direction denoted by a unit vector \(\vec{u}\). Consider two cases:
(a) Vector \(\vec{u}\) is fixed to the global coordinate system.
(b) Vector \(\vec{u}\) is fixed to the body coordinate system.

\section*{5}

\section*{A FORTRAN Program for Analysis of Planar Kinematics}

In this chapter, a FORTRAN program for planar kinematic analysis is presented. The program contains only two types of lower-pair joints - a revolute joint and a translational joint. With only these two basic joints, a large class of mechanisms can be analyzed. The program is organized in a form that can be expanded to include other kinematic pairs, such as those formulated in Chap. 4. The exercises given at the end of this chapter provide a pattern to assist in expansion of the program.

The subroutines LINEAR and LU that were discussed in Chap. 3, are used in this program. They can be appended to this program without any modification. The program is written in standard FORTRAN and should run on most FORTRAN compilers without any difficulty. However, it is possible that some compilers will require minor modification of the program.

\subsection*{5.1 KINEMATIC ANALYSIS PROGRAM (KAP)}

The main routine of the kinematic analysis program performs three major tasks:
1. Reads input data, either directly or by making calls to other input subroutines.
2. Splits the working arrays \(A\) and IA into smaller subarrays by defining pointers (addresses in arrays).
3. Makes calls to subroutine KINEM for kinematic analysis.

A detailed explanation of these three major tasks follows.

Input/Output. Input is provided to the program through the console in unformatted form: \(\operatorname{READ}(1, *) \ldots\) The output is written on the console screen with specified format; e.g., WRITE (1,230)....

Working Arrays. There are two main working arrays in this program-one real array A and one integer IA. The two arrays are dimensioned to 500 and 200, respectively (MAXA \(=500\) and MAXIA \(=200\) ), which is sufficient for most small problems. For larger problems - those with larger numbers of bodies or kinematic joints, for example - these dimensions can be increased. The program computes the minimum dimensions that are required for the two arrays and provides a warning message when the existing dimensions are not sufficient.

Number of Elements. The first request the program makes is to

> ENTER NB, NR, NT, NG, NS, ND, NP

These data are defined as follows:
NB Number of bodies in the system, including ground
NR Number of revolute joints in the system
NT Number of translational joints in the system
NG Number of bodies that are attached to (or considered as) ground
NS Number of simple constraints in the system
ND Number of driving constraints (driving links)
NP Number of points of interest
Further explanation of these variables and their functions is given in the following sections. The program computes the number of coordinates N and the total number of constraint equations M , including the driving constraints, from this first set of data. If N is not equal to \(\mathbf{M}\), then an error message is given by the program.

Subarrays. The working arrays A and IA are divided into smaller subarrays, according to the number of elements in the problem. The subarrays and their corresponding pointers and lengths are shown in Fig. 5.1. Pointers N1, N2, . . locate the beginnings of subarrays of \(A\), and pointers M1, M2, . . locate the beginnings of subarrays IA; e.g., subarray TJ begins at \(\mathrm{A}(\mathrm{N} 2)\), so \(\mathrm{TJ}(2)\) is the same as \(\mathrm{A}(\mathrm{N} 2+1)\), and so forth. Some pointers that have not been used, such as N8, N9, or M8, or other additional pointers can be included for further expansion of the program. The function of the subarrays is explained in the following sections.

Input Data. The main program makes calls to other subroutines, such as INBODY and INRVLT, to read additional information for the problem at hand. These subroutines are discussed in Sec. 5.1.1.

Time Parameters. The last prompt given by the program is ENTER STARTING TIME, FINAL TIME, AND TIME INCREMENT

These values are assigned to the variable names TO, TE, and DT. During the analysis, the time parameter T is incremented by DT from T 0 to TE.


Figure 5.1 Subarrays of A and IA with their corresponding pointers and lengths.

Kinematic Analysis. After all of the input data are read by the program, the main program makes a call to subroutine KINEM for kinematic analysis. Prior to this call, three variables are defined and given specific values:
\begin{tabular}{ll} 
NRMAX & \begin{tabular}{c} 
Maximum number of iterations allowed for the Newton-Raphson \\
algorithm at each time step
\end{tabular} \\
FEPS & Maximum error allowed for constraint violation \\
EPSLU & Error tolerance for the L-U factorization algorithm
\end{tabular}

The user may modify the program to assign values to these variables, by a READ statement, depending on the accuracy required in the response computation.

The main routine of the KAP program is as follows:
```

C. . . . . . . . .KINEMATIC ANALYS IS
C
C. ...........Main Program.
C
OCMMON /OONST / NRMAX,FEPS,EPSLU
OCMMDN /MPNTR / M1,M2,M3,M4 ,M5 ,M6,M7,M8,M9 ,M10
OCMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,ND,NP
COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N14,N15,N16
COMMON /RONOOL/ IR,IC,M,N
COMMON /TIME / TO,TE,DT,T
DIMENSION A(500),IA(200)
C.....For more space in A and IA arrays, increase the dimension and
C.....update MAXA and MAXIA accordingly
MAXA =500
MAXIA=200
C.....Read number of bodies, revolute joints, translational joints,
C....grounded bodies, simple constraints, drivers, points of interest
10 WRITE (1,200)
READ(1,* ) NB,NR,NT,NG,NS,ND,NP
C....Determine number of coordinates N and number of constraints M
N=3*NB
M=2*(NR+NT)+3*NG+NS+ND
C....N must be equal to M (including the driver constraints)
IF (M.EQ.N) GOTO 20
WRITE(1,210) N,M
GOTO 10
C.....Define pointers and split A and IA into subarrays
C.....Subarrays for storing input data
20 N1=1
N2=N1+4*NR
N3-N2+7*NT
N4=N3+3*NG
N5=N4+ NS
N6=N5+3*ND
N7=N6+2*NB
M1=1
M2=M1+2*NR
M3=M2+2*NT
M4=M3+6*NG
M5=M4+2*NS
M7-M5+2*ND
C.....Subarrays for vector of coordinates, velocities, etc.
N10=N7+2*NP
N11=N10+N
N120N11+N
N14=N12+N

```
```

        N15=N14+N*M
        N16-N15+M
        NUSED-N16+M-1
        M10=M7+NP
        MUSED=M10+N-1
    C.....Check for sufficient storage space in A and IA arrays
IF(NUSED.LE.MAXA .AND. MUSED.LE.MAXIA)GOTO }3
WRITE(1, 220) NUSED,MUSED
STOP
C.....Read initial estimates on the coordinates
30 CALL INBODY (A(N10),NB)
C.....Read revolute joints data
IF (NR.GT.0) CALL INRVLT (A(N1),IA(M1),NR)
C....Read translational joints data
IF (NT.GT.0) CALL INTRAN (A(N2),IA(M2),NT,A(N10),NB)
C.....Read ground constraints data
NG3-3*NG
IF (NG.GT.0) CALL INGRND (A(N3),IA(M3),NG,A(N10),NG3,NB)
C.....Read simple constraints data
IF (NS.GT.0) CALL INSMPL (A(N4),IA(M4),NS,A(N10),NB)
C.....Read driver constraints dataA
IF (ND.GT.0) CALL INDRVR (A(N5),IA(M5),ND)
C.....Read special points of interest data
IF (NP.GT.0) CALL INPOIN (A(N7),IA(M7),NP)
C.....Read initial time, final time, and the time increment
WRITE(1,230)
READ(1,* ) T0,TE,DT
C.....End of input data
NRMAX=20
FEPS =0.001
EPSLU=0.001
C.....Start KINEMATIC ANALYSIS......
CALL. KINEM (A,IA,MAXA,MAXIA)
STOP
200 FORMAT (5X, 'ENTER NB,NR,NT,NG,NS,ND,NP')
210 FORMAT(5X,'***INPUT ERROR*** N=',I 3,' M=',I3)
220 FORMAT(5X,'***DIMENSION ON A AND/OR IA ARRAYS NOT SUFFICIENT***',
+ /,10X,'MINIMMM DIMENSION ON A MUST BE',I5,
+ /,10X,'MINIMMM DIMENSION ON IA MUST BE', I5)
230 FORMAT(5X,'ENTER STARTING TIME, FINAL TIME, AND DT')
END

```

\subsection*{5.1.1 Model-Description Subroutines}

The following subroutines are called by the main routine of KAP to read the description of the model.

Subroutine INBODY. This subroutine reads initial estimates of \(x_{i}, y_{i}\) and \(\phi_{i}\) for each body with a prompt:

FOR BODY i ENTER INITIAL EST. ON X, Y, PHI
This information is saved in array Q , dimensioned as \(\mathrm{Q}(3, \mathrm{NB})\). For example, for body \(i\),
\[
\begin{aligned}
& x_{i} \rightarrow \mathrm{Q}(1, \mathrm{I}) \\
& y_{i} \rightarrow \mathrm{Q}(2, \mathrm{I}) \\
& \phi_{i} \rightarrow \mathrm{Q}(3, \mathrm{I})
\end{aligned}
\]

Subroutine INBODY is as follows:
```

    SUBROUTINE INBODY (Q,NB)
    DIMENSION Q(3,NB)
    DO 10 I=1,NB
        WRITE(1,200) I
        READ(1,*) (Q(J,I), J=1,3)
    10 OONTINUE
REIURN
200 FORMAT(5X,'FOR BODY',I4,' ENTER INITIAL EST. ON X, Y, PHI')
END

```

Subroutine INRVLT. This subroutine is called if NR \(>0\), to read information describing the revolute joints in the system (refer to Sec. 4.2.1). The prompt given by this subroutine is
```

FOR REV. JOINT NO. k ENTER
BODY NOS. I AND J, THEN
XI-P-I, ETA-P-I, XI-P-J, ETA-P-J

```

This prompt is repeated for \(k=1, \ldots\), NR. Body numbers \(i\) and \(j\) that are connected by revolute joint \(k\) are stored in array IRJ, dimensioned as IRJ(NR,2).

The local components of vectors that locate the revolute joints on the bodies are stored in array RJ, dimensioned as \(\operatorname{RJ}(\mathrm{NR}, 4)\). For example, for the \(k\) th revolute joint in a system, connecting bodies \(i\) and \(j\), the entries for the IRJ and RJ arrays are
\[
\begin{aligned}
I & \rightarrow \operatorname{IRJ}(\mathrm{~K}, 1) & J & \rightarrow \operatorname{IRJ}(\mathrm{~K}, 2) \\
\xi_{i}^{P} & \rightarrow \mathrm{RJ}(\mathrm{~K}, 1) & \xi_{j}^{P} & \rightarrow \mathrm{RJ}(\mathrm{~K}, 3) \\
\eta_{i}^{P} & \rightarrow \mathrm{RJ}(\mathrm{~K}, 2) & \eta_{j}^{P} & \rightarrow \mathrm{RJ}(\mathrm{~K}, 4)
\end{aligned}
\]

Subroutine INRVLT is as follows:
```

    SUBROUTINE INRVLT (RJ,IRJ,NR)
    DIMENSION RJ(NR,4),IRJ(NR,2)
    DO 10 K=1,NR
        WRITE(1,200) K
        READ(1,*) (IRJ (K,L) , L=1 , 2),(RJ (K,L) ,L=1,4)
    10 OONTINUE
RETURN
200 FORMAT(5X,'FOR REV. JOINT NO.',I3,' ENTER BODY NOS. I AND J',/,
+
END

```

Subroutine INTRAN. The subroutine INTRAN is called if NT \(>0\), to read information describing the translational joints in the system (refer to Sec. 4.2.1). The prompt given by this subroutine is
```

FOR TRAN. JOINT NO. k ENTER
BODY NOS. I AND J, THEN
XI-P-I, ETA-P-I, XI-Q-I, ETA-Q-I
XI-P-J, ETA-P-J

```

This prompt is repeated for \(k=1, \ldots, \mathrm{NT}\). Body numbers \(i\) and \(j\) that are connected by the \(k\) th translational joint are stored in array ITJ, dimensioned as ITJ(NT,2). The local
coordinates of points \(P_{i}\) and \(P_{j}\) are stored in array TJ, dimensioned as TJ(NT,7). The local coordinates of point \(Q_{i}\) are not saved directly. Instead,
\[
\alpha_{i}=\xi_{i}^{P}-\xi_{i}^{Q}, \quad \beta_{i}=\eta_{i}^{P}-\eta_{i}^{Q}
\]
are stored in vector TJ. For the \(k\) th translational joint in the system, the elements of the ITJ and TJ arrays are:
\[
\begin{array}{rlrl}
I & \rightarrow \mathrm{ITJ}(\mathrm{~K}, 1) & J & \rightarrow \mathrm{ITJ}(\mathrm{~K}, 2) \\
\xi_{i}^{P} & \rightarrow \mathrm{TJ}(\mathrm{~K}, 1) & \xi_{j}^{P} & \rightarrow \mathrm{TJ}(\mathrm{~K}, 5) \\
\eta_{i}^{P} & \rightarrow \mathrm{TJ}(\mathrm{~K}, 2) & \eta_{j}^{P} & \rightarrow \mathrm{TJ}(\mathrm{~K}, 6) \\
\alpha_{i} & \rightarrow \mathrm{TJ}(\mathrm{~K}, 3) & \phi_{i}^{0}-\phi_{j}^{0} & \rightarrow \mathrm{TJ}(\mathrm{~K}, 7) \\
\beta_{i} & \rightarrow \mathrm{TJ}(\mathrm{~K}, 4) &
\end{array}
\]

Subroutine INTRAN is as follows:
```

    SUBROUTINE INTRAN (TJ,ITJ,NT,Q,NB)
    DIMENS ION TJ (NT, 7), ITJ (NT, 2) , Q (3, NB)
    DO \(10 \mathrm{~K}=1\), NT
        WRITE (1,200) K
        \(\operatorname{READ}(1, *)(I T J(K, L), L=1,2),(T J(K, L), L=1,6)\)
        \(\mathrm{TJ}(\mathrm{K}, 3)=\mathrm{TJ}(\mathrm{K}, 1)-\mathrm{TJ}(\mathrm{K}, 3)\)
        \(\mathrm{TJ}(\mathrm{K}, 4)=\mathrm{TJ}(\mathrm{K}, 2)-\mathrm{TJ}(\mathrm{K}, 4)\)
    $10 \mathrm{TJ}(\mathrm{K}, 7)=\mathrm{Q}(3,1 \mathrm{TJ}(\mathrm{K}, 1))-\mathrm{Q}(3,1 \mathrm{TJ}(\mathrm{K}, 2))$
RETURN
200 FORMAT(5X,'FOR TRAN. JOINT NO.', 13 ,' ENTER BODY NOS. I AND J', /,
+10X, 'XI-P-I ,ETA-P-I ,XI-Q-I ,ETA-Q-I ,XI-P-J ,ETA-P-J')
END

```

Subroutine INGRND. The subroutine INGRND is called if NG \(>0\), to read the number(s) of a body or bodies constrained to the ground by giving the prompt

ENTER BODY NO. FOR NO. \(k\) GROUNDED BODY
This prompt is repeated NG times. This subroutine also transfers initial estimates of the coordinates for the grounded bodies from array \(Q\) to array GR, dimensioned as \(\mathrm{GR}(3 * \mathrm{NG})\). Each body that is constrained to the ground is treated as having three simple constraints on its \(x, y\), and \(\phi\) motion. This information is stored in an integer array IGR, dimensioned as \(\operatorname{IGR}(3 * \mathrm{NG}, 2)\). The way the data are stored in the IGR and GR arrays is similar to the storage of data in the SM and ISM arrays for simple constraints, as will be shown in the next subsection.

Subroutine INGRND is as follows:
SUBROUTINE INGRND (GR, IGR,NG, Q,NG3,NB)
DIMENSION GR (NG3) , IGR (NG3, 2) , Q (3, NB)
DO \(10 \mathrm{~K}=1\), NG
WRITE (1,200) K
\(\operatorname{READ}(1, *) 1 B\)
DO \(10 \quad \mathrm{~J}=1,3\)
\(\mathrm{KK}=(\mathrm{K}-1) * 3+\mathrm{J}\)
IGR(KK, 1)=IB
IGR (KK, 2 ) =J
\(\mathrm{GR}(\mathrm{KK})=\mathrm{Q}(\mathrm{J}, \mathrm{IB})\)
10 CONIINUE
RETURN
200 FORMAT (5X, 'ENTER BODY NO. FOR NO.', I 3,' GROUNDED BODY') END

Subroutine INSMPL. This subroutine is called if NS \(>0\), to read information on the bodies that have simple constraints (refer to Eqs. 4.31, 4.32, and 4.33). The prompt given by this subroutine is

FOR SIMPLE CONSTRAINT NO. k ENTER
BODY NO. AND 1, 2, OR 3 FOR \(X, Y\), OR PHI CONSTRAINT DIRECTION

This prompt is repeated NS times. The information is stored in array ISM, dimensioned as ISM(NS,2). In addition, this subroutine transfers the initial values of the coordinates from array \(Q\) to array SM, dimensioned as \(\operatorname{SM}(\mathrm{NS})\). For example, if the \(k\) th simple constraint acts in the \(y\) direction on body \(i\), then
\[
\mathrm{I} \rightarrow \mathrm{ISM}(\mathrm{~K}, 1), 2(\text { for } \mathrm{y}) \rightarrow \mathrm{ISM}(\mathrm{~K}, 2), \mathrm{Q}(2,1) \rightarrow \mathrm{SM}(\mathrm{~K})
\]
where \(\mathrm{Q}(2, \mathrm{I})\) contains the initial value on the \(y\) coordinate of body \(i\).
The entries of array SM will be used as the constant \(c_{1}, c_{2}\), or \(c_{3}\) in Eq. 4.31, 4.32, or 4.33.

Subroutine INSMPL is as follows:
```

        SUBROUTINE INSMPL (SM, ISM,NS,Q,NB)
        DIMENSION SM(NS),ISM(NS,2),Q(3,NB)
        DO 10 K=1,NS
            WRITE(1,200) K
            READ(1,*) (ISM(K,L),L=1,2)
            SM(K)=Q(ISM(K,2),I SM(K,1))
        10 CONTINUE
            RETURN
    200 FORMAT(5X,'FOR SIMPLE OONSTRAINT NO.',I 3,' ENTER BODY NO.',/,
+ 10X,'AND 1,2,OR 3 FOR X,Y,OR PHI OONSTRAINT DIRECTION')
END

```

Subroutine INDRVR. This subroutine is called if ND \(>0\), to read information on the driving link(s) (refer to Sec. 4.2.8). The prompt given by this subroutine is

This prompt is repeated ND times. The body number(s) and indices indicating \(x, y\), or \(\phi\) are constrained and stored in an integer array IDR, dimensioned as IDR(ND,2). The initial value of \(x, y\), or \(\phi\) and its velocity and acceleration at \(t=0\) are stored in array DR, dimensioned as \(\operatorname{DR}(\mathrm{ND}, 3)\). For example, if the \(k\) th driver constraint is acting on the rotation of body \(i\) as follows:
\[
\phi_{i}-\phi_{i}^{0}-\omega t-0.5 \alpha t^{2}=0
\]
then
\[
\begin{aligned}
I \rightarrow \operatorname{IDR}(\mathrm{~K}, 1), \quad 3(\text { for } \phi) \rightarrow \operatorname{IDR}(\mathrm{K}, 2), \\
\phi_{i}^{0} \rightarrow \operatorname{DR}(\mathrm{~K}, 1), \quad \omega \rightarrow \operatorname{DR}(\mathrm{K}, 2), \quad \alpha \rightarrow \operatorname{DR}(\mathrm{K}, 3)
\end{aligned}
\]

Subroutine INDRVR is as follows:
```

            SUBROUTINE INDRVR (DR,IDR,ND)
            DIMENSION DR(ND, 3),IDR(ND, 2)
            DO 10 K=1,ND
            WRITE(1,200) K
            READ(1,*) (IDR(K,L),L=1,2),(DR(K,L) ,L=1,3)
    10 OONTINUE
RETURN
200 FORMAT( 5X,'FOR DRIVER NO.',I 3,' ENTER BODY NO.',/,
+ 10X,'1, 2, OR 3 FOR X, Y, OR PHI',/,
+ 10X,'INITIAL POSITION, VELOCITY, AND ACCELERATION')
END

```

Subroutine INPOIN. This subroutine is called if NP \(>0\), to read information on points of interest defined by the user. The user may specify some points of interest on one or more bodies (refer to Eqs. 4.1, 4.43, and 4.45). Hence, the program will report the global position, velocity, and acceleration for these points at every time step. Note that the program reports the coordinates of all the bodies in the system automatically, and therefore there is no need to specify the centroid of a body as a point of interest. The prompt from this subroutine is

FOR POINT OF INTEREST NO. k ENTER
BODY NO., THEN XI-P AND ETA-P COORDINATES
This prompt is repeated NP times. The body numbers are stored in array IPI, dimensioned as \(\operatorname{IPI}(\mathrm{NP})\). The \(\xi_{i}^{P}\) and \(\eta_{i}^{P}\) coordinates are stored in array PI, dimensioned as PI(NP,2).

Subroutine INPOIN is as follows:
SUBROUTINE INPOIN (PI,IPI,NP)
DIMENSION PI (NP,2),IPI(NP)
DO \(10 \mathrm{~K}=1\), NP
WRITE \((1,200) \mathrm{K}\)
\(\operatorname{READ}(1, *) \operatorname{IPI}(\mathrm{K}),(\mathrm{PI}(\mathrm{K}, \mathrm{L}), \mathrm{L}=1,2)\)
10 CONTINUE
RETURN
200 FORMAT( 5X, 'FOR POINT OF INTEREST NO.', I 3,' ENTER BODY NO.' ,/, \(+10 X\), 'XI-P AND ETA-P COORDINATES') END

\subsection*{5.1.2 Kinematic Analysis}

Following the process of inputting data to describe a model, the main routine calls subroutine KINEM. This subroutine is organized according to Algorithm K-II of Sec. 3.2.2, with a few minor additional steps.

Subroutine KINEM. This subroutine increments the time variable T from the initial time T0 to final time TE by steps DT. At each time step, the subroutine performs position, velocity, and acceleration analysis, and reports the results by making calls to other subroutines. Two flags are set by this subroutine:

JACOB A flag for Jacobian matrix evaluation.
EQ.0: Jacobian matrix does not need to be evaluated.
EQ.1: Jacobian matrix must be evaluated.

IFNCT A flag for evaluation of constraint equations.
EQ.0: No function needs to be evaluated.
EQ.1: Constraint equations must be evaluated.
EQ.2: The right side of the velocity equations must be evaluated.
EQ.3: The right side of the acceleration equations must be evaluated.
Position Analysis. The flags JACOB and IFNCT are both set to 1 and a call is made to subroutine NUTON2 for position analysis.

Velocity Analysis. The flag JACOB is set to 1 and IFNCT is set to 2. A call to subroutine FUNCT evaluates the Jacobian matrix and the right side of the velocity equations. At this step, subroutine LINEAR (refer to Sec. 3.3.5) is used to solve for the velocities according to Eq. 3.14.

Acceleration Analysis. The flag JACOB is set to 0 , since the Jacobian matrix and its corresponding \(\mathbf{L}\) and \(\mathbf{U}\) matrices are still valid from the velocity analysis step, and IFNCT is set to 3 . A call to subroutine FUNCT evaluates the right side of acceleration equations. At this step, subroutine LINEAR is used to solve for the acceleration, according to Eq. 3.16.

The velocities and accelerations determined by subroutine LINEAR are originally stored in array \(F\). The velocities and accelerations are then transferred from array F to arrays QD and QDD, respectively, by subroutine TRANSF. The contents of arrays QD and QDD are organized in a way similar to the arrangements of array Q .

Computationally, the evaluation of trigonometric functions is time-consuming. Each time new values for coordinates are calculated, the sine and cosine of the rotational coordinates are computed and stored in array RB by subroutine TRIG. Array RB is dimensioned \(\mathrm{RB}(\mathrm{NB}, 2)\), and, for example, for body \(i\),
\[
\sin \phi_{i} \rightarrow \mathrm{RB}(\mathrm{I}, 1), \quad \cos \phi_{i} \rightarrow \mathrm{RB}(\mathrm{I}, 2)
\]

The contents of array RB are used in several other subroutines.
Subroutines KINEM, TRANSF, and TRIG are as follows:
```

        SUBROUTINE KINEM (A,IA,MAXA,MAXIA)
        COMMON /ANALYS/ JACOB,IFNCT
        COMMON /OONST / NRMAX,FEPS,EPSLU
        COMMDN /MPNTR / M1 ,M2 ,M3 ,M4,M5 ,M6 ,M7 ,M8 ,M9 ,M10
        COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,ND,NP
        COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N14,N15,N16
        COMMON /RONOOL/ IR,IC,M,N
        COMMDN /TIME / TO,TE,DT,T
        DIMENSION A(MAXA),IA(MAXIA)
        WRITE (1,200)
    C.....Initia1ize time variable
        I STEP=0
        T=T0
    C....Calculate sine and cosine of rotational coordinates
CALL TRIG (A(N6),A(N10),NB)
C.....Position (coordinate) analysis......
10 JACOB=1
IFNCT=1
CALL NUTON2 (A, IA,MAXA,MAXIA,A(N10),A(N15),NB, JACOB)
C....Velocity analysis.
IFNCT=2
CALL FUNCT (A,IA,MAXA,MAXIA,A(N10),A(N11),A(N14),A(N15),JACOB)
CALL LINEAR (A(N14),A(N15),A(N16),IA(M10),M,JAOOB,EPSLU)

```
```

C.....Transfer velocities from F array to QD array
CALL TRANSF (A(N11),A(N15),N)
C.....Acceleration analysis
JACOB=0
I FNCT=3
CALL FUNCT (A,IA,MAXA,MAXIA, (N10),A(N11),A(N14),A(N15), JACOB)
CALL LINEAR (A(N14),A(N15),A(N16),IA(M10),M, JAOOB,EPSLU)
C.....Transfer accelerations from F array to QDD array
CALL TRANSF (A(N12),A(N15),N)
C.....Report the result for this time step
CALL REPORT (A(N6),A(N7),A(N10),A(N11),A(N12),IA(M7),NB,NP,T)
C.....Increment the time variable
I STEP=I STEP+1
T=TO+DT*FLOAT( I STEP)
IF (T.GT.TE.OR.DT.EQ.0.0) STOP
GO TO 10
200 FORMAT(///,10X,'***** KINEMATIC ANALYSIS
*****',//)
END
SUBROUTINE TRANSF (B,F,N)
DIMENSION B(N),F(N)
DO 10 I=1,N
10
B(I)=F(I)
RETURN
END
SUBROUTINE TRIG (RB,Q,NB)
DIMENS ION RB(NB,2),Q(3,NB)
DO 10 I=1,NB
RB(I,1)=SIN(Q(3,I))
RB(I, 2)=COS(Q(3,I))
10 OONTINUE
RETURN
END

```

Subroutine NUTON2. Subroutine NUTON2 is similar to subroutine NEWTON of Sec. 3.4.3, with minor modifications. This subroutine is called by subroutine KINEM for position analysis when \(J A C O B=1\) and IFNCT \(=1\). This subroutine calls subroutine FUNCT to evaluate the Jacobian matrix and constraint equations. If NewtonRaphson iteration fails to converge in NRMAX iterations, analysis will be terminated with the message

\section*{***CONVERGENCE FAILED***}

This failure may be caused by either of the following:
1. Large time increments DT (numerical failure)
2. Nonexistence of a solution (physical impossibility)

Subroutine NUTON2 is as follows:
```

SUBROUTINE NUTON2 (A, IA,MAXA,MAXIA,Q,F,NB, JACOB)
OOMMON /OONST / NRMAX,FEPS,EPSLU
COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
COMMDN /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N14,N15,N16
OCMMON /ROWOOL/ IR,IC,M,N
DIMENSION Q(N),A(MAXA),IA(MAXIA),F(M)
DO 20 I=1,NRMAX
CALL FUNCT (A, IA,MAXA,MAXIA,A(N10),A(N11),A(N14),A(N15),JACOB)
CALL LINEAR (A(N14),A(N15),A(N16),IA(M10),M,JAODB,EPSLU)

```
```

            IOONVR=0
            DO 10 J=1,N
        IF(ABS(F(J )).GT.FEPS ) ICONVR=1
                        Q(J)=Q(J )-F(J )
                                CALL TRIG (A(N6),Q,NB)
            IF (ICONVR) 30,30,20
        20 CONTINUE
        WRITE(*, 200)
        STOP
    30 RETURN
    200 FORMAT (5X,'***CONVERGENCE FAILED***')
END

```

Subroutine REPORT. This subroutine reports the values of coordinates for all bodies in the system at the end of each time step. In addition, the global coordinates, velocities, and accelerations of the special points of interest are computed and reported.

Subroutine REPORT is as follows:
```

    SUBROUTINE REPORT (RB,PI,Q,QD,QDD,IPI,NB,NP,T)
    DIMENSION Q(3,NB),QD(3,NB),QDD(3,NB),PI(NP,2),IPI (NP),RB(NB,2)
    WRITE (1, 200) T
    DO 10 I=1,NB
    10 \operatorname{MRITE}(1,210)I,(Q(J,I), J=1,3),(QD(J,I ), J=1,3),(QDD(J,I ), J=1 , 3)
15 IF (NP.EQ.0) RETURN
WRITE(1,220)
DO 20 K=1,NP
I=IPI (K)
XPMX=PI(K,1)*RB(I,2)-PI(K,2)*RB(I,1)
YPMY=PI (K,1)*RB(I, 1)+PI (K, 2)*RB(I, 2)
XP =Q(1,I )+XPMX
YP =Q(2,I)+YPMY
XDP =QD(1,I)-YPMY*QD(3,I)
YDP =QD(2,I)+XPMX*QD(3,I)
XDDP=QDD(1,I)-XPMX*QD(3,I)**2-YPMY*QDD(3,I)
YDDP=QDD (2,I) -YPMY*QD (3,I)**2+XPMX*QDD (3,I)
WRITE (1, 230) K,XP,YP, XDP ,YDP, XDDP , YDDP
20 OONTINUE
RETURN
200 FORMAT(/,'TIME =',F10.4,/,'-------------..','/,
+ ' BODY',5X,'X',7X,'Y',5X,'PHI',6X, 'XD',6X,'YD',4X,'PHID',
+ 'XD',6X,'YD',6X, 'XDD',6X, 'YDD')
210 FORMAT(I3,6F8.3,3F9.3)
220 FORMAT('POINTS OF INTEREST',/,' NO.',5X,'X',7X,'Y',6X,
'XD',6X, 'YD',6X, 'XDD',6X, 'YDD')
230 FORMAT(13,4F8.3,2F9.3)
END

```

\subsection*{5.1.3 Function Evaluation}

Subroutine FUNCT. The constraint equations, the right sides of the velocity and acceleration equations, and the Jacobian matrix are evaluated by calling subroutine FUNCT. This subroutine initializes a row number counter IR to zero. This counter gives the total number of functions or rows in the Jacobian matrix. The counter is incremented by subroutines RVLT, TRAN, SMPL, and DRVR. After all of the constraint equations have been considered, IR is equal to M (or N ).

Prior to evaluation of the entries in the Jacobian matrix, the entries in the matrix FQ are initialized to zero. Subroutine FUNCT calls subroutines RVLT, TRAN, SMPL (twice), and DRVR to evaluate those equations and the Jacobian matrix corresponding
to revolute joints, translational joints, ground, simple constraints, and driver links. The order of calling these subroutines has been written arbitrarily and does not have any significance.

Subroutine FUNCT is as follows:
```

SUBROUTINE FUNCT (A,IA,MAXA,MAXIA,Q,QD, FQ,F, JACOB)
OCMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,ND,NP
CCMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N14,N15,N16
COMMON /RONOOL/ IR,IC,M,N
DIMENS ION A(MAXA),IA(MAXIA),Q(N),QD(N),FQ(M,N),F(M)
IR=0
IF (JACOB.EQ.0) GOTO 20
DO 10 I=1,M
DO 10 J=1,N
FQ(I, J )=0.0
20 IF (NR.GT.0) CALL RVLT (A(N1),IA(M1),A(N6),Q,QD,FQ,F,NR,NB)
IF (NT.GT.0) CALL TRAN (A(N2),IA(M2),A(N6),Q,QD,FQ,F,NT,NB)
IF (NG.GT.0) CALL SMPL (A(N3),IA(M3),Q,FQ,F,NG3,NB)
IF (NS.GT.0) CALL SMPL (A(N4),IA(M4),Q,FQ,F,NS,NB)
IF (ND.GT.0) CALL DRVR (A(N5),IA(M5),Q,FQ,F,ND,NB)
RETURN
END

```
10

Subroutine RVLT. Subroutine RVLT is called by subroutine FUNCT when \(\mathrm{NR}>0\). It evaluates the constraint equation violations, the right sides of the velocity and acceleration equations, and the entries of the Jacobian matrix corresponding to the revolute joints in the system (refer to Sec. 4.2.1). Array F is used to store the constraint equation violations and the right sides of the velocity and acceleration equations, depending on the value of the flag IFNCT. The nonzero entries of the Jacobian matrix are stored in matrix form in array FQ , when \(\mathrm{JACOB}=1\).

Subroutine RVLT is as follows:
```

            SUBROUTINE RVLT (RJ,IRJ,RB,Q,QD,FQ,F,NR,NB)
            OCMMDN /ANALYS/ JAOOB,IFNCT
            COMMDN /ROWOOL/ IR,IC,M,N
            DIMENSION RJ (NR,4),IRJ(NR,2),RB(NB,2),Q(3,NB),QD(3,NB),
            + FQ(M,N),F(M)
            DO 100 K=1,NR
            I=IRJ(K,1)
            J=IRJ(K,2)
            XPIMXI= RJ(K,1)*RB(I, 2)-RJ(K,2)*RB(I,1)
            YPIMYI= RJ(K,1)*RB(I,1)+RJ(K,2)*RB(I,2)
            XPJMXJ= RJ (K,3)*RB(J,2)-RJ(K,4)*RB(J,1)
            YPJMYJ= RJ (K,3)*RB(J,1)+RJ(K,4)*RB(J,2)
            GOTO ( 10,20,30) , IFNCT
    C......Constraint equations
10 F(IR+1)= Q(1,I )+XPIMXI -Q(1,J)-XPJMXJ
F(IR+2)=Q(2,I )+YPIMYI-Q(2,J)-YPJMYJ
GOTO 40
C....Right-hand-side of velocity equations
20 F(IR+1)=0.0
F(IR+2)=0.0
GOTO 40
C.....Right-hand-side of acceleration equations
30 F(IR+1)= XPIMXI*QD(3,I)**2-XPJMXJ*QD(3,J)**2
F(IR+2)= YPIMYI*QD(3,I)**2-YPJMYJ*QD(3,J)**2
40 IF (JAOOB) 60,60,50

```
```

C....Jacobian matrix nonzero entries
50 ICI= 3*(I-1)
ICJ= 3*(J-1)
FQ(IR+1,ICI+1)=1.0
FQ(IR+1,ICI+3)=-YPIMYI
FQ(IR+1,ICJ+1)=-1.0
FQ(IR+1,ICJ+3)= YPJMYJ
FQ(IR+2,ICI+2)=1.0
FQ(IR+2,ICI+3)= XPIMXI
FQ(IR+2,ICJ+2 )=-1.0
FQ(IR+2,ICJ + 3)=-XP JMXJ
60 IR=IR+2
100 CONTINUE
REIURN
END

```

Subroutine TRAN. This subroutine is called by subroutine FUNCT when NT \(>0\). It evaluates the constraint equation violations, the right sides of the velocity and acceleration equations, and the entries in the Jacobian matrix corresponding to the translational joints in the system (refer to Secs. 4.2.1 and 4.3.1). The organization of this subroutine is similar to that of subroutine RVLT.

Subroutine TRAN is as follows:
```

        SUBROUTINE TRAN (TJ,ITJ,RB,Q,QD,FQ,F,NT,NB)
        COMMON /ANALYS/ JACOB,IFNCT
        COMMON /ROMCOL/ IR,IC,M,N
        DIMENSION TJ (NT,7),ITJ (NT,2),RB(NB,2),Q(3,NB),QD(3,NB),
            + DO 100 K=1 FQ(M,N),F(M)
        DO 100 K=1,NT
        I=ITJJ(K,1)
        J=1TJ(K,2)
        XPIMXI = TJ (K,1)*RB(I,2)-TJ (K, 2)*RB(I , 1)
        YPIMYI =TJ (K,1)*RB(I, 1)+TJ (K, 2)*RB(I , 2)
        XPJMXJ = TJ (K,5)*RB(J,2)-TJ (K,6)*RB(J,1)
        YPJMYJ = TJ (K,5)*RB(J, 1) +TJ (K,6)*RB(J,2)
        XPIMXQI=TJ (K,3)*RB(I, 2)-TJ(K,4)*RB(I, 1)
        YPIMYQI= TJ (K,3)*RB(I,1)+TJ(K,4)*RB(I , 2)
        GOTO (10, 20,30) , IFNCT
    C......Constraint equations
10F(IR+1)= XPIMXQI* (Q(2, J )+YPJMYJ-Q(2,I)-YPIMYI )
+ -YPIMYQI*(Q(1,J )+XPJMXJ-Q(1,I )-XPIMXI)
F(IR+2)= Q(3,I)-Q(3,J)-TJ (K,7)
GOTO 40
C.....Right-hand-side of velocity equations
20 F
F(IR+2)=0.0
GOTO 40
C.....Right-hand-side of acceleration equations
30 F(IR+1)=-QD(3,I)*(2.*(XPIMXQI* (QD(1,I)-QD(1, J ))
+ +YPIMYQI * (QD(2,I)-QD(2,J)))
+ +QD(3,I)*(XPIMXQI*(Q (2,I) -Q (2,J))
+ -YPIMYQI*(Q (1,I)-Q (1,J))))
F(IR+2)=0.0
40 IF (JAOOB) 60,60,50
C.....Jacobian matrix nonzero entries
50. ICI= 3*(I-1)
ICJ= 3* (J-1)
FQ(IR+1,ICI +1)= YPIMYQI
FQ(IR+1,ICI +2 ) = -XPIMXQI
FQ(IR+1,ICI+3)=-(Q(1,J)+XPJMXJ -Q(1,I))*XPIMXQI
+
-(Q(2,J)+YPJMYJ -Q(2,I))*YPIMYQI

```
```

                FQ(IR+1,ICJ+1)=-YP IMYQI
                FQ(IR+1,ICJ+2)= XPIMXQI
                FQ(IR+1,ICJ+3)= XP JMXJ *XPIMXQI +YP JMYJ *YPIMYQI
                FQ(IR+2,ICI+3)=1.0
                FQ(IR+2,ICJ+3)=-1.0
    6 0
                IR=IR+2
    100 OONTINUE
RETURN
END

```

Subroutine SMPL. Subroutine SMPL evaluates the constraint equation violations, the right sides of the velocity and acceleration equations, and the entries in the Jacobian matrix for simple constraints (refer to Sec. 4.2.7). It is called by subroutine FUNCT when NG \(>0\) and NS \(>0\). Grounded bodies are treated as bodies with three simple constraints. The organization of this subroutine is similar to that of subroutine RVLT.

Subroutine SMPL is as follows:
SUBROUTINE SMPL (SM, ISM,Q,FQ,F,NS,NB)
OCMMON /ANALYS/ JACOB,IFNCT
COMMON /ROMOOL/ IR,IC,M,N
DIMENS ION SM(NS), ISM(NS,2), Q(3,NB),FQ(M,N),F(M)
DO \(100 \mathrm{~K}=1\), NS
\(\mathrm{I}=\mathrm{I} \operatorname{SM}(\mathrm{K}, 1)\)
\(\mathrm{L}=\mathrm{I} \operatorname{SM}(\mathrm{K}, 2)\)
GOTO (10, 20, 20) , IFNCT
C. ..... Constraint equation
\(10 \quad \mathrm{~F}(\mathrm{IR}+1)=\mathrm{Q}(\mathrm{L}, \mathrm{I})-\mathrm{SM}(\mathrm{K})\) GOTO 40
C......Right-hand-side of velocity/acceleration equation
\(20 \quad F(I R+1)=0.0\)
40 IF (JAOOB) 60,60,50
C......Jacobian matrix nonzero entry
\(50 \quad \mathrm{ICI}=3^{*}(\mathrm{I}-1)\)
\(\mathrm{FQ}(\mathrm{IR}+1, \mathrm{ICI}+\mathrm{L})=1.0\)
\(60 \quad I R=I R+1\)
100 OONTINUE
RETURN
END
Subroutine DRVR. This subroutine is called by subroutine FUNCT when \(\mathrm{ND}>0\). It evaluates constraint equation violations, the right sides of the velocity and acceleration equations, and the entries in the Jacobian matrix for the driver links in the system (refer to Sec. 4.2.8). This is the only function evaluation subroutine in which the time variable T is used explicitly.

Subroutine DRVR is as follows:
```

    SUBROUTINE DRVR (DR,IDR,Q,FQ,F,ND,NB)
    OOMMDN /ANALYS/ JAOOB,IFNCT
    CCMMON /RONOOL/ IR,IC,M,N
    COMMON /TIME / TO,TE,DT,T
    DIMENSION DR(ND, 3),IDR(ND, 2),Q(3,NB),FQ(M,N),F(M)
    DO 100 K=1,ND
        I=IDR(K,1)
        L=IDR(K,2)
        GOTO (10,20,30) , IFNCT
    C......Constraint equation
10 F(IR+1)=Q(L,I)-DR(K,1)-T*(DR(K,2)+T*DR(K,3)/2.0)
GOTO 40

```
```

C......Right-hand-side of velocity equation
GOTO 40
C......Righ-hand-side of acceleration equation
30 F(IR+1)= DR(K,3)
40 IF (JACOB) 60,60,50
C.....Jacobian matrix nonzero entry
50 ICI= 3*(I-1)
FQ(IR+1,ICI+L)=1.0
60 IR=IR+1
100 CONTINUE
RETURN
END

```

\subsection*{5.1.4 Input Prompts}

A list of all the prompts given by the program KAP is given here. The prompts are labeled for easy reference, from (a) through (i). In examples that follow, each prompt is referred to by its corresponding label. The parameter k in the prompts is problemdependent. For example, in a model with two revolute joints, prompt (c) is repeated twice and \(k\) takes on values of 1 and 2.

The prompts given are as follows:
(a) ENTER NB, NR, NT, NG, NS, ND, NP
(b) FOR BODY k ENTER INITIAL EST. ON X, Y, PHI
(C) FOR REV. JOINT NO. k ENTER BODY NOS. I AND J XI-P-I, ETA-P-I, XI-P-J, ETA-P-J
(d) FOR TRAN. JOINT NO. \(k\) ENTER BODY NOS. I AND J XI-P-I, ETA-P-I, XI-Q-I, ETA-Q-I, XI-P-J, ETA-P-J
(e) ENTER BODY NO. FOR NO. k GROUNDED BODY
(f) FOR SIMPLE CONSTRAINT NO. \(k\) ENTER BODY NO. AND 1, 2, OR 3 FOR \(\mathrm{X}, \mathrm{Y}, \mathrm{OR}\) PHI CONSTRAINT DIRECTION
(9) FOR DRIVER NO. k ENTER BODY NO.

1, 2, OR 3 FOR X, Y, OR PHI INITIAL POSITION, VELOCITY AND ACCELERATION
(h) FOR POINT OF INTEREST NO. \(k\) ENTER BODY NO. XI-P AND ETA-P COORDINATES
(i) ENTER STARTING TIME, FINAL TIME, AND TIME INCREMENT

\subsection*{5.2 SIMPLE EXAMPLES}

In the following sections, several simple examples are presented. The steps needed to set up a model for each mechanism are explained. Input data for the kinematic analysis
program are listed for each example. Similar steps can be followed to analyze many other mechanisms using this program.

\subsection*{5.2.1 Four-Bar Linkage}

The four-bar linkage shown in Fig. 5.2(a) is considered for kinematic analysis. This mechanism consists of four bodies, including ground; four revolute joints; one driver; and one point of interest (point \(P\) ). The body numbers and their corresponding coordinate systems and the revolute joint numbers are shown in Fig. 5.2(b).

Initial estimates for the coordinates are as follows:
\[
\begin{array}{lll}
x_{1}=0.0, & y_{1}=0.0, & \phi_{1}=0.0 \\
x_{2}=0.5, & y_{2}=0.8, & \phi_{2}=60.0^{\circ} \\
x_{3}=2.6, & y_{3}=2.6, & \phi_{3}=30.0^{\circ} \\
x_{4}=3.5, & y_{4}=1.8, & \phi_{4}=60.0^{\circ}
\end{array}
\]

The local coordinates for the revolute joints are
\[
\begin{array}{llll}
\xi_{1}^{A}=0.0, & \eta_{1}^{A}=0.0, & \xi_{2}^{A}=-1.0, & \eta_{2}^{A}=0.0 \\
\xi_{2}^{B}=1.0, & \eta_{2}^{B}=0.0, & \xi_{3}^{B}=-2.0, & \eta_{3}^{B}=0.0 \\
\xi_{3}^{C}=2.0, & \eta_{3}^{c}=0.0, & \xi_{4}^{C}=2.0, & \eta_{4}^{C}=0.0 \\
\xi_{4}^{D}=-2.0, & \eta_{4}^{D}=0.0, & \xi_{1}^{D}=2.5, & \eta_{1}^{D}=0.0
\end{array}
\]

Body 2 is the driver link, and its corresponding driving variable is \(\phi_{2}\), with \(\phi_{2}^{0}=1.0472 \mathrm{rad}\left(60^{\circ}\right)\) and a constant angular velocity of \(\omega=6.2832 \mathrm{rad} / \mathrm{s}\). For one complete revolution, one second of simulation time is required. A time increment of 0.025 s results in increments of \(9^{\circ}\) in \(\phi_{2}\).


Figure 5.2 (a) A four-bar linkage, and (b) the corresponding joint numbers and body-fixed coordinate systems.

The point of interest (point \(P\) ) is located on body 3. The local coordinates of this point are
\[
\xi_{3}^{P}=0.5, \quad \eta_{3}^{P}=1.5
\]

When KAP is executed, the following sequence of prompts is given by the program (refer to Sec. 5.1.4), followed by inputs from the user to describe the model:
\begin{tabular}{|c|c|c|}
\hline Prompt & \(\underline{\text { k }}\) & Input \\
\hline (a) & & 4,4,0,1,0,1,1 \\
\hline (b) & 1 & 0.0,0.0,0.0 \\
\hline (b) & 2 & 0.5,0.8,1.047 \\
\hline (b) & 3 & 2.6,2.6,0.5 \\
\hline (b) & 4 & 3.5,1.8,1 \\
\hline (c) & 1 & 1,2,0.0,0.0,-1.0,0.0 \\
\hline (c) & 2 & 2,3,1.0,0.0,-2.0,0.0 \\
\hline (c) & 3 & 3,4,2.0,0.0,2.0,0.0 \\
\hline (c) & 4 & 4,1,-2.0,0.0,2.5,0.0 \\
\hline (e) & 1 & 1 \\
\hline (8) & 1 & 2,3,1.0472,6.2832,0.0 \\
\hline (h) & 1 & 3,0.5,1.5 \\
\hline (i) & & 0.0,1.0,0.025 \\
\hline
\end{tabular}

Note that all of the angles are given in radians. The coordinates are given as estimates, except for \(\phi_{2}\) (since this is the independent variable) and the coordinates of the nonmoving body 1.

The output of this simulation for one second, with increments of 0.025 s , provides 41 time steps (including \(t=0 \mathrm{~s}\) ). A portion of the output for the first two time steps is as follows:

KINEMATIC ANALYSIS

TIME \(=\quad .0000\)
\begin{tabular}{lrrrrrrrrr} 
\\
BODY & X & Y & PHI & XD & YD & PHID & XDD & YDD & PHIDD \\
1 & .000 & .000 & .000 & .000 & .000 & .000 & .000 & .000 & .000 \\
2 & .500 & .866 & 1.047 & -5.441 & 3.142 & 6.283 & -19.739 & -34.190 & .000 \\
3 & 2.824 & 2.553 & .423 & -11.085 & 6.732 & .246 & -52.441 & -39.898 & 15.646 \\
4 3.574 & 1.687 & 1.004 & -5.644 & 3.590 & 3.344 & -32.702 & -5.709 & 12.264 \\
POINTS OF INTEREST & & YD & YD & XDD & YDD & & & \\
NO. & X & Y & XD & & & \\
1 & 2.663 & 4.126 & -11.472 & 6.692 & -77.042 & -42.500 & & &
\end{tabular}


It can be observed that at \(t=0\), the program corrects the initial estimates on the coordinates. The value of \(\phi_{2}\) is kept constant according to the value given with the driving constraint. Also, the coordinates of body 1 remain unchanged, since body 1 is the ground. At each time step, the body numbers and the corresponding coordinates, velocities, and accelerations of their points of interest are reported. Figure 5.3 shows the path taken by point \(P\) for one revolution of the crank.


Figure 5.3 Path covered by point of interest \(P\).

\subsection*{5.2.2 Slider-Crank Mechanism}

The slider-crank mechanism of Sec. 4.4 .1 is now considered for kinematic analysis. The mechanism is modeled in two slightly different forms.

Model 1. The first model considers the mechanism as discussed in Sec. 4.4.1. There are four bodies, three revolute joints, and one translational joint in this model. The flow of input data that describe this model is as follows:
\begin{tabular}{lll} 
Prompt & \(\underline{k}\) & \multicolumn{1}{c}{ Input } \\
\cline { 1 - 1 } (a) & & \(4,3,1,1,0,1,0\) \\
(b) & 1 & \(0.0,0.0,0.0\) \\
(b) & 2 & \(-86.6,50.0,5.76\) \\
(b) & 3 & \(-467.0,40.0,0.2\) \\
(b) & 4 & \(-663.1,0.0,0.0\) \\
(c) & 1 & \(4,3,0.0,0.0,-200.0,0.0\) \\
(c) & 2 & \(3,2,300.0,0.0,-100.0,0.0\)
\end{tabular}
(c) 3
\(2,1,100.0,0.0,0.0,0.0\)
(d) 1
4,1,0.0,0.0,100.0,0.0,0.0,0.0
(e) 1
1
(9) 1
2,3,5.76,-1.2,0.0
(i)
\(0.0,5.3,0.1\)

Figure 5.4 shows acceleration of the slider for one complete revolution of the crank.


Figure 5.4 Acceleration of the slider versus time.

Model 2. The translational joint of the first model is replaced in model 2 by two simple constraints. The slider, body 4 , is constrained in the \(y\) and \(\phi\) directions. The flow of input data that describe this model is as follows:
\begin{tabular}{lll} 
Prompt & k & Input \\
(a) & & \(4,3,0,1,2,1,0\) \\
(b) & 1 & \(0,0,0.0,0.0\)
\end{tabular}
\begin{tabular}{lll} 
(b) & 2 & \(-86.6,50.0,5.76\) \\
(b) & 3 & \(-467.0,40.0,0.2\) \\
(b) & 4 & \(-663.1,0.0,0.0\) \\
c & 1 & \(4,3,0.0,0.0,-200.0,0.0\) \\
c & 2 & \(3,2,300.0,0.0,-100.0,0.0\) \\
(c) & 3 & \(2,1,100.0,0.0,0.0,0.0\) \\
(e & 1 & 1 \\
(f & 1 & 4,2 \\
f & 2 & 4,3 \\
( \()\) & 1 & \(2,3,5.76,-1.2,0.0\) \\
(i) & \(0.0,5.3,0.1\)
\end{tabular}

\subsection*{5.2.3 Quick-Return Mechanism}

Consider the quick-return mechanism of Sec. 4.4.2. Model 1 consists of six bodies, five revolute joints, and two translational joints. The flow of input data for this model is as follows:
\begin{tabular}{|c|c|c|}
\hline Prompt & k & Input \\
\hline (a) & & 6,5,2,1,0,1,0 \\
\hline (b) & 1 & 0.0,-300.0,0.0 \\
\hline (b) & 2 & 70.0,-60.0,1.1 \\
\hline (b) & 3 & 50.0,40.0,0.5 \\
\hline (b) & 4 & 90.0,60.0,1.1 \\
\hline (b) & 5 & 80.0,180.0,6.0 \\
\hline (b) & 6 & 0.0,200.0,0.0 \\
\hline (c) & 1 & 3,4,50.0,0.0,0.0,0.0 \\
\hline (c) & 2 & 2,5,250.0,0.0,60.0,0.0 \\
\hline c & 3 & 1,2,0.0,0.0,-250.0,0.0 \\
\hline c & 4 & 5,6,-60.0,0.0,0.0,0.0 \\
\hline (c) & 5 & 1,3,0.0,300.0,-50.0,0.0 \\
\hline (d) & 1 & 4,2,0.0,0.0,-10.0,0.0,0.0,0.0, \\
\hline (d) & 2 & 6,1,0.0,0.0,-10.0,0.0, 0.0,500.0 \\
\hline e & 1 & 1 \\
\hline (B) & 1 & 3,3,0.52,3.0,0.0 \\
\hline (i) & & 0.0,2.1,0.025 \\
\hline
\end{tabular}

The velocity of body 6 (the slider) is plotted from the output in Fig. 5.5.


Figure 5.5 Slider velocity of a quick-return mechanism versus time.

\subsection*{5.3 PROGRAM EXPANSION}

The computer program that is listed in the opening of Sec. 5.1 and in subsections 5.1.1 through 5.1.3 has been presented in its simplest form. This program may be expanded to accommodate, for example, other forms of input/output or other types of kinematic joints. These improvements to KAP are suggested at the end of this chapter, through a series of exercises (problems). These suggestions are only guidelines - the modifications can also take other forms. Actual use of the program, for modeling and analyzing a wide range of examples, will provide the user with additional ideas for expansion and modification.

\section*{PROBLEMS}

The following problems provide examples that can be simulated by using a kinematic analysis program such as KAP. Many of the problems can be simulated on the existing listed version of KAP. Other problems may require some modifications or extensions to the program. Guidelines
for improving the versatility and increasing the capability of KAP are included for some of those problems. The majority of the problems involve the analysis of four-bar mechanisms. The fourbar mechanism is one of the simplest closed-loop mechanisms, since it has only 1 DOF. A wide variety of motion can be generated by this mechanism. In the following four-bar mechanisms, \(a\), \(b, c\), and \(d\) denote, respectively, the length of the crank, coupler, follower, and frame. The fourbar mechanisms are classified under various types. The following nomenclature is used to describe each type of four-bar mechanism: \({ }^{17}\)
\[
\begin{aligned}
s & =\text { length of shortest link } \\
l & =\text { length of longest link } \\
m, n & =\text { lengths of the other two links }
\end{aligned}
\]
5.1 A double-crank four-bar mechanism is shown in Fig. P.5.1. For this class, \(s+l<m+n\) and the shortest link is the frame. Assume \(a=2.1, b=1.6, c=2.6\), and \(d=0.9\) (any unit). Drive link \(A B\) through \(360^{\circ}\) with a constant angular velocity. Monitor the path, velocity, and acceleration of points \(B\) and \(C\).
5.2 A rocker-crank four-bar mechanism is shown in Fig. P.5.2. For this class, \(s+l<m+n\) and the shortest link is one of the side links. Assume that \(a=0.8, b=2.1, c=1.6\), and \(d=2.5\). Drive link \(A B\) through \(360^{\circ}\) and find the forbidden zones for the rocker \(C D\) and the angle of oscillation \(\alpha\).


Figure P.5.1


Figure P.5.2
5.3 A double-rocker four-bar mechanism is shown in Fig. P.5.3. For this class, \(s+l<m+n\) and the shortest link is the coupler. Assume that \(a=2.1, b=0.8, c=2.5\), and \(d=1.6\). Drive the coupler \(B C\) through one revolution. Determine the forbidden zones and the range of the angles of oscillation for the two rockers \(A B\) and \(C D\).


Figure P.5.3
5.4 Repeat Prob. 5.3 but instead of driving the coupler \(B C\), drive one of the rockers:
(a) Drive the rocker \(A B\) counterclockwise.
(b) Drive the rocker \(A B\) clockwise.
(c) Determine the forbidden zones and the range of the angle of oscillation, and then compare the result with the result of Prob. 5.3.
5.5 Figure P.5.5 shows a change-point four-bar mechanism. For this class, \(s+l=m+n\). Assume that \(a=1.2, b=1.4, c=1.6\), and \(d=1.0\). Drive link \(A B\) through \(720^{\circ}\). Observe that after one revolution of \(A B\), point \(C\) finds a new position at \(C^{\prime}\). After the second revolution of \(A B\), point \(C\) returns to its initial position. Plot the rotational acceleration of \(C D\) versus its rotational angle.
5.6 The Galloway mechanism (Fig. P.5.6) is another example of a change-point four-bar mechanism. In this mechanism, \(a=d, b=c\), and \(a<b\). Assume that \(a=1.0\) and \(b=1.5\). Drive \(A B\) through \(720^{\circ}\) and observe that \(C D\) executes only \(360^{\circ}\).


Figure P.5.5


Figure P.5.6
5.7 Figure P.5.7 shows another example of a change-point four-bar mechanism. Configuration (a) is usually called parallelogram linkage and configuration (c) is called antiparallel linkage. Configuration (b) is the change-point state. A mechanism starting in configuration (a) and going through configuration (b) may end up in configuration (a) or (c). During position analysis, if the mechanism is near the change-point state, the Newton-Raphson iteration may converge to any of the two solutions. Drive \(A B\) through \(360^{\circ}\) and monitor the position of point \(C\). Repeat for different values of time increment DT.


Figure P.5.7

Note: A mechanism at a change-point state becomes kinematically indeterminate. During numerical analysis, if the constraint equations are assembled in the exact change-point state, the Jacobian matrix loses rank; i.e., the number of degrees of freedom increases at the change-point state. Since the Jacobian matrix becomes singular, a solution cannot be obtained at such a state.
5.8 It would be useful to save the input data in a file, in the same sequence in which it is entered interactively. This input data file could then be used if further simulation of the same problem were needed.
(a) Include any required statements in the program to save a copy of the interactively entered data in a disk file. The name of the file should be specified by the user.
(b) The program should be able to accept data either interactively or from a file.
(c) Include a WRITE statement after each READ statement to echo a copy of the input data to the output unit.
5.9 Provide the capability for the program to save the output response at every time step in an output file.
5.10 If the output is written into the terminal, the flow of data may be fast. In subroutine KINEM, after reporting the response and before incrementing T, include a WRITE and a READ statement making the program pause. The write statement should prompt the user to enter a C for continue or a T to terminate.
5.11 Modify the program to accept data on angles either in radians or in degrees. If the data are given in degrees, then the program must convert them to radians. Similarly, the program should provide the output in either radians or degrees.
5.12 For some mechanisms, the user may be interested in a complete kinematic analysis. However, sometimes position analysis or position and velocity analysis may be sufficient. It would be more efficient to avoid unwanted levels of analysis. Modify the program by introducing a flag as follows:

IANAL \(=1:\) Perform position analysis.
IANAL \(=2\) : Perform position and velocity analysis.
IANAL \(=3:\) Perform position, velocity, and acceleration analysis.
5.13 Formulate additional kinematic joints in the program. Follow the form of the subroutines for the revolute joint that already exists in KAP. Include the following joints:
(a) Revolute-revolute joint (Eq. 4.14.)
(b) Revolute-translational joint (Eq. 4.16)
(c) Spur gears (Eq. 4.18)
5.14 Include other types of driver constraints in the program. Some useful constraints are:
(a) \(\phi_{i}-\phi_{j}-c_{1}(t)=0\)
(b) \(x_{i}-x_{j}-c_{2}(t)=0\)
(c) \(y_{i}-y_{j}-c_{3}(t)=0\)
(d) Eq. 4.39
(e) Eq. 4.40

The time-dependent functions can be provided either as a table of data or as a closed-form function such as a sine or a cosine function.
5.15 In position analysis, the Newton-Raphson algorithm uses values of the coordinates from the previous time step as an estimate on \(\mathbf{q}^{i}\) to start the iteration. These estimates can be improved as follows:
\[
\mathbf{q}^{i} \simeq \mathbf{q}^{i-1}+\Delta t \dot{\mathbf{q}}^{i-1}+\frac{1}{2} \Delta t^{2} \ddot{\mathbf{q}}^{i-1}
\]

Include this modification in subroutine KINEM to improve the efficiency of the program.
5.16 Employ a Gaussian elimination (or L-U factorization) algorithm with full or partial (row interchange) pivoting in KAP to identify any redundant constraint equation. This process can be performed once on the Jacobian matrix before the start of the kinematic analysis.
5.17 Simple but general algorithms and program listings for spline functions can be found in most numerical computation textbooks. Include such a program in KAP. These subroutines may be called for interpolation when a curve is defined in discrete form.
5.18 When nonstandard constraint equations are to be included in KAP to model a particular mechanism, it is not efficient to modify and compile the entire program. In this case a usersupplied subroutine can be written to implement the constraints. In subroutine FUNCT, include a call to a subroutine USRCON before the return statement, as follows:

CALL USRCON (A, IA, MAXA, MAXIA)
Then, include a dummy user-supplied subroutine:

> SUBROUTINE USRCON (A, IA, MAX, MAXIA)
> C.....Include all of the common blocks DIMENSION A(MAXA), IA(MAXIA)
> C.....Insert new constraints, define the Jacobian, etc.
> RETURN
> END

This subroutine will be called by subroutine FUNCT at every time step. If there is no nonstandard function in the model, this subroutine will not affect the simulation. However, if equations are inserted in this subroutine and it is linked to KAP, then each time this subroutine is called, the nonstandard constraints will be included in the model.
5.19 When constraint equations are formulated in terms of a set of Cartesian coordinates, the resultant Jacobian is a sparse matrix; i.e., most of the elements of the matrix are zero. There are matrix factorization algorithms designed specifically for sparse matrices: \({ }^{\dagger}\) These algorithms are much more efficient than the standard full-matrix algorithms, such as subroutine LU used in KAP. If you have access to such sparse-matrix algorithms, employ them in KAP to improve the efficiency of the program.
5.20 The linkage shown in Fig. P.5.20 is used to advance a film strip intermittently by the motion of point \(P\), which periodically engages and disengages from the sprocket holes in the filmstrip as crank \(C D\) rotates clockwise at \(1800 \mathrm{rpm} .{ }^{17}\)
(a) Plot the path of point \(P\).
(b) How far apart should the sprocket holes be placed on the film strip?
(c) What is the mean speed of the film strip?


Figure P.5.20
5.21 Assign values to the lengths of the links of the four-bar mechanism shown in Fig. P.5.21. Rotate the crank and monitor the path of several points on the coupler, such as \(P_{1}, P_{2}, \ldots\).
\({ }^{\dagger}\) The HARWELL Subroutine Library offers FORTRAN subroutines for sparse-matrix factorization. For more detail, write to HARWELL Subroutine Library, United Kingdom Atomic Energy Authority, Computer Science and System Division, AERE Harwell, Oxfordshire.


Figure P.5.21


Figure P.5.22
5.22 The crank-rocker four-bar mechanism shown in Fig. P.5.22 is a web cutter. \({ }^{17}\) Link \(A B\) rotates with a constant angular velocity of 100 rpm .
(a) Determine the angle of the crank at the time of cutting.
(b) What should be the velocity of the web at the time of cutting?
5.23 Figure P.5.23 shows an elliptic trammel. Any point \(P\) on the link \(K L\) traces out an ellipse. Define several points \(P_{1}, P_{2}, \ldots\) as special points of interest on link \(K L\), and then rotate this link through \(360^{\circ}\) and plot the paths of these points.
5.24 A dough-kneader mechanism is shown in Fig. P.5.24. \({ }^{\ddagger}\) The crank \(A B\) rotates through \(360^{\circ}\). Note that in order to model this mechanism two revolute joints are needed at \(B\). Plot the path of point \(P\). Assume \(A B=6, B C=E F=13, B E=C F=6, D C=15, A D=18\), \(B P=26\).


Figure P.5. 23


Figure P.5.24
\({ }^{\ddagger}\) These mechanisms have been adopted for kinematic simulation from the following reference: Artobolevsky, I. I., Mechanisms in Modern Engineering Design, Vol. I, MIR Publishers, Moscow, 1975. Many other simple to complicated mechanisms can be found in this and subsequent volumes of the same reference.
5.25 Figure P.5.25 shows a translator mechanism for a drafting table. \({ }^{\ddagger}\) The lengths of the links comply with the conditions \(A B=D C, A D=B C, F G=E H\), and \(E F=H G\). Verify that the system has 2 DOF. Also show that parallel lines can be drawn by link \(p\). This can be done in several ways; for example, keep the angle between \(E F\) and \(E H\) constant (include the constraint \(\phi_{i}-\phi_{j}-c=0\) ), and then rotate link \(A B\).


Figure P.5. 25
5.26 Repeat Prob. 5.25 and assume that ring \(q\) consists of two rings \(q_{1}\) and \(q_{2}\) as shown in Fig. P.5.26. The two rings are connected by revolute joint \(I\). Angle \(\alpha\) between the two rings can be varied in order to change the angle of link \(p\). While keeping \(\alpha\) constant during any given simulation, perform several simulations for different values of \(\alpha\).


Figure P.5. 26
5.27 Figure P.5.27 shows two pantograph mechanisms where \(T\) is the tracing point and \(D\) is the drawing point. Since \(O, D\), and \(T\) are on a straight line, the paths traced by \(T\) and \(D\) are


Figure P.5.27
geometrically similar. In order to set up a simulation model, the following tasks must be performed:
(a) Carry out Prob. 5.12 for position analysis only.
(b) Carry out Prob. 5.14(d) and (e).

Select an outline to be traced; for example, a known geometrical shape. Discretize the outline into several points, and then find the \(x\) and \(y\) coordinates of these points with respect to the global \(x y\) coordinate system. These coordinates provide driver constraints on \(x^{T}\) and \(y^{T}\).
5.28 A pantograph mechanism can be modified easily to draw nonsimilar or warped traces. For example if point \(D\) is moved to position \(D_{1}\) (see Fig. P.5.28), the shape of the drawn path will be different from the traced path. Similarly, if \(D\) is moved to position \(D_{2}\), then the drawn path will be quite different from the traced path. By moving \(D\) to different positions, completely unfamiliar shapes can be generated. If the result is displayed on a graphics terminal, interesting paths may be observed. Make this problem a computer graphics oriented project.
5.29 The mechanism shown in Fig. P.5.29 is known as a Kostitsyn approximate straight-line system, where \(b=4 a, c=5 a\), and \(A D=b .^{\ddagger}\) When link \(B F\) rotates, point \(E\) describes a path


Figure P.5.28


Figure P.5.29

\footnotetext{
\({ }^{\ddagger}\) See footnote to Prob. 5.24.
}
of which a portion is an approximate straight line. Verify this and note that the linkage \(A B C D\) is a change-point four-bar mechanism.
5.30 Figure P.5.30 shows the Peaucellier-Lipkin exact straight-line mechanism. \({ }^{\ddagger}\) Choose \(e=1.2, f=3.0\), and \(g=1.4\). Rotate link \(R E\) and observe that point \(S\) describes a straight line. Determine the length of the straight line drawn by point \(S\).
5.31 The Peaucellier-Lipkin circle inversion mechanism is shown in Fig. P.5.31. \({ }^{\ddagger}\) Assume that \(a=2.0, b=1.6\), and \(c=1.4\), where \(F E<c\). Rotate link \(E R\) and observe that point \(S\) describes a circle \(c\) centered at point \(O\). Verify that \(F O=F E\left(a^{2}-b^{2}\right) /\left(c^{2}-F E^{2}\right)\) and that the radius of the circle is \(r=c(F O / F E)\) (for simulation assume \(F E=0.8\) ).


Figure P.5.30


Figure P.5.31
5.32 The Hart mechanism for drawing ellipses is shown in Figure P.5.32. The lengths of the links comply with the conditions \(n=2 m\) and \(e=\sqrt{2} \bar{m}\). Verify that when link \(M R\) rotates, point \(S\) describes a straight line and any intermediatc point \(P\) on link \(R S\) describes an ellipse.
5.33 The Chebyshev six-bar reversing mechanism is shown in Fig. P.5.33. Verify that one revoIution of crank \(E F\) corresponds to one revolution of link \(J K\) in the opposite direction. Assume \(E F=1.08, F G=G H=G I=2, I J=J K=1.14, H K=2.78\), and \(E H=2.66\).
5.34 The mechanism shown in Fig. P.5.34 is a simplified version of a thread-and-needlc guiding system of a sewing machine. \({ }^{\ddagger}\) Rotate crank \(c\) and plot the path of point \(H\). Assume \(A B=10\), \(B E=9, D E=D F=6.5, E F=5, F G=17.5, a=f=10, b=7, d=1\), and \(e=4\).
5.35 For the mechanism shown in Fig. P.5.35, verify that when the crank rotates, the output link dwells. Plot the angular velocity of the output link versus the crank angle. Assume that \(O A=0.8, A B=3.6, B C=2.3, C H=D E=E F=3.5, O H=1.5, A D=C G=4\), \(B D=2\), and \(F G=1\).
5.36 For the mechanism shown in Fig. P.5.36, find that portion of the crank orientation for which the output link dwells. Plot the path of the motion of the slider. Assume \(O A=1, A B=4\), \(B C=O E=3.5, C E=5\), and \(B D=O F=2.5\).
5.37 A Geneva-wheel mechanism is shown in Fig. P.5.37. For a specific design, the following information is known: the radius of the crank \(\rho\); the number of slots \(n\), the half angle between two adjacent slots \(\beta=\pi / n\); the center distance \(d=\rho / \sin \beta\). It is also known that the crank axis is perpendicular to the slot axis at the moment of engagement or disengage-
\({ }^{*}\) See footnote to Prob. 5.24.


Figure P.5.32


Figure P.5.34


Figure P.5.33


Figure P.5.35


Figure P.5.36


Figure P.5.37
ment. In order to model this mechanism, two types of constraints must be employed interchangeably:
(a) When the pin is disengaged from the wheel, the wheel does not rotate.
(b) When the pin is engaged with a slot, the pin-slot combination can be modeled as a revo-lute-translational joint. It is clear that the axis of the translational joint will be different from one revolution to the next.
Include the constraint equations and the proper logic for switching between these equations in KAP.
5.38 A walking robot mechanism is shown in Fig. P.5.38(a). \({ }^{\ddagger}\) The mechanism consists of two identical \(\Pi\)-shaped legs, each having two feet. When one pair of feet is on the ground, the


\footnotetext{
\({ }^{\dagger}\) See footnote to Prob. 5.24.
}


Figure P.5.38
other pair moves forward by the rotation of two parallel cranks \(A_{1} C^{\prime} A_{2}\) and \(A_{3} C_{1}^{\prime} A_{4}\). The lengths of the links comply with the conditions:
\[
\begin{aligned}
A_{i} B_{i} & =B_{i} D_{i}=1 \quad i=1,2,3,4 \\
B_{1} C & =B_{2} C=B_{3} C_{1}=B_{4} C_{1}=1 \\
A_{1} C^{\prime} & =A_{2} C^{\prime}=A_{3} C_{1}^{\prime}=A_{4} C_{1}^{\prime}=0.355 \\
A_{2} A_{4} & =A_{1} A_{3}=C^{\prime} C_{1}^{\prime}=0.634 \\
C C^{\prime} & =C_{1} C_{1}^{\prime}=0.785
\end{aligned}
\]

A kinematic analysis of this system requires the following:
(a) If all of the bodies and revolute joints that are shown in the figure are included in the model, redundant constraint equations will result. Therefore, the redundant equations must be eliminated.
(b) During one-half of the crank revolution, one pair of feet must be constrained to the ground, and during the other half of the crank revolution, the other pair of feet must be constrained to the ground. This process must be performed interchangeably in every revolution of the crank.
If the result of the simulation is viewed on a graphics terminal, a sequence of graphs, as shown in Fig. P.5.38(b), will be observed.
5.39 Some relatively complex mechanisms can become interesting simulation projects. For the more complex systems, graphics output can be a valuable analysis and design tool. Examples of such mechanisms can be found in the following:
(a) Recliner chairs
(b) Sofa-sleepers
(c) Crank mechanisms for automobile windows
(d) Convertible tops in some automobiles

Find data on these mechanisms and try to simulate their kinematics.

\section*{6}

\section*{Euler Parameters}

In this chapter and the next, spatial kinematics is discussed. Although the analytical procedure in spatial kinematics is the same as in the planar case, spatial kinematic analysis requires much more powerful mathematical techniques than planar kinematics, particularly for describing the angular orientation of a body in a global coordinate system. Therefore, this chapter is mostly devoted to developing the techniques involved in describing the angular orientation of bodies in space, without being concerned with the translation.

As its title suggests, this chapter concentrates on a set of orientational coordinates known as Euler parameters, \({ }^{\dagger}\) which are free of some of the deficiencies of other commonly used angular coordinates, such as Euler angles. At the beginning, it may appear that Euler parameters have no physical significance and that they are just mathematical tools. However, when the subject is thoroughly understood, their physical relevance will also become evident. Furthermore, for large-scale computer programs that treat the angular orientation of bodies, either rigid or deformable, the use of Euler parameters may drastically simplify the mathematical formulations.

\subsection*{6.1 COORDINATES OF A BODY}

An unconstrained body in space requires six independent coordinates to determine its configuration - three coordinates specify translation and three specify rotation. The six coordinates define the location of a Cartesian coordinate system that is fixed in the body (i.e., the location of its local, or body-fixed, coordinates) relative to the global (refer-
\({ }^{\dagger}\) Euler parameters are a normalized form of parameters known as quaternions. \({ }^{19}\)
ence or inertial) coordinate axes. Since all points in the body may be located relative to this body-fixed coordinate system, the global locations of all points in the body can thus be determined from the six coordinates. The coordinates of the origin of the body-fixed axes are the translational coordinates. Rotational coordinates are then needed to define the orientation of the local axes relative to the global coordinate axes. Throughout this text, the body-fixed axes will be denoted as \(\xi \eta \zeta\) axes and the global axes will be denoted as \(x y z\) axes.

Figure 6.1(a) shows how the configuration of the \(\xi \eta \zeta\) axes with respect to the \(x y z\) axes can be considered a translation ( \(x y z\) to \(x^{\prime} y^{\prime} z^{\prime}\) ) and a rotation ( \(x^{\prime} y^{\prime} z^{\prime}\) to \(\xi \eta \zeta\) ). However, for purposes of finding only the angular orientation of the \(\xi \eta \zeta\) axes relative to the \(x y z\) system, the origins of the two systems may be considered to coincide, as shown in Fig. 6.1(b).


Figure 6.1 Configuration of Cartesian coordinate systems: (a) translation and rotation; (b) rotation only.
A vector \(\vec{s}\) from the origin to a point \(P\), as shown in Fig. 6.2, can be expanded in either of the two coordinate systems. If unit vectors \(\vec{u}_{(\xi)}, \vec{u}_{(\eta)}\), and \(\vec{u}_{(\xi)}\) are defined along the \(\xi \eta \zeta\) axes and \(\vec{u}_{(x)}, \vec{u}_{(y)}\), and \(\vec{u}_{(z)}\) are defined along the \(x y z\) axes, then:
\[
\begin{equation*}
\vec{s}=s_{(x)} \vec{u}_{(x)}+s_{(y)} \vec{u}_{(y)}+s_{(z)} \overrightarrow{\vec{u}}_{(z)} \tag{6.1}
\end{equation*}
\]
or
\[
\begin{equation*}
\vec{s}=s_{(\xi)} \vec{u}_{(\xi)}+s_{(\eta)} \vec{u}_{(\eta)}+s_{(\xi)} \vec{u}_{(\xi)} \tag{6.2}
\end{equation*}
\]


Figure 6.2 Unit vectors along the axes of the local and global coordinate systems.
where
\[
s_{(x)}=\vec{s} \cdot \vec{u}_{(x)}, \quad s_{(y)}=\vec{s} \cdot \vec{u}_{(y)}, \quad s_{(z)}=\vec{s} \cdot \vec{u}_{(z)}
\]
and
\[
s_{(\xi)}=\vec{s} \cdot \vec{u}_{(\xi)}, \quad s_{(\eta)}=\vec{s} \cdot \vec{u}_{(\eta)}, \quad s_{(\xi)}=\vec{s} \cdot \vec{u}_{(\xi)}
\]

The component vectors that define \(\vec{s}\) in the two coordinate systems are
\[
\mathbf{s}=\left[s_{(x)}, s_{(y)}, s_{(z)}\right]^{T}
\]
in the \(x y z\) system and
\[
\mathbf{s}^{\prime}=\left[s_{(\xi)}, s_{(\eta)}, s_{(\zeta)}\right]^{T}
\]
in the \(\xi \eta \zeta\) system. It is clear that there is a relation between \(s\) and \(s^{\prime}\), since they are uniquely defined by the same vector \(\vec{s}\). To find this relation, the \(\vec{u}_{(\xi)}, \vec{u}_{(\eta)}\), and \(\vec{u}_{(\xi)}\) unit vectors are defined in terms of the \(\vec{u}_{(x)}, \vec{u}_{(y)}\), and \((\vec{u})_{(z)}\) unit vectors as follows:
\[
\begin{align*}
& \vec{u}_{(\xi)}=a_{11} \vec{u}_{(x)}+a_{21} \vec{u}_{(y)}+a_{31} \vec{u}_{(z)} \\
& \vec{u}_{(\eta)}=a_{12} \vec{u}_{(x)}+a_{22} \vec{u}_{(y)}+a_{32} \vec{u}_{(z)}  \tag{6.3}\\
& \vec{u}_{(y)}=a_{13} \vec{u}_{(x)}+a_{23} \vec{u}_{(y)}+a_{33} \vec{u}_{(z)}
\end{align*}
\]
where \(a_{i j}, i, j=1,2,3\), are the direction cosines that can be expressed as
\[
\begin{align*}
& a_{11}=\vec{u}_{(\xi)} \cdot \vec{u}_{(x)}=\cos \left(\vec{u}_{(\xi)}, \vec{u}_{(x)}\right) \\
& a_{21}=\vec{u}_{(\xi)} \cdot \vec{u}_{(y)}=\cos \left(\vec{u}_{(\xi)}, \vec{u}_{(y)}\right) \\
& a_{31}=\vec{u}_{(\xi)} \cdot \vec{u}_{(z)}=\cos \left(\vec{u}_{(\xi)}, \vec{u}_{(z)}\right) \\
& a_{12}=\vec{u}_{(\eta)} \cdot \vec{u}_{(x)}=\cos \left(\vec{u}_{(\eta)}, \vec{u}_{(x)}\right) \\
& a_{22}=\vec{u}_{(\eta)} \cdot \vec{u}_{(y)}=\cos \left(\vec{u}_{(\eta)}, \vec{u}_{(y)}\right)  \tag{6.4}\\
& a_{32}=\vec{u}_{(\eta)} \cdot \vec{u}_{(z)}=\cos \left(\vec{u}_{(\eta)}, \vec{u}_{(z)}\right) \\
& a_{13}=\vec{u}_{(\xi)} \cdot \vec{u}_{(x)}=\cos \left(\vec{u}_{(\xi)}, \vec{u}_{(x)}\right) \\
& a_{23}=\vec{u}_{(\xi)} \cdot \vec{u}_{(y)}=\cos \left(\vec{u}_{(\xi)}, \vec{u}_{(y)}\right) \\
& a_{33}=\vec{u}_{(\xi)} \cdot \vec{u}_{(z)}=\cos \left(\vec{u}_{(\xi)}, \vec{u}_{(z)}\right)
\end{align*}
\]

Substituting from Eq. 6.3 into Eq. 6.2 yields
\[
\begin{align*}
\vec{s}= & \left(a_{11} s_{(\xi)}+a_{12} s_{(\eta)}+a_{13} s_{(\xi)}\right) \vec{u}_{(x)} \\
& +\left(a_{21} s_{(\xi)}+a_{22} s_{(\eta)}+a_{23} s_{(\xi)}\right) \vec{u}_{(y)} \\
& +\left(a_{31} s_{(\xi)}+a_{32} s_{(\eta)}+a_{33} s_{(\xi)}\right) \vec{u}_{(z)} \tag{6.5}
\end{align*}
\]

By equating the right sides of Eqs. 6.1 and 6.5 , it is found that
\[
\begin{aligned}
s_{(x)} & =a_{11} s_{(\xi)}+a_{12} s_{(\eta)}+a_{13} s_{(\xi)} \\
s_{(y)} & =a_{21} s_{(\xi)}+a_{22} s_{(\eta)}+a_{23} s_{(\xi)} \\
s_{(z)} & =a_{31} s_{(\xi)}+a_{32} s_{(\eta)}+a_{33} s_{(\xi)}
\end{aligned}
\]
or, in matrix form,
\[
\begin{equation*}
\mathbf{s}=\mathbf{A} \mathbf{s}^{\prime} \tag{6.6}
\end{equation*}
\]
where the matrix \(\mathbf{A}\) of direction cosines is
\[
\mathbf{A}=\left[\begin{array}{lll}
a_{11} & a_{12} & a_{13}  \tag{6.7}\\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}\right]
\]

The matrix A has a special property. If the \(x y z\) components of unit vectors \(\vec{u}_{(\xi)}\), \(\vec{u}_{(\eta)}\), and \(\vec{u}_{(6)}\) are denoted by \(\mathbf{u}_{(\xi)}, \mathbf{u}_{(\eta)}\), and \(\mathbf{u}_{(\xi)}\) and the \(x y z\) components of vectors \(\vec{u}_{(x)}\), \(\vec{u}_{(y)}\), and \(\vec{u}_{(z)}\) are denoted by \(\mathbf{u}_{(x)}, \mathbf{u}_{(y)}\), and \(\mathbf{u}_{(z)}\), it is clear that
\[
\mathbf{u}_{(x)}=\left[\begin{array}{l}
1  \tag{6.8}\\
0 \\
0
\end{array}\right], \quad \mathbf{u}_{(y)}=\left[\begin{array}{l}
0 \\
1 \\
0
\end{array}\right], \quad \mathbf{u}_{(z)}=\left[\begin{array}{l}
0 \\
0 \\
1
\end{array}\right]
\]

Equation 6.4 indicates that \(a_{11}\) is the \(x\) component of \(\mathbf{u}_{(\xi)}, a_{21}\) is the \(y\) component of \(\mathbf{u}_{(\xi)}\), and so forth. Therefore,
\[
\mathbf{u}_{(\xi)}=\left[\begin{array}{l}
a_{11} \\
a_{21} \\
a_{31}
\end{array}\right] \quad \mathbf{u}_{(\eta)}=\left[\begin{array}{l}
a_{12} \\
a_{22} \\
a_{32}
\end{array}\right] \quad \mathbf{u}_{(6)}=\left[\begin{array}{l}
a_{13} \\
a_{23} \\
a_{33}
\end{array}\right]
\]
and the matrix \(\mathbf{A}\) can be written as follows:
\[
\begin{equation*}
\mathbf{A}=\left[\mathbf{u}_{(\xi)}, \mathbf{u}_{(\eta)}, \mathbf{u}_{(\xi)}\right] \tag{6.9}
\end{equation*}
\]

Since the unit vectors \(\mathbf{u}_{(\xi)}, \mathbf{u}_{(\eta)}\), and \(\mathbf{u}_{(\xi)}\) are orthogonal,
\[
\begin{equation*}
\mathbf{A}^{T} \mathbf{A}=\mathbf{I} \tag{6.10}
\end{equation*}
\]

Thus, \(\mathbf{A}^{r}=\mathbf{A}^{-1}\), and the matrix \(\mathbf{A}\) is also orthogonal. This special property permits an easy inversion of Eq. 6.6, to obtain
\[
\begin{equation*}
\mathbf{s}^{\prime}=\mathbf{A}^{T} \mathbf{s} \tag{6.11}
\end{equation*}
\]

The nine direction cosines in \(\mathbf{A}\) define the orientation of the \(\xi \eta \zeta\) axes relative to the \(x y z\) axes, but they are not independent. Substituting Eq. 6.4 into Eq. 6.10 provides six equations (three of the nine equations are repeated twice) among the nine direction cosines. Thus, only three direction cosines are independent. While the nine direction cosines, subject to six constraints, could be adopted as rotational coordinates, this is neither practical nor convenient. Thus, other orientation coordinates are sought.

When the origins of the \(x y z\) and \(\xi \eta \zeta\) coordinate systems do not coincide, as is the case in Fig. 6.1(a), the foregoing analysis is applied between the \(x^{\prime} y^{\prime} z^{\prime}\) and \(\xi \eta \zeta\) systems. If the component vector \(\mathbf{s}^{\prime P}\) locates a point \(P\) in the \(\xi \eta \zeta\) coordinate system, as it does in Fig. 6.3, then in the \(x^{\prime} y^{\prime} z^{\prime}\) system this vector is just \(\mathbf{A s}^{\prime P}\), and in global \(x y z\) coordinates,
\[
\begin{equation*}
\mathbf{r}^{P}=\mathbf{r}+\mathbf{A s}^{\prime P} \tag{6.12}
\end{equation*}
\]
where \(\mathbf{r}\) is the vector from the origin of the \(x y z\) system to the origin of the \(\xi \eta \zeta\) system.


Figure 6.3 Translation and rotation of a body in three-dimensional space.

\subsection*{6.1.1 Euler's Theorem on the Motion of a Body}

At any instant in time, the orientation of a body can be specified by a transformation matrix, the elements of which may be expressed in terms of suitable sets of coordinates. As time progresses, the orientation of the body will change. Hence the transformation matrix will be a function of time. Since the motion of the body is continuous, the transformation matrix must be a continuous function of time. The transformation may thus be said to evolve continuously. With this method of describing the motion, Euler's theorem \({ }^{4}\) can be stated as follows:

Euler's theorem: The general displacement of a body with one point fixed is a rotation about some axis.

The theorem indicates that the orientation of the body-fixed axes at any time \(t\) can be obtained by an imaginary rotation of these axes from an orientation coincident with the global axes. This imaginary axis of rotation is not the so-called instantaneous axis of rotation of the body - in this text we will call it the orientational axis of rotation. It is important to note that any vector lying along the orientational axis of rotation is left unaffected by this imaginary rotation - it must have the same components in both the reference and the body-fixed coordinates. The other necessary condition for rotation, that the magnitude of vectors undergoing the imaginary rotation be unaffected, is automatically satisfied.

An immediate corollary of Euler's theorem, known as Chasles's theorem, \({ }^{4}\) is stated as follows:

Chasles's theorem: The most general displacement of a body is a translation plus a rotation.
This theorem simply states that removing the constraint of motion with one point fixed introduces three translatory degrees of freedom for the origin of the body-fixed axes.

\subsection*{6.1.2 Active and Passive Points of View}

A change in the angular orientation may be interpreted from an active point of view or from a passive point of view. Symbolically, a transformation may be written as
\[
\begin{equation*}
\mathbf{s}=\mathbf{A} \mathbf{s}^{\prime} \tag{6.13}
\end{equation*}
\]

According to the active point of view, the operator \(\mathbf{A}\) relates two vectors of equal length, \(\vec{s}\) and \(\vec{s}^{\prime}\), expressed in terms of the global coordinate system only, as shown in Fig. 6.4(a). On the other hand, the passive point of view describes only a single vector \(\vec{s}\) and introduces a new local coordinate system to account for the change in orientation, as shown in Fig. 6.4(b). In this case the operator \(\mathbf{A}\) relates the global components of the vector \(\vec{s}\) to its local components; i.e., \(\mathbf{s}\) and \(\mathbf{s}^{\prime}\). Whereas one rotates the coordinate system counterclockwise (positive sense of rotation), according to the passive point of view, one rotates vector \(\vec{s}\) clockwise by the same angle from the active point of view, to obtain a new vector \(\vec{s}^{\prime}\) in the same coordinate system. The algebra is the same when either of the two points of view is followed.

In the following sections, rotational coordinates known as Euler parameters are discussed. The Euler parameter set employs the active point of view for determination of the transformation matrix \(\mathbf{A}\). A discussion of two other sets of commonly used coordinates, known as Euler angles and Bryant angles, can be found in Appendix A. The pas-


Figure 6.4 Coordinate system rotation: (a) active point of view; (b) passive point of view.
sive point of view is employed to determine the transformation matrix in terms of Euler and Bryant angles.

\subsection*{6.1.3 Euler Parameters}

Euler's theorem states that a coordinate transformation can be accomplished by a single rotation about a suitable axis. It is natural, therefore, to seek a representation of the coordinate transformation in terms of parameters of this rotation, namely, the angle of rotation and the direction cosines of the orientational axis of rotation. \({ }^{4}\)

In Fig. 6.5 the initial position of the vector \(\vec{s}\) is denoted by \(\overrightarrow{O P}\) and the final position \(\vec{s}^{\prime}\) is denoted by \(\overrightarrow{O P} \vec{P}^{\prime}\). The unit vector along the orientational axis of rotation is denoted by \(\vec{u}\). Vector \(\vec{s}\) can be expressed as the sum of three vectors:
\[
\begin{equation*}
\vec{s}=\overrightarrow{O N}+\overrightarrow{N Q}+\overrightarrow{Q P} \tag{a}
\end{equation*}
\]


Figure 6.5 Vector diagram for derivation of rotation formula.

The direct distance between points \(O\) and \(N\) is \(\vec{u} \cdot \vec{s}^{\prime}\), so vector \(\overrightarrow{O N}\) can be written as follows:
\[
\begin{equation*}
\overrightarrow{O N}=\vec{u}\left(\vec{u} \cdot \vec{s}^{\prime}\right) \tag{b}
\end{equation*}
\]

Vector \(\overrightarrow{N P}^{\prime}\) can also be described as follows:
\[
\overrightarrow{N P^{\prime}}=\vec{s}^{\prime}-\overrightarrow{O N}=\vec{s}^{\prime}-\vec{u}\left(\vec{u} \cdot \vec{s}^{\prime}\right)
\]

Hence,
\[
\begin{equation*}
\overrightarrow{N Q}=\left[\vec{s}^{\prime}-\vec{u}\left(\vec{u} \cdot \vec{s}^{\prime}\right)\right] \cos \phi \tag{c}
\end{equation*}
\]

The magnitude of vector \(\overrightarrow{N P}^{\prime}\) is the same as that of vectors \(\overrightarrow{N P}\) and \(\vec{u} \times \vec{s}^{\prime}\). Therefore, vector \(\overrightarrow{Q P}\) may be expressed as
\[
\begin{equation*}
\overrightarrow{Q P}=\vec{u} \times \vec{s}^{\prime} \sin \phi \tag{d}
\end{equation*}
\]

Substitution of Eqs. \(b, c\), and \(d\) into Eq. \(a\), together with a slight rearrangement of terms, leads to the rotation formula : \(^{4}\)
\[
\begin{equation*}
\vec{s}=\vec{s}^{\prime} \cos \phi+\vec{u}\left(\vec{u} \cdot \vec{s}^{\prime}\right)(1-\cos \phi)+\vec{u} \times \vec{s}^{\prime} \sin \phi \tag{6.14}
\end{equation*}
\]

By means of the standard trigonometric relationships
\[
\begin{aligned}
\cos \phi & =2 \cos ^{2} \frac{\phi}{2}-1 \\
\sin \phi & =2 \sin \frac{\phi}{2} \cos \frac{\phi}{2} \\
1-\cos \phi & =2 \sin ^{2} \frac{\phi}{2}
\end{aligned}
\]
and the new quantities
\[
\begin{align*}
e_{0} & =\cos \frac{\phi}{2} \\
\vec{e} & =\vec{u} \sin \frac{\phi}{2} \tag{6.15}
\end{align*}
\]
the rotation formula of Eq. 6.14 can be put in a more useful form:
\[
\begin{equation*}
\vec{s}=\left(2 e_{0}^{2}-1\right) \vec{s}^{\prime}+2 \vec{e}\left(\vec{e} \cdot \vec{s}^{\prime}\right)+2 e_{0} \vec{e} \times \vec{s}^{\prime} \tag{6.16}
\end{equation*}
\]

Algebraic representation of Eq. 6.16, using the component form \(\mathbf{e}=\left[e_{1}, e_{2}, e_{3}\right]^{T}\) of \(\vec{e}\), yields
\[
\mathbf{s}=\left(2 e_{0}^{2}-1\right) \mathbf{s}^{\prime}+2 \mathbf{e}\left(\mathbf{e}^{T} \mathbf{s}^{\prime}\right)+2 e_{0}\left(\widetilde{\mathbf{e}^{\prime}}\right)
\]
or
\[
\begin{equation*}
\mathbf{s}=\left[\left(2 e_{0}^{2}-1\right) \mathbf{I}+2 \mathbf{e} \mathbf{e}^{T}+2 e_{0} \tilde{\mathbf{e}}\right] \mathbf{s}^{\prime} \tag{6.17}
\end{equation*}
\]
where \(\mathbf{I}\) is the \(3 \times 3\) identity matrix and, by the definition in Eq. 2.43,
\[
\tilde{\mathbf{e}}=\left[\begin{array}{rrr}
0 & -e_{3} & e_{2} \\
e_{3} & 0 & -e_{1} \\
-e_{2} & e_{1} & 0
\end{array}\right]
\]

The term in brackets in Eq. 6.17 is thus the transformation matrix of Eq. 6.13; i.e.,
\[
\begin{equation*}
\mathbf{A}=\left(2 e_{0}^{2}-1\right) \mathbf{I}+2\left(\mathbf{e}^{T}+e_{0} \tilde{\mathbf{e}}\right) \tag{6.18}
\end{equation*}
\]

More explicitly,
\[
\mathbf{A}=2\left[\begin{array}{lll}
e_{0}^{2}+e_{1}^{2}-\frac{1}{2} & e_{1} e_{2}-e_{0} e_{3} & e_{1} e_{3}+e_{0} e_{2}  \tag{6.19}\\
e_{1} e_{2}+e_{0} e_{3} & e_{0}^{2}+e_{2}^{2}-\frac{1}{2} & e_{2} e_{3}-e_{0} e_{1} \\
e_{1} e_{3}-e_{0} e_{2} & e_{2} e_{3}+e_{0} e_{1} & e_{0}^{2}+e_{3}^{2}-\frac{1}{2}
\end{array}\right]
\]

Taking the transpose of both sides of Eq. 6.18 yields
\[
\begin{equation*}
\mathbf{A}^{T}=\left(2 e_{0}^{2}-1\right) \mathbf{I}+2\left(\mathbf{e} \mathbf{e}^{T}-e_{0} \tilde{\mathbf{e}}\right) \tag{6.20}
\end{equation*}
\]

The four quantities \(e_{0}, e_{1}, e_{2}\), and \(e_{3}\) are called Euler parameters. Equation 6.15 indicates that the Euler parameters are not independent. Since \(\cos ^{2}(\phi / 2)+\mathbf{u}^{T} \mathbf{u} \sin ^{2}(\phi / 2)=\) 1 , then
\[
\begin{equation*}
\boldsymbol{e}_{0}^{2}+\mathbf{e}^{T} \mathbf{e}=1 \tag{6.21}
\end{equation*}
\]
i.e.,
\[
e_{0}^{2}+e_{1}^{2}+e_{2}^{2}+e_{3}^{2}=1
\]

If the four Euler parameters are put in a 4 -vector as follows:
\[
\begin{align*}
\mathbf{p} & =\left[e_{0}, e^{T}\right]^{T} \\
& =\left[e_{0}, e_{1}, e_{2}, e_{3}\right]^{T} \tag{6.22}
\end{align*}
\]
then Eq. 6.21 is written as
\[
\begin{equation*}
\mathbf{p}^{T} \mathbf{p}-1=0 \tag{6.23}
\end{equation*}
\]

According to Euler's theorem, any vector lying along the orientational axis of rotation must have the same components in both initial and final coordinate systems. This statement may be verified by finding the local and global components of the vector \(\vec{e}\). Assume that \(\mathbf{e}=\left[e_{1}, e_{2}, e_{3}\right]^{T}\) consists of the global components of \(\vec{e}\). The transformation matrix \(\mathbf{A}\) can be used to obtain the local components of \(\vec{e}\); i.e., \(\mathbf{e}^{\prime}\), as follows:
\[
\begin{aligned}
\mathbf{e}^{\prime} & =\mathbf{A}^{T} \mathbf{e} \\
& =\left(2 e_{0}^{2}-1\right) \mathbf{e}+2\left(\mathbf{\mathbf { e e } ^ { T }}-e_{0} \tilde{\mathbf{e}}\right) \mathbf{e} \\
& =\left(2 e_{0}^{2}-1\right) \mathbf{e}+2 \mathbf{e}\left(1-e_{0}^{2}\right) \\
& =\left(2 e_{0}^{2}-1\right) \mathbf{e}+2\left(1-e_{0}^{2}\right) \mathbf{e} \\
& =\left(2 e_{0}^{2}-1+2-2 e_{0}^{2}\right) \mathbf{e} \\
& =\mathbf{e}
\end{aligned}
\]
where Eqs. 6.18 and 6.21 and the identity \(\tilde{e} \mathbf{e}=\boldsymbol{0}(\) Eq. 2.48) have been used. This result shows that the global components and the local components of \(\vec{e}\) are the same. Figure 6.6 illustrates the projection of \(\vec{e}\) on both the \(\xi \eta \zeta\) and the \(x y z\) axes.

\subsection*{6.1.4 Determination of Euler Parameters}

From the transformation matrix of Eq. 6.19, it is possible to derive explicit formulas for the Euler parameters in terms of the elements of the transformation matrix. Assume that the nine direction cosines of a transformation matrix are given as in Eq. 6.7:
\[
\mathbf{A}=\left[\begin{array}{lll}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}\right]
\]


Figure 6.6 Projection of vector \(\vec{e}\) on \(\xi, \eta, \zeta, x, y\), and \(z\) axes.

The trace of \(\mathbf{A}\), denoted by \(\operatorname{tr} \mathbf{A}\), is defined as follows:
\[
\begin{equation*}
\operatorname{tr} \mathbf{A}=a_{11}+a_{22}+a_{33} \tag{6.24}
\end{equation*}
\]

From the transformation matrix of Eq. 6.19 it is found that
\[
\begin{aligned}
\operatorname{tr} \mathbf{A} & =2\left(3 e_{0}^{2}+e_{1}^{2}+e_{2}^{2}+e_{3}^{2}\right)-3 \\
& =2\left(2 e_{0}^{2}+1\right)-3 \\
& =4 e_{0}^{2}-1
\end{aligned}
\]
where Eq. 6.21 has been employed. This equation can be written as
\[
\begin{equation*}
e_{0}^{2}=\frac{\operatorname{tr} \mathbf{A}+1}{4} \tag{6.25}
\end{equation*}
\]

Substituting this into the diagonal elements of \(\mathbf{A}\) results in
\[
\begin{aligned}
a_{11} & =2\left(e_{0}^{2}+e_{1}^{2}\right)-1 \\
& =2\left(\frac{\operatorname{tr} \mathbf{A}+1}{4}+e_{1}^{2}\right)-1
\end{aligned}
\]
or
\[
\begin{equation*}
e_{1}^{2}=\frac{1+2 a_{11}-\operatorname{tr} \mathbf{A}}{4} \tag{6.26a}
\end{equation*}
\]
and similarly,
\[
\begin{equation*}
e_{2}^{2}=\frac{1+2 a_{22}-\operatorname{tr} \mathbf{A}}{4} \tag{6.26b}
\end{equation*}
\]
and
\[
\begin{equation*}
e_{3}^{2}=\frac{1+2 a_{33}-\operatorname{tr} \mathbf{A}}{4} \tag{6.26c}
\end{equation*}
\]

In contrast to Euler and Bryant angles (see Appendix A), or any other set of three rotational coordinates, there are no critical cases in which these inverse formulas are singular.

It is interesting and computationally important to note that Eqs. 6.25 and 6.26 determine only the magnitudes of the Euler parameters, in terms of only the diagonal elements of the direction-cosine matrix \(\mathbf{A}\). To find the algebraic signs of the Euler parameters, off-diagonal terms must be used. Equation 6.21 indicates that at least one Euler parameter must be nonzero, e.g., \(e_{0}\). The sign of \(e_{0}\) may be selected as positive or negative. Subtracting symmetrically placed off-diagonal terms of matrix \(\mathbf{A}\) in Eqs. 6.7 and 6.19 yields
\[
\begin{aligned}
& a_{32}-a_{23}=4 e_{0} e_{1} \\
& a_{13}-a_{31}=4 e_{0} e_{2} \\
& a_{21}-a_{12}=4 e_{0} e_{3}
\end{aligned}
\]
or
\[
\begin{align*}
& e_{1}=\frac{a_{32}-a_{23}}{4 e_{0}} \\
& e_{2}=\frac{a_{13}-a_{31}}{4 e_{0}}  \tag{6.27}\\
& e_{3}=\frac{a_{21}-a_{12}}{4 e_{0}}
\end{align*}
\]

If \(e_{0}\), calculated from Eq. 6.25 , is nonzero, then Eq. 6.27 can be used to determine \(e_{1}, e_{2}\), and \(e_{3}\). Suppose that for an assumed sign for \(e_{0}\), and for the computed values of \(e_{1}, e_{2}\), and \(e_{3}\), the angle of rotation and the axis of rotation are determined to be \(\phi\) and \(\vec{e}\), respectively. If the sign of \(e_{0}\) is inverted, the signs of \(e_{1}, e_{2}\), and \(e_{3}\) are inverted also. Changing the signs of all four parameters does not influence the transformation matrix, since the matrix is quadratic.

If \(e_{0}\), calculated from Eq. 6.25 , is found to be zero, then Eqs. \(6.26 \mathrm{a}-\mathrm{c}\) can be used to calculate \(e_{1}, e_{2}\), and \(e_{3}\). Since \(e_{0}=0\), Eq. 6.15 indicates that \(\phi=k \pi, k= \pm 1\), \(\pm 3, \ldots\) Therefore, the sign of \(\phi\) is immaterial; e.g., \(+\pi\) and \(-\pi\) are the same. To find the algebraic sign of \(e_{1}, e_{2}\), and \(e_{3}\), symmetrically placed off-diagonal terms of matrix \(\mathbf{A}\) are added to obtain
\[
\begin{align*}
& a_{21}+a_{12}=4 e_{1} e_{2} \\
& a_{31}+a_{13}=4 e_{1} e_{3}  \tag{6.28}\\
& a_{32}+a_{23}=4 e_{2} e_{3}
\end{align*}
\]

At least one of the three Euler parameters \(e_{1}, e_{2}\), and \(e_{3}\) must be nonzero. Its sign may be selected as positive or negative. Then, Eq. 6.28 can be used to determine the magnitude and the sign of the other two parameters.

\section*{Example 6.1}

Nine direction cosines of a transformation matrix \(\mathbf{A}\) are given as follows:
\[
\mathbf{A}=\left[\begin{array}{rrr}
0.5449 & -0.5549 & 0.6285 \\
0.3111 & 0.8299 & 0.4629 \\
-0.7785 & -0.0567 & 0.6249
\end{array}\right]
\]

Determine the four Euler parameters corresponding to this transformation.
Solution The trace of \(\mathbf{A}\) is calculated from Eq. 6.24:
\[
\operatorname{tr} \mathbf{A}=0.5449+0.8299+0.6249=1.9997
\]

Then, Eq. 6.25 yields \(e_{0}^{2}=0.7499\). Selecting the positive sign for \(e_{0}\), we find that \(e_{0}=0.866\). From Eq. 6.27,
\[
\begin{gathered}
e_{1}=\frac{-0.0567-0.4629}{4.0(0.866)}=-0.15 \\
e_{2}=\frac{0.6285+0.7785}{4.0(0.866)}=0.406 \\
e_{3}=\frac{0.3111+0.5549}{4.0(0.866)}=0.25
\end{gathered}
\]

A test can be performed to check that the four parameters satisfy the constraint of Eq. 6.21. Either the four parameters are \(\mathbf{p}=[0.866,-0.15,0.406,0.25]^{T}\), or, if the sign of \(e_{0}\) is changed, the four parameters become \(\mathbf{p}=[-0.866,0.15\), \(-0.406,-0.25]^{T}\).

\section*{Example 6.2}

Determine the four Euler parameters for transformation matrix
\[
\mathbf{A}=\left[\begin{array}{rrr}
-0.280 & -0.600 & -0.749 \\
-0.600 & -0.500 & 0.625 \\
-0.749 & 0.625 & -0.220
\end{array}\right]
\]

Solution The trace of \(\mathbf{A}\) is found from Eq. 6.24:
\[
\operatorname{tr} \mathbf{A}=-0.280-0.500-0.220=-1.0
\]

Then, Eq. 6.25 yields \(e_{0}=0.0\). From Eq. 6.26 it is found that
\[
e_{1}^{2}=\frac{1.0+2.0(-0.28)+1.0}{4.0}=0.36
\]

Therefore, \(e_{1}= \pm 0.6\). If the positive sign is selected for \(e_{1}\), then, Eq. 6.28 yields
\[
\begin{aligned}
& e_{2}=\frac{-0.6-0.6}{4.0(0.6)}=-0.5 \\
& e_{3}=\frac{-0.749-0.749}{4.0(0.6)}=-0.624
\end{aligned}
\]

The vector of the Euler parameters is \(\mathbf{p}=[0.0,0.6,-0.5,-0.624]^{T}\) or \(\mathbf{p}=[0.0\), \(-0.6,0.5,0.624]^{T}\).

When the angle of rotation is \(\phi=k \pi, k= \pm 1, \pm 3, \ldots\), then \(e_{0}\) is zero. Therefore, the transformation matrix of Eq. 6.19 becomes
\[
\mathbf{A}=2\left[\begin{array}{lll}
e_{1}^{2}-\frac{1}{2} & e_{1} e_{2} & e_{1} e_{3}  \tag{6.29}\\
e_{1} e_{2} & e_{2}^{2}-\frac{1}{2} & e_{2} e_{3} \\
e_{1} e_{3} & e_{2} e_{3} & e_{3}^{2}-\frac{1}{2}
\end{array}\right]
\]
which is symmetric. This property was observed in Example 6.2.

\subsection*{6.1.5 Determination of the Direction Cosines}

It was shown in Section 6.1.4 that the Euler parameters can be determined if the direction cosines are known. This section considers methods to determine the direction cosines.

One method for determining the nine direction cosines that describe the orientation of a body-fixed coordinate system with respect to the reference coordinates is to use Euler angles. If the three Euler angles can be determined (refer to Appendix A), then the elements of the transformation matrix can be computed. A direct calculation of the four Euler parameters in terms of the three Euler angles is given in Appendix B. This method may seem to be simple and straightforward; however, determination of the three Euler angles is difficult, and, for general cases, impractical.

A second method for determining the nine direction cosines is discussed here. Two points \(A\) and \(B\) are located on the \(\xi\) and \(\eta\) axes, as shown in Fig. 6.7. The \(x y z\) coordinates of \(A\) and \(B\) and the origin \(O\) can be found by measurements taken on the actual system or on an illustration, or by some other means. The coordinates of points \(O, A\), and \(B\) are denoted by \(\mathbf{r}, \mathbf{r}^{A}\), and \(\mathbf{r}^{B}\), respectively. Vectors \(\vec{a}\) and \(\vec{b}\) shown in Fig. 6.7 have the global component vectors
\[
\begin{align*}
& \mathbf{a}=\mathbf{r}^{A}-\mathbf{r} \\
& \mathbf{b}=\mathbf{r}^{B}-\mathbf{r} \tag{6.30}
\end{align*}
\]
and magnitudes \(a\) and \(b\), respectively. Vectors \(\vec{a}\) and \(\vec{b}\) must be orthogonal; i.e., \(\mathbf{a}^{\boldsymbol{T}} \mathbf{b}\) must equal 0 . This rule is an important test of the accuracy of the measured data.


Figure 6.7 Points defining the \(\xi \eta \zeta\) axes.

Unit vectors \(\vec{u}_{(\xi)}\) and \(\vec{u}_{(\eta)}\) may now be defined as follows:
\[
\begin{align*}
& \mathbf{u}_{(\xi)}=\frac{\mathbf{a}}{a} \\
& \mathbf{u}_{(\eta)}=\frac{\mathbf{b}}{b} \tag{6.31}
\end{align*}
\]

The third unit vector, \(\vec{u}_{(\zeta)}\) on the \(\zeta\) axis, can be found by noting that \(\vec{u}_{(\zeta)}=\vec{u}_{(\xi)} \times \vec{u}_{(\eta)}\); i.e.,
\[
\begin{equation*}
\mathbf{u}_{(\zeta)}=\tilde{\mathbf{u}}_{(\xi)} \mathbf{u}_{(\eta)} \tag{6.32}
\end{equation*}
\]

Then, from Eq. 6.9, the nine direction cosines are found to be
\[
\begin{equation*}
\mathbf{A}=\left[\mathbf{u}_{(\xi)}, \mathbf{u}_{(\eta)}, \mathbf{u}_{(\xi)}\right] \tag{6.33}
\end{equation*}
\]

\section*{Example 6.3}

Points \(A\) on the \(\xi\) axis, \(B\) on the \(\eta\) axis, and \(O\) (the origin of the \(\xi \eta \zeta\) axes) have coordinates \(\mathbf{r}^{A}=[0.977,1.665,2.916]^{T}, \mathbf{r}^{B}=[-0.573,2.539,-0.709]^{T}\), and \(\mathbf{r}=[-0.10,0.30,0.25]^{T}\). Determine the nine direction cosines and the four Euler parameters.
Solution From Eq. 6.30, it is found that
\[
\begin{aligned}
& \mathbf{a}=[1.077,1.365,2.666]^{T} \\
& \mathbf{b}=[-0.473,2.239,-0.959]^{T}
\end{aligned}
\]

A test for orthogonality shows that \(\mathbf{a}^{T} \mathbf{b}=-0.0099 \simeq 0.0\), which is acceptable. The magnitudes of \(\mathbf{a}\) and \(\mathbf{b}\) are calculated to be \(a=3.183\) and \(b=2.481\). Then, Eq. 6.31 determines the unit vectors as
\[
\begin{aligned}
& \mathbf{u}_{(\xi)}=[0.338,0.429,0.838]^{T} \\
& \mathbf{u}_{(\eta)}=[-0.191,0.902,-0.387]^{T}
\end{aligned}
\]

The third unit vector is found from Eq. 6.32:
\[
\mathbf{u}_{(6)}=\left[\begin{array}{ccc}
0 & -0.838 & 0.429 \\
0.838 & 0 & -0.338 \\
-0.429 & 0.338 & 0
\end{array}\right]\left[\begin{array}{r}
-0.191 \\
0.902 \\
-0.387
\end{array}\right]=\left[\begin{array}{r}
-0.922 \\
-0.293 \\
0.387
\end{array}\right]
\]

Hence
\[
\mathbf{A}=\left[\begin{array}{rrr}
0.338 & -0.191 & -0.922 \\
0.429 & 0.902 & -0.293 \\
0.838 & -0.387 & 0.387
\end{array}\right]
\]
which yields, according to the process of Sec. \(6.1 .4, \mathbf{p}=[0.810,-0.029,-0.543\), \(0.191]^{T}\). The sum of the squares of the four Euler parameters is \(\mathbf{p}^{T} \mathbf{p}=0.988 \simeq 1\).

In most practical problems, the choice of how to embed a body-fixed coordinate system in a body (a link) is open. The \(\xi \eta \zeta\) axes may be embedded in a body according

(a)

(b)

(c)


(d)


Figure 6.8 Orientation of body-fixed coordinate system in special cases. (a) \(\xi \eta \xi \| x y z\), (b) \(\xi \| x\), (c) \(\eta \| y\), and (d) \(\zeta \| z\).
to any of the configurations shown in Fig. 6.8. If the \(\xi \eta \zeta\) axes are parallel to the \(x y z\) axes, as shown in Fig. 6.8(a), then
\[
\begin{equation*}
\mathbf{p}=[1,0,0,0]^{T} \quad \xi \eta \zeta \| x y z \tag{6.34a}
\end{equation*}
\]

If the \(\xi\) axis is parallel to the \(x\) axis and the angle of rotation is \(\phi\), as shown in Fig. 6.8(b), then
\[
\begin{equation*}
\mathbf{p}=\left[\cos \frac{\phi}{2}, \sin \frac{\phi}{2}, 0,0\right]^{T} \quad \xi \| x \tag{6.34b}
\end{equation*}
\]

Similarly, for the orientations shown in Fig. 6.8(c) and (d),
\[
\begin{equation*}
\mathbf{p}=\left[\cos \frac{\phi}{2}, 0, \sin \frac{\phi}{2}, 0\right]^{T} \quad \eta \| y \tag{6.34c}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{p}=\left[\cos \frac{\phi}{2}, 0,0, \sin \frac{\phi}{2}\right]^{T} \quad \zeta \| z \tag{6.34d}
\end{equation*}
\]

In these special cases, it is relatively simple to determine the angle of rotation and then to calculate the Euler parameters.

\subsection*{6.2 IDENTITIES WITH EULER PARAMETERS}

In this section, important formulas and identities between Euler parameters, their time derivatives, and their transformation matrices are derived. Derivation of some of the identities is shown in the text. However, to avoid extensive proofs in the text, several
problems are given instead at the end of this chapter. These identities are useful in the derivation of spatial constraint equations and equations of spatial motion. \({ }^{15}\)

The product \(\mathbf{p p}^{T}\) is a \(4 \times 4\) matrix that can be written in the form
\[
\begin{align*}
\mathbf{p p}^{T} & =\left[\begin{array}{c}
e_{0} \\
\mathbf{e}
\end{array}\right]\left[e_{0}, \mathbf{e}^{T}\right] \\
& =\left[\begin{array}{ll}
e_{0}^{2} & e_{0} \mathbf{e}^{T} \\
e_{0} \mathbf{e} & \mathbf{e e}^{T}
\end{array}\right] \tag{6.35}
\end{align*}
\]

From Eqs. 2.48 and 2.50, it is found that
\[
\begin{equation*}
\tilde{\mathbf{e} e}=0 \tag{6.36}
\end{equation*}
\]
and
\[
\begin{align*}
\tilde{\mathbf{e}} \tilde{\mathbf{e}} & =\mathbf{e}^{T}-\mathbf{e}^{T} \mathbf{e} \mathbf{I} \\
& =\mathbf{e}^{T}-\left(1-e_{0}^{2}\right) \mathbf{I} \tag{6.37}
\end{align*}
\]

A pair of \(3 \times 4\) matrices \(\mathbf{G}\) and \(\mathbf{L}\) are defined as \({ }^{\dagger}\)
\[
\begin{align*}
\mathbf{G} & =\left[\begin{array}{rrrr}
-e_{1} & e_{0} & -e_{3} & e_{2} \\
-e_{2} & e_{3} & e_{0} & -e_{1} \\
-e_{3} & -e_{2} & e_{1} & e_{0}
\end{array}\right] \\
& =\left[-\mathbf{e}, \tilde{\mathbf{e}}+e_{0} \mathbf{I}\right] \tag{6.38}
\end{align*}
\]
and
\[
\begin{align*}
\mathbf{L} & =\left[\begin{array}{rrrr}
-e_{1} & e_{0} & e_{3} & -e_{2} \\
-e_{2} & -e_{3} & e_{0} & e_{1} \\
-e_{3} & e_{2} & -e_{1} & e_{0}
\end{array}\right] \\
& =\left[-\mathbf{e},-\tilde{\mathbf{e}}+e_{0} \mathbf{I}\right] \tag{6.39}
\end{align*}
\]

Each row of \(\mathbf{G}\) and \(\mathbf{L}\) is orthogonal to \(\mathbf{p}\); i.e.,
\[
\begin{align*}
\mathbf{G p} & =\left[-\mathbf{e}, \tilde{\mathbf{e}}+e_{0} \mathbf{I}\right]\left[\begin{array}{l}
e_{0} \\
\mathbf{e}
\end{array}\right] \\
& =\left[-e_{0} \mathbf{e}+\tilde{\mathbf{e}} \mathbf{e}+e_{0} \mathbf{e}\right]=\mathbf{0} \tag{6.40}
\end{align*}
\]
where Eq. 6.36 has been used. Similarly,
\[
\begin{equation*}
\mathbf{L p}=0 \tag{6.41}
\end{equation*}
\]

A direct calculation reveals that the rows of \(\mathbf{G}\) are orthogonal, as are also the rows of \(\mathbf{L}\); i.e.,
\[
\begin{equation*}
\mathbf{G G}^{T}=\mathbf{I} \tag{6.42}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{L L}^{T}=\mathbf{I} \tag{6.43}
\end{equation*}
\]
\({ }^{\dagger}\) These matrices will be used extensively in the formulations that follow in this text. In Sec. 6.4 and some other sections it will be seen that \(\mathbf{G}\) and \(\mathbf{L}\) are transformation matrices dealing with global and local components of vectors.
so that
\[
\begin{equation*}
\mathbf{G} \mathbf{G}^{T}=\mathbf{L} \mathbf{L}^{T} \tag{6.44}
\end{equation*}
\]

However, \(\mathbf{G}^{T} \mathbf{G}\) is of the form
\[
\begin{align*}
\mathbf{G}^{T} \mathbf{G} & =\left[\begin{array}{c}
-\mathbf{e}^{T} \\
-\tilde{\mathbf{e}}+e_{0} \mathbf{I}
\end{array}\right]\left[-\mathbf{e}, \tilde{\mathbf{e}}+e_{0} \mathbf{I}\right] \\
& =\left[\begin{array}{ll}
\mathbf{e}^{T} \mathbf{e} & -\mathbf{e}^{T} \tilde{\mathbf{e}}-e_{0} \mathbf{e}^{T} \\
\tilde{\mathbf{e}} \mathbf{e}-e_{0} \mathbf{e} & -\tilde{\mathbf{e}} \tilde{\mathbf{e}}+e_{0} \tilde{\mathbf{e}}-e_{0} \tilde{\mathbf{e}}+e_{0}^{2} \mathbf{I}
\end{array}\right] \\
& =\left[\begin{array}{ll}
1-e_{0}^{2} & -e_{0} \mathbf{e}^{T} \\
-e_{0} \mathbf{e} & -\mathbf{e}^{T}+\mathbf{I}
\end{array}\right] \\
& =-\left[\begin{array}{ll}
e_{0}^{2} & e_{0} \mathbf{e}^{T} \\
e_{0} \mathbf{e} & \mathbf{e d}^{T}
\end{array}\right]+\mathbf{I}^{*} \\
& =-\mathbf{p} \mathbf{p}^{T}+\mathbf{I}^{*} \tag{6.45}
\end{align*}
\]
where Eq. 6.37 has been used and \(\mathbf{I}^{*}\) is the \(4 \times 4\) identity matrix. Similarly, it can be shown that
\[
\begin{equation*}
\mathbf{L}^{T} \mathbf{L}=-\mathbf{p} \mathbf{p}^{T}+\mathbf{I}^{*} \tag{6.46}
\end{equation*}
\]
so that
\[
\begin{equation*}
\mathbf{G}^{T} \mathbf{G}=\mathbf{L}^{T} \mathbf{L} \tag{6.47}
\end{equation*}
\]

A very interesting relationship can be found by evaluating the matrix product \(\mathbf{G} \mathbf{L}^{T}\) :
\[
\begin{align*}
\mathbf{G} \mathbf{L}^{T} & =\left[-\mathbf{e}, \tilde{\mathbf{e}}+e_{0} \mathbf{I}\right]\left[\begin{array}{c}
-\mathbf{e}^{T} \\
\mathbf{e}+e_{0} \mathbf{I}
\end{array}\right] \\
& =\mathbf{e e}^{T}+\left(\tilde{\mathbf{e}}+e_{0} \mathbf{I}\right)\left(\tilde{\mathbf{e}}+e_{0} \mathbf{I}\right) \\
& =\left(2 e_{0}^{2}-1\right) \mathbf{I}+2\left(\mathbf{e e}^{T}+e_{0} \tilde{\mathbf{e}}\right) \tag{6.48}
\end{align*}
\]

Comparing Eq. 6.48 with the transformation matrix \(\mathbf{A}\) of Eq. 6.18 reveals that
\[
\begin{equation*}
\mathbf{A}=\mathbf{G} \mathbf{L}^{T} \tag{6.49}
\end{equation*}
\]

Equation 6.49 demonstrates that the quadratic transformation matrix \(\mathbf{A}\) can be treated as the result of two successive linear transformations. This is one of the most useful relationships between the \(\mathbf{G}\) and \(\mathbf{L}\) matrices and is a powerful property of Euler parameters. The first time derivative of Eq. 6.23 yields
\[
\begin{equation*}
\mathbf{p}^{T} \dot{\mathbf{p}}=\dot{\mathbf{p}}^{T} \mathbf{p}=0 \tag{6.50}
\end{equation*}
\]

Similarly, the first time derivatives of Eqs. 6.40 and 6.41 result in the identities
\[
\begin{equation*}
\mathbf{G} \dot{\mathbf{p}}=-\dot{\mathbf{G}} \mathbf{p} \tag{6.51}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{L} \dot{\mathbf{p}}=-\dot{\mathbf{L}} \mathbf{p} \tag{6.52}
\end{equation*}
\]

The product \(\dot{\mathbf{G}} \dot{\mathbf{p}}\) may be calculated, using Eq. 6.38, as follows:
\[
\begin{align*}
\dot{\mathbf{G}} \dot{\mathbf{p}} & =\left[-\dot{\mathbf{e}}, \tilde{\mathbf{e}}+\dot{e}_{0} \mathbf{I}\right]\left[\begin{array}{l}
\dot{e}_{0} \\
\dot{\mathbf{e}}
\end{array}\right] \\
& =-\dot{e}_{0} \dot{\mathbf{e}}+\dot{e}_{0} \dot{\mathbf{e}}+\tilde{\mathbf{e}} \dot{\mathbf{e}}=0 \tag{6.53}
\end{align*}
\]
since the vector product of \(\dot{\mathbf{e}}\) by itself is zero. Similarly,
\[
\begin{equation*}
\dot{\mathrm{L}} \dot{\mathbf{p}}=0 \tag{6.54}
\end{equation*}
\]

Equation 2.53 can be employed, with Eqs. 6.38 and 6.39, to show that
\[
\begin{equation*}
\mathbf{G} \dot{\mathbf{L}}^{T}=\dot{\mathbf{G}} \mathbf{L}^{T} \tag{6.55}
\end{equation*}
\]

The time derivative of Eq. 6.49 yields
\[
\begin{align*}
\dot{\mathbf{A}} & =\dot{\mathbf{G}} \mathbf{L}^{T}+\mathbf{G} \dot{\mathbf{L}}^{T}=2 \mathbf{G} \dot{\mathbf{L}}^{T} \\
& =2 \mathbf{G} \mathbf{L}^{T} \tag{6.56}
\end{align*}
\]

The product \(\mathbf{G} \dot{\mathbf{p}}\) can be expanded as follows:
\[
\begin{aligned}
\mathbf{G} \dot{\mathbf{p}} & =\left[-\mathbf{e}, \tilde{\mathbf{e}}+e_{0} \mathbf{I}\right]\left[\begin{array}{c}
\dot{e}_{0} \\
\dot{\mathbf{e}}
\end{array}\right] \\
& =-\dot{e}_{0} \mathbf{e}+\tilde{\mathbf{e}} \dot{\mathbf{e}}+e_{0} \dot{\mathbf{e}}
\end{aligned}
\]

Transforming both sides of the equation to skew-symmetric matrices, by the operation shown in Eq. 2.43 of Chap. 2, yields
\[
\begin{align*}
\widetilde{\mathbf{G} \mathbf{p}} & =-\dot{e}_{0} \tilde{\mathbf{e}}+\widetilde{\mathbf{e}} \widetilde{\mathbf{e}}+e_{\tilde{e}}^{\tilde{\mathbf{e}}} \\
& =-\dot{e}_{0} \tilde{\mathbf{e}}+\tilde{\mathbf{e}} \tilde{\tilde{e}}-\tilde{\tilde{\mathbf{e}} \tilde{\mathbf{e}}}+e_{0}^{\tilde{\mathbf{e}}} \\
& =-\dot{e}_{0} \tilde{\mathbf{e}}+\tilde{\mathbf{e}} \tilde{\mathbf{e}}-\dot{\mathbf{e}}^{T}+\dot{\mathbf{e}}^{T} \mathbf{I} \mathbf{I}+e_{0} \tilde{\mathbf{e}} \\
& =-\dot{e}_{0} \tilde{\mathbf{e}}+\tilde{\mathbf{e}} \tilde{\mathbf{e}}-\mathbf{e} \dot{\mathbf{e}}^{T}-e_{0} \dot{e}_{0} \mathbf{I}+e_{0} \tilde{\mathbf{e}} \\
& =\left[\mathbf{e},-\tilde{\mathbf{e}}-e_{0} \mathbf{I}\right]\left[\begin{array}{c}
-\dot{\mathbf{e}}^{T} \\
-\tilde{\mathbf{e}}+\dot{e}_{0} \mathbf{I}
\end{array}\right] \\
& =-\mathbf{G} \dot{\mathbf{G}}^{T} \tag{6.57}
\end{align*}
\]
where Eqs. 2.52, 2.50, and the identity \(e_{0} \dot{e}_{0}+\mathbf{e}^{T} \dot{\mathbf{e}}=0\) (Eq. 6.50) have been used. Similarly,
\[
\begin{equation*}
\widetilde{\mathbf{L}}=\mathbf{L} \dot{\mathbf{L}}^{T} \tag{6.58}
\end{equation*}
\]

Two more identities can be derived using Eqs. 6.51, 6.52, 6.57, and 6.58:
\[
\begin{align*}
\mathbf{G} \dot{\mathbf{G}}^{T} & =-\dot{\mathbf{G}} \mathbf{G}^{T}  \tag{6.59}\\
\mathbf{L \dot { L }}^{T} & =-\dot{\mathbf{L}} \mathbf{L}^{T} \tag{6.60}
\end{align*}
\]

Furthermore, the time derivative of Eq. 6.50 yields
\[
\begin{equation*}
\mathbf{p}^{T} \ddot{\mathbf{p}}+\dot{\mathbf{p}}^{T} \dot{\mathbf{p}}=0 \tag{6.61}
\end{equation*}
\]

The time derivative of Eq. 6.56 results in
\[
\begin{align*}
\ddot{\mathbf{A}} & =2 \dot{\mathbf{G}} \dot{\mathbf{L}}^{T}+2 \mathbf{G} \ddot{\mathbf{L}}^{T} \\
& =2 \dot{\mathbf{G}} \dot{\mathbf{L}}^{T}+2 \mathbf{\mathbf { G }} \mathbf{L}^{T} \tag{6.62}
\end{align*}
\]
from which it is seen that
\[
\begin{equation*}
\ddot{\mathbf{G}} \mathbf{L}^{T}=\mathbf{G} \ddot{\mathbf{L}}^{T} \tag{6.63}
\end{equation*}
\]

At this time, it may not be apparent how useful these identities can be. However, later in this chapter and in the next several chapters, these identities will be used extensively.

\subsection*{6.2.1 Identities with Arbitrary Vectors}

Additional useful identities between Euler parameters, transformation matrices, and arbitrary vectors are derived here for later use. \({ }^{15}\) Consider an arbitrary 3 -vector a. Two \(4 \times 4\) matrices \(\stackrel{+}{\mathbf{a}}\) and \(\overline{\mathbf{a}}\) are defined as follows:
\[
\stackrel{+}{\mathbf{a}} \equiv\left[\begin{array}{cc}
0 & -\mathbf{a}^{T}  \tag{6.64}\\
\mathbf{a} & \tilde{\mathbf{a}}
\end{array}\right]
\]
and
\[
\overline{\mathbf{a}} \equiv\left[\begin{array}{ll}
0 & -\mathbf{a}^{T}  \tag{6.65}\\
\mathbf{a} & -\tilde{\mathbf{a}}
\end{array}\right]
\]

The overhead plus or minus refers to the sign of the skew-symmetric matrix \(\overline{\mathbf{a}}\) in the definitions. Since \(\stackrel{+}{\mathbf{a}}\) and \(\overline{\mathbf{a}}\) are skew-symmetric,
\[
\begin{equation*}
\stackrel{+}{\mathbf{a}}^{T}=-\stackrel{+}{\mathbf{a}} \tag{6.66}
\end{equation*}
\]
and
\[
\begin{equation*}
\overline{\mathbf{a}}^{T}=-\overline{\mathbf{a}} \tag{6.67}
\end{equation*}
\]

To illustrate the importance and convenience of this notation, the matrix product \(\mathbf{G}^{T} \mathbf{a}\) may be evaluated as follows:
\[
\begin{align*}
\mathbf{G}^{T} \mathbf{a} & =\left[\begin{array}{c}
-\mathbf{e}^{T} \\
-\tilde{\mathbf{e}}+e_{0} \mathbf{I}
\end{array}\right] \mathbf{a} \\
& =\left[\begin{array}{c}
-\mathbf{e}^{T} \mathbf{a} \\
-\tilde{\mathbf{e}} \mathbf{a}+e_{0} \mathbf{a}
\end{array}\right] \\
& =\left[\begin{array}{c}
-\mathbf{a}^{T} \mathbf{e} \\
\mathbf{a} e_{0}+\tilde{\mathbf{a}} \mathbf{e}
\end{array}\right] \\
& =\left[\begin{array}{cc}
0 & -\mathbf{a}^{T} \\
\mathbf{a} & \mathbf{a}
\end{array}\right]\left[\begin{array}{l}
e_{0} \\
\mathbf{e}
\end{array}\right] \\
& =\stackrel{\mathbf{a}}{\mathbf{p}} \tag{6.68}
\end{align*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\mathbf{L}^{T} \mathbf{a}=\overline{\mathbf{a}} \mathbf{p} \tag{6.69}
\end{equation*}
\]

The product Gà is evaluated as follows:
\[
\begin{align*}
\mathbf{G} \mathbf{a} & =\left[-\mathbf{e}, \tilde{\mathbf{e}}+e_{0} \mathbf{I}\right]\left[\begin{array}{cc}
0 & -\mathbf{a}^{T} \\
\mathbf{a} & \tilde{\mathbf{a}}
\end{array}\right] \\
& =\left[\tilde{\mathbf{e}} \mathbf{a}+e_{0} \mathbf{a}, \mathbf{e} \mathbf{e}^{T}+\tilde{\mathbf{e n}} \mathbf{a}+e_{0} \tilde{\mathbf{a}}\right] \\
& =\left[\tilde{\mathbf{e}} \mathbf{a}, \tilde{\mathbf{a}} \tilde{\mathbf{e}}+e_{0} \tilde{\mathbf{a}}\right]+\left[e_{0} \mathbf{a}, \mathbf{a e}^{T}\right] \tag{a}
\end{align*}
\]
where Eq. 2.53 has been used. It can be shown that \(\left[\mathbf{e} \mathbf{a}, \tilde{\mathbf{a}} \tilde{\mathbf{e}}+e_{0} \tilde{\mathbf{a}}\right]=\tilde{\mathbf{a}} \mathbf{G}\), and hence Eq. \(a\) is reduced to
\[
\begin{equation*}
\mathbf{G} \mathbf{a}=\tilde{\mathbf{a}} \mathbf{G}+\mathbf{a p} \mathbf{p}^{T} \tag{6.70}
\end{equation*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\mathbf{L} \overline{\mathbf{a}}=-\tilde{\mathbf{a}} \mathbf{L}+\mathbf{a} \mathbf{p}^{T} \tag{6.71}
\end{equation*}
\]

The time derivative of Eq. 6.68 can be written as
\[
\begin{equation*}
\dot{\mathbf{G}}^{T} \mathbf{a}+\mathbf{G}^{T} \dot{\mathbf{a}}=\stackrel{+}{\mathbf{a}} \mathbf{p}+\stackrel{+}{\mathbf{a}} \mathbf{p} \tag{b}
\end{equation*}
\]

Since \(\mathbf{a}_{+}\)is an arbitrary vector, Eq. 6.68 can be evaluated with the vector \(\dot{\mathbf{a}}\), to obtain \(\mathbf{G}^{T} \dot{\mathbf{a}}=\stackrel{+}{\mathbf{a}} \dot{\mathbf{p}}\). This result can be used in Eq. \(b\) to obtain
\[
\begin{equation*}
\dot{\mathbf{G}}^{T} \mathbf{a}=\stackrel{+}{\mathbf{a}} \dot{\mathbf{p}} \tag{6.72}
\end{equation*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\dot{\mathbf{L}}^{T} \mathbf{a}=\overline{\mathbf{a}} \dot{\mathbf{p}} \tag{6.73}
\end{equation*}
\]

Postmultiplying Eq. 6.56 by a and using Eqs. 6.73 and 6.69 yields
\[
\begin{equation*}
\dot{\mathbf{A}} \mathbf{a}=2 \mathbf{G} \overline{\mathbf{a}} \dot{\mathbf{p}} \tag{6.74}
\end{equation*}
\]
and
\[
\begin{equation*}
\dot{\mathbf{A}} \mathbf{a}=2 \dot{\mathbf{G}} \mathbf{a} \mathbf{p} \tag{6.75}
\end{equation*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\dot{\mathbf{A}}^{T} \mathbf{a}=2 \mathbf{L} \mathbf{a} \dot{\mathbf{p}} \tag{6.76}
\end{equation*}
\]
and
\[
\begin{equation*}
\dot{\mathbf{A}}^{T} \mathbf{a}=2 \dot{\mathbf{L}} \mathbf{a} \mathbf{p} \tag{6.77}
\end{equation*}
\]

The time derivative of Eq. 6.72 can be written as
\[
\begin{equation*}
\ddot{\mathbf{G}}^{T} \mathbf{a}+\dot{\mathbf{G}}^{T} \dot{\mathbf{a}}=\stackrel{+}{\mathbf{a}} \dot{\mathbf{p}}+\stackrel{+}{\mathbf{a}} \ddot{\mathbf{p}} \tag{6.78}
\end{equation*}
\]

Since \(\mathbf{a}\) is an arbitrary vector, Eq. 6.72 is also valid as \(\dot{\mathbf{G}}^{T} \dot{\mathbf{a}}=\stackrel{+}{\mathbf{a}} \dot{\mathbf{p}}\). Hence, Eq. 6.78 becomes
\[
\begin{equation*}
\ddot{\mathbf{G}}^{T} \mathbf{a}=\stackrel{+}{\mathbf{a}} \ddot{\mathbf{p}} \tag{6.79}
\end{equation*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\ddot{\mathbf{L}}^{T} \mathbf{a}=\overline{\mathbf{a}} \ddot{\mathbf{p}} \tag{6.80}
\end{equation*}
\]

Equation 6.62 is postmultiplied by a to obtain
\[
\begin{equation*}
\ddot{\mathbf{A}} \mathbf{a}=2 \dot{\mathbf{G}} \dot{\mathbf{L}}^{T} \mathbf{a}+2 \mathbf{G} \dot{\mathbf{L}}^{T} \mathbf{a} \tag{6.81}
\end{equation*}
\]

From Eqs. 6.73 and 6.80 , Eq. 6.81 becomes
\[
\begin{equation*}
\ddot{\mathbf{A}} \mathbf{a}=2 \dot{\mathbf{G}} \dot{\mathbf{L}}^{T} \mathbf{a}+2 \mathbf{G} \overline{\mathbf{a}} \ddot{\square} \tag{6.82}
\end{equation*}
\]
or
\[
\begin{equation*}
\ddot{\mathbf{A}} \mathbf{a}=2 \dot{\mathbf{G}} \overline{\mathbf{a}} \dot{\mathbf{p}}+2 \mathbf{G} \overline{\mathbf{a}} \ddot{\mathbf{p}} \tag{6.83}
\end{equation*}
\]

Similarly, the product \(\ddot{\mathbf{A}}^{T} \mathbf{a}\) can be written as follows:
\[
\begin{equation*}
\ddot{\mathbf{A}}^{T} \mathbf{a}=2 \dot{\mathbf{L}} \dot{\mathbf{G}}^{T} \mathbf{a}+2 \mathbf{L} \dot{\mathbf{a}} \ddot{\mathbf{p}} \tag{6.84}
\end{equation*}
\]
or
\[
\begin{equation*}
\ddot{\mathbf{A}}^{T} \mathbf{a}=2 \dot{\mathbf{L}} \dot{\mathbf{a}} \dot{\mathbf{p}}+2 \mathbf{L}^{+} \ddot{\mathbf{p}} \tag{6.85}
\end{equation*}
\]

The partial derivative of the matrix product Aa with respect to \(\mathbf{p}\) is expanded as follows:
\[
\begin{align*}
\frac{\partial}{\partial \mathbf{p}}(\mathbf{A a}) & =\frac{\partial}{\partial \mathbf{p}}\left[\left(2 e_{0}^{2}-1\right) \mathbf{a}+2 \mathbf{e e}^{T} \mathbf{a}+2 e_{0} \tilde{\mathbf{e}} \mathbf{a}\right] \\
& =2\left[2 e_{0} \mathbf{a}+\tilde{\mathbf{e}} \mathbf{a}, \mathbf{e}^{T} \mathbf{a} \mathbf{I}+\mathbf{e a}^{T}-e_{0} \tilde{\mathbf{a}}\right] \tag{c}
\end{align*}
\]

By using Eq. 2.50, we can write this partial derivative thus:
\[
\begin{align*}
\frac{\partial}{\partial \mathbf{p}}(\mathbf{A} \mathbf{a}) & =2\left[2 e_{0} \mathbf{a}+\tilde{\mathbf{e}} \mathbf{a}, \mathbf{a} \mathbf{e}^{T}-\tilde{\mathbf{e}} \tilde{\mathbf{a}}+\mathbf{e a}^{T}-e_{0} \tilde{\mathbf{a}}\right] \\
& =2\left[e_{0} \mathbf{a}+\tilde{\mathbf{e}} \mathbf{a},-\tilde{\mathbf{e}} \tilde{\mathbf{a}}+\mathbf{e a}^{T}-e_{0} \tilde{\mathbf{a}}\right]+2\left[e_{0} \mathbf{a}, \mathbf{a e}^{T}\right] \\
& =2\left[-\mathbf{e}, \tilde{\mathbf{e}}+e_{0} \mathbf{I}\right]\left[\begin{array}{cc}
0 & -\mathbf{a}^{T} \\
\mathbf{a} & -\tilde{\mathbf{a}}
\end{array}\right]+2 \mathbf{a} \mathbf{p}^{T} \\
& =2 \mathbf{G} \mathbf{a}+2 \mathbf{a} \mathbf{p}^{T} \tag{6.86}
\end{align*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\frac{\partial}{\partial \mathbf{p}}\left(\mathbf{A}^{T} \mathbf{a}\right)=2 \mathbf{L}^{+} \mathbf{a}+2 \mathbf{a} \mathbf{p}^{T} \tag{6.87}
\end{equation*}
\]

The following identity is valid for the transformation matrix \(\mathbf{A}\) - which may be described in terms of Euler parameters or any other set of rotational coordinates-and any vector \(\vec{s}\). If the vector product of vector \(\vec{s}\) and an arbitrary vector \(\vec{a}\) is a vector \(\vec{b}\), then in terms of global and local components, this vector product is expressed as
\[
\begin{equation*}
\mathbf{b}=\tilde{\mathbf{s}} \mathbf{a} \tag{d}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{b}^{\prime}==\tilde{\mathbf{s}}^{\prime} \mathbf{a}^{\prime} \tag{e}
\end{equation*}
\]

Since \(\mathbf{a}=\mathbf{A} \mathbf{a}^{\prime}\) and \(\mathbf{b}=\mathbf{A b}^{\prime}\), Eq. \(d\) becomes
\[
\begin{equation*}
\mathbf{A} \mathbf{b}^{\prime}=\tilde{\mathbf{s}} \mathbf{A} \mathbf{a}^{\prime} \tag{f}
\end{equation*}
\]

Substituting Eq. \(e\) into Eq. \(f\) and eliminating the arbitrary vector \(\mathbf{a}^{\prime}\) from both sides yields
\[
\begin{equation*}
\mathbf{A} \tilde{\mathbf{s}}^{\prime}=\tilde{\mathbf{s}} \mathbf{A} \tag{6.88}
\end{equation*}
\]

Postmultiplying both sides of Eq. 6.88 by \(\mathbf{A}^{T}\) yields
\[
\begin{equation*}
\tilde{\mathbf{s}}=\mathbf{A} \tilde{\mathbf{s}}^{\prime} \mathbf{A}^{T} \tag{6.89}
\end{equation*}
\]

Equation 6.89 will be found useful in many derivations.

\subsection*{6.3 THE CONCEPT OF ANGULAR VELOCITY}

Consider the \(\xi \eta \zeta\) coordinate system shown in Fig. 6.9(a), with its origin constrained to the origin of the nonrotating \(x y z\) coordinate system, but otherwise free to rotate. The global location of a point \(P\) that is fixed in the \(\xi \eta \zeta\) coordinate system is given by the equation
\[
\mathbf{s}^{P}=\mathbf{A s}^{{ }^{P}}
\]

Differentiating this equation with respect to time yields
\[
\dot{\mathbf{s}}^{P}=\dot{\mathbf{A}} \mathbf{s}^{P}+\mathbf{A} \dot{\mathbf{s}}^{\prime P}
\]

Since \(\vec{s}^{P}\) is fixed in the \(\xi \eta \zeta\) axes, \(\dot{\mathbf{s}}^{\prime P}=\mathbf{0}\), and therefore
\[
\begin{equation*}
\dot{\mathbf{s}}^{P}=\dot{\mathbf{A}} \mathbf{s}^{\prime P} \tag{6.90}
\end{equation*}
\]

(a)

(b)

Figure 6.9 (a) Rotating \(\xi \eta \zeta\) coordinate system, (b) Rotating and translating \(\xi \eta \zeta\) coordinate system.
At this point, the objective is to express the elements of matrix \(\dot{\mathbf{A}}\) in terms of the elements of matrix \(\mathbf{A}\). Two linear relationships between \(\dot{\mathbf{A}}\) and \(\mathbf{A}\) may be expressed as
\[
\begin{equation*}
\dot{\mathbf{A}}=\boldsymbol{\Omega} \mathbf{A} \tag{6.91}
\end{equation*}
\]
or
\[
\begin{equation*}
\dot{\mathbf{A}}=\mathbf{A} \boldsymbol{\Omega}^{\prime} \tag{6.92}
\end{equation*}
\]
where \(\boldsymbol{\Omega}\) and \(\boldsymbol{\Omega}^{\prime}\) are two \(3 \times 3\) coefficient matrices. What the two coefficient matrices are and how they are related will be answered in the remainder of this section. \({ }^{19}\)

Differentiating the identity \(\mathbf{A}^{T} \mathbf{A}=\mathbf{I}\) with respect to time yields
\[
\begin{equation*}
\dot{\mathbf{A}}^{T} \mathbf{A}+\mathbf{A}^{T} \dot{\mathbf{A}}=\mathbf{0} \tag{a}
\end{equation*}
\]

Substituting Eq. 6.91 into Eq. \(a\) results in
\[
\begin{equation*}
\mathbf{A}^{T} \boldsymbol{\Omega}^{T} \mathbf{A}+\mathbf{A}^{T} \boldsymbol{\Omega} \mathbf{A}=\mathbf{0} \tag{b}
\end{equation*}
\]

Premultiplying Eq. \(b\) by \(\mathbf{A}\) and then postmultiplying the result by \(\mathbf{A}^{T}\) yields \(\boldsymbol{\Omega}^{T}+\) \(\boldsymbol{\Omega}=\mathbf{0}\), or
\[
\begin{equation*}
\boldsymbol{\Omega}=-\boldsymbol{\Omega}^{T} \tag{6.93}
\end{equation*}
\]

Equation 6.93 indicates that \(\boldsymbol{\Omega}\) is a skew-symmetric matrix. Assume that \(\boldsymbol{\Omega}\) is composed of the elements of a \(\mathbf{3}\)-vector \(\boldsymbol{\omega}\) so that \(\boldsymbol{\Omega}=\tilde{\boldsymbol{\omega}}\). Then Eq. 6.91 becomes
\[
\begin{equation*}
\dot{\mathbf{A}}=\tilde{\boldsymbol{\omega}} \mathbf{A} \tag{6.94}
\end{equation*}
\]

Similarly, substituting Eq. 6.92 into Eq. \(a\) results in
\[
\begin{equation*}
\mathbf{\Omega}^{\prime T} \mathbf{A}^{T} \mathbf{A}+\mathbf{A}^{T} \mathbf{A} \mathbf{\Omega}^{\prime}=\mathbf{0} \tag{c}
\end{equation*}
\]
or
\[
\begin{equation*}
\mathbf{\Omega}^{\prime}=-\mathbf{\Omega}^{\prime T} \tag{6.95}
\end{equation*}
\]

Therefore \(\boldsymbol{\Omega}^{\prime}\) is also a skew-symmetric matrix. Assume that \(\boldsymbol{\Omega}^{\prime}\) is composed of the elements of a 3 -vector \(\boldsymbol{\omega}^{\prime}\) so that \(\boldsymbol{\Omega}^{\prime}=\overline{\boldsymbol{\omega}}^{\prime}\). Then Eq. 6.92 becomes
\[
\begin{equation*}
\dot{\mathbf{A}}=\mathbf{A} \bar{\omega}^{\prime} \tag{6.96}
\end{equation*}
\]

Comparing Eqs. 6.94 and 6.96 gives
\[
\begin{equation*}
\tilde{\boldsymbol{\omega}} \mathbf{A}=\mathbf{A} \tilde{\boldsymbol{\omega}}^{\prime} \tag{6.97}
\end{equation*}
\]

Equation 6.97 is identical in form with Eq. 6.88 ; i.e., \(\tilde{\mathbf{s}} \mathbf{A}=\mathbf{A} \tilde{\boldsymbol{s}}^{\prime}\). Therefore, it can be deduced that \(\omega\) and \(\omega^{\prime}\) are the global and the local components of the same vector \(\vec{\omega}\). The vector \(\vec{\omega}\) is defined as the angular velocity of the \(\xi \eta \zeta\) coordinate system. The components of vector \(\vec{\omega}\) may be expressed as
\[
\begin{equation*}
\boldsymbol{\omega}=\left[\omega_{(x)}, \omega_{(y)}, \omega_{(z)}\right]^{T} \tag{6.98}
\end{equation*}
\]
and
\[
\begin{equation*}
\boldsymbol{\omega}^{\prime}=\left[\omega_{(\xi)}, \omega_{(\eta)}, \omega_{(\xi)}\right]^{T} \tag{6.99}
\end{equation*}
\]

By substituting Eq. 6.94 in Eq. 6.90 , it is found that \({ }^{\dagger}\)
\[
\begin{align*}
\dot{\mathbf{S}}^{P} & =\tilde{\boldsymbol{\omega}} \mathbf{A \mathbf { s } ^ { \prime }} \\
& =\tilde{\boldsymbol{\omega}} \mathbf{S}^{P} \tag{6.100}
\end{align*}
\]

In vector form Eq. 6.100 is expressed as
\[
\vec{s}^{P}=\vec{\omega} \times \vec{s}^{P}
\]

For any vector \(\vec{s}\) attached to the \(\xi \eta \zeta\) coordinate system, like that in Fig. 6.8(a), Eq. 6.100 can be written as
\[
\begin{equation*}
\dot{\mathbf{s}}=\tilde{\omega} \mathbf{s} \tag{6.101}
\end{equation*}
\]

For a \(\xi \eta \zeta\) coordinate system that rotates and translates relative to the nonmoving \(x y z\) axes, the velocity of a point \(P\) that is fixed in the \(\xi \eta \zeta\) system can be determined. As shown in Fig. 6.9(b), we may employ a translating coordinate system such as \(x^{\prime} y^{\prime} z^{\prime}\) whose origin coincides with the origin of the \(\xi \eta \zeta\) coordinate axes. The \(\xi \eta \zeta\) system rotates relative to the \(x^{\prime} y^{\prime} z^{\prime}\) system, which only translates relative to the \(x y z\) system. Point \(P\) can be located in the \(x y z\) system by the relation
\[
\mathbf{r}^{P}=\mathbf{r}+\mathbf{s}^{P}
\]

The time derivative of this equation gives the velocity of point \(P\) as
\[
\begin{align*}
\dot{\mathbf{r}}^{P} & =\dot{\mathbf{r}}+\dot{\mathbf{s}}^{P} \\
& =\dot{\mathbf{r}}+\tilde{\boldsymbol{\omega}} \mathbf{s}^{P} \tag{6.102}
\end{align*}
\]

\subsection*{6.3.1 Time Derivatives of Euler Parameters}

In this section, identities between the time derivatives of Euler parameters and angular velocity vectors \(\boldsymbol{\omega}\) and \(\boldsymbol{\omega}^{\prime}\) are derived. These identities can be used for conversion from \(\boldsymbol{\omega}\) or \(\boldsymbol{\omega}^{\prime}\) to \(\dot{\mathbf{p}}\) and vice versa.

Postmultiplying Eq. 6.94 by \(\mathbf{A}^{T}\) yields
\[
\begin{equation*}
\dot{\mathbf{A}} \mathbf{A}^{T}=\tilde{\boldsymbol{\omega}} \tag{6.103}
\end{equation*}
\]
\({ }^{\dagger}\) By substituting Eq. 6.96 in Eq. 6.90 , it is found that \(\dot{s}^{P}=\mathbf{A} \bar{\omega}^{\prime} \mathbf{s}^{\prime P}\). The global and local components of vector \(\dot{s}^{P}\) are denoted by \(\dot{\mathbf{s}}^{P}\) and \((\mathbf{s})^{\prime P}\) where \((\dot{\mathbf{s}})^{\prime P}=\mathbf{A}(\dot{\mathbf{s}})^{\prime P}\), and thus \((\dot{\mathbf{s}})^{\prime P}=\overline{\boldsymbol{\omega}}^{\prime} \mathbf{s}^{\prime P}\). This equation is the same as Eq. 6.100 , but expressed in terms of the local components of the vectors. Note that \((\dot{\mathbf{s}})^{\prime P} \neq \dot{\mathbf{s}}^{\prime P}\). Vector \(\dot{\mathbf{s}}^{P}\) is defined as the time derivative of a constant vector \(\mathbf{s}^{\prime P}\), and so \(\dot{\mathbf{s}}^{\prime P}=\boldsymbol{0}\). However, \((\mathbf{s})^{\prime P}\) is defined as the local components of vector \(\vec{s}\), and if \(\vec{s} \neq \overrightarrow{0}\), then ( \(\vec{s})^{P P}\) can be nonzero.

From Eqs. 6.56 and 6.49 , Eq. 6.103 becomes \(2 \dot{\mathbf{G}} \mathbf{L}^{T} \mathbf{L G}{ }^{T}=\tilde{\boldsymbol{\omega}}\), which, upon application of Eqs. 6.46 and 6.40 , results in \(2 \dot{\mathbf{G}} \mathbf{G}^{T}=\boldsymbol{\omega}\). Finally, substituting Eqs. 6.59 and 6.57 into this last equation gives \(2 \widetilde{\mathbf{G} \mathbf{p}}=\tilde{\boldsymbol{\omega}}\), or
\[
\begin{equation*}
\omega=2 \mathbf{G} \dot{\mathbf{p}} \tag{6.104}
\end{equation*}
\]

In expanded form, Eq. 6.104 is
\[
\left[\begin{array}{c}
\omega_{(x)}  \tag{6.105}\\
\omega_{(y)} \\
\omega_{(z)}
\end{array}\right]=2\left[\begin{array}{rrrr}
-e_{1} & e_{0} & -e_{3} & e_{2} \\
-e_{2} & e_{3} & e_{0} & -e_{1} \\
-e_{3} & -e_{2} & e_{1} & e_{0}
\end{array}\right]\left[\begin{array}{l}
\dot{e}_{0} \\
\dot{e}_{1} \\
\dot{e}_{2} \\
\dot{e}_{3}
\end{array}\right]
\]

Premultiplying Eq. 6.104 by \(\mathbf{G}^{T}\) yields \(\mathbf{G}^{T} \boldsymbol{\omega}=2 \mathbf{G}^{T} \mathbf{G} \dot{\mathbf{p}}\), which, upon application of Eqs. 6.45 and 6.50 , results in the inverse transformation
\[
\begin{equation*}
\dot{\mathbf{p}}=\frac{1}{2} \mathbf{G}^{T} \boldsymbol{\omega} \tag{6.106}
\end{equation*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\boldsymbol{\omega}^{\prime}=2 \mathbf{L} \dot{\mathbf{p}} \tag{6.107}
\end{equation*}
\]

In expanded form, Eq. 6.107 is
\[
\left[\begin{array}{c}
\omega_{(\xi)}  \tag{6.108}\\
\omega_{(\eta)} \\
\omega_{(\xi)}
\end{array}\right]=2\left[\begin{array}{rrrr}
-e_{1} & e_{0} & e_{3} & -e_{2} \\
-e_{2} & -e_{3} & e_{0} & e_{1} \\
-e_{3} & e_{2} & -e_{1} & e_{0}
\end{array}\right]\left[\begin{array}{l}
\dot{e}_{0} \\
\dot{e}_{1} \\
\dot{e}_{2} \\
\dot{e}_{3}
\end{array}\right]
\]
\({ }^{\prime}\) The inverse transformation of Eq. 6.107 is found to be
\[
\begin{equation*}
\dot{\mathbf{p}}=\frac{1}{2} \mathbf{L}^{T} \boldsymbol{\omega}^{\prime} \tag{6.109}
\end{equation*}
\]

Differentiating Eq. 6.104 with respect to time yields \(\dot{\boldsymbol{\omega}}=2 \mathbf{G} \ddot{\mathbf{p}}+2 \dot{\mathbf{G}} \mathbf{p}\), which, upon application of Eq. 6.53, becomes
\[
\begin{equation*}
\dot{\omega}=2 \mathbf{G} \ddot{\mathbf{p}} \tag{6.110}
\end{equation*}
\]

Similarly, differentiating Eq. 6.107 with respect to time and using Eq. 6.54 results in
\[
\begin{equation*}
\dot{\boldsymbol{\omega}}^{\prime}=2 \mathbf{L} \ddot{\boldsymbol{p}} \tag{6.111}
\end{equation*}
\]

Vectors \(\dot{\omega}\) and \(\dot{\omega}^{\prime}\) are the global and local components of a vector \(\overrightarrow{\dot{\omega}}\) defined as the angular acceleration of the \(\xi \eta \zeta\) coordinate system. It can be shown that the inverses of Eqs. 6.110 and 6.111 are
\[
\begin{equation*}
\ddot{\mathbf{p}}=\frac{1}{2} \mathbf{G}^{T} \dot{\boldsymbol{\omega}}-\frac{1}{4}\left(\boldsymbol{\omega}^{T} \boldsymbol{\omega}\right) \mathbf{p} \tag{6.112}
\end{equation*}
\]
and
\[
\begin{equation*}
\ddot{\mathbf{p}}=\frac{1}{2} \mathbf{L}^{T} \dot{\boldsymbol{\omega}}^{\prime}-\frac{1}{4}\left(\boldsymbol{\omega}^{\prime T} \boldsymbol{\omega}^{\prime}\right) \mathbf{p} \tag{6.113}
\end{equation*}
\]

It is clear that \(\omega^{T} \omega=\omega^{\prime T} \omega^{\prime}=\omega^{2}\), where \(\omega\) is the magnitude of \(\vec{\omega}\). Furthermore, it can be shown that the scalar product \(\omega^{T} \omega-\omega^{2}=0\) yields
\[
\begin{equation*}
4 \dot{\mathbf{p}}^{T} \dot{\mathbf{p}}-\omega^{2}=0 \tag{6.114}
\end{equation*}
\]

\subsection*{6.4 SEMIROTATING COORIDINATE SYSTEMS}

The concept of Euler parameters as rotational coordinates may appear, to the uninitiated reader, as a mathematical tool without any physical meaning. However, careful study of these parameters will prove the contrary. Physical interpretation of Euler parameters is simple and is more natural to implement than any other set of rotational coordinates, such as Euler or Bryant angles.

The angular orientation of one coordinate system relative to another can be looked upon by Euler's theorem as the result of a single rotation about an orientational axis of rotation by an angle \(\phi\). A viewer may observe a rotation in different ways; three cases are considered here.

Case 1. Consider an observer standing along the axis of rotation in the global \(x y z\) system. If the \(x y z\) and \(\xi \eta \zeta\) coordinates are initially coincident, then as the \(\xi \eta \zeta\) system finds its orientation, it will have rotated by an angle \(\phi\) as seen by the observer. A positive rotation may be seen by the observer as a clockwise rotation of \(\xi \eta \zeta\) about \(\vec{u}\).

Case 2: The observer is in the \(\xi \eta \zeta\) coordinate system. In this case the rotation described in case 1 will be viewed as a counterclockwise rotation of the \(x y z\) system by an angle \(\phi\) about \(\vec{u}\).

Case 3: The observer is in a semirotating coordinate system designated \(\alpha \beta \gamma\). In this case the same rotation will be viewed as a clockwise rotation of the \(\xi \eta \zeta\) system about \(\vec{u}\) by an angle \(\phi / 2\) and a simultaneous counterclockwise rotation of the \(x y z\) system by an angle \(\phi / 2\).

The three cases are illustrated in Fig. 6.10(a-c) for the special case of a planar system. It is assumed that the axis of rotation is outside the plane, along the \(z\) (or \(\zeta\) or \(\gamma\) ) axis. The same example for the general case of a spatial system is illustrated in Fig. 6.10(d--f).

Equation 6.49 states that the transformation matrix \(\mathbf{A}\) is the result of two successive transformations; i.e., \(\mathbf{A}\) can be expressed as the product of two \(3 \times 4\) matrices \(\mathbf{G}\) and \(\mathbf{L}\) as
\[
\mathbf{A}=\mathbf{G} \mathbf{L}^{T}
\]

The components of a vector \(\vec{s}\) are transformed from the \(\xi \eta \zeta\) coordinate system to the \(x y z\) coordinate system as follows:
\[
\mathbf{s}=\mathbf{A} \mathbf{s}^{\prime}
\]

This process can be performed in two steps:
\[
\begin{gathered}
\mathbf{s}^{*}=\mathbf{L}^{T} \mathbf{s}^{\prime} \\
\mathbf{s}=\mathbf{G} \mathbf{S}^{*}
\end{gathered}
\]
where \(\mathbf{s}^{*}\) is a 4-vector. Matrix \(\mathbf{L}^{T}\) can be interpreted as transforming \(\mathbf{s}^{\prime}\) from the \(\xi \eta \zeta\) coordinates by a semirotation to an intermediate 4 -space semirotating coordinate system, instead of the 3 -space semi-rotating \(\alpha \beta \gamma\) system. Hence, \(\mathbf{s}^{*}\) is transformed from the 4 -, space semirotating system to the \(x y z\) system by a second semirotation through matrix \(\mathbf{G}\).

The transformation matrices \(\mathbf{G}\) and \(\mathbf{L}\) are linear in terms of the Euler parameters. The linearity of \(\mathbf{G}\) and \(\mathbf{L}\) is due to the fact that they perform a coordinate transformation between the local and global systems via a four-dimensional semirotating coordinate system. However, if the semirotating coordinate system is defined in a three-dimen-


Figure 6.10 Observer's interpretation of angle and orientational axis of rotation as seen from a point on the orientational axis of rotation and in a fixed orientation relative to ( \(a, \mathrm{~d}\) ) the xyz axes, (b, e) the \(\xi \eta \zeta\) axes, and ( \(\mathrm{c}, \mathrm{f}\) ) the \(\alpha \beta \gamma\) axes.
sional space, its corresponding semirotational transformation matrices will be nonlinear in terms of the rotational coordinates.

\subsection*{6.5 RELATIVE AXIS OF ROTATION}

The Euler parameters defined in Sec. 6.1.3 describe the angular orientation of a bodyfixed coordinate system with respect to a global coordinate system. It may be advantageous to describe the orientation of a body-fixed coordinate system relative to another body-fixed coordinate system. In doing so, we need to find an axis about which one of the coordinate systems may be rotated by some angle to become parallel to the other coordinate system.

Assume that a \(\xi_{i} \eta_{i} \zeta_{i}\) coordinate system with respect to the global \(x y z\) coordinate system is described by Euler parameters \(\mathbf{p}_{i}\) and transformation matrix \(\mathbf{A}_{i}\). Similarly, assume that the orientation of the \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate system with respect to the global coordinate system is described by Euler parameters \(\mathbf{p}_{j}\) and transformation matrix \(\mathbf{A}_{j}\). A vector \(\vec{s}\) with \(\mathbf{s}_{i}^{\prime}\) components in the \(\xi_{i} \eta_{i} \zeta_{i}\) system has global components
\[
\begin{equation*}
\mathbf{s}=\mathbf{A}_{i} \mathbf{s}_{i}^{\prime} \tag{a}
\end{equation*}
\]

The global components of vector \(\vec{s}\), i.e., s, can be transformed in terms of the \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate system as follows:
\[
\begin{equation*}
\mathbf{s}_{j}^{\prime}=\mathbf{A}_{j}^{T} \mathbf{s} \tag{b}
\end{equation*}
\]

Substitution of Eq. \(a\) into Eq. \(b\) yields
\[
\begin{equation*}
\mathbf{s}_{j}^{\prime}=\mathbf{A}_{j}^{T} \mathbf{A}_{i} \mathbf{s}_{i}^{\prime} \tag{c}
\end{equation*}
\]

Equation \(c\) may be written as
\[
\begin{equation*}
\mathbf{s}_{j}^{\prime}=\mathbf{A}_{i j} \mathbf{s}_{i}^{\prime} \tag{6.115}
\end{equation*}
\]
where
\[
\begin{equation*}
\mathbf{A}_{i j}=\mathbf{A}_{j}^{T} \mathbf{A}_{i} \tag{6.116}
\end{equation*}
\]

The product \(\mathbf{A}_{j}^{T} \mathbf{A}_{i}\) or \(\mathbf{A}_{i j}\) is the transformation matrix from \(\xi_{i} \eta_{i} \zeta_{i}\) coordinates to \(\xi_{j} \boldsymbol{\eta}_{j} \zeta_{j}\) coordinates. At this point, the objective is to find a set of Euler parameters
\[
\begin{equation*}
\mathbf{p}_{i j}=\left[e_{0}, \mathbf{e}^{T}\right]_{i j}^{T}=\left[e_{0}, e_{1}, e_{2}, e_{3}\right]_{i j}^{T} \tag{6.117}
\end{equation*}
\]
that define the matrix \(\mathbf{A}_{i j}\) in terms of Euler parameters \(\mathbf{p}_{i}\) and \(\mathbf{p}_{j}\).
Before attempting to find Euler parameters \(\mathbf{p}_{i j}\), we present two identities. The product \(\mathbf{L}_{i} \mathbf{p}_{j}\) is expanded as follows:
\[
\begin{align*}
\mathbf{L}_{i} \mathbf{p}_{j} & =\left[-\mathbf{e}_{i},-\tilde{\mathbf{e}}_{i}+e_{0 i} \mathbf{I}\right]\left[\begin{array}{l}
e_{0 j} \\
e_{j}
\end{array}\right] \\
& =\left[-e_{0 j} \mathbf{e}_{i}-\tilde{\mathbf{e}}_{i} \mathbf{e}_{j}+e_{0 i} \mathbf{e}_{j}\right] \\
& =\left[e_{0 i} \mathbf{e}_{j}+\tilde{\mathbf{e}}_{j} \mathbf{e}_{i}-e_{0 j} \mathbf{e}_{i}\right] \\
& =\left[\mathbf{e}_{j}, \tilde{\mathbf{e}}_{j}-e_{0 j} \mathbf{I}\right]\left[\begin{array}{c}
e_{0 i} \\
\mathbf{e}_{i}
\end{array}\right] \\
& =-\mathbf{L}_{j} \mathbf{p}_{i} \tag{6.118}
\end{align*}
\]

Similarly, it can be shown that
\[
\begin{equation*}
\mathbf{G}_{i} \mathbf{p}_{j}=-\mathbf{G}_{j} \mathbf{p}_{i} \tag{6.119}
\end{equation*}
\]

Now if \(\mathbf{A}_{i}\) in Eq. 6.116 is replaced by \(\mathbf{G}_{i} \mathbf{L}_{i}^{T}\) from Eq. 6.49, then postmultiplying by \(\mathbf{L}_{i}\) yields
\[
\begin{align*}
\mathbf{A}_{i j} \mathbf{L}_{i} & =\mathbf{A}_{j}^{T} \mathbf{G}_{i} \mathbf{L}_{i}^{T} \mathbf{L}_{i} \\
& =\mathbf{A}_{j}^{T} \mathbf{G}_{i}\left(-\mathbf{p}_{i} \mathbf{p}_{i}^{T}+\mathbf{I}^{*}\right) \\
& =\mathbf{A}_{j}^{T} \mathbf{G}_{i} \tag{d}
\end{align*}
\]
where Eqs. 6.45 and 6.40 have been used. Postmultiplying Eq. \(d\) by \(\mathbf{p}_{j}\) yields
\[
\begin{aligned}
\mathbf{A}_{i j} \mathbf{L}_{i} \mathbf{p}_{j} & =\mathbf{A}_{j}^{T} \mathbf{G}_{i} \mathbf{p}_{j} \\
& =-\mathbf{L}_{j} \mathbf{G}_{j}^{T} \mathbf{G}_{j} \mathbf{p}_{i} \\
& =-\mathbf{L}_{j}\left(-\mathbf{p}_{j} \mathbf{p}_{j}^{T}+\mathbf{I}^{*}\right) \mathbf{p}_{i} \\
& =-\mathbf{L}_{j} \mathbf{p}_{i}
\end{aligned}
\]
or
\[
\begin{equation*}
\mathbf{A}_{i j} \mathbf{L}_{j} \mathbf{p}_{i}=\mathbf{L}_{j} \mathbf{p}_{i} \tag{e}
\end{equation*}
\]
where Eqs. \(6.49,6.119,6.45,6.41\), and 6.118 have been employed, in that order. Equation \(e\) may be rewritten as
\[
\begin{equation*}
\mathbf{A}_{i j} \mathbf{b}=\mathbf{b} \tag{f}
\end{equation*}
\]
where
\[
\begin{equation*}
\mathbf{b}=\mathbf{L}_{j} \mathbf{p}_{i} \tag{g}
\end{equation*}
\]

Equation \(g\) shows that the transformation matrix \(\mathbf{A}_{i j}\) does not change the components of vector \(\vec{b}\). Therefore, vector \(\vec{b}\) must be located along a relative orientational axis of rotation between the \(\xi_{i} \eta_{i} \zeta_{i}\) and \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate systems. Since vector \(\vec{e}_{i j}\) also lies along the same relative orientational axis of rotation, \(\vec{e}_{i j}\) and \(\vec{b}\) must be collinear.

According to Eq. 6.25, the Euler parameter \(e_{0 i j}\) can be evaluated from
\[
\begin{equation*}
e_{0 i j}^{2}=\frac{\operatorname{tr} \mathbf{A}_{i j}+1}{4} \tag{h}
\end{equation*}
\]

The trace of matrix \(\mathbf{A}_{i j}\) can be found by substituting the elements of matrices \(\mathbf{A}_{i}\) and \(\mathbf{A}_{j}\) from Eq. 6.19 into Eq. 6.116. If the matrix product is carried out and the trace of the resultant \(3 \times 3\) matrix is formed and simplified, \({ }^{\dagger}\) the trace of \(\mathbf{A}_{i j}\) is found to be
\[
\begin{equation*}
\operatorname{tr} \mathbf{A}_{i j}=4\left(\mathbf{p}_{j}^{T} \mathbf{p}_{i}\right)^{2}-1 \tag{i}
\end{equation*}
\]

Substitution of Eq. \(i\) into Eq. \(h\) yields
\[
e_{0 i j}^{2}=\left(\mathbf{p}_{j}^{T} \mathbf{p}_{i}\right)^{2}
\]
or
\[
\begin{equation*}
e_{0 i j}=\mathbf{p}_{j}^{T} \mathbf{p}_{i} \tag{6.120}
\end{equation*}
\]
where, according to the discussion of Sec. 6.1.4, the positive sign is chosen.
Calculating the sum of the squares of \(e_{0 i j}\) and the components of vector \(\mathbf{b}\) reveals that
\[
\begin{align*}
e_{0 i j}^{2}+\mathbf{b}^{T} \mathbf{b} & =\left(\mathbf{p}_{j}^{T} \mathbf{p}_{i}\right)^{2}+\mathbf{p}_{i}^{T} \mathbf{L}_{j}^{T} \mathbf{L}_{j} \mathbf{p}_{i} \\
& =\left(\mathbf{p}_{j}^{T} \mathbf{p}_{i}\right)^{2}+\mathbf{p}_{i}^{T}\left(-\mathbf{p}_{j} \mathbf{p}_{j}^{T}+\mathbf{I}^{*}\right) \mathbf{p}_{i} \\
& =\left(\mathbf{p}_{j}^{T} \mathbf{p}_{i}\right)^{2}-\left(\mathbf{p}_{j}^{T} \mathbf{p}_{i}\right)^{2}+\mathbf{p}_{i}^{T} \mathbf{p}_{i} \\
& =1 \tag{j}
\end{align*}
\]
where Eqs. \(6.46,2.41\), and 6.23 have been used. Since it is already known that \(\mathbf{e}_{i j}\) and \(\mathbf{b}\) are parallel, then a comparison of Eq. \(j\) and Eq. 6.21 indicates that \(\mathbf{e}_{i j}=\mathbf{b}\), or
\[
\begin{equation*}
\mathbf{e}_{i j}=\mathbf{L}_{j} \mathbf{p}_{i} \tag{6.121}
\end{equation*}
\]

Hence, the Euler parameters \(\mathbf{p}_{i j}\) are
\[
\mathbf{p}_{i j}=\left[\begin{array}{l}
\boldsymbol{e}_{0} \\
\mathbf{e}
\end{array}\right]_{i j}=\left[\begin{array}{c}
\mathbf{p}_{j}^{T} \mathbf{p}_{i} \\
\mathbf{L}_{j} \mathbf{p}_{i}
\end{array}\right]=\left[\begin{array}{l}
\mathbf{p}^{T} \\
\mathbf{L}
\end{array}\right]_{j} \mathbf{p}_{i}
\]
or
\[
\begin{equation*}
\mathbf{p}_{i j}=\mathbf{L}_{j}^{*} \mathbf{p}_{i} \tag{6.122}
\end{equation*}
\]
where \(\mathbf{L}_{j}^{*}\) is a \(4 \times 4\) matrix defined as
\[
\mathbf{L}_{j}^{*}=\left[\begin{array}{l}
\mathbf{p}^{T}  \tag{6.123}\\
\mathbf{L}
\end{array}\right]_{j}
\]

Equation 6.122 shows that if the Euler parameters describing the orientations of two bodies with respect to a global coordinate system are known, then the Euler parameters describing the orientation of one of the bodies with respect to the other can be determined.

Two more identities are stated here that can be verified easily:
\[
\begin{equation*}
\mathbf{L}_{i} \dot{\mathbf{p}}_{j}=-\dot{\mathbf{L}}_{j} \mathbf{p}_{i} \tag{6.124}
\end{equation*}
\]
\({ }^{\dagger}\) Since the calculation of \(\operatorname{tr} \mathbf{A}_{i j}\) is too extensive to be listed in detail, only the final result is presented.
and
\[
\begin{equation*}
\mathbf{G}_{i} \dot{\mathbf{p}}_{j}=-\dot{\mathbf{G}}_{j} \mathbf{p}_{i} \tag{6.125}
\end{equation*}
\]

These identities relate the Euler parameters of one body and the time derivative of the Euler parameters of another body.

\subsection*{6.5.1 Intermediate Axis of Rotation}

In Section 6.5, a relative orientational axis of rotation between \(\xi_{i} \eta_{i} \zeta_{i}\) and \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate systems was found when the Euler parameters that describe the orientation of the two systems with respect to the global coordinate system were known. This method can be stated in another form, but identical in principle, as follows:

Find an intermediate orientational axis of rotation about which a body-fixed coordinate system at time \(\mathrm{t}^{\mathrm{k}}\) can be rotated to become parallel to the coordinate system describing the orientation of the same body at time \(\mathrm{t}^{\mathrm{l}}\).

If the coordinate system of body \(i\) at times \(t^{k}\) and \(t^{l}\) is denoted by \(\xi_{i} \eta_{i} \zeta_{i}^{k}\) and \(\xi_{i} \eta_{i} \zeta_{i}^{l}\), the Euler parameters of the body at \(t^{k}\) and \(t^{\prime}\) are denoted by \(\mathbf{p}_{i}^{k}\) and \(\mathbf{p}_{i}^{l}\), respectively. Similarly, the intermediate set of Euler parameters between orientations at \(t^{k}\) and \(t^{l}\), which is denoted by \(\mathbf{p}_{i}^{k l}\), can be written from Eq. 6.122 as
\[
\begin{equation*}
\mathbf{p}_{i}^{k l}=\mathbf{L}_{i}^{* l} \mathbf{p}_{i}^{k} \tag{6.126}
\end{equation*}
\]
where
\[
\mathbf{L}_{i}^{* l}=\left[\begin{array}{l}
\mathbf{p}^{T}  \tag{6.127}\\
\mathbf{L}
\end{array}\right]_{i}^{\prime}
\]

\subsection*{6.6 FINITE ROTATION}

Consider the two bodies \(i\) and \(j\) shown in Fig. 6.11 (these may also be interpreted as two different configurations of the same body). The translational vectors for the two bodies are \(\mathbf{r}_{i}\) and \(\mathbf{r}_{j}\), and the translational vector between the two bodies is denoted by \(\mathbf{r}_{i j}\). It is clear that
\[
\begin{aligned}
\mathbf{r}_{j} & =\mathbf{r}_{i}+\mathbf{r}_{j i} \\
& =\mathbf{r}_{j i}+\mathbf{r}_{i}
\end{aligned}
\]


Figure 6.11 Two bodies with different translational and rotational configurations.

This indicates that translational vectors follow the commutative law of vector summation, and therefore a translational vector is a true vector quantity. However, it will be seen that this is not a characteristic of the rotation of a body. If the transformation matrices for the two bodies are \(\mathbf{A}_{i}\) and \(\mathbf{A}_{j}\) and the transformation matrix of body \(j\) with respect to body \(i\) is \(\mathbf{A}_{j i}\), then from Eq. 6.116 (by reversing the indices) it is found that \(\mathbf{A}_{j i}=\mathbf{A}_{i}^{T} \mathbf{A}_{j}\), or
\[
\begin{align*}
\mathbf{A}_{j} & =\mathbf{A}_{i} \mathbf{A}_{j i} \\
& \neq \mathbf{A}_{j i} \mathbf{A}_{i} \tag{a}
\end{align*}
\]

This is obvious, since matrix multiplication is not commutative. This means that in two successive rotations, the order of rotations cannot be reversed. Assume that a finite rotation is denoted by a rotational vector in the direction of the orientational axis of rotation, having a magnitude proportional to the angle of rotation, e.g., \(\mathbf{e}=\mathbf{u} \sin \phi / 2\). Then three rotational vectors \(\mathbf{e}_{i}, \mathbf{e}_{j}\), and \(\mathbf{e}_{j i}\) can be defined and it can be deduced that
\[
\mathbf{e}_{j} \neq \mathbf{e}_{i}+\mathbf{e}_{j i}
\]

In contrast to the rotational vector of a finite rotation, the angular velocity vector is a true vector quantity. The time derivative of Eq. \(a\) is
\[
\begin{equation*}
\dot{\mathbf{A}}_{j}=\dot{\mathbf{A}}_{i} \mathbf{A}_{j i}+\mathbf{A}_{i} \dot{\mathbf{A}}_{j i} \tag{b}
\end{equation*}
\]

From Eq. 6.94, Eq. \(b\) becomes
\[
\begin{equation*}
\tilde{\boldsymbol{\omega}}_{j} \mathbf{A}_{j}=\tilde{\boldsymbol{\omega}}_{i} \mathbf{A}_{i} \mathbf{A}_{j i}+\mathbf{A}_{i} \tilde{\boldsymbol{\omega}}_{j i}^{\prime} \mathbf{A}_{j i} \tag{c}
\end{equation*}
\]

Note that \(\tilde{\boldsymbol{\omega}}_{j i}^{\prime}\) represents the components of \(\vec{\omega}_{j i}\) with respect to the \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate system. Substituting Eq. \(a\) in Eq. \(c\) and simplifying the result yields
\[
\begin{aligned}
\tilde{\boldsymbol{\omega}}_{j} & =\tilde{\boldsymbol{\omega}}_{i}+\mathbf{A}_{i} \tilde{\boldsymbol{\omega}}_{j i}^{\prime} \mathbf{A}_{i}^{T} \\
& =\tilde{\boldsymbol{\omega}}_{i}+\tilde{\boldsymbol{\omega}}_{j i}
\end{aligned}
\]

Therefore,
\[
\omega_{j}=\omega_{i}+\omega_{j i}
\]

This is the proof that the angular velocity is a true vector quantity.

\section*{PROBLEMS}
6.1 Three vectors \(\vec{a}, \vec{b}\), and \(\vec{c}\) are defined along the positive \(\xi\) axis, \(\eta\) axis, and \(\zeta\) axis, respectively. The global components of these vectors are
\[
\mathbf{a}=\left[\begin{array}{r}
0.0776 \\
-1.8833 \\
-0.6685
\end{array}\right] \quad \mathbf{b}=\left[\begin{array}{r}
0.6410 \\
1.0038 \\
-2.7535
\end{array}\right] \quad \mathbf{c}=\left[\begin{array}{r}
1.4642 \\
-0.0537 \\
0.3213
\end{array}\right]
\]
(a) Test these three vectors for orthogonality.
(b) Determine the global components of the three unit vectors \(\mathbf{u}_{(\xi)}, \mathbf{u}_{(\eta)}\), and \(\mathbf{u}_{(\zeta)}\) along the \(\xi \eta \zeta\) axes.
(c) Determine the nine direction cosines of matrix \(\mathbf{A}\).
6.2 Using Eq. 6.3, find six constraint equations between the nine direction cosines.
6.3 Two vectors \(\vec{a}\) and \(\vec{b}\) are defined along the positive \(\xi\) and \(\eta\) axes, respectively. The global components of these vectors are
\[
\mathbf{a}=\left[\begin{array}{l}
0.1107 \\
0.3924 \\
1.1286
\end{array}\right] \quad \mathbf{b}=\left[\begin{array}{r}
-1.9450 \\
1.5330 \\
-0.3422
\end{array}\right]
\]
(a) Test these two vectors for orthogonality.
(b) Determine the global components of the three unit vectors \(\mathbf{u}_{(\xi)}, \mathbf{u}_{(\eta)}\), and \(\mathbf{u}_{(\xi)}\) along the \(\xi \eta \zeta\) axes.
(c) Determine the clements of matrix \(\mathbf{A}\).
6.4 Two vectors \(\vec{a}\) and \(\vec{c}\) are defined along the positive \(\xi\) and \(\zeta\) axes, respectively. The global components of these vectors are
\[
\mathbf{a}=\left[\begin{array}{r}
0.6438 \\
2.3930 \\
-1.6909
\end{array}\right] \quad \mathbf{c}=\left[\begin{array}{r}
-0.7796 \\
-0.2077 \\
-0.5908
\end{array}\right]
\]

Determine the elements of matrix \(A\).
6.5 A vector \(\vec{a}\) along the positive \(\xi\) axis and a vector \(\vec{d}\) on the \(\xi \eta\) plane have the following components:
\[
\mathbf{a}=\left[\begin{array}{r}
-1.0 \\
1.2 \\
0.5
\end{array}\right] \quad \mathbf{d}=\left[\begin{array}{r}
1.3 \\
-0.6 \\
0.8
\end{array}\right]
\]
(a) Determine the three unit vectors along the \(\xi \eta \zeta\) axes.
(b) Find the elements of matrix \(\mathbf{A}\).
(c) Is the solution to this problem unique?
6.6 Determine the four Euler parameters for the transformation matrices \(\mathbf{A}\) in
(a) Prob. 6.1
(b) Prob. 6.3
(c) Prob. 6.4
(d) Prob. 6.5
6.7 Determine the four Euler parameters for the transformation matrices
(a)
\[
\mathbf{A}=\left[\begin{array}{rrr}
-0.4590 & 0.8376 & -0.2962 \\
0.4908 & 0.5170 & 0.7014 \\
0.7406 & 0.1766 & -0.6483
\end{array}\right]
\]
(b)
\[
\mathbf{A}=\left[\begin{array}{rrr}
-0.4590 & 0.4908 & 0.7406 \\
0.8376 & 0.5170 & 0.1766 \\
-0.2962 & 0.7014 & -0.6483
\end{array}\right]
\]

Compare the results of parts (a) and (b). What do you conclude?
6.8 A transformation matrix \(\mathbf{A}\) is given as follows:
\[
\mathbf{A}=\left[\begin{array}{rrr}
0.0319 & -0.8506 & 0.5249 \\
-0.8506 & -0.2988 & -0.4327 \\
0.5249 & -0.4327 & -0.7330
\end{array}\right]
\]
(a) Test matrix \(\mathbf{A}\) for orthogonality, using the identity \(\mathbf{A}^{T} \mathbf{A}=\mathbf{I}\).
(b) Determine the four Euler parameters for matrix \(\mathbf{A}\).
6.9 By using Eq. 6.18, show that if the signs of all four Euler parameters are reversed, i.e., if \(\mathbf{p} \rightarrow-\mathbf{p}\), then the transformation matrix \(\mathbf{A}\) is not affected.
6.10 If the angular orientation of a body-fixed coordinate system is described in terms of three Euler angles \(\psi=40^{\circ}, \theta=30^{\circ}\), and \(\sigma=-25^{\circ}\), find its corresponding set of Euler parameters.
6.11 Determine the global coordinates of the point \([2,6,8]^{T}\) in a rotating \(\xi \eta \zeta\) system, where the Euler angles are \(\psi=45^{\circ}, \theta=45^{\circ}\), and \(\sigma=30^{\circ}\) and the origins of the two coordinate systems coincide.
6.12 Determine the coordinates of a point in a rotating \(\xi \eta \zeta\) system if its global coordinates are \([3,3,1]^{T}\). The Euler angles are \(\psi=30^{\circ}, \theta=30^{\circ}\), and \(\sigma=60^{\circ}\) and the origins of the two systems coincide.
6.13 A sequence of two rotations is required to uniquely locate the longitudinal axis of a vehicle. Consider the sequence \(\psi, \theta\) as shown in Fig. P.6.13. The first rotation is a positive rotation about the \(x\) axis through an angle \(\psi\); the second, a positive rotation about the \(\zeta^{\prime}\) axis through an angle \(\theta\).
(a) Determine the elements of a transformation matrix \(\mathbf{A}\).
(b) Test A for orthogonality.
6.14 Find the Euler angles describing the rotation shown in Fig. P.6. 14 for \(\varepsilon=30^{\circ}\).


Figure P.6.13


Figure P.6.14
6.15 Consider the spinning top shown in Fig. P.6.15. Assume Euler angle rates of \(\dot{\psi}=2 \mathrm{rad} / \mathrm{s}\), \(\dot{\theta}=0\), and \(\dot{\sigma}=125 \mathrm{rad} / \mathrm{s}\) when \(\psi=120^{\circ}, \theta=30^{\circ}\), and \(\sigma=90^{\circ}\).
(a) Determine the corresponding values of \(\omega_{(\xi)}, \omega_{(\eta)}\), and \(\omega_{(\xi)}\).
(b) For what values of \(\theta\) would the inverse of the transformation in (a) be nonexistent?


Figure P.6.15
6.16 For the spinning top of Prob. 6.15, assume angular rate components of \(\omega_{(\xi)}=0\), \(\omega_{(\eta)}=2 \mathrm{rad} / \mathrm{s}\), and \(\omega_{(\xi)}=20 \mathrm{rad} / \mathrm{s}\) when \(\psi=60^{\circ}, \theta=30^{\circ}\), and \(\sigma=120^{\circ}\).
(a) Determine \(\omega_{(x)}\), \(\omega_{(y)}\), and \(\omega_{(z)}\).
(b) Determine \(\dot{\psi}, \dot{\theta}\), and \(\dot{\sigma}\).
6.17 Find expressions to convert Bryant angles into equivalent Euler parameters.
6.18 Find expressions to convert Euler parameters to Bryant angles.
6.19 Show that the rotation angle \(\phi\) and the components of the unit vector \(\mathbf{u}\) along the orientational axis of rotation are determined by the equations
\[
\begin{gathered}
\cos \phi=\frac{\operatorname{tr} \mathbf{A}-1}{2} \\
u_{i}^{2}=\frac{a_{i i}-\cos \phi}{1-\cos \phi} \quad i=1,2,3
\end{gathered}
\]
where \(a_{i i}\) is an element of \(\mathbf{A}\).
6.20 Point \(P\) is located from the origin of a coordinate system by a vector having components \([3,4,12]^{T}\). The vector is subjected to a rotation of \(30^{\circ}\) positive about an axis passing through the origin and a point \([2,-3,2]^{T}\). Find the components of the vector in its new orientation.
6.21 The Euler parameters describing the orientation of a body-fixed coordinate system with respect to a global coordinate system are \(e_{0}=e_{3}=0.6533, e_{1}=0.3827\), and \(e_{2}=0\). Find the three Euler angles describing this orientation.
6.22 Four Euler parameters \(\mathbf{p}=[0.8,0,0,0.6]^{T}\) describe the angular orientation of the \(\xi \eta \zeta\) axes with respect to the \(x y z\) axes. Determine the equivalent set of Euler angles for this orientation. What do you conclude?
6.23 Points \(D\) and \(F\) are located on the \(\eta\) and \(\zeta\) axes, respectively, and have local coordinates \(\mathbf{s}^{\prime D}=[0,2,0]^{T}\) and \(\mathbf{s}^{\prime F}=[0,0,1]^{T}\). The global components of vectors \(\vec{s}^{D}\) and \(\vec{s}^{F}\) are \(s^{D}=[-1.1098,1.6598,-0.1134]^{T}\) and \(s^{F}=[0.6285,0.4629,0.6249]^{T}\). Find the elements of matrix A describing this orientation.
6.24 Verify Eq. 6.41.
6.25 Verify Eq. 6.42 (or 6.43 ),
(a) Using the expanded form of matrix \(\mathbf{G}\) (or \(\mathbf{L}\) )
(b) Using the compact form of matrix \(\mathbf{G}\) (or \(\mathbf{L}\) )
6.26 Verify the following identities:
(a) Eq. 6.46
(b) Eq. 6.54
(c) Eq. 6.55
(d) Eq. 6.58
(e) Eq. 6.59
(f) Eq. 6.60
6.27 Verify the following identities:
(a) Eq. 6.69
(b) Eq. 6.71
(c) Eq. 6.73
(d) Eq. 6.80
(e) Eqs. 6.84 and 6.85
(f) Eq. 6.87
6.28 Start with Eq. 6.96 and obtain Eq. 6.107.
6.29 Show that the inverse transformation of Eq. 6.109 is valid.
6.30 Verify the following identities:
(a) Eq. 6.112
(b) Eq. 6.113
(c) Eq. 6.119
6.31 Show that \(\operatorname{tr} \mathbf{A}_{i j}=4\left(\mathbf{p}_{j}^{T} \mathbf{p}_{i}\right)^{2}-1\) by determining the diagonal elements of the matrix product \(\mathbf{A}_{j}^{T} \mathbf{A}_{i}\).
6.32 Derive the inverse transformation of Eq. 6.122; i.e., calculate \(\mathbf{p}_{i}\) when \(\mathbf{p}_{j}\) and \(\mathbf{p}_{i j}\) are known (Hint: Start with Eq. 6.121).

\section*{Spatial Kinematics}

This chapter derives the spatial kinematic constraint equations for several standard kinematic pairs. Those for other standard pairs, or for special-purpose kinematic pairs, may be formulated similarly. Euler parameters are used to define the angular orientation of bodies. The methodology will remain the same if other sets of rotational coordinates are used. However, the quadratic nature of the transformation matrix, the absence of trigonometric functions, and the singularity-free aspect of the Euler parameters make them more attractive than other sets of rotational coordinates. Another advantage of Euler parameter formulation is that it allows kinematic relationships for different pairs to be written in compact matrix form, so that compact and efficient computational algorithms can be developed.

\subsection*{7.1 RELATIVE CONSTRAINTS BETWEEN TWO VECTORS}

In this section algebraic relations between two vectors are derived to provide the basis for subsequent constraint equation formulation. Most kinematic constraints require that two vectors remain parallel or perpendicular. A vector may have fixed length, e.g., if it connects points that are fixed in the same body; or it may have variable length, if, for example, it connects points that are fixed in different bodies.

In constraint equation formulation, it is necessary to express the components of all vectors in the same coordinate system, the most natural being the global coordinate system. The global components of a vector that is fixed in a body may be obtained from the vector's local components or they may be obtained from the global coordinates of its endpoints.

Vector \(\vec{s}_{i}\) in Fig. 7.1 is fixed in body \(i\). Thus, its magnitude is constant and its orientation relative to the \(\xi_{i} \eta_{i} \zeta_{i}\) axes does not change. The global components of \(\vec{s}_{i}\) can be obtained from any of the following:
\[
\begin{align*}
\mathbf{s}_{i} & =\mathbf{s}_{i}^{B}-\mathbf{s}_{i}^{C} \\
& =\mathbf{A}_{i} \mathbf{s}_{i}^{\prime B}-\mathbf{A}_{i} \mathbf{s}_{i}^{C} \\
& =\mathbf{A}_{i}\left(\mathbf{s}_{i}^{\prime B}-\mathbf{s}_{i}^{\prime C}\right) \tag{7.1}
\end{align*}
\]
where \(\mathbf{s}_{i}^{\prime B}=\left[\xi^{B}, \eta^{B}, \zeta^{B}\right]_{i}^{T}\) and \(\mathbf{s}_{i}^{\prime C}=\left[\xi^{C}, \eta^{C}, \zeta^{C}\right]_{i}^{T}\) are known constant quantities. It can be observed from Eq. 7.1 that the global components of a vector that is fixed in body \(i\) do not depend on the global location of the body, i.e., on the vector \(\mathbf{r}_{i}\).

When a vector connects two points on bodies \(i\) and \(j\), such as vector \(\vec{d}\) in Fig. 7.1, its global components are written as
\[
\begin{align*}
& \mathbf{d}=\left(\mathbf{r}_{j}+\mathbf{s}_{j}^{B}\right)-\left(\mathbf{r}_{i}+\mathbf{s}_{i}^{B}\right) \\
& =\mathbf{r}_{j}+\mathbf{A}_{j} \mathbf{s}_{j}^{\prime B}-\mathbf{r}_{i}-\mathbf{A}_{i} \mathbf{s}_{i}^{\prime B} \tag{7.2}
\end{align*}
\]
where \(\mathbf{s}_{j}^{\prime B}=\left[\xi^{B}, \eta^{B}, \zeta^{B}\right]_{j}^{T}\) is constant. It is clear that the global components of a vector that connects points on two bodies depend on the global position of the bodies, i.e., on vectors \(\mathbf{r}_{i}\) and \(\mathbf{r}_{j}\).

In the following subsections, constraint equations are derived by imposing conditions between vectors in adjacent bodies. In general, either the vectors are of constant magnitude and are embedded in different bodies, e.g., \(\vec{s}_{i}\) and \(\vec{s}_{j}\) in Fig. 7.1; or one vector is embedded in one body and the other vector is connected between points on it and an adjacent body and may have fixed or variable magnitude, e.g., vectors \(\vec{s}_{i}\) and \(\vec{d}\) or \(\vec{s}_{j}\) and \(\vec{d}\). If we write a constraint between two vectors having constant magnitudes, then we will refer to it as type \(I\) constraint. If the constraint is between two vectors, one having fixed magnitude and the other being variable, then we will refer to it as type 2 constraint. Constraint equations in this chapter are assigned a superscript with two indices. The first index denotes the type of constraint, and the second index defines the number of independent equations in the expression.


Figure 7.1 Vectors with constant and with varying magnitudes.

\subsection*{7.1.1 Two Perpendicular Vectors}

To specify that two vectors must remain perpendicular (normal) at all times, we need one constraint relation. If the direction of one vector is specified, the second vector can translate and rotate only in planes perpendicular to the first vector. Vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) shown in Fig. 7.1 remain perpendicular if their scalar product is zero, i.e., if
\[
\begin{align*}
\Phi^{(n 1,1)} & \equiv \mathbf{s}_{i}^{T} \mathbf{s}_{j} \\
& =\mathbf{s}_{i}^{T} \mathbf{A}_{i}^{T} \mathbf{A}_{j} \mathbf{s}_{j}^{\prime}=0 \tag{7.3}
\end{align*}
\]

Note that the superscription \(\Phi\) indicates that this is a normal type 1 constraint having 1 equation.

If vector \(\vec{d}\) in Fig. 7.1, which is connected between bodies \(i\) and \(j\), is to remain perpendicular to \(\vec{s}_{i}\) (normal type 2), then
\[
\begin{align*}
\Phi^{(n 2,1)} & \equiv \mathbf{s}_{i}^{T} \mathbf{d} \\
& =\mathbf{s}_{i}{ }^{T} \mathbf{A}_{i}^{T}\left(\mathbf{r}_{j}+\mathbf{A}_{j} \mathbf{s}_{j}^{\prime B}-\mathbf{r}_{i}-\mathbf{A}_{i} \mathbf{s}_{i}^{\prime B}\right)=0 \tag{7.4}
\end{align*}
\]

\subsection*{7.1.2 Two Parallel Vectors}

For two vectors to remain parallel, two constraint equations are required. The two constraint equations are derived by setting the vector product of the two vectors to zero. The vector product yields three algebraic equations, of which only two are independent; i.e., one of the equations can be derived by combining the remaining two equations. Therefore, two of the equations can serve as the constraint equations.

For two vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) that are embedded in corresponding bodies, the constraint equations imposing parallelism (parallel type 1) are
\[
\begin{align*}
\boldsymbol{\Phi}^{(p!, 2)} & \equiv \tilde{\mathbf{s}}_{i} \mathbf{S}_{j} \\
& =\mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime} \mathbf{A}_{i}^{T} \mathbf{A}_{j} \mathbf{s}_{j}^{\prime}=\mathbf{0} \tag{7.5}
\end{align*}
\]
where Eq. 6.89 has been employed. For a vector \(\vec{s}_{i}\) with constant magnitude and a vector \(\vec{d}\), the constraint equations (parallel type 2 ) are written as
\[
\begin{align*}
\boldsymbol{\Phi}^{(p 2,2)} & \equiv \tilde{\mathbf{s}}_{i} \mathbf{d} \\
& =\mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime} \mathbf{A}_{i}^{T}\left(\mathbf{r}_{j}+\mathbf{A}_{j} \mathbf{s}_{j}^{\prime B}-\mathbf{r}_{i}-\mathbf{A}_{i} \mathbf{s}_{i}^{\prime B}\right)=\mathbf{0} \tag{7.6}
\end{align*}
\]

Note that \(\boldsymbol{\Phi}^{(p 1,2)}=0\) and \(\boldsymbol{\Phi}^{(p 2,2)}=0\) provide three equations each. However, the sets of equations each have only two independent equations. There exists a critical case that is associated with selection of two equations from Eq. 7.5 or 7.6. The critical case occurs when the two vectors become parallel to one of the global coordinate axes. To show how this critical case arises, consider Eq. 7.5 in component form:
\[
\begin{align*}
-s_{(z) i} s_{(y) j}+s_{(y) j} s_{(z) j} & =0  \tag{a}\\
s_{(z) j} s_{(x) j}-s_{(x) j} s_{(z) j} & =0  \tag{b}\\
-s_{(y) i} s_{(x) j}+s_{(x) i} s_{(y) j} & =0 \tag{c}
\end{align*}
\]
where \(\mathbf{s}_{i} \equiv\left[s_{(x)}, s_{(y)}, s_{(z)}\right]_{i}^{T}\) and \(\mathbf{s}_{j} \equiv\left[s_{(x)}, s_{(y)}, s_{(z)}\right]_{j}^{T}\). Assume as an example that vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) are located on bodies \(i\) and \(j\) of a cylindrical joint as shown in Fig. 7.2. At time \(t\) the joint axis, and hence vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\), has become parallel to the \(z\) coordinate axis.


Figure 7.2 The axis of the cylindrical joint may become parallel to 0 . f the coordinate axes, e.g., the \(z\) axis.

Then \(s_{(x) i}=s_{(y) i}=0\) and \(s_{(x) j}=s_{(y) j)}=0\). It is clear that Eqs. \(a, b\), and \(c\) are satisfied. However, for position, velocity, and acceleration analysis the Jacobian matrix of the constraint equations is needed. In this matrix, the row associated with Eq. \(c\) appears as
\[
\partial\left(\text { Eq. c)/ } \cdots\left[\begin{array}{ccc}
\mathbf{p}_{i} & & \mathbf{p}_{i} \\
\vdots & & \vdots \\
\cdots & (1) \cdots \cdots & (2) \cdots \\
\vdots & & \vdots
\end{array}\right]\right.
\]

The possible nonzero entries (1) and (2) are
\[
\begin{aligned}
& \text { (1) }=-\frac{\partial s_{(y) j}}{\partial \mathbf{p}_{i}} s_{(x) j}+\frac{\partial s_{(x) j}}{\partial \mathbf{p}_{i}} s_{(y) j} \\
& \text { (2) }=-\frac{\partial s_{(x) j}}{\partial \mathbf{p}_{j}} s_{(y) i}+\frac{\partial s_{(y) j}}{\partial \mathbf{p}_{j}} s_{(y) i}
\end{aligned}
\]

These entries for this example are identical to zero, and hence Eq. \(c\) leads to a reduction in the row rank of the Jacobian matrix and to numerical difficulties. Therefore, Eqs. \(a\) and \(b\) must be selected, since both contain the nonzero components \(s_{(2) j}\) and \(s_{(z z)}\). A technique for the selection of a proper set of equations can be stated as follows:

Compare the absolute values of \(\mathrm{s}_{(x, j)}, \mathrm{s}_{(y)}\), and \(\mathrm{s}_{(2) \mathrm{i}}\) and select the two equations (out of three) having the largest terms.

\subsection*{7.2 RELATIVE CONSTRAINTS BETWEEN TWO BODIES}

In the following subsections constraint equations for several commonly used lowerpair kinematic joints are derived. These equations fall under the category of holonomic constraints.

\subsection*{7.2.1 Spherical, Universal, and Revolute Joints (LP)}

A spherical or ball joint between two adjacent bodies \(i\) and \(j\) is shown in Fig. 7.3. The center of the spherical joint, point \(P\), has constant coordinates with respect to the \(\xi_{i} \eta_{i} \zeta_{i}\) and \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate systems. There are three algebraic equations for this joint; they can be found from the vector equation \(\vec{r}_{i}+\vec{s}_{i}^{P}-\vec{s}_{j}^{P}-\vec{r}_{j}=\overrightarrow{0}\), as follows:
\[
\begin{equation*}
\boldsymbol{\Phi}^{(s, 3)} \equiv \mathbf{r}_{i}+\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P}-\mathbf{A}_{j} \mathbf{s}_{j}^{\prime P}-\mathbf{r}_{j}=0 \tag{7.7}
\end{equation*}
\]

There are three relative degrees of freedom between two bodies that are connected by a spherical joint.

A universal or Hooke joint between bodies \(i\) and \(j\) is shown in Fig. 7.4(a). One bar of the cross can be considered part of body \(i\) and the other bar can be considered an extension of body \(j\). Point \(P\), the intersection of the axes of the bars, has constant coordinates with respect to both body-fixed coordinate systems. Therefore, at point \(P\), the constraint of Eq. 7.7 can be applied. The remaining constraint is that the two vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\), arbitrarily placed on the cross axes, remain perpendicular. Therefore, the constraint equations for a universal joint are
\[
\begin{gather*}
\boldsymbol{\Phi}^{(s, 3)}=\mathbf{0} \\
\boldsymbol{\Phi}^{(n 1,1)} \equiv \mathbf{s}_{i}^{T} \mathbf{s}_{j}=0 \tag{7.8}
\end{gather*}
\]

There are two relative degrees of freedom between a pair of bodies that are connected by a universal joint.

The constraint formulation of Eq. 7.8 is for the general case of a universal joint between two bodies. This formulation can be simplified for special cases. For example, consider the configuration in Fig. 7.4(b), where body-fixed coordinates are embedded in the bodies in such a way that the \(\xi_{i}\) axis and \(\zeta_{j}\) axis are parallel to their corresponding bars of the cross and therefore \(\xi_{i}\) and \(\zeta_{j}\) must remain perpendicular. Two unit vectors \(\mathbf{u}_{i}^{\prime}=[1,0,0]^{T}\) and \(\mathbf{u}_{j}^{\prime}=[0,0,1]^{T}\) can be defined on the \(\xi_{i}\) and \(\zeta_{j}\) axes, respectively. Hence, \(\mathbf{u}_{i}=\mathbf{A}_{i} \mathbf{u}_{i}^{\prime}\) and \(\mathbf{u}_{j}=\mathbf{A}_{j} \mathbf{u}_{j}^{\prime}\) form one constraint equation,
\[
\begin{equation*}
\Phi^{(n 1,1)} \equiv \mathbf{u}_{i}^{T} \mathbf{u}_{j}=0 \tag{7.9}
\end{equation*}
\]


Figure 7.3 A spherical joint.


Figure 7.4 A universal joint: (a) general case, and (b) special case where \(\xi_{i}\) axis and \(\zeta_{j}\) axis are parallel to their corresponding bars of the cross.
that replaces the constraint \(\mathbf{s}_{i}^{T} \mathbf{s}_{j}=0\) of Eq. 7.8. For this special case, only point \(P\) needs to be defined--there is no need to define points \(Q_{i}\) and \(Q_{j}\).

A revolute joint between bodies \(i\) and \(j\) is shown in Fig. 7.5(a). Any point on the revolute-joint axis has constant coordinates in both local coordinate systems. Equation 7.7 can be imposed on an arbitrary point \(P\) on the joint axis. Two other points, \(Q_{i}\) on body \(i\) and \(Q_{j}\) on body \(j\), are also chosen arbitrarily on the joint axis. It is clear that vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) must remain parallel. Therefore, there are five constraint equations for a revolute joint:
\[
\begin{align*}
\boldsymbol{\Phi}^{(\mathbf{s}, 3)} & =\mathbf{0} \\
\boldsymbol{\Phi}^{(p 1,2)} & =\tilde{\mathbf{s}}_{i} \mathbf{s}_{j}=\mathbf{0} \tag{7.10}
\end{align*}
\]

There is only one relative degree of freedom between two bodies connected by a revolute joint.

The constraint formulation for a revolute joint may be simplified for special cases. Consider as an example the configuration in Fig. 7.5(b), where the body-fixed coordinates are placed in such a way that the \(\zeta_{i}\) and \(\zeta_{j}\) axes are parallel to the revolute-joint axis. In such a case, the two unit vectors \(\mathbf{u}_{i}^{\prime}=[0,0,1]^{T}\) and \(\mathbf{u}_{j}^{\prime}=[0,0,1]^{T}\) must remain parallel at all times; i.e.,
\[
\begin{equation*}
\boldsymbol{\Phi}^{(p 1,2)} \equiv \overline{\mathbf{u}}_{i} \mathbf{u}_{j}=0 \tag{7.11}
\end{equation*}
\]

This equation replaces the \(\tilde{s}_{i} \mathbf{s}_{j}=0\) constraints in Eq. 7.10. For this or other, similar special cases, only one point on the joint axis (point \(p\) ) needs to be defined.


Figure 7.5 A revolute joint: (a) general case, and (b) special case when the \(\zeta_{i}\) axis, the \(\zeta_{j}\) axis, and the joint axis are parallel.

For the special case of Fig. 7.5(b), another method can be used to keep the \(\zeta_{i}\) and \(\zeta_{j}\) axes parallel. Since these two axes are parallel to the joint axis, the joint axis is the relative orientational axis of rotation. If a relative set of Euler parameters is defined as \(\mathbf{p}_{i j}=\left[e_{0}, e_{1}, e_{2}, e_{3}\right]_{i j}^{T}\), Eq. 6.122 must hold; i.e.,
\[
\mathbf{p}_{i j}=\mathbf{L}_{j}^{*} \mathbf{p}_{i}
\]

For this configuration, since the relative axis of rotation is parallel to the \(\zeta\) axes, \(e_{1 i j}=\) \(e_{2 i j}=0\). Therefore two algebraic equations are obtained:
\[
\boldsymbol{\Phi}^{(e,, 2)} \equiv\left[\begin{array}{cccc}
-e_{1} & e_{0} & e_{3} & -e_{2}  \tag{7.12}\\
-e_{2} & -e_{3} & e_{0} & e_{1}
\end{array}\right]_{j} \mathbf{p}_{i}=\mathbf{0}
\]

These two equations may be used instead of the \(\boldsymbol{\Phi}^{(p, 2)}\) constraints of Eq. 7.10.
The special-case configurations of Figs. 7.4(b) and 7.5(b) are not unique. Other special-case configurations may be defined in order to simplify the constraint formulation.

\subsection*{7.2.2 Cylindrical, Translational, and Screw Joints (LP)}

A cylindrical joint constrains two bodies \(i\) and \(j\) to move along a common axis, but allows relative rotation about this axis. To derive equations of constraint for this joint, four points, \(P_{i}\) and \(Q_{i}\) on body \(i\) and \(P_{j}\) and \(Q_{j}\) on body \(j\), are arbitrarily chosen on the joint axis, \({ }^{\dagger}\) as shown in Fig. 7.6(a). It is required that the vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) of constant

\footnotetext{
\({ }^{\dagger}\) Points defined on a body may not be physically located on that body. For example, points \(P_{i}\) and \(Q_{i}\) are inside the hollow cylinder of body \(i\). They are defined with respect to the \(\xi_{i} \eta_{i} \zeta_{i}\) coordinate axes and they move with body \(i\).
}
magnitude and \(\vec{d}\) of variable magnitude remain collinear. Therefore, four constraint equations are needed to define a cylindrical joint; they can be found from two vector product conditions:
\[
\begin{align*}
& \boldsymbol{\Phi}^{(p 1,2)} \equiv \tilde{\mathbf{s}}_{i} \mathbf{S}_{j}=0 \\
& \boldsymbol{\Phi}^{(p 2,2)} \equiv \tilde{\mathbf{s}}_{i} \mathbf{d}=\mathbf{0} \tag{7.13}
\end{align*}
\]

Thus, there are two relative degrees of freedom between bodies connected by a cylindrical joint.

The constraint formulation for cylindrical joints may be simplified in special cases. A special case is shown in Fig. 7.6(b), where the \(\zeta_{i}\) and \(\zeta_{j}\) axes are parallel to the joint axis. Only two points, \(P_{i}\) and \(P_{j}\), are placed on the joint axis to define a vector \(\vec{d}\). This vector must remain parallel to two unit vectors \(\vec{u}_{i}\) and \(\vec{u}_{j}\). Therefore, in this case only four constraint equations are needed:
\[
\begin{align*}
& \boldsymbol{\Phi}^{(p 1,2)} \equiv \tilde{\mathbf{u}}_{i} \mathbf{u}_{j}=\mathbf{0} \\
& \boldsymbol{\Phi}^{(p 2,2)} \equiv \tilde{\mathbf{u}}_{i} \mathbf{d}=\mathbf{0} \tag{7.14}
\end{align*}
\]

Another special case with even simpler formulation is shown in Fig. 7.6(c), where the \(\zeta\) axes coincide with the joint axis. Vector \(\mathbf{d}=\mathbf{r}_{i}-\mathbf{r}_{j}\) must remain parallel to the unit vectors \(\mathbf{u}_{i}\) and \(\mathbf{u}_{j}\). This can be established by the constraints of Eq. 7.14. For this special case, there is no need to define any arbitrary points on the joint axis.

A translational or prismatic joint is similar to a cylindrical joint with the exception that the two bodies cannot rotate relative to each other. Therefore, the cylindricaljoint equations apply and one additional equation is required. Two perpendicular vectors, \(\vec{h}_{i}\) and \(\vec{h}_{j}\) on bodies \(i\) and \(j\), as shown in Fig. 7.7, must remain perpendicular. Therefore, there are five constraint equations for a translational joint:
\[
\begin{align*}
\boldsymbol{\Phi}^{(p 1,2)} & \equiv \tilde{\mathbf{s}}_{i} \mathbf{s}_{j}=\mathbf{0} \\
\mathbf{\Phi}^{(p 2,2)} & \equiv \tilde{\mathbf{s}}_{\boldsymbol{i}} \mathbf{d}=\mathbf{0}  \tag{7.15}\\
\Phi^{(n 1,1)} & \equiv \mathbf{h}_{i}^{T} \mathbf{h}_{j}=0
\end{align*}
\]

The vectors \(\vec{h}_{i}\) and \(\vec{h}_{j}\) are located so they are perpendicular to the line of translation. The relative number of degrees of freedom between two bodies that are connected by a translational joint is 1.

The body-fixed coordinates can be embedded in bodies \(i\) and \(j\) in special-case configurations, much as in the special cases of cylindrical joints shown in Fig. 7.6(b) and (c). If \(\xi_{i} \eta_{i} \zeta_{i}\) and \(\xi_{j} \eta_{j} \zeta_{j}\) are parallel and \(\zeta_{i}\) and \(\zeta_{j}\) are also parallel to the joint axis, then the constraint \(\mathbf{h}_{i}^{T} \mathbf{h}_{j}=0\) of Eq. 7.15 can be replaced by a similar constraint, but without defining any additional points such as \(R_{i}\) and \(R_{j}\). Since the \(\xi_{i}\) axis is perpendicular to the \(\eta_{j}\) axis, then unit vectors on these axes must remain perpendicular at all times.

Figure 7.8 illustrates a screw joint between bodies \(i\) and \(j\), which can rotate and translate about a common axis. However, the rotation and translation are related to each other by the pitch of the screw. To formulate this joint, four constraint equations for an equivalent cylindrical joint can be used. A fifth constraint equation must be supplied, to provide the relation between relative translation and rotation of the bodies. For this purpose, two unit vectors \(\vec{u}_{i}\) and \(\vec{u}_{j}\) perpendicular to the joint axis are embedded in bodies \(i\) and \(j\), respectively, as shown in Fig. 7.8. If the initial angle between \(\vec{u}_{i}\) and \(\vec{u}_{j}\) is \(\theta^{0}\) and the instantaneous angle between the two vectors is denoted by \(\theta\), then \(\theta-\theta^{0}\) is the relative rotation between the two bodies. Similarly, if the initial magnitude of vector \(\vec{d}\) is \(d^{0}\) and its instantaneous magnitude is denoted by \(d\), then \(d-d^{0}\) is the relative translational

(a)

(b)

(c)

Figure 7.6 A cylindrical joint: (a) general case; (b, c) special cases.


Figure 7.7 A translational joint.


Figure 7.8 A screw joint: (a) side view and (b) top view.
displacement between bodies \(i\) and \(j\). Therefore, the five constraint equations can be written as follows:
\[
\begin{align*}
\boldsymbol{\Phi}^{(p 1,2)} & \equiv \tilde{\mathbf{s}}_{i} \mathbf{s}_{j}=\mathbf{0} \\
\boldsymbol{\Phi}^{(p 2,2)} & \equiv \tilde{\mathbf{s}}_{i} \mathbf{d}=0  \tag{7.16}\\
\boldsymbol{\Phi}^{(r, 1)} & \equiv\left(d-d^{0}\right)-\alpha\left(\theta-\theta^{0}\right)=0
\end{align*}
\]
where \(\alpha\) is the pitch rate of the screw joint.

The angle \(\theta\) for a screw joint can be treated as an artificial coordinate. Therefore, one additional constraint equation must be considered with the five constraints of Eq. \(7.16{ }^{\dagger}\)
\[
\begin{equation*}
\Phi^{(\theta, 1)} \equiv \mathbf{u}_{i}^{T} \mathbf{u}_{j}-\cos \theta=0 \tag{7.17}
\end{equation*}
\]

If body-fixed coordinates are embedded in the bodies as was done in the special case of Fig. 7.6(c), then vector \(\mathbf{d}=\mathbf{r}_{i}-\mathbf{r}_{j}\) is obtained easily. Furthermore, since both \(\zeta\) axes are parallel to the joint axis, the joint axis is the relative axis of rotation between bodies \(i\) and \(j\). Therefore, from Eq. 6.120, we can write one constraint equation,
\[
\begin{equation*}
\Phi^{(\theta, 1)} \equiv \mathbf{p}_{i}^{T} \mathbf{p}_{j}-\cos \frac{\theta}{2}=0 \tag{7.18}
\end{equation*}
\]
that can be used instead of Eq. 7.17. In Eq. 7.18, \(\theta\) is the relative angle of rotation between the bodies, and it is assumed that the two body-fixed coordinates are initially parallel, i.e., that \(\theta^{0}=0\).

\subsection*{7.2.3 Composite Joints}

Kinematic joints can be combined and modeled as composite joints in order to reduce the number of coordinates and constraint equations. Several examples of such composite joints are shown in this section.

Figure 7.9 illustrates two bodies connected by a rigid link that contains two spherical joints; the entire system is called a spherical-spherical joint. Only one constraint equation is required for this joint; it may be written in the form
\[
\begin{equation*}
\Phi^{(s s, 1)} \equiv \mathbf{d}^{T} \mathbf{d}-l^{2}=0 \tag{7.19}
\end{equation*}
\]
where
\[
\begin{equation*}
\mathbf{d}=\mathbf{r}_{j}+\mathbf{A}_{j} \mathbf{s}_{j}^{\prime P}-\mathbf{r}_{i}-\mathbf{A}_{i} \mathbf{s}_{i}^{\prime P} \tag{7.20}
\end{equation*}
\]
and \(l\) is the actual length of the link.


Figure 7.9 A spherical-spherical joint.
\({ }^{\dagger}\) If \(\theta\) is not treated as an artifical variable, then Eq. 7.17 is not needed. In this case, \(\theta\) may be calculated from \(\theta=\cos ^{-1}\left(\mathbf{u}_{i}^{T} \mathbf{u}_{j}\right)\) for \(0 \leq \theta \leq \pi\); and when \(\pi<\theta<2 \pi\), additional tests are required. This approach is cumbersome and computationally inefficient.

A spherical-revolute joint is shown in Fig. 7.10. Two points \(P_{j}\) and \(Q_{j}\) are defined on the revolute-joint axis in such a way that vector \(\vec{d}\) is perpendicular to vector \(\vec{s}_{j}\) as shown. If the distance between points \(P_{i}\) and \(P_{j}\) is to remain equal to \(l\), then two constraint equations are written:
\[
\begin{align*}
& \Phi^{(s-s, 1)} \equiv \mathbf{d}^{T} \mathbf{d}-l^{2}=0 \\
& \Phi^{(n 2,1)} \equiv \mathbf{d}^{T} \mathbf{s}_{j}=0 \tag{7.21}
\end{align*}
\]

Two revolute-revolute joints are shown in Fig. 7.11. In the configuration of Fig. 7.11(a), the two revolute-joint axes are assumed to be perpendicular. Four constraint equations for this joint may be written, as follows:
\[
\begin{align*}
& \Phi^{(s-s, 1)} \equiv \mathbf{d}^{T} \mathbf{d}-l^{2}=0 \\
& \Phi^{(n 2,1)} \equiv \mathbf{d}^{T} \mathbf{s}_{i}=0 \\
& \Phi^{(2,1)} \equiv \mathbf{d}^{T} \mathbf{s}_{j}=0  \tag{7.22}\\
& \Phi^{(1,1)} \equiv \mathbf{s}_{i}^{T} \mathbf{s}_{j}=0
\end{align*}
\]

When the revolute-joint axes are parallel, as shown in Fig. 7.11(b), the four constraint equations are
\[
\begin{align*}
& \Phi^{(s-s, 1)} \equiv \mathbf{d}^{T} \mathbf{d}-l^{2}=0 \\
& \Phi^{(n, 1)} \equiv \mathbf{d}^{\top} \mathbf{s}_{i}=0 \\
& \Phi^{(n, 1)} \equiv \mathbf{d}^{T} \mathbf{s}_{j}=0  \tag{7.23}\\
& \boldsymbol{\Phi}^{(p 1,1)} \equiv \overline{\mathbf{s}}_{i} \mathbf{s}_{j}=\mathbf{0}
\end{align*}
\]
where \(\Phi^{(p, 1)}\) indicates that only one out of three equations of \(\Phi^{(p 1)}\) is needed. The reason is that the equations \(\mathbf{d}^{T} \mathbf{s}_{i}=0, \mathbf{d}^{T} \mathbf{s}_{j}=0\), and only one of the three equations from \(\tilde{\mathbf{s}}_{i} \mathbf{s}_{j}=\mathbf{0}\) are independent. However, it is possible that one of the constraint equations


Figure 7.10 A spherical-revolute joint.


Figure 7.11 Revolute-revolute joint links: (a) perpendicular and (b) parallel.
\(\mathbf{d}^{T} \mathbf{s}_{i}=0\) or \(\mathbf{d}^{T} \mathbf{s}_{j}=0\) could be ignored, so that two equations from \(\boldsymbol{\Phi}^{(p 1,2)}\) could be selected. In this case, either of the following sets may be used:
\[
\begin{align*}
& \Phi^{(s-s, 1)}=0 \\
& \Phi^{(n 2,1)} \equiv \mathbf{d}^{T} \mathbf{s}_{i}=0  \tag{7.24}\\
& \boldsymbol{\Phi}^{(p 1,2)} \equiv \tilde{\mathbf{s}}_{i} \mathbf{s}_{j}=\mathbf{0}
\end{align*}
\]
or
\[
\begin{align*}
& \Phi^{(s-s, 1)}=0 \\
& \Phi^{(n 2,1)} \equiv \mathbf{d}^{T} \mathbf{s}_{j}=0  \tag{7.25}\\
& \Phi^{(p 1,2)} \equiv \tilde{\mathbf{s}}_{i} \mathbf{S}_{j}=0
\end{align*}
\]

A revolute-cylindrical joint that connects two bodies is shown in Fig. 7.12. Formulation of this joint as a revolute joint and a cylindrical joint involving three bodies would require nine constraint equations. This would result in the elimination of a total of 9 degrees of freedom, leaving 3 relative degrees of freedom between bodies \(i\) and \(j\). This joint may also be effectively formulated as two bodies connected by a composite revolutecylindrical joint. In the configuration shown in Fig. 7.12, the cylindrical-joint axis is perpendicular to the revolute-joint axis and the two axes intersect. Three vectors, \(\vec{s}_{i}, \vec{s}_{j}\), and \(\vec{d}\), can be defined for the constraint formulation. Vector \(\vec{s}_{i}\) is located on the revolute-joint axis on body \(i\), vector \(\dot{\vec{s}}_{j}\) is on the cylindrical-joint axis on body \(j\), and vec-


Figure 7.12 A revolute-cylindrical joint.
tor \(\vec{d}\) connects point \(P_{i}\) on body \(i\) to point \(P_{j}\) on body \(j\). The constraint equations are written as follows:
\[
\begin{align*}
& \boldsymbol{\Phi}^{(n 1,1)} \equiv \mathbf{s}_{i}^{T} \mathbf{s}_{j}=0 \\
& \boldsymbol{\Phi}^{(p 2,2)} \equiv \tilde{\mathbf{s}}_{j} \mathbf{d}=\mathbf{0} \tag{7.26}
\end{align*}
\]

The first equation requires that the two axes remain perpendicular. The remaining equation guarantees that the two axes intersect at point \(P_{i}\).

Other composite joints, similar to those in the preceding examples, may be defined and formulated. The formulation of composite joints may be simplified if bodyfixed coordinates are placed on the bodies in such a way that unit vectors on the coordinate axes are used instead of arbitrary vectors on the joint axes. This process is similar to the special cases shown in Secs. 7.2.1 and 7.2.2.

\subsection*{7.2.4 Simplified Constraints}

Translational constraint equations on the global coordinates of a body can be formulated in a manner similar to the process used for planar motion in Section 4.2.7. Such constraints are
\[
\begin{align*}
& \Phi \equiv x_{i}-c_{1}=0  \tag{7.27}\\
& \Phi \equiv y_{i}-c_{2}=0  \tag{7.28}\\
& \Phi \equiv z_{i}-c_{3}=0  \tag{7.29}\\
& \Phi \equiv x_{i}^{P}-c_{4}=0  \tag{7.30}\\
& \Phi \equiv y_{i}^{P}-c_{5}=0  \tag{7.31}\\
& \Phi \equiv z_{i}^{P}-c_{6}=0 \tag{7.32}
\end{align*}
\]
where \(c_{1}\) through \(c_{6}\) are constants. Equations 7.27 through 7.29 constrain the origin of the body-fixed coordinate system, and Eqs. 7.30 through 7.32 constrain a point \(P_{i}\) on
body \(i\), in both cases relative to the global axes. The constants \(c_{1}\) through \(c_{6}\) can be replaced by time-dependent quantities for driving constraints.

\subsection*{7.3 POSITION, VELOCITY, AND ACCELERATION ANALYSIS}

The computational aspects of spatial kinematics are the same as those of planar kinematic analysis. For a mechanism with \(b\) bodies, the vector of coordinates is represented as \(\mathbf{q}=\left[\mathbf{q}_{1}^{T}, \mathbf{q}_{2}^{T}, \ldots, \mathbf{q}_{b}^{T}\right]^{T}\). Vector \(\mathbf{q}_{i}\) is expressed in the form \(\mathbf{q}_{i}=\left[x, y, z, e_{0}, e_{1}, e_{2}, e_{3}\right]_{i}^{T}\) or \(\mathbf{q}_{i}=\left[\mathbf{r}^{T}, \mathbf{p}^{T}\right]_{i}^{T}\). If any artificial variables are used in the constraint formulation (for example, for screw joints), the artificial variables are also included in vector \(\mathbf{q}\). The constraint equations are assumed to consist of \(m\) equations in the form
\[
\begin{equation*}
\boldsymbol{\Phi} \equiv \boldsymbol{\Phi}(\mathbf{q})=0 \tag{7.33}
\end{equation*}
\]

Equation 7.33 contains \(m-b\) kinematic constraints and \(b\) mathematical constraints between the Euler parameters (one equation per body) in the form given by Eq. 6.23.

Kinematic analysis with the appended driving constraint method requires a formulation identical to that shown in Sec. 3.2.2. For kinematic analysis, Jacobian matrix \(\boldsymbol{\Phi}_{\mathbf{q}}\) must be evaluated.

\section*{Example 7.1}

Determine the nonzero entries of the Jacobian matrix for constraint \(\Phi^{(n 1,1)}\) of Eq. 7.3.

Solution The nonzero entries are determined by evaluating the partial derivatives of Eq. 7.3 with respect to the coordinates of bodies \(i\) and \(j\). Since \(\mathbf{s}_{i}^{T} \mathbf{s}_{j}\) is not a function of \(\mathbf{r}_{i}\), its partial derivative with respect to \(\mathbf{r}_{i}\) is zero:
\[
\begin{equation*}
\Phi_{\mathbf{r}_{i}}^{(n 1,1)}=\mathbf{0}^{T} \tag{1}
\end{equation*}
\]

The partial derivative of Eq. 7.3 can be found by writing the equation as \(\mathbf{s}_{j}^{T} \mathbf{s}_{i}\) or \(\mathbf{s}_{j}^{T} \mathbf{A}_{i} \mathbf{s}_{i}^{\prime}\) and employing Eq. 6.86 to obtain
\[
\begin{equation*}
\Phi_{\mathbf{p}_{i}}^{(n 1,1)}=2 \mathbf{s}_{j}^{T}\left(\mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}+\mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T}\right) \tag{2}
\end{equation*}
\]

Similarly, the partial derivatives of Eq. 7.3 with respect to \(\mathbf{r}_{j}\) and \(\mathbf{p}_{j}\), by keeping the equation in the form \(\mathbf{s}_{i}^{T} \mathbf{A}_{j} \mathbf{s}_{j}^{\prime}\), are found to be
\[
\begin{gather*}
\Phi_{\mathbf{r}_{j}}^{(n 1,1)}=\boldsymbol{0}^{T}  \tag{3}\\
\Phi_{\mathbf{p}_{j}}^{(n 1,1)}=2 \mathbf{s}_{i}^{T}\left(\mathbf{G}_{j} \overrightarrow{\mathbf{s}}_{j}^{\prime}+\mathbf{s}_{j}^{\prime} \mathbf{p}_{j}^{T}\right) \tag{4}
\end{gather*}
\]

Equations 2 and 4 each provide four nonzero entries in the columns of the Jacobian matrix associated with \(\mathbf{p}_{i}\) and \(\mathbf{p}_{j}\).

Table 7.1 summarizes elements of the Jacobian matrix for Eqs. 7.3 through 7.7 and Eq. 7.19. These equations are the essential building blocks for nearly all of the constraints developed in this chapter and for most constraint relations that can be developed between adjacent bodies. The fact that all terms can be written in compact vector or matrix form demonstrates an additional advantage of the Euler parameter formulation for developing compact computational algorithms.

TABLE 7.1 Components of the Jacobian Matrix of the Most Common Constraints \({ }^{15}\)
\begin{tabular}{|c|c|c|c|c|}
\hline \(\Phi\) & \(\boldsymbol{\Phi}_{\mathrm{r}_{i}}\) & \(\boldsymbol{\Phi}_{\mathrm{p}_{i}}\) & \(\boldsymbol{\Phi}_{\mathrm{r} j}\) & \(\boldsymbol{\Phi}_{\mathrm{P} j}\) \\
\hline \(\Phi^{(n 1,1)}\) & \(0^{T}\) & \(\mathbf{s}_{j}^{T} \mathrm{C}_{i}\) & \(0^{T}\) & \(\mathbf{s}^{T} \mathrm{C}_{j}\) \\
\hline \(\Phi^{(n 2,1)}\) & \(-\mathbf{s}^{T}\) & \(-s_{i}^{T} \mathbf{B}_{i}+\mathbf{d}^{T} \mathrm{C}_{i}\) & \(\mathbf{s}_{i}^{T}\) & \(\mathbf{s}^{T} \mathbf{B}_{j}\) \\
\hline \(\Phi^{(p 1,2)}\) & 0 & \(-\tilde{\mathbf{s}}_{j} \mathrm{C}_{i}\) & 0 & \(\tilde{s}_{i} \mathrm{C}_{j}\) \\
\hline \(\Phi^{(p 2,2)}\) & \(-\tilde{\mathbf{s}}_{i}\) & \(-\tilde{\mathbf{s}}_{i} \mathbf{B}_{i}-\overline{\mathbf{d}}^{\text {C }}\) & \(\tilde{\mathbf{s}}_{i}\) & \(\bar{s}_{i} \mathbf{B}_{j}\) \\
\hline \(\boldsymbol{\Phi}^{(s, 3)}\) & I & C & - I & \(-\mathrm{C}_{j}\) \\
\hline \(\Phi^{(s-5,1)}\) & \(-2 \mathbf{d}^{T}\) & \(-2 \mathbf{d}^{\top} \mathbf{B}_{i}\) & \(2 \mathrm{~d}^{7}\) & \(2 \mathbf{d}^{T} \mathbf{B}_{j}\) \\
\hline
\end{tabular}
where:
\[
\mathbf{B}_{k}=2\left(\mathbf{G}_{k} \overline{\mathbf{s}}_{k}^{\prime B}+\mathbf{s}_{k}^{\prime B} \mathbf{p}_{k}^{T}\right), \mathbf{C}_{k}=2\left(\mathbf{G}_{k} \overline{\mathbf{s}}_{k}^{\prime}+\mathbf{s}_{k}^{\prime} \mathbf{p}_{k}^{T}\right), k=i, j
\]

\subsection*{7.3.1 Modified Jacobian Matrix and Modified Vector \(\gamma\)}

For most common constraints, the components of the Jacobian matrix of Table 7.1 can be employed directly. However, when the velocity and acceleration equations are considered, some equivalent terms from both sides of the equations can be eliminated. Hence, the final forms of the resultant equations are simpler, having fewer terms than the original equations.

\section*{Example 7.2}

Consider a vector \(\vec{s}_{i}\) whose magnitude and direction must remain fixed in a given problem. The constraints describing this condition can be expressed as
\[
\begin{align*}
\boldsymbol{\Phi} & \equiv \mathbf{s}_{i}-\mathbf{c} \\
& =\mathbf{A}_{i} \mathbf{s}_{i}^{\prime}-\mathbf{c}=\mathbf{0} \tag{1}
\end{align*}
\]
where \(\mathbf{c}\) contains three constant components. The entries of the Jacobian matrix for Eq. 1 are found, from Eq. 6.86, to be
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{p}_{i}}=2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}+2 \mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T} \tag{2}
\end{equation*}
\]

According to the first equation of Eq. 3.13, the velocity equation is
\[
\begin{equation*}
\left(2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}+2 \mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T}\right) \dot{\mathbf{p}}_{i}=\mathbf{0} \tag{3}
\end{equation*}
\]

Using the identity \(\mathbf{p}_{i}^{T} \dot{\mathbf{p}}_{i}=\boldsymbol{0}\) in Eq. 3 yields
\[
\begin{equation*}
2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime} \dot{\mathbf{p}}_{i}=\mathbf{0} \tag{4}
\end{equation*}
\]

Similarly, according to the first equation of Eq. 3.15, the acceleration equation is
\[
\begin{align*}
\left(2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}+2 \mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T}\right) \ddot{\mathbf{p}}_{i} & =-\left[\left(2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}+2 \mathbf{s}_{\mathbf{s}}^{\prime} \mathbf{i}_{i}^{T}\right) \dot{\mathbf{p}}_{i}\right]_{p_{i}} \dot{\mathbf{p}}_{i} \\
& =\left(-2 \dot{\mathbf{G}}_{i} \overline{\mathbf{s}}_{i}^{\prime} \mathbf{p}_{i}-2 \mathbf{s}_{i}^{\prime} \dot{\mathbf{p}}_{i}^{T} \mathbf{p}_{i}\right)_{\mathbf{p}_{i}} \dot{\mathbf{p}}_{i} \\
& =-2 \dot{\mathbf{G}}_{i} \mathbf{s}_{i}^{\prime} \dot{\mathbf{p}}_{i}-2 \mathbf{s}_{i}^{\prime} \dot{\mathbf{p}}_{i}^{T} \mathbf{p}_{i} \tag{5}
\end{align*}
\]
where Eqs. 6.73, 6.54, and 6.69 have been employed, in that order. From Eq. 6.61, i.e., \(\mathbf{p}_{i}^{T} \dot{\mathbf{p}}_{i}+\dot{\mathbf{p}}_{i}^{T} \dot{\mathbf{p}}_{i}=0\), Eq. 5 becomes
\[
\begin{align*}
2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime} \ddot{\mathbf{p}}_{i} & =-2 \dot{\mathbf{G}}_{i} \overline{\mathbf{s}}_{i}^{\prime} \dot{\mathbf{p}}_{i} \\
& =-\left(2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime} \dot{\mathbf{p}}_{i}\right)_{\mathbf{p}_{\mathbf{i}}} \dot{\mathbf{p}}_{i} \tag{6}
\end{align*}
\]

For velocity and acceleration analysis, Eqs. 4 and 6 can be used instead of Eqs. 3 and 5. By defining a modified Jacobian matrix as
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{p}_{i}}^{(m)}=2 \mathbf{G}_{i} \overline{\mathbf{s}}_{\mathbf{i}}^{\prime} \tag{7}
\end{equation*}
\]
we can write Eqs. 4 and 6 as
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{p}_{i}}^{(m)} \dot{\mathbf{p}}_{i}=0 \tag{8}
\end{equation*}
\]
and
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{p}_{i}}^{(m)} \ddot{\mathbf{p}}_{i}=-\left(\boldsymbol{\Phi}_{\mathbf{p}_{i}}^{(m)} \dot{\mathbf{i}}_{i}\right)_{\mathbf{p}_{\mathbf{i}}} \dot{\mathbf{i}}_{i} \tag{9}
\end{equation*}
\]

The simplification of the velocity and acceleration equations shown in Example 7.2 can be applied to all of the common constraint equations; it enables the entries of the modified Jacobian matrix and the components of the vector \(\gamma\) to be found. Another method for finding the modified Jacobian and vector \(\gamma\) is to use the second time derivative of the constraint equations and employ Eq. 6.82.

\section*{Example 7.3}

Apply Eq. 6.82 to find the second time derivative of Eq. 7.3. Obtain the modified Jacobian matrix and modified vector \(\gamma\) for this constraint.
Solution The second time derivative of Eq. 7.3, i.e., \(\mathbf{s}_{i}^{T} \mathbf{s}_{j}=0\), is
\[
\mathbf{s}_{j}^{T} \ddot{\mathbf{j}}_{i}+\mathbf{s}_{i}^{T} \ddot{\mathbf{j}}_{j}+2 \dot{\mathbf{s}}_{i}^{T} \dot{\mathbf{s}}_{j}=0
\]
or
\[
\mathbf{s}_{j}^{T} \ddot{\mathbf{A}}_{i} \mathbf{s}_{i}^{\prime}+\mathbf{s}_{i}^{T} \ddot{\mathbf{A}}_{j} \mathbf{s}_{j}^{\prime}+2 \dot{\mathbf{s}}_{i}^{T} \dot{\mathbf{j}}_{j}=0
\]

With Eq. 6.82, the above equation yields
\[
2 \mathbf{s}_{j}^{T}\left(\dot{\mathbf{G}}_{i} \dot{\mathbf{L}}_{i}^{T} \mathbf{s}_{i}^{\prime}+\mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime} \ddot{\mathbf{P}}_{i}\right)+2 \mathbf{s}_{i}^{T}\left(\dot{\mathbf{G}}_{j} \dot{\mathbf{L}}_{j}^{T} \mathbf{s}_{j}^{\prime}+\mathbf{G}_{j} \mathbf{s}_{j}^{\prime} \ddot{\mathbf{p}}_{j}\right)+2 \dot{\mathbf{s}}_{i}^{T} \dot{\mathbf{s}}_{j}=0
\]

This equation can be rearranged as follows:
\[
\left[2 \mathbf{s}_{j}^{T} \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}, 2 \mathbf{s}_{i}^{T} \mathbf{G}_{j} \overline{\mathbf{s}}_{j}^{\prime}\right]\left[\begin{array}{c}
\ddot{\mathbf{p}}_{i} \\
\ddot{\mathbf{p}}_{j}
\end{array}\right]=-2\left(\mathbf{s}_{j}^{T} \dot{\mathbf{G}}_{i} \dot{\mathbf{L}}_{i} \mathbf{s}_{j}^{\prime}+\mathbf{s}_{i}^{T} \dot{\mathbf{G}}_{j} \dot{\mathbf{L}}_{j} \mathbf{s}_{j}^{\prime}+\dot{\mathbf{s}}_{i}^{T} \mathbf{\dot { \mathbf { s } }}_{j}\right)
\]

The matrix on the left side of this equation shows the nonzero entries of the modified Jacobian, and the expression on the right side of this equation is the modified \(\gamma\) vector.

Table 7.2 summarizes the elements of the modified Jacobian matrix and modified vector \(\gamma\) for Eqs. 7.3 through 7.7 and Eq. 7.19. The components of Table 7.2 can be used in a computer program for kinematic analysis instead of those of Table 7.1. Examples 7.2 and 7.3 show how the entries of the Jacobian matrix can be modified (simplified) for velocity and acceleration analysis. It should be noted that the modified Jacobian matrix can also be used for position analysis. The first step of the NewtonRaphson iteration is given by Eq. 3.42 as
\[
\begin{equation*}
\Phi_{q} \Delta \mathbf{q}=-\Phi \tag{a}
\end{equation*}
\]

TABLE 7.2 Components in the Expansion of the Most Common Constraints \({ }^{15}\)
\begin{tabular}{|c|c|c|c|c|c|}
\hline \(\Phi\) & \(\boldsymbol{\Phi}_{\mathbf{r}_{i}}^{(m)}\) & \(\Phi_{\mathbf{p}_{i}\left({ }^{(m)}\right.}\) & \(\boldsymbol{\Phi}_{\mathrm{r}_{j}\left({ }^{(m)}\right.}\) & \(\Phi_{\mathbf{p}_{j}(\underline{m)}}\) & \(\gamma^{(m)}\) \\
\hline \(\Phi^{(m 1,1)}\) & \(0^{T}\) & \(2 s_{j}^{T} \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}\) & \(\boldsymbol{0}^{T}\) &  & \(\mathbf{s}_{i}^{T} \mathbf{h}_{j}+\mathbf{s}^{T} \mathbf{h}_{\mathbf{h}}-2 \dot{\mathbf{s}}_{i} \mathrm{~s}_{\mathbf{j}_{j}}\) \\
\hline \(\Phi^{(12,1)}\) & \(-\mathbf{s}_{i}^{T}\) & \(-2 \mathbf{s}_{i}^{T} \mathbf{G}_{\mathbf{i}} \overline{\mathbf{s}}_{i}^{\prime B}+2 \mathbf{d}^{T} \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}\) & \(\mathbf{s}_{i}^{T}\) & \(2 \mathbf{2 s}_{i}^{T} \mathbf{G}_{j} \mathbf{s}_{j}^{\prime B}\) & \(-\mathbf{s}_{i}^{T}\left(\mathbf{h}_{i}^{B}-\mathbf{h}_{j}^{B}\right)+\mathbf{d}^{T} \mathbf{h}_{i}-2 \dot{s}_{i} \mathbf{d}^{\dot{d}}\) \\
\hline \(\Phi^{(p 1,2)}\) & 0 & \(-2 \bar{s}_{j} \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}\) & 0 & \(2 \tilde{\mathbf{s}}_{i} \mathbf{G}_{j} \overline{\mathbf{s}}_{j}^{\prime}\) & \[
\tilde{\mathbf{s}}_{i} \mathbf{h}_{j}-\tilde{\mathbf{s}}_{j} \mathbf{h}_{i}-2 \tilde{\mathbf{s}}_{i} \dot{\mathbf{s}}_{j}
\] \\
\hline \(\Phi^{(p 2,2)}\) & \({ }_{-\overline{\mathbf{s}}_{i}}\) &  & \(\tilde{\mathbf{s}}_{i}\) & \(2 \overline{\mathbf{s}}_{i} \mathbf{G}_{\mathbf{j}} \mathrm{s}_{j}{ }^{\prime \prime}\) & \(-\tilde{\mathbf{s}}_{i}\left(\mathbf{h}_{j}^{B}-\mathbf{h}_{i}^{B}\right)-\bar{d} h_{i}-2 \tilde{\mathbf{s}_{i}} \dot{\mathrm{~d}}\) \\
\hline \(\Phi^{(s, 3)}\) & I & \(2 \mathbf{G}_{i} \mathbf{s}_{i}^{\prime \prime}{ }^{\prime \prime}\) & -I & \(-2 \mathbf{G}_{j} \mathbf{s}_{j}^{\prime p}\) & \(\mathbf{h}_{i}^{p}-\mathbf{h}_{j}^{P}\) \\
\hline \(\Phi^{(s-5,1)}\) & \(-2 \mathbf{d}^{T}\) & \(-4 \mathbf{d}^{T} \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\text {'P }}\) & \(2 \mathbf{d}^{T}\) & \(4 \mathbf{d}^{T} \mathbf{G}_{\mathbf{j}} \overline{\mathbf{s}}_{j}^{\prime \prime}\). & \(2 \mathbf{d}^{T}\left(\mathbf{h}_{i}^{P}-\mathbf{h}_{j}^{P}\right)-2 \dot{\mathbf{d}}^{T} \dot{\mathbf{d}}\) \\
\hline
\end{tabular}
where:
\[
\mathbf{h}_{k}=-2 \dot{\mathbf{G}}_{k} \dot{\mathbf{L}}_{k}^{T} \mathbf{s}_{k}^{\prime}, \mathbf{h}_{k}^{B}=-2 \dot{\mathbf{G}}_{k} \dot{\mathbf{L}}_{k}^{T} \mathbf{s}_{k}^{\prime B}, \mathbf{h}_{k}^{P}=-2 \dot{\mathbf{G}}_{k} \dot{\mathbf{L}}_{k}^{T} \mathbf{s}_{k}^{\prime P}, k=i, j
\]

This operation involves the product of the Jacobian matrix and a vector \(\Delta \mathbf{q}\) which contains the infinitesimal changes in the coordinates. Since the modification on the Jacobian has affected only the columns associated with the Euler parameters, not those for the translational coordinates, we need only test the part corresponding to \(\boldsymbol{\Phi}_{\mathrm{q}} \Delta \mathbf{p}\). The first variation of the constraint equation on the Euler parameters (Eq. 6.23) is
\[
\begin{equation*}
2 \mathbf{p}^{T} \Delta \mathbf{p}=0 \tag{b}
\end{equation*}
\]

If we consider the entries of the Jacobian from Table 7.1 in the columns associated with \(\mathbf{p}_{i}\) and \(\mathbf{p}_{j}\) and employ Eq. \(b\), we will obtain the modified matrix. For example, the columns of the Jacobian associated with \(\mathbf{p}_{i}\) of \(\Phi^{(n i, 1)}\) yield
\[
\begin{aligned}
\Phi_{\mathbf{p}_{i}} & =\left(\mathbf{s}_{j}^{T} \mathbf{C}_{i}\right) \Delta \mathbf{p}_{i} \\
& =2 \mathbf{s}_{j}^{T}\left(\mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}+\mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T}\right) \Delta \mathbf{p}_{i} \\
& =2 \mathbf{s}_{j}^{T} \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime} \Delta \mathbf{p}_{i} \\
& =\Phi_{\mathbf{p}_{i}}^{(m)} \Delta \mathbf{p}_{i}
\end{aligned}
\]

This result can also be obtained by considering Eq. 6.86. This equation can be simplified in the Newton-Raphson algorithm as follows:
\[
\begin{aligned}
{\left[\frac{\partial}{\partial \mathbf{p}}(\mathbf{A a})\right] \Delta \mathbf{p} } & =\left(2 \mathbf{G} \overline{\mathbf{a}}+2 \mathbf{a} \mathbf{p}^{T}\right) \Delta \mathbf{p} \\
& =2 \mathbf{G} \overline{\mathbf{a}} \Delta \mathbf{p}
\end{aligned}
\]

This equation shows that the term 2Gā in Eq. 6.86 (or the term 2La+ in Eq. 6.87) yields the entries of the modified Jacobian matrix. Therefore, two more identities can be stated:
\[
\begin{equation*}
\left[\frac{\partial}{\partial \mathbf{p}}(\mathbf{A a})\right]^{(m)}=2 \mathbf{G} \overline{\mathbf{a}} \tag{7.34}
\end{equation*}
\]
and
\[
\begin{equation*}
\left[\frac{\partial}{\partial \mathbf{p}}(\mathbf{A a})\right]^{(m)}=2 \mathbf{L}^{+} \tag{7.35}
\end{equation*}
\]

\section*{PROBLEMS}
7.1 Two points \(B\) and \(C\) are located on body \(i\). The coordinates of these points with respect to a set of \(\xi_{i} \eta_{i} \zeta_{i}\) axes are \([1.2,0.5,-0.1]^{T}\) and \([-0.3,-0.8,2.1]^{T}\), respectively. The Euler parameters describing the angular orientation of body \(i\) with respect to a global coordinate system are \(\mathbf{p}_{i}=[0.860,-0.150,0.420,0.248]^{\mathrm{T}}\).
(a) Find the local components of vector \(\vec{s} \equiv \overrightarrow{C B}\).
(b) Find the global components of vector \(\vec{s}\).
(c) Is it possible to find the global coordinates of points \(B\) and \(C\) ? Explain.
7.2 Point \(D\) is located on body \(i\) with local coordinates \([1.5,-1.6,0.2]^{T}\). The origin of the \(\xi_{i} \eta_{i} \zeta_{i}\) coordinate system is located with respect to the global coordinates by vector \(\mathbf{r}_{i}=[3.3,1.4\), \(2.0]^{T}\). The Euler parameters describing the angular orientation of body \(i\) are \(\mathbf{p}_{i}=[0.810,-0.029,-0.543,0.220]^{T}\). Determine the following:
(a) Local components of vector \(\vec{s}_{i}^{D}\) connecting the origin of the \(\xi_{i} \eta_{i} \zeta_{i}\) axes to point \(D\)
(b) Global components of vector \(\vec{s}_{i}^{D}\)
(c) Global coordinates of point \(D\)
7.3 Point \(B\) is located on body \(i\) with local coordinates [1.0,1.0, -0.5] \({ }^{T}\), and point \(C\) is located on body \(j\) with local coordinates \([-2.0,1.5,-1.0]^{T}\). The origins of the bodies are located by vectors \(\mathbf{r}_{i}=[-1.2,0.4,3.1]^{T}\) and \(\mathbf{r}_{j}=[0.4,4.5,0.5]^{T}\) with respect to the global axes. The Euler parameters for the two bodies are \(\mathbf{p}_{i}=[0.343,-0.564,0.604,0.447]^{T}\) and \(\mathbf{p}_{j}=[0.270,0.732,-0.331,0.531]^{T}\). Find the following:
(a) Global coordinates of points \(B_{i}\) and \(C_{j}\)
(b) Global components of vector \(\vec{d} \equiv \overrightarrow{B_{i} C_{j}}\)
(c) Local components of vector \(\vec{d}\) with respect to the \(\xi_{i} \eta_{i} \zeta_{i}\) axes
(d) Local components of vector \(\vec{d}\) with respect to the \(\xi_{j} \eta_{j} \zeta_{j}\) axes
7.4 The orientations of bodies \(i\) and \(j\) are defined by Euler parameters \(\mathbf{p}_{i}=[-0.667\), \(-0.427,0.241,-0.561]^{T}\) and, \(\mathbf{p}_{j}=[0.223,0.549,-0.623,0.511]^{T}\). Vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) are fixed vectors on the bodies with local components \(\mathbf{s}_{i}^{\prime}=[1,1,2]^{T}\) and \(\mathbf{s}_{j}^{\prime}=[-1,1, \alpha]^{T}\), where \(\alpha\) is unknown. Determine \(\alpha\), knowing that \(\vec{s}_{i}\) and \(\vec{s}_{j}\) are perpendicular.
7.5 What must the condition be between Euler parameters \(\mathbf{p}_{i}\) and \(\mathbf{p}_{j}\) for the two vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) to remain perpendicular? Assume \(\mathbf{s}_{i}^{\prime}=[1,1,2]^{T}\) and \(\mathbf{s}_{j}^{\prime}=[2,-1,-2]^{T}\).
7.6 Points \(B\) and \(C\) are located on body \(i\) with local coordinates \([-0.2,-0.7,0.6]^{T}\) and \([2.7,-0.7,-0.1]^{T}\). Vector \(\vec{s}_{i}\) connects points \(B\) and \(C\). Find a condition on the Euler parameters of this body for which vector \(\vec{s}_{i}\) remains parallel to the \(x y\) plane.
7.7 Show that the three equations in \(\tilde{\mathbf{s}}_{i} \mathbf{s}_{j}=\mathbf{0}\) are not independent.
7.8 Vectors \(\mathbf{r}_{i}=[0.5,0.5,0.9]^{T}\) and \(\mathbf{r}_{j}=[0.5,-0.2,1.3]^{T}\) locate the fixed origins of bodies \(i\) and \(j\) with respect to the global axes. Vector \(\vec{s}_{i}^{C}\) locates point \(C\) on body \(i\) as \(\mathbf{s}_{i}{ }^{c}=\) \([-0.7,1.6,0.8]^{T}\). Point \(B\) is located on body \(j\) by vector \(\mathrm{s}_{j}^{\prime B}=[0.8,-0.6,-0.5]^{T}\). Write the constraint equation that makes vector \(\vec{s}_{i}^{C}\) perpendicular to vector \(\vec{d} \equiv \overrightarrow{C_{i} B_{j}}\).
7.9 Repeat Prob. 7.8 but assume that the origins of bodies \(i\) and \(j\) are not fixed; i.e., that vectors \(\mathbf{r}_{i}\) and \(\mathbf{r}_{j}\) are variables.
7.10 Write the condition for the two vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) to be parallel. If the condition yields more than two equations, then choose the best two equations as constraint equations:
(a) \(\mathbf{s}_{i}^{\prime}=[-0.448,0.399,1.700]^{T}, \boldsymbol{p}_{i}=[0.860,-0.150,0.420,0.248]^{T}\),
\[
\mathbf{s}_{j}^{\prime}=[-0.4131,1.690,0.634]^{T}, \mathbf{p}_{j}=[0.810,0.029,-0.543,0.220]^{T} .
\]
(b) \(\mathbf{s}_{i}^{\prime}=[1.976,0.874,1.825]^{T}, \mathbf{p}_{i}=[0.270,-0.732,0.331,-0.531]^{T}\), \(\mathbf{s}_{j}^{\prime}=[-1.143,-0.295,-1.220]^{T}, \mathbf{p}_{j}=[0.564,-0.447,-0.343,0.604]^{T}\).
(c) \(\mathbf{s}_{i}^{\prime}=[-0.132,-0.089,-0.324]^{T}, \mathbf{p}_{i}=[-0.667,-0.427,0.241,-0.561]^{T}\), \(\mathbf{s}_{j}^{\prime}=[-0.446,0.121,-0.082]^{T}, \mathbf{p}_{j}=[0.223,0.549,-0.623,0.511]^{T}\).
(d) \(\mathbf{s}_{i}^{\prime}=[1.64,3,-1.52]^{T}, \mathbf{p}_{i}=[0.6,0,-0.8,0]^{T}\), \(\mathbf{s}_{j}^{\prime}=[4.4,-6,0.8]^{T}, \mathbf{p}_{j}=[0,0.8,0,0.6]^{T}\).
7.11 What is (are) the necessary and sufficient condition(s) on the Euler parameters for the \(\xi_{i} \eta_{i} \zeta_{i}\) and \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate systems to remain parallel?
7.12 For the \(\xi_{i} \eta_{i} \zeta_{i}\) and \(\xi_{j} \eta_{j} \zeta_{j}\) coordinate systems, what is (are) the necessary and sufficient condition(s) on the Euler parameters for the following conditions to be true:
(a) The \(\xi_{i}\) and \(\xi_{j}\) axes remain parallel.
(b) The \(\zeta_{i}\) and \(\eta_{j}\) axes remain parallel.
(c) The \(\eta_{i}\) and \(\eta_{j}\) axes remain perpendicular.
7.13 Find the constraint equation for point \(P\) on body \(i\) to remain on the plane \(z=6\). The local coordinates of \(P\) are \(\mathbf{s}_{i}^{\prime P}=[2,-2,-1]^{T}\).
7.14 Repeat Prob. 7.13 and constrain point \(P\) to move on the line described by the intersection of the two planes \(z=6\) and \(x=2\).
7.15 Figure P.7.15 shows a mechanism consisting of two bodies connected by a spherical joint. The vectors locating the center of the joint have components \(s_{i}^{\prime P}=[0.4,1.4,-0.5]^{T}\) and \(\mathbf{s}_{j}^{\prime P}=[0.9,-0.3,-0.2]^{T}\). Write the constraint equations for this joint. If the two bodies are not connected to any other bodies, how many degrees of freedom does the system have?


Figure P. 7.15
7.16 If two points \(P\) and \(Q\) are chosen arbitrarily on a revolute-joint axis connecting two bodies \(i\) and \(j\), Eq. 7.7 may be repeated twice - for \(P\) and for \(Q\) - to yield six constraint equations. These equations can be used instead of Eq. 7.10. However, since there is one relative degree of freedom between the two bodies, one of the equations must be redundant. Define a strategy to eliminate one of the six equations efficiently in every possible configuration.
7.17 Two bodies are connected by a revolute joint. Find the constraint equation for this joint, using the spherical joint constraint and scalar products, instead of the vector product of Eq. 7.10. Hint: Initially define two vectors on one of the bodies perpendicular to the joint axis.
7.18 Add one equation to the cylindrical-joint constraints of Eq. 7.13 to allow them to be used as revolute-joint constraint equations.
7.19 The two bodies shown in Fig. P.7.19 can slide relative to each other without separation. Formulate constraint equations describing this type of joint.

7.20 Use the concept of relative axis of rotation (relative set of Euler parameters) to find an additional equation to convert cylindrical-joint constraints to translational-joint constraints.
7.21 Consider the system shown in Fig. 7.9, which contains two spherical joints. Determine the number of degrees of freedom if:
(a) The system is modeled by three bodies and two spherical joints.
(b) The system is modeled by two bodies and one spherical-spherical joint.

Compare the results from (a) and (b). Explain why they are different.
7.22 An A-arm suspension system contains a link connecting the main chassis (body \(i\) ) to the wheel (body \(j\) ) by two revolute joints as shown in Fig. P.7.22. The two joint axes intersect at an angle \(\theta=90^{\circ}\). Determine a set of constraint equations to model this composite revolute-revolute joint.


Figure P. 7.22
7.23 Repeat Prob. 7.22 for \(\theta \neq 90^{\circ}\).
7.24 The steering command in automobile simulations can be provided as a time-dependent constraint equation. This is usually done when the simulation of the actual steering mechanism is not of interest. Assume that body \(i\) is the main chassis, with \(\eta_{i}\) along the longitudinal direction and body \(j\) as one of the front wheels. In Figure P.7.24, the suspension mechanism between the wheel and the chassis is not shown. If the steering command is described as \(c(t)\), derive a constraint equation between two unit vectors along the \(\eta_{i}\) and \(\eta_{j}\) axes.

7.25 Verify the entries of the Jacobian matrix listed in Table 7.1.
7.26 Derive expressions for vector \(\gamma\) for the constraint equations that are listed in Table 7.1
7.27 Verify the entries of the modified Jacobian matrix and modified vector \(\boldsymbol{\gamma}\) listed in Table 7.2.
7.28 The road wheels of a tracked vehicle are connected to the chassis by road arms as shown in Fig. P.7.28. A road arm can be modeled as a composite revolute-revolute joint with parallel axes. The constraint equations for this composite joint may be simplified by locating the local coordinate systems on the chassis and on the wheels so that they have parallel axes; e.g., the \(\xi\) axes could be parallel as shown coming out of the plane. Derive the simplified constraint equations.


Figure P. 7.28

\title{
Basic Concepts
}

\section*{in}

\section*{Dynamics}

The basic concepts and laws of dynamics are best introduced by beginning with particle dynamics. To derive the equations of motion for both unconstrained and constrained systems of bodies, only two of Newton's laws of motion for a single particle are needed as postulates.

\subsection*{8.1 DYNAMICS OF A PARTICLE}

The simplest body arising in the study of motion is a particle, or point mass, defined here as a mass concentrated at a point. While mass in reality is generally distributed in space, the notion that a body has all its mass concentrated at a point is an adequate approximation for many purposes. Further, as will be seen shortly, the centroid of a complex distribution of mass behaves as a point mass. Thus, analysis of the behavior of a point mass leads to useful results, even for complex systems.

Newton's first law of motion for a particle relates the total force \(\vec{f}\) acting on the particle, the mass \(m^{(p)}\) of the particle, and the acceleration \(\vec{a}\) of the particle, as follows (see Fig. 8.1):
\[
\begin{equation*}
\vec{f}=m^{(p)} \vec{a} \tag{8.1}
\end{equation*}
\]


Figure 8.1 A particle moving in a global coordinate system.
where it is assumed that a consistent system of units is used. From this equation, a condition for particle equilibrium may be deduced. A particle remains at rest (in equilibrium) or in a state of constant velocity if and only if the total force \(\vec{f}\) acting on the particle is \(\overrightarrow{0}\).

Newton's second law of motion, which is the law of action and reaction, states that when two particles exert forces on each other, these interacting forces are equal in magnitude, opposite in sense, and directed along the straight line joining the particles.

The vector form of Newton's law of motion for a particle in Eq. 8.1 can be written in terms of the components of vectors \(\vec{f}\) and \(\vec{a}\). If we denote the force \(\vec{f}\) as \(\mathbf{f}=\) [ \(\left.f_{(x)}, f_{(y)}, f_{(z)}\right]^{T}\), then for a particle located in the \(x y z\) coordinate system with position vector \(\vec{r}\), Eq. 8.1 becomes
\[
\begin{equation*}
\mathbf{f}=m^{(p)} \ddot{\mathbf{r}} \tag{8.2}
\end{equation*}
\]

Throughout the preceding discussion of Newton's laws of motion, it is presumed that position, and hence acceleration, is measured in an inertial reference frame (a global \(x y z\) ) coordinate system. Such a reference frame should technically be defined as a coordinate system fixed in the stars. For most engineering purposes, an adequate reference frame is an earth-fixed reference system. It is important, however, to note that for applications concerning space dynamics, or even in long-range trajectories, the rotation of the earth has a significant effect on the precision with which points can be located by means of Newton's equations of motion, and an earth-fixed reference system may be inadequate.

\subsection*{8.2 DYNAMICS OF A SYSTEM OF PARTICLES}

The governing laws of the dynamics of individual particles are now extended to systems of interacting particles. The equations of motion for such systems can be written simply as the collection of equations of motion for all the particles taken individually. If the forces acting between particles are readily characterized, this method is practical and direct. The more common situation in mechanical system dynamics, however, involves constraints among systems of particles; thus the forces acting between particles are usually not so readily determined. For this reason we introduce the concept of the center of mass.

Consider the system of \(p\) particles shown schematically in Fig. 8.2. Particle \(i\) has mass \(m_{i}^{(p)}\) and is located by a position vector \(\vec{r}_{i}\) directed from the origin of an inertial reference frame to the particle. The forces acting on each particle include an externally applied force \(\vec{f}_{i}\) and internal forces of interaction between particles \(\vec{f}_{i j}, j \neq i\), where \(\vec{f}_{i i}=\overrightarrow{0}\). The total force acting on the \(i\) th particle is the summation of external and internal forces. Thus, for body \(i\), Newton's first law of motion (Eq. 8.2) becomes
\[
\begin{equation*}
m_{i}^{(p)} \ddot{\mathbf{r}}_{i}=\mathbf{f}_{i}+\sum_{j=1}^{p} \mathbf{f}_{i j}, \quad i=1, \ldots, p \tag{a}
\end{equation*}
\]

This system of equations describes the motion of the system of \(p\) particles.
Since the forces of interaction between bodies in a system must satisfy Newton's law of action and reaction, the force on body \(i\) due to body \(j\) must be equal to the negative of the force acting on body \(j\) due to body \(i\); i.e.,
\[
\begin{equation*}
\mathbf{f}_{i j}=-\mathbf{f}_{j i} \quad i, j=1, \ldots, p \tag{b}
\end{equation*}
\]

Note that since \(\mathbf{f}_{i i}=0\), Eq. \(b\) holds also for \(i=j\).


Figure 8.2 System of \(p\) particles.
If the external forces acting on each particle are known and the nature of the force acting between bodies \(i\) and \(j\) is known, the system of equations in Eq. \(a\) may be written explicitly. However, the force of interaction between particles will generally depend on the positions of the particles.

Since Eq. \(a\) must hold for each particle in the system, this system of equations can be summed to obtain
\[
\begin{equation*}
\sum_{i=1}^{p} m_{i}^{(p)} \ddot{\mathbf{i}}_{i}=\sum_{i=1}^{p} \mathbf{f}_{i}+\sum_{i=1}^{p} \sum_{j=1}^{p} \mathbf{f}_{i j} \tag{c}
\end{equation*}
\]
which is valid for the entire system of particles. The double sum of Eq. c contains both \(\mathbf{f}_{i j}\) and \(\mathbf{f}_{j i}\), and hence from Eq. \(b\) it is found that
\[
\begin{equation*}
\sum_{i=1}^{p} \sum_{j=1}^{p} \mathbf{f}_{i j}=\mathbf{0} \tag{d}
\end{equation*}
\]

Thus, Eq. \(c\) reduces to
\[
\begin{equation*}
\sum_{i=1}^{p} m_{i}^{(p)} \mathbf{r}_{i}=\sum_{i=1}^{p} \mathbf{f}_{i} \tag{e}
\end{equation*}
\]

We define the total mass of the system as the sum of the individual masses,
\[
\begin{equation*}
m=\sum_{i=1}^{p} m_{i}^{(p)} \tag{8.3}
\end{equation*}
\]
and the center of mass, or centroid, of the system of particles as
\[
\begin{equation*}
\mathbf{r}=\frac{1}{m} \sum_{i=1}^{p} m_{i}^{(p)} \mathbf{r}_{i} \tag{8.4}
\end{equation*}
\]
where \(\vec{r}\) is the vector from the origin to the center of mass. Since the total mass and each of the component masses are constant, Eq. 8.4 can be differentiated twice with respect to time to obtain
\[
\begin{equation*}
m \ddot{\mathbf{r}}=\sum_{i=1}^{p} m_{i}^{(p)} \ddot{\mathbf{r}}_{i} \tag{f}
\end{equation*}
\]

Finally, if the total external force acting on the system of particles is defined as
\[
\begin{equation*}
\mathbf{f}=\sum_{i=1}^{p} \mathbf{f}_{i} \tag{8.5}
\end{equation*}
\]
then Eqs. \(e\) and \(f\), and Eq. 8.5 yield
\[
\begin{equation*}
m \ddot{\mathbf{r}}=\mathbf{f} \tag{8.6}
\end{equation*}
\]

This result states that the resultant of the external forces on any system of mass equals the total mass of the system times the acceleration of the center of mass. That is, the center of mass moves as if it were a particle of mass \(m\) under the action of the force \(\mathbf{f}\).

\subsection*{8.3 DYNAMICS OF A BODY}

A body can be regarded as a collection of a very large number of particles. In addition, from the definition of rigidity, the location of the particles in a body relative to one another remains unchanged.

In the discussion of the dynamics of a system of particles, the translational equation of motion was derived as
\[
\mathbf{f}=m \ddot{\mathbf{r}}
\]

Since a body is a particular case of a system of particles, Eq. 8.5 also applies to bodies.
The definition of the center of mass or centroid of a body is found from Eq. 8.4. The summation over the particles is replaced by an integral over the body volume, and the mass of the individual particle is replaced by the infinitesimal mass \(d m\) :
\[
\begin{equation*}
\mathbf{r}=\frac{1}{m} \int_{(v)} \mathbf{r}^{p} d m \tag{8.7}
\end{equation*}
\]
where \(\mathbf{r}^{p}\) locates an infinitesimal mass, as shown in Fig. 8.3. Vector \(\mathbf{r}^{p}\) is the sum of two vectors:
\[
\begin{equation*}
\mathbf{r}^{P}=\mathbf{r}+\mathbf{s} \tag{8.8}
\end{equation*}
\]


Figure 8.3 A body as a collection of infinitesimal masses.

Substituting Eq. 8.8 into Eq. 8.7 yields
\[
\begin{aligned}
\mathbf{r} & =\frac{1}{m} \int_{(v)}(\mathbf{r}+\mathbf{s}) d m \\
& =\mathbf{r}+\frac{1}{m} \int_{(v)} \mathbf{s} d m
\end{aligned}
\]
or
\[
\begin{equation*}
\int_{(v)} \mathbf{s} d m=0 \tag{8.9}
\end{equation*}
\]

The first and second time derivatives of Eq. 8.9 are
\[
\begin{equation*}
\int_{(v)} \dot{\mathbf{s}} d m=0 \tag{8.10}
\end{equation*}
\]
and
\[
\begin{equation*}
\int_{(v)} \ddot{\mathbf{s}} d m=\mathbf{0} \tag{8.11}
\end{equation*}
\]

Equations 8.9 through 8.11 will become useful in the following sections in deriving the equations of motion.

In addition to its tendency to move a body in the direction of its application, a force also tends to rotate the body about any axis that does not intersect the line of action of the force and which is not parallel to it. The measure of this tendency is known as the moment of the force about the given axis. The moment of a force is also frequently referred to as torque. The rotational equation of motion for a body will be derived in Sec. 8.3.2.

\subsection*{8.3.1 Moments and Couples}

Consider a force \(\vec{f}_{i}\) acting on a body, as shown in Fig. 8.4, and a point \(O\) not on the line of action of the force. If a vector \(\vec{s}\) is introduced from \(O\) to any point on the line of action of \(\vec{f}_{i}\), the moment \(\vec{n}_{i}^{O}\) is found from the vector product:
\[
\begin{equation*}
\mathbf{n}_{i}^{O}=\tilde{\mathbf{s}_{i}} \tag{8.12}
\end{equation*}
\]

The principle of moments is easily proved by applying the distributive law for the sum of vector products. A system of forces \(\vec{f}_{\mathrm{i}}, i=1, \ldots, k\), is shown in Fig. 8.5(a) concur-


Figure 8.4 The moment of a force \(\vec{f}_{i}\) about point \(O\).


Figure 8.5 (a) A force system acting at point \(A\). (b) The equivalent system.
rent at point \(A\) whose position vector from point \(O\) is \(\vec{s}\). The sum of moments about \(O\) is found to be
\[
\begin{equation*}
\mathbf{n}^{o}=\tilde{\mathbf{s}} \mathbf{f} \tag{8.13}
\end{equation*}
\]
where \(n^{o}=\sum_{i=1}^{k} \mathbf{n}_{i}^{o}\) and \(\mathbf{f}=\sum_{i=1}^{k} \mathbf{f}_{i}\). Thus the sum of the moments of a system of concurrent forces (forces all of which act at a point) about a given point is equal to the moment of their sum about the same point, as shown in Fig. 8.5 (b).

The moment produced by two equal and opposite and noncollinear forces is known as a couple. Couples have certain unique properties. Figure 8.6 shows two equal and opposite forces \(\vec{f}\) and \(-\vec{f}\) acting on a body. The vector \(\vec{s}\) joins any point \(B\) on the line of action of \(-\vec{f}\) to any point \(A\) on the line of action of \(\vec{f}\). Points \(A\) and \(B\) are located by position vectors \(\vec{s}^{A}\) and \(\vec{s}^{B}\), from any point \(O\). The combined moment of the two forces about \(O\) is
\[
\begin{aligned}
\mathbf{n} & =\tilde{\mathbf{s}}^{A} \mathbf{f}+\tilde{\mathbf{s}}^{B}(-f) \\
& =\left(\tilde{\mathbf{s}}^{A}-\tilde{\mathbf{s}}^{B}\right) \mathbf{f}
\end{aligned}
\]

Since \(\mathbf{s}^{A}-\mathbf{s}^{B}=\mathbf{s}\), the moment of the couple becomes
\[
\begin{equation*}
\mathbf{n}=\tilde{\mathbf{s}} \mathbf{f} \tag{8.14}
\end{equation*}
\]

Thus, the moment of a couple is the same about all points. Note that the magnitude of \(\mathbf{n}\) is \(n=f d\), where \(d\) is the perpendicular distance between the lines of action of the two


Figure 8.6 The moment of a couple.
forces. It is clear that the moment of a couple is a free vector, whereas the moment of a force about a point, which is also the moment about a defined axis through the point, is a sliding vector whose direction is along the axis through the point.

The effect of a force acting on a body has been described in terms of its tendency to move the body in the direction of the force and to rotate the body about any axis that does not intersect the line of the force. The representation of this dual effect is often facilitated by replacing the given force by an equal parallel force and a couple to compensate for the change in the moment of the force. This resolution of a force into a force and a couple is illustrated in Fig. 8.7, where the couple has a magnitude \(n=f d\).


Figure 8.7 (a) A force \(\vec{f}\) acting at point \(A\). (b) Two forces \(\vec{f}\) and \(-\vec{f}\) acting at point \(B\) have no effect on the body, since they cancel each other. (c) The couple \(\vec{f}\) acting at \(A\) and \(-\vec{f}\) acting at \(B\) are replaced by the moment \(\vec{n}\).

To study the motion of a body, it is often convenient to replace the forces acting on the body by an equivalent system consisting of one force and one couple, as shown in Fig. 8.8. The force \(\mathbf{f}\) that acts at the centroid of the body is equal to the sum of all \(k\) forces acting on the body; i.e.,
\[
\begin{equation*}
\mathbf{f}=\sum_{i=1}^{k} \mathbf{f}_{i} \tag{8.15}
\end{equation*}
\]

(a)

(b)

Figure 8.8 (a) A system of forces acting on a body. (b) Equivalent force-couple system.

The moment of the couple is the sum of the moments of the individual forces with respect to the centroid \(C\); i.e.,
\[
\begin{equation*}
\mathbf{n}=\sum_{i=1}^{k} \mathbf{n}_{i}=\sum_{i=1}^{k} \bar{\Phi}_{i} \mathbf{f}_{i} \tag{8.16}
\end{equation*}
\]

Note that \(\boldsymbol{f}\) is acting at the centroid but \(\mathbf{n}\) is a free vector.

\subsection*{8.3.2 Rotational Equations of Motion}

When the origin of the body-fixed \(\xi \eta \zeta\) coordinate system is located at the center of mass of a body, the system is what is known as a centroidal coordinate system. Consider the body shown in Fig. 8.9, where the only external force is a force \(\vec{f}\) acting on the \(i\) th particle of the body. This particle is located with respect to the \(x y z\) coordinate system by
\[
\begin{equation*}
\mathbf{r}_{i}^{p}=\mathbf{r}+\mathbf{s}_{i} \tag{a}
\end{equation*}
\]
or, by expanding the vectors of both sides of Eq. \(a\) into skew-symmetric matrices yields
\[
\begin{equation*}
\tilde{\mathbf{r}}_{i}^{p}=\tilde{\mathbf{r}}+\overline{\mathbf{s}}_{i} \tag{b}
\end{equation*}
\]

Postmultiplying this equation by \(\mathbf{f}\) yields
\[
\begin{equation*}
\tilde{\mathbf{r}}_{i}^{P} \mathbf{f}=\tilde{\mathbf{r}} \mathbf{f}+\tilde{\mathbf{s}}_{i} \mathbf{f} \tag{c}
\end{equation*}
\]
or
\[
\begin{equation*}
\mathbf{n}^{G}=\overline{\mathbf{r}} \mathbf{f}+\mathbf{n} \tag{8.17}
\end{equation*}
\]
where \(\mathbf{n}^{G}\) and \(\mathbf{n}\) are the moments of \(\mathbf{f}\) with respect to the origin of the \(x y z\) coordinate system and the body centroid, respectively.

In addition to the external force \(\vec{f}\) acting on the \(i\) th particle, there are internal forces acting between the particles, such as \(\vec{f}_{\mathrm{ij}}\) acting on the \(i\) th particle by the \(j\) th particle. The equation of motion for the \(i\) th particle can be written as
\[
\begin{equation*}
\mathbf{f}+\left(\mathbf{f}_{i j}+\cdots\right)=m_{i}^{(p) \dot{r}_{i}^{P}} \tag{d}
\end{equation*}
\]


Figure 8.9 A body as a collection of particles under the action of a force \(\vec{f}\).
where the terms in the parentheses represent the reaction forces from all other particles of the body acting on the \(i\) th particle. Premultiplying Eq. \(d\) by \(\tilde{\mathbf{r}}_{i}^{P}\) results in
\[
\begin{equation*}
\tilde{\mathbf{r}}_{i}^{P} \mathbf{f}+\left(\tilde{\mathbf{r}}_{i}^{P} \mathbf{f}_{i j}+\cdots\right)=m_{i}^{(p)} \tilde{\mathbf{r}}_{i}^{P} \ddot{\mathbf{r}}_{i}^{P} \tag{e}
\end{equation*}
\]

For the other particles in the body, such as the \(j\) th particle, on which we have assumed only reaction forces are acting, the equations of motion have the general form
\[
\begin{equation*}
\left(\mathbf{f}_{j i}+\cdots\right)=m_{j}^{(p) \ddot{r}_{j}^{P}}, \quad j=1, \ldots, k-1, j \neq i \tag{f}
\end{equation*}
\]

Premultiplying Eq. \(f\) by \(\tilde{\mathbf{r}}_{j}^{P}\) yields
\[
\begin{equation*}
\left(\tilde{\mathbf{r}}_{j}^{P} \mathbf{f}_{j i}+\cdots\right)=m_{j}^{(p)} \tilde{\mathbf{r}}_{j}^{P} \dot{\mathbf{r}}_{j}^{P}, \quad j=1, \ldots, k-1, j \neq i \tag{g}
\end{equation*}
\]

Summing Eq. \(e\) and all of the \(k-1\) equations in Eq. \(g\) results in
\[
\begin{equation*}
\tilde{\mathbf{r}}_{i}^{P} \mathbf{f}+\left(\tilde{\mathbf{r}}_{i}^{P} \mathbf{f}_{i j}+\tilde{\mathbf{r}}_{j}^{P} \mathbf{f}_{j i}+\cdots\right)=m_{i}^{(p)} \tilde{\mathbf{r}}_{i}^{P} \dot{\mathbf{r}}_{i}^{P}+m_{j}^{(p)} \tilde{\mathbf{r}}_{j}^{P} \dot{\mathbf{r}}_{j}^{P}+\cdots \tag{h}
\end{equation*}
\]

For every \(\mathbf{f}_{i j}\), there is an \(\mathbf{f}_{j i}=-\mathbf{f}_{i j}\), and therefore Eq. \(h\) becomes
\[
\begin{equation*}
\mathbf{r}_{i}^{P} \mathbf{f}+\left[\left(\tilde{\mathbf{r}}_{i}^{P}-\tilde{\mathbf{r}}_{j}^{P}\right) \mathbf{f}_{i j}+\cdots\right]=m_{i}^{(p)} \tilde{\mathbf{r}}_{i}^{P} \ddot{\mathbf{r}}_{i}^{P}+m_{j}^{(p)} \dot{\mathbf{r}}_{j}^{P} \dot{\mathbf{r}}_{j}^{P}+\cdots \tag{i}
\end{equation*}
\]

All vector product terms in the parentheses of Eq. \(i\) are identical to zero, since any typical term \(\left(\tilde{\mathbf{r}}_{i}^{P}-\tilde{\mathbf{r}}_{j}^{P}\right) \mathbf{f}_{i j}\) is zero (vectors \(\mathbf{f}_{i j}\) and \(\mathbf{r}_{i}^{P}-\mathbf{r}_{j}^{P}\) are collinear, as illustrated in Fig. 8.10). Therefore, Eq. \(i\) is simplified to
\[
\begin{aligned}
\tilde{\mathbf{r}}_{i}^{P} \mathbf{f} & =m_{i}^{(p)} \tilde{\mathbf{r}}_{i}^{P} \ddot{\mathbf{r}}_{i}^{P}+m_{j}^{(p)} \tilde{\mathbf{r}}_{j}^{P} \ddot{\mathbf{r}}_{j}^{P}+\cdots \\
& =\sum_{j=1}^{k} m_{j}^{(p)} \tilde{\mathbf{r}}_{j}^{P} \mathbf{r}_{j}^{P}
\end{aligned}
\]
or
\[
\begin{equation*}
\mathbf{n}^{G}=\sum_{j=1}^{k} m_{j}^{(p)} \dot{\mathbf{r}}_{j}^{p} \ddot{\mathbf{r}}_{j}^{p} \tag{j}
\end{equation*}
\]

If the mass of any typical particle is replaced by an infinitesimal mass \(d m\), then the summation can be replaced by an integral over the volume of the body. Hence Eq. \(j\) becomes
\[
\begin{equation*}
\mathbf{n}^{G}=\int_{(v)} \tilde{\mathbf{r}}^{P} \dot{\mathbf{r}}^{P} d m \tag{k}
\end{equation*}
\]


Figure 8.10 Reaction forces between two particles.

Taking \(\mathbf{r}^{P}=\mathbf{r}+\mathbf{s}\) and substituting in Eq. \(k\) yields
\[
\begin{align*}
\mathbf{n}^{G} & =\int_{(v)}(\tilde{\mathbf{r}}+\tilde{\mathbf{s}})(\ddot{\mathbf{r}}+\ddot{\mathbf{s}}) d m \\
& =\tilde{\mathbf{r}} \ddot{\mathbf{r}} \int_{(v)} d m+\int_{(v)} \tilde{\mathbf{s}} \ddot{\mathbf{s}} d m \\
& =m \tilde{\mathbf{r}} \ddot{\mathbf{r}}+\int_{(v)} \tilde{\mathbf{s} \mathbf{S}} d m \tag{8.18}
\end{align*}
\]
where Eqs. 8.9 and 8.11 have been employed.
From Eq. 6.101, i.e., \(\dot{\mathbf{s}}=\boldsymbol{\omega} s\), it is found that
\[
\begin{align*}
\ddot{\mathbf{s}} & =\tilde{\boldsymbol{\omega}} \mathbf{S}+\tilde{\boldsymbol{\omega}} \dot{\mathbf{S}} \\
& =-\tilde{\mathbf{s}} \dot{\boldsymbol{\omega}}+\tilde{\boldsymbol{\omega}} \tilde{\boldsymbol{\omega}} \mathbf{S} \tag{8.19}
\end{align*}
\]

Premultiplying Eq. 8.19 by \(\tilde{\mathbf{s}}\) and rearranging one term yields
\[
\tilde{\mathbf{s}} \tilde{\mathbf{S}}=-\tilde{\mathbf{s}} \tilde{\boldsymbol{s}} \dot{\omega}-\tilde{\mathbf{s}} \tilde{\boldsymbol{\omega}} \tilde{\mathbf{S}} \boldsymbol{\omega}
\]

From Eq. 2.52, it is found that
\[
\begin{aligned}
\tilde{\mathbf{s}} \tilde{\boldsymbol{\omega}} \tilde{\mathbf{s}} \boldsymbol{\omega} & =[\tilde{\boldsymbol{\omega}} \tilde{\mathbf{s}}+(\tilde{\mathbf{s}} \boldsymbol{\omega})] \tilde{\mathbf{s}} \boldsymbol{\omega} \\
& =\tilde{\boldsymbol{\omega}} \tilde{\mathbf{s}} \boldsymbol{\tilde { s }} \boldsymbol{\omega}
\end{aligned}
\]

Hence
\[
\begin{equation*}
\tilde{\mathbf{s}} \tilde{\mathbf{S}}=-\tilde{\mathbf{s}} \tilde{\mathbf{s}} \dot{\omega}-\tilde{\omega} \tilde{\mathbf{s}} \tilde{\mathbf{s}} \boldsymbol{\omega} \tag{l}
\end{equation*}
\]

Substitution of Eq. \(l\) into Eq. 8.18 yields
\[
\begin{align*}
\mathbf{n}^{G} & =m \tilde{\mathbf{r}} \ddot{\mathbf{r}}+\left(-\int_{(v)} \tilde{\mathbf{s}} \tilde{\mathbf{s}} d m\right) \dot{\boldsymbol{\omega}}+\tilde{\boldsymbol{\omega}}\left(-\int_{(v)} \tilde{\mathbf{s}} \tilde{\mathbf{s}} d m\right) \boldsymbol{\omega} \\
& =\tilde{\mathbf{r}} \mathbf{f}+\mathbf{J} \dot{\boldsymbol{\omega}}+\tilde{\boldsymbol{\omega}} \mathbf{J} \boldsymbol{\omega} \tag{8.20}
\end{align*}
\]
where \(\mathbf{J}=-\int_{(v)} \tilde{\mathbf{s}} \tilde{\mathbf{s}} d m\) is defined as the global inertia tensor for the body. Comparison of Eq. 8.17 and 8.20 results in
\[
\begin{equation*}
\mathbf{n}=\mathbf{J} \dot{\omega}+\tilde{\omega} \mathbf{J} \boldsymbol{\omega} \tag{8.21}
\end{equation*}
\]

Equation 8.21 is the rotational equation of motion for a body.

\subsection*{8.3.3 The Inertia Tensor}

Figure 8.11 shows a body with its centroidal body-fixed coordinate system. It is assumed that the body has volume \(v\). Vector \(\vec{s}\) in the local coordinate system is described as \(\mathbf{s}^{\prime}=\left[s_{(\xi)}, s_{(\eta)}, s_{(\xi)}\right]^{T}\). The inertia tensor is defined as the integral
\[
\begin{equation*}
\mathbf{J}^{\prime}=-\int_{(v)} \tilde{\mathbf{s}}^{\prime} \tilde{\mathbf{s}}^{\prime} d m \tag{8.22}
\end{equation*}
\]
which can be written in expanded form as
\[
\mathbf{J}^{\prime}=\int_{(v)}\left[\begin{array}{lll}
s_{(\eta)}^{2}+s_{(\xi)}^{2} & -s_{(\xi)} s_{(\eta)} & -s_{(\xi)} s_{(\xi)}  \tag{8.23}\\
-s_{\left.(\eta)^{\prime}\right)}^{s_{(\xi)}} & s_{(\xi)}^{2}+s_{(\xi)}^{2} & -s_{(\eta)^{\prime}(\xi)}^{s_{(\xi)}} \\
-s_{(\zeta)} s_{(\xi)} & -s_{(\zeta)} s_{(\eta)} & s_{(\xi)}^{2}+s_{(\eta)}^{2}
\end{array}\right] d m
\]


Figure 8.11 A body with centroidal coordinate system.

The following individual integrals are defined:
\[
\begin{align*}
& j_{\xi \xi}=\int_{(v)}\left(s_{(\eta)}^{2}+s_{(\xi)}^{2}\right) d m \\
& j_{\eta \eta}=\int_{(v)}\left(s_{(\xi)}^{2}+s_{(\xi)}^{2}\right) d m \\
& j_{\zeta \zeta}=\int_{(v)}\left(s_{(\xi)}^{2}+s_{(\eta)}^{2}\right) d m  \tag{8.24}\\
& j_{\xi \eta}=j_{\eta \xi}=-\int_{(v)} s_{(\xi)} s_{(\eta)} d m \\
& j_{\eta \zeta}=j_{\zeta \eta}=-\int_{(v)} s_{(\eta)} s_{(\zeta)} d m \\
& j_{\zeta \xi}=j_{\xi \zeta}=-\int_{(v)} s_{(\xi)} s_{(\xi)} d m
\end{align*}
\]
where \(j_{\xi \xi}, j_{\eta \eta}\), and \(j_{\xi \zeta}\) are called the moments of inertia and \(j_{\xi \eta}, j_{\eta \xi}, j_{\zeta \eta}, j_{\eta \zeta}, j_{\zeta \xi}\), and \(j_{\xi \xi}\) are called the products of inertia. Then Eq. 8.22 is written as
\[
\mathbf{J}^{\prime}=\left[\begin{array}{lll}
j_{\xi \xi} & j_{\xi \eta} & j_{\xi \zeta}  \tag{8.25}\\
j_{\eta \xi} & j_{\eta \eta} & j_{\eta \zeta} \\
j_{\zeta \xi} & j_{\zeta \eta} & j_{\zeta \zeta}
\end{array}\right]
\]

The matrix \(\mathbf{J}^{\prime}\) is called the inertia tensor (inertia matrix) for the body.
If the orientation of the centroidal \(\xi \eta \zeta\) body-fixed axes is changed, the moments and products of inertia will change in value. There is one unique orientation of the \(\xi \eta \zeta\) axes for which the products of inertia vanish. For this orientation the inertia matrix takes the diagonal form
\[
\begin{equation*}
\mathbf{J}^{\prime}=\operatorname{diag}\left[j_{\xi \xi}, j_{\eta \eta}, j_{\xi \xi}\right] \tag{8.26}
\end{equation*}
\]

The \(\xi \eta \zeta\) axes for which the products of inertia vanish are called the principal axes of inertia.

Another form of the global inertia tensor \(\mathbf{J}\) as defined in Eq. 8.20 can be derived as follows:
\[
\begin{align*}
\mathbf{J} & =-\int_{(v)} \tilde{\mathbf{s}} \tilde{\boldsymbol{s}} d m \\
& =-\int_{(v)} \mathbf{A} \tilde{\mathbf{s}}^{\prime} \mathbf{A}^{T} \mathbf{A} \tilde{\mathbf{s}}^{\prime} \mathbf{A}^{T} d m \\
& =\mathbf{A}\left(-\int_{\tilde{\mathbf{s}}^{\prime} \tilde{\mathbf{s}}^{\prime}} d m\right) \mathbf{A}^{T} \\
& =\mathbf{A} \mathbf{J}^{\prime} \mathbf{A}^{T} \tag{8.27}
\end{align*}
\]

In contrast to \(\mathbf{J}^{\prime}\), which is a constant matrix, \(\mathbf{J}\) is a function of the angular orientation of the body.

The time derivatives of \(\mathbf{J}^{\prime}\) and \(\mathbf{J}\) are
\[
\begin{equation*}
\dot{J}^{\prime}=\mathbf{0} \tag{8.28}
\end{equation*}
\]
and
\[
\begin{align*}
\dot{\mathbf{J}} & =\dot{\mathbf{A}} \mathbf{J}^{\prime} \mathbf{A}^{T}+\mathbf{A} \mathbf{J}^{\prime} \dot{\mathbf{A}}^{T} \\
& =\tilde{\boldsymbol{\omega}} \mathbf{A} \mathbf{J}^{\prime} \mathbf{A}^{T}+\mathbf{A} \mathbf{J}^{\prime} \mathbf{A}^{T} \tilde{\boldsymbol{\omega}}^{T} \\
& =\tilde{\boldsymbol{\omega}} \mathbf{J}-\mathbf{J} \tilde{\boldsymbol{\omega}} \tag{8.29}
\end{align*}
\]
where Eq. 6.94 has been employed.
The rotational equations of motion for a body are given by Eq. 8.21 in terms of the global inertia tensor. The equations represented there can be converted to use the local components of the vectors by taking \(\mathbf{n}=\mathbf{A n} \mathbf{n}^{\prime}, \boldsymbol{\omega}=\mathbf{A} \boldsymbol{\omega}^{\prime}\), and \(\dot{\boldsymbol{\omega}}=\dot{\mathbf{A}} \boldsymbol{\omega}^{\prime}+\mathbf{A} \dot{\boldsymbol{\omega}}^{\prime}=\) \(\mathbf{A} \tilde{\boldsymbol{\omega}}^{\prime} \boldsymbol{\omega}^{\prime}+\mathbf{A} \dot{\boldsymbol{\omega}}^{\prime}=\mathbf{A} \dot{\boldsymbol{\omega}}^{\prime}\) and substituting in Eq. 8.21 to get
\[
\mathbf{A} \mathbf{n}^{\prime}=\mathbf{J} \mathbf{A} \dot{\boldsymbol{\omega}}^{\prime}+\mathbf{A} \tilde{\boldsymbol{\omega}}^{\prime} \mathbf{A}^{\tau} \mathbf{J} \mathbf{A} \boldsymbol{\omega}^{\prime}
\]

Premultiplication by \(\mathbf{A}^{T}\) and application of Eq. 8.27 yield
\[
\begin{equation*}
\mathbf{n}^{\prime}=\mathbf{J}^{\prime} \dot{\boldsymbol{\omega}}^{\prime}+\dot{\boldsymbol{\omega}}^{\prime} \mathbf{J}^{\prime} \boldsymbol{\omega}^{\prime} \tag{8.30}
\end{equation*}
\]

This represents the rotational equations of motion for a body, which are known as Euler's equations of motion.

\subsection*{8.3.4 An Unconstrained Body}

Consider the body drawn solid in Fig. 8.12(a), which has no contact with any other body except through force elements. There is no kinematic joint attached to this body to eliminate any of its degrees of freedom. A typical free-body diagram for this body is shown in Fig. 8.12(b). If the sum of all forces acting on the body is denoted by \(\mathbf{f}_{i}\) and the sum of the moment of \(\mathbf{f}_{i}\) and any other pure moments acting on the body is denoted by \(\mathbf{n}_{i}\), then the translational and rotational equations of motion for this body are given, from Eqs. 8.6 and 8.30, as
\[
\begin{gather*}
m_{i} \ddot{\mathbf{r}}_{i}=\mathbf{f}_{i}  \tag{8.31}\\
\mathbf{J}_{i}^{\prime} \dot{\boldsymbol{\omega}}_{i}^{\prime}+\tilde{\boldsymbol{\omega}}_{i}^{\prime} \mathbf{J}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime}=\mathbf{n}_{i}^{\prime} \tag{8.32}
\end{gather*}
\]


Figure 8.12 (a) An unconstrained body, and (b) its equivalent free-body diagram.
Equations 8.31 and 8.32 are the so-called Newton-Euler equations of motion for an unconstrained body.

Equations 8.31 and 8.32 can be expressed in matrix form as
\[
\left[\begin{array}{ll}
\mathbf{N} & \mathbf{0}  \tag{8.33}\\
\mathbf{0} & \mathbf{J}^{\prime}
\end{array}\right]_{i}\left[\begin{array}{l}
\ddot{\mathbf{r}} \\
\dot{\boldsymbol{\omega}}^{\prime}
\end{array}\right]_{i}+\left[\begin{array}{c}
\boldsymbol{0} \\
\overline{\boldsymbol{\omega}}^{\prime} \mathbf{J}^{\prime} \boldsymbol{\omega}^{\prime}
\end{array}\right]_{i}=\left[\begin{array}{l}
\mathbf{f} \\
\mathbf{n}^{\prime}
\end{array}\right]_{i}
\]
where
\[
\begin{equation*}
\mathbf{N}_{i}=\operatorname{diag}[m, m, m]_{i} \tag{8.34}
\end{equation*}
\]

Equation 8.33 can also be written in the compact form
\[
\begin{equation*}
\mathbf{M}_{i} \dot{\mathbf{h}}_{i}+\mathbf{b}_{i}=\mathbf{g}_{i} \tag{8.35}
\end{equation*}
\]
where
\[
\mathbf{M}_{i}=\left[\begin{array}{cc}
\mathbf{N} & \mathbf{0}  \tag{8.36}\\
\mathbf{0} & \left.\mathbf{J}^{\prime}\right]_{i}
\end{array}\right.
\]
is the body mass matrix, where
\[
\mathbf{h}_{i}=\left[\begin{array}{c}
\dot{\mathbf{r}}  \tag{8.37}\\
\boldsymbol{\omega}^{\prime}
\end{array}\right]_{i}
\]
is the body velocity vector, where
\[
\mathbf{b}_{i}=\left[\begin{array}{c}
0  \tag{8.38}\\
\tilde{\boldsymbol{\omega}}^{\prime} \mathbf{J}^{\prime} \boldsymbol{\omega}^{\prime}
\end{array}\right]_{i}
\]
contains the quadratic velocity terms, and where
\[
\mathbf{g}_{i}=\left[\begin{array}{l}
\mathbf{f}  \tag{8.39}\\
\mathbf{n}^{\prime}
\end{array}\right]_{i}
\]
is the body force vector.

\subsection*{8.4 DYNAMICS OF A SYSTEM OF BODIES}

A system of bodies making up a mechanical system can be regarded as a collection of individual bodies interconnected by kinematic joints and/or force elements. If there are no kinematic joints in the system, it is called a system of unconstrained bodies. If there are one or more kinematic joints in the system, it is referred to as a system of constrained bodies.

\subsection*{8.4.1 A System of Unconstrained Bodies}

Consider the system of unconstrained bodies shown in Fig. 8.13. It is assumed that there are \(b\) bodies in this system connected to one another by various force elements. In addition, other forces, either constant or time-dependent, may act on the bodies.

The equations of motion for the \(i\) th body were given by Eq. 8.35:
\[
\begin{equation*}
\mathbf{M}_{i} \dot{\mathbf{h}}_{i}+\mathbf{b}_{i}=\mathbf{g}_{i} \tag{a}
\end{equation*}
\]

Equation \(a\) can be repeated for \(i=1, \ldots, b\) to obtain
\[
\left[\begin{array}{cccc}
\mathbf{M}_{1} & & &  \tag{8.40}\\
& \mathbf{M}_{2} & & \\
& & \ddots & \\
& & & \mathbf{M}_{b}
\end{array}\right]\left[\begin{array}{c}
\dot{\mathbf{h}}_{1} \\
\dot{\mathbf{h}}_{2} \\
\vdots \\
\vdots \\
\mathbf{h}_{b}
\end{array}\right]+\left[\begin{array}{c}
\mathbf{b}_{1} \\
\mathbf{b}_{2} \\
\vdots \\
\vdots \\
\mathbf{b}_{b}
\end{array}\right]=\left[\begin{array}{c}
\mathbf{g}_{1} \\
\mathbf{g}_{2} \\
\vdots \\
\mathbf{g}_{b}
\end{array}\right]
\]
or
\[
\begin{equation*}
\mathbf{M} \dot{\mathbf{h}}+\mathbf{b}=\mathbf{g} \tag{8.41}
\end{equation*}
\]
where
\[
\begin{equation*}
\mathbf{M}=\operatorname{diag}\left[\mathbf{M}_{1}, \mathbf{M}_{2}, \ldots, \mathbf{M}_{b}\right] \tag{8.42}
\end{equation*}
\]
is the system mass matrix, where
\[
\begin{equation*}
\mathbf{h}=\left[\mathbf{h}_{1}^{T}, \mathbf{h}_{2}^{T}, \ldots, \mathbf{h}_{b}^{T}\right] \tag{8.43}
\end{equation*}
\]


Figure 8.13 A system of unconstrained bodies.
is the system velocity vector, where
\[
\begin{equation*}
\mathbf{b}=\left[\mathbf{b}_{1}^{T}, \mathbf{b}_{2}^{T}, \ldots, \mathbf{b}_{b}^{T}\right]^{T} \tag{8.44}
\end{equation*}
\]
contains the quadratic velocity terms, and where
\[
\begin{equation*}
\mathbf{g}=\left[\mathbf{g}_{1}^{T}, \mathbf{g}_{2}^{T}, \ldots, \mathbf{g}_{b}^{T}\right]^{T} \tag{8.45}
\end{equation*}
\]
is the system force vector. Equation 8.41 represents the equations of motion for a system of unconstrained bodies. Vector \(\mathbf{g}\) contains all of the external and internal forces and moments. Gravitational force is considered an external force, whereas the force elements within the system, such as springs, are considered internal forces. \({ }^{\dagger}\)

\subsection*{8.4.2 A System of Constrained Bodies}

In a system of constrained bodies, two or more of the bodies are interconnected by kinematic joints. In addition to the kinematic joints, force elements are usually present, as shown in Fig. 8.14.


Figure 8.14 A system of constrained bodies.

If the system vector of coordinates for \(b\) bodies is denoted by \(\mathbf{q}\), then the kinematic joints in the system can be represented as \(m\) independent constraints, normally nonlinear equations in terms of \(\mathbf{q}\), as:
\[
\begin{equation*}
\Phi \equiv \Phi(\mathbf{q})=0 \tag{8.46}
\end{equation*}
\]

Each kinematic joint introduces reaction forces between connecting bodies. These reaction forces, which are also referred to as constraint forces, are denoted by vector \(\mathbf{g}^{(c)}\) :
\[
\begin{equation*}
\mathbf{g}^{(c)}=\left[\mathbf{g}_{1}^{(c)^{T}}, \mathbf{g}_{2}^{(c)^{T}}, \ldots, \mathbf{g}_{b}^{(c) T}\right]^{T} \tag{8.47}
\end{equation*}
\]
where \(\mathbf{g}_{i}^{(c)}, i=1, \ldots, b\), is the vector of joint reaction forces acting on body \(i\). The sum of the constraint forces \(\mathbf{g}^{(c)}\) and external forces \(\mathbf{g}\) provides the total of forces acting on the system. Hence, Eq. 8.41 can be modified to read
\[
\begin{equation*}
\mathbf{M} \dot{\mathbf{h}}+\mathbf{b}=\mathbf{g}+\mathbf{g}^{(c)} \tag{8.48}
\end{equation*}
\]

Equations 8.46 and 8.48 together present the equations of motion for a system of constrained bodies.

\footnotetext{
\({ }^{\dagger}\) In the formulation of the equations of motion, springs, dampers, and actuators are not treated as bodies but rather as abstract force elements. This is not an unrealistic assumption, since the mass of these elements is usually negligible as compared to the mass of the connecting bodies.
}

The constraint force vector \(\mathbf{g}^{(c)}\) can be expressed in terms of the constraint equations. However, this task cannot be accomplished at this point, since no rotational coordinates have been defined for the bodies. Equation 8.46 is expressed in terms of the vector of coordinates \(\mathbf{q}\), whereas Eq. 8.48 is expressed in terms of the vector of velocities \(\mathbf{h}\). For body \(i\) the relationship between \(\dot{\mathbf{q}}_{i}\) and \(\mathbf{h}_{i}\) has not yet been defined. The ambiguity lies with the rotational coordinates, not the translational coordinates. Vector \(\mathbf{h}_{i}\) is defined as \(\mathbf{h}_{i}^{T}=\left[\dot{\mathbf{r}}^{T}, \boldsymbol{\omega}^{\prime T}\right]_{i}^{T}\), while vector \(\mathbf{q}_{i}\) is \(\mathbf{q}^{T}=\left[\mathbf{r}^{T}, ?\right]_{i}^{T}\). This ambiguity will be clarified when proper sets of rotational coordinates for planar and spatial motions are defined in Chaps. 9 and 11.

\subsection*{8.4.3 Constraint Reaction Forces}

It is possible to obtain a relationship between the constraint reaction forces and the constraint equations if (1) a proper vector of coordinates is defined and (2) the constraint forces are expressed with respect to the same coordinate system as the vector of coordinates. For example, assume that Euler parameters are used as rotational coordinates; in that case, vector \(\mathbf{q}_{i}=\left[\mathbf{r}^{T}, \mathbf{p}^{T}\right]_{i}^{T}\) is the vector of coordinates for body \(i\). Vector \(\dot{\mathbf{q}}_{i}=\left[\dot{\mathbf{r}}^{T}, \dot{\mathbf{p}}^{T}\right]_{i}^{T}\) is different from vector \(\mathbf{h}_{i}=\left[\dot{\mathbf{r}}^{T}, \boldsymbol{\omega}^{\prime}\right]_{i}^{T}\); however, the identity \(\boldsymbol{\omega}_{i}^{\prime}=2 \mathbf{L}_{i} \dot{\mathbf{p}}_{i}\) can transform \(\mathbf{h}_{i}\) to \(\dot{\mathbf{q}}_{i}\) or vice versa. Similarly, for the entire system, \(\mathbf{h}\) can be transformed to \(\dot{\mathbf{q}}\). The constraint reaction force vector \(\mathbf{g}^{(c)}\) and the velocity vector \(\mathbf{h}\) are expressed in the same coordinate system. It will be seen in Chap. 11 that vector \(\mathbf{g}^{(c)}\) can be transformed to another coordinate system consistent with \(\mathbf{q}\).

At this point it will be assumed that \(\mathbf{g}^{(c)}\) can be transformed to a coordinate system consistent with \(\mathbf{q}\) and denoted as \(\mathbf{g}^{(*)}\). It will further be assumed that there are \(m\) independent constraint equations
\[
\begin{equation*}
\Phi \equiv \Phi(\mathbf{q})=0 \tag{a}
\end{equation*}
\]

If the joints are assumed to be frictionless, the work \({ }^{\dagger}\) done by the constraint forces in a virtual (infinitesimal) displacement \({ }^{\ddagger} \delta \mathbf{q}\) is zero; i.e.,
\[
\begin{equation*}
\mathbf{g}^{(*)^{T}} \delta \mathbf{q}=0 \tag{b}
\end{equation*}
\]

Since the virtual displacement \(\delta \mathbf{q}\) must be consistent with the constraints, Eq. \(a^{8}\) yields
\[
\begin{equation*}
\Phi_{q} \delta q=0 \tag{c}
\end{equation*}
\]

The vector of \(n\) coordinates \(\mathbf{q}\) may be partitioned into a set of \(m\) dependent coordinates \(\mathbf{u}\), and a set of \(n-m\) independent coordinates \(\mathbf{v}\), as \(\mathbf{q} \equiv\left[\mathbf{u}^{T}, \mathbf{v}^{T}\right]^{T}\). This yields a partitioned vector of virtual displacements \(\delta \mathbf{q} \equiv\left[\delta \mathbf{u}^{T}, \delta \mathbf{v}^{T}\right]^{T}\) and a partitioned Jacobian
\({ }^{\dagger}\) The work done by a force \(\mathbf{f}\) acting on a system and causing a displacement \(\mathbf{q}\) is defined as \(w=\mathbf{f}^{T} \mathbf{q}\).
\({ }^{\ddagger}\) A virtual displacement of a system is defined as an infinitesimal change in the coordinates of the system consistent with the constraints and forces imposed on the system at time \(t\). The displacement is called a virtual one to distinguish it from an actual displacement of the system occurring in a time interval \(d t\), during which the constraints and forces may change.
\({ }^{8}\) The Taylor series expansion of Eq. \(a\) about \(\mathbf{q}\) is
\[
\boldsymbol{\Phi}(\mathbf{q}+\delta \mathbf{q})=\boldsymbol{\Phi}(\mathbf{q})+\boldsymbol{\Phi}_{q} \delta \mathbf{q}+\text { higher-order terms }
\]

A displacement \(\delta \mathbf{q}\) consistent with the constraints yields \(\boldsymbol{\Phi}(\mathbf{q}+\delta \mathbf{q})=\boldsymbol{0}\). Using \(\boldsymbol{\Phi}(\mathbf{q})=\boldsymbol{0}\) and eliminating the higher-order terms for infinitesimal \(\delta \mathbf{q}\), we find that \(\boldsymbol{\Phi}_{\mathrm{q}} \delta \mathbf{q}=\boldsymbol{0}\).
matrix \(\boldsymbol{\Phi}_{\mathbf{q}}=\left[\boldsymbol{\Phi}_{\mathbf{u}}, \boldsymbol{\Phi}_{\mathbf{v}}\right]\). The matrix \(\boldsymbol{\Phi}_{\mathbf{u}}\) is \(m \times m\) and nonsingular, since the constraint equations are assumed to be independent. If vector \(\mathbf{g}^{(*)}\) is also partitioned as \(\mathbf{g}^{(*)} \equiv\left[\mathbf{g}_{(\mu)}^{(*)^{T}}, \mathbf{g}_{(v)^{(*)}}{ }^{(*)}\right]^{T}\), then Eq. \(b\) can be written as
\[
\mathbf{g}_{(*)}^{(*)^{\top}} \delta \mathbf{u}+\mathbf{g}_{(v)}^{(*)} \delta \mathbf{v}=0
\]
or
\[
\begin{equation*}
\mathbf{g}_{(w)^{*}}^{(* T} \delta \mathbf{u}=-\mathbf{g}_{(v)}^{(*)} \delta \mathbf{v} \tag{d}
\end{equation*}
\]

Similarly, Eq. \(c\) yields
\[
\begin{equation*}
\Phi_{\mathbf{u}} \delta \mathbf{u}=-\boldsymbol{\Phi}_{\mathbf{v}} \delta \mathbf{v} \tag{8.49}
\end{equation*}
\]

If Eq. \(d\) is appended to the system of equations represented by Eq. 8.49, the result can be written as
\[
\left[\begin{array}{l}
\mathbf{g}_{(u)^{\prime}}^{(*)}  \tag{e}\\
\boldsymbol{\Phi}_{\mathbf{u}}
\end{array}\right] \delta \mathbf{u}=-\left[\begin{array}{l}
\mathbf{g}_{\left.()^{(*)}\right)^{r}}^{\boldsymbol{\Phi}_{\mathbf{v}}}
\end{array}\right] \delta \mathbf{v}
\]

The matrix to the left in Eq. \(e\) is an \((m+1) \times m\) matrix. Since \(\boldsymbol{\Phi}_{\mathbf{u}}\) is an \(m \times m\) nonsingular matrix, the first row of the \((m+1) \times m\) matrix, i.e., \(\mathbf{g}_{(a)^{\left(*^{4} T^{T}\right.}}\), can be expressed as a linear combination of the other rows of the matrix:
\[
\begin{equation*}
\mathbf{g}_{(u)}^{(*)}=\boldsymbol{\Phi}_{\mathrm{u}}^{T} \boldsymbol{\lambda} \tag{f}
\end{equation*}
\]
where \(\boldsymbol{\lambda}\) is an \(m\)-vector of multipliers known as Lagrange multipliers. Substitution of Eq. \(f\) in Eq. \(d\) yields
\[
\begin{equation*}
\boldsymbol{\lambda}^{T} \boldsymbol{\Phi}_{\mathbf{u}} \delta \mathbf{u}=-\mathbf{g}_{(v)^{*}}^{(*)^{T}} \delta \mathbf{v} \tag{g}
\end{equation*}
\]
or
where Eq. 8.49 has been employed. Vector \(\delta \mathbf{v}\) is an arbitrary (independent) vector. The consistency of the constraints for virtual displacements \(\delta \boldsymbol{q}\) is guaranteed by solving Eq. 8.49 for \(\delta \mathbf{u}\). Since Eq. \(g\) must hold for any arbitrary \(\delta \mathbf{v}\), then
\[
\boldsymbol{\lambda}^{T} \boldsymbol{\Phi}_{\mathbf{v}}=\mathbf{g}_{(v)^{(*)}}{ }^{T}
\]
or
\[
\begin{equation*}
\mathbf{g}_{(v)}^{(*)}=\boldsymbol{\Phi}_{\mathbf{v}}^{T} \boldsymbol{\lambda} \tag{h}
\end{equation*}
\]

Appending Eq. \(f\) to Eq. \(h\) yields
\[
\begin{equation*}
\mathbf{g}^{(*)}=\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda} \tag{8.50}
\end{equation*}
\]

Equation 8.50 expresses the constraint reaction forces in terms of the constraint equations and a vector of multipliers.

\subsection*{8.5 CONDITIONS FOR PLANAR MOTION}

The equations of motion for an unconstrained body can be simplified if the motion is planar. Assume, without any loss of generality, that the centroidal body-fixed \(\xi_{i} \eta_{i} \zeta_{i}\) coordinates are attached to body \(i\) in such a way that the \(\zeta\) axis is parallel to the \(z\) axis. Furthermore, assume that the center of mass of the body is located in the \(x y\) plane. The vector of translational coordinates for the body is then \(\mathbf{r}_{i}=[x, y, 0]_{i}^{T}\). It can be assumed that the \(\xi \eta\) and \(x y\) planes remain coincident, and therefore that \(\ddot{z}_{i}=0\) at all times. If the angle \(\phi_{i}\) is taken as the rotational coordinate, then vectors of angular velocity and angu-
lar acceleration for the body can be written, respectively, as \(\boldsymbol{\omega}_{i}=\boldsymbol{\omega}_{i}^{\prime}=[0,0, \dot{\phi}]_{i}^{T}\) and \(\dot{\boldsymbol{\omega}}_{i}=\dot{\boldsymbol{\omega}}_{i}^{\prime}=[0,0, \ddot{\phi}]_{i}^{T}\).

Assume that a force \(\vec{f}_{i}\) is acting at point \(P_{i}\), as shown in Fig. 8.15. The translational equation of motion given in Eq. 8.31 can be written, where \(\ddot{z}_{i}=0\), as
\[
\left[\begin{array}{lll}
m & &  \tag{8.51}\\
& m & \\
& & m
\end{array}\right]_{i}\left[\begin{array}{l}
\ddot{x} \\
\ddot{y} \\
0
\end{array}\right]_{i}=\left[\begin{array}{l}
f_{(x)} \\
f_{(y)} \\
f_{(z)}
\end{array}\right]_{i}
\]
which yields \(f_{\left.()_{i}\right)}=0\).
Condition 1. In planar motion, the forces acting on a body must remain parallel to the plane of the body.

If it is assumed that \(P_{i}\) is in the plane, then the moment \(\mathbf{n}_{i}=\tilde{\mathbf{s}}_{i}^{P} \mathbf{f}_{i}\) can have a nonzero component only in the \(z\) (or \(\zeta\) ) direction: \(\mathbf{n}_{i}=\mathbf{n}_{i}^{\prime}=[0,0, n]_{i}^{T}\), where \(n_{i}\) is the magnitude of the moment. Hence, the system of rotational equations of motion, given in Eq. 8.32, is written as
\[
\mathbf{J}_{i}^{\prime}\left[\begin{array}{c}
0 \\
0 \\
\ddot{\phi}
\end{array}\right]_{i}+\left[\begin{array}{rrr}
0 & -\dot{\phi} & 0 \\
\dot{\phi} & 0 & 0 \\
0 & 0 & 0
\end{array}\right]_{i} \mathbf{J}_{i}^{\prime}\left[\begin{array}{l}
0 \\
0 \\
\dot{\phi}
\end{array}\right]_{i}=\left[\begin{array}{l}
0 \\
0 \\
n
\end{array}\right]_{i}
\]
or, by using the elements of \(\mathbf{J}^{\prime}\) from Eq. 8.25, it is found that
\[
\begin{align*}
j_{\xi \zeta_{i}} \ddot{\phi}_{i}-j_{\eta \zeta_{i}} \dot{\phi}_{i}^{2} & =0  \tag{a}\\
j_{\eta \zeta_{i}} \ddot{\phi}_{i}+\dot{j}_{\xi \zeta_{i}} \dot{\phi}_{i}^{2} & =0  \tag{b}\\
j_{\zeta \zeta_{i}} \ddot{\phi}_{i} & =n_{i} \tag{8.52}
\end{align*}
\]

Equations \(a\) and \(b\) can be written as
\[
\left[\begin{array}{rr}
j_{\xi \zeta} & -j_{\eta \zeta}  \tag{c}\\
j_{\eta \zeta} & j_{\xi \zeta}
\end{array}\right]_{i}\left[\begin{array}{c}
\ddot{\phi} \\
\dot{\phi}^{2}
\end{array}\right]_{i}=\left[\begin{array}{l}
0 \\
0
\end{array}\right]
\]

In order to have a nontrivial solution for Eq. \(c\), i.e., nonzero \(\ddot{\phi}_{i}\) and \(\dot{\phi}_{i}\), the matrix at the left in Eq. \(c\) must be singular. Therefore, \(j_{\xi \zeta_{i}}^{2}+j_{\eta 5_{i}}^{2}=0\), which yields
\[
\begin{equation*}
j_{\xi 5_{i}}=j_{\eta \zeta_{i}}=0 \tag{8.53}
\end{equation*}
\]

Condition 2. A moment about the \(\zeta\) axis causes the body to rotate only about that axis if the products of inertia \(j_{\xi \xi}\) and \(j_{\eta \xi}\) are zero.


Figure 8.15 A body with centroidal coordinate system in planar motion.

The condition shown by Eq. 8.53 applies to bodies in which the plane of motion is geometrically the plane of symmetry, where uniform distribution of mass is assumed. However, if either of the products of inertia \(j_{\xi \zeta}\) and \(j_{\eta \zeta}\) is nonzero, a moment about the \(\zeta\) axis causes the body to rotate about an axis nonparallel to \(\zeta\).

An unconstrained body can experience a nonplanar motion if conditions 1 and 2 are not met. However, for a constrained body, kinematic joints may be aligned in such a way that the body would move only in a plane, without conditions 1 and 2 being satisfied. In such cases, if condition 1 is not met, joint reaction forces are developed that are not in the plane of motion, and if condition 2 is not met, joint reaction moments are developed that are not about the axis perpendicular to the plane of motion.

\section*{9}

\section*{Planar Dynamics}

In this chapter, the equations of motion for both unconstrained and constrained mechanical systems undergoing planar motion are developed in a form adequate for computer programming. Suitable equations are formulated for a variety of forces commonly encountered in mechanical systems, such as gravity and the forces of springs and dampers. The kinematic constraint equations of Chap. 4 are applied to complete the equations of motion.

\subsection*{9.1 EQUATIONS OF MOTION}

Translational and rotational equations of motion for an unconstrained body are written from Eqs. 8.51 and 8.52, as follows:
\[
\begin{aligned}
m_{i} \ddot{x}_{i} & =f_{(x)_{i}} \\
m_{i} \ddot{y}_{i} & =f_{(y)_{i}} \\
j_{\zeta \zeta_{i}} \ddot{\phi}_{i} & =n_{i}
\end{aligned}
\]
or
\[
\left[\begin{array}{lll}
m & &  \tag{9.1}\\
& m & \\
& & \mu
\end{array}\right]_{i}\left[\begin{array}{l}
\ddot{x} \\
\ddot{y} \\
\ddot{\phi}
\end{array}\right]_{i}=\left[\begin{array}{l}
f_{(x)} \\
f_{(y)} \\
n
\end{array}\right]_{i}
\]
where for notational simplicity the polar moment of inertia \(j_{\zeta \zeta}\) of a body is denoted by \(\mu\). Equation 9.1 may also be expressed as
\[
\begin{equation*}
\mathbf{M}_{i} \ddot{\mathbf{\varphi}}_{i}=\mathbf{g}_{i} \tag{9.2}
\end{equation*}
\]
where
\[
\begin{aligned}
\mathbf{M}_{i} & =\operatorname{diag}[m, m, \mu]_{i} \\
\mathbf{q}_{i} & =[x, y, \phi]_{i}^{T} \\
\mathbf{g}_{i} & =\left[f_{(x)}, f_{(y)}, n\right]_{i}^{T}
\end{aligned}
\]

A comparison of Eqs. 9.2 and 8.35 reveals that in planar motion \(\mathbf{b}_{i}=0\) and \(\dot{\mathbf{h}}_{i}=\ddot{\mathbf{q}}_{i}\) or \(\mathbf{h}_{i}=\dot{\mathbf{q}}_{i}\). The ambiguity that was mentioned in Sec. 8.4 .2 between \(\mathbf{h}_{i}\) and \(\dot{\mathbf{q}}_{i}\), for general motion of a body, does not exist when planar motion is considered. In planar motion, the rotational velocity vector \(\omega_{i}^{\prime} \equiv[0,0, \dot{\phi}]_{i}^{T}\) is the time derivative of a rotational coordinate vector \([0,0, \phi]_{i}^{T}\).

For a system of \(b\) unconstrained bodies, Eq. 9.2 is repeated \(b\) times as
\[
\begin{equation*}
\mathbf{M} \ddot{\mathbf{q}}=\mathbf{g} \tag{9.3}
\end{equation*}
\]
where
\[
\begin{aligned}
\mathbf{M} & =\operatorname{diag}\left[\mathbf{M}_{1}, \mathbf{M}_{2}, \ldots, \mathbf{M}_{b}\right] \\
\mathbf{q} & =\left[\mathbf{q}_{1}^{T}, \mathbf{q}_{2}^{T}, \ldots, \mathbf{q}_{b}^{T}\right]^{T} \\
\mathbf{g} & =\left[\mathbf{g}_{1}^{T}, \mathbf{g}_{2}^{T}, \ldots, \mathbf{g}_{b}^{T}\right]^{T}
\end{aligned}
\]

The system mass matrix \(\mathbf{M}\) is a \(3 b \times 3 b\) constant diagonal matrix, and vectors \(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}\), and g are \(3 b\)-vectors.

For a system of \(b\) constrained bodies, the equations of motion can be written as
\[
\begin{equation*}
\mathbf{M} \ddot{\mathbf{q}}=\mathbf{g}+\mathbf{g}^{(c)} \tag{9.4}
\end{equation*}
\]
where \(\mathbf{g}^{(c)}\) is the vector of constraint reaction forces. Since Eq. 9.4, and hence \(\mathbf{g}^{(c)}\), is described in the same coordinate system as \(\mathbf{q}\), then from Eq. 8.50 it is found that
\[
\begin{equation*}
\mathbf{g}^{(c)}=\boldsymbol{\Phi}_{\mathrm{q}}^{T} \boldsymbol{\lambda} \tag{9.5}
\end{equation*}
\]
where \(\boldsymbol{\Phi}=0\) represents \(m\) independent constraint equations. Substitution of Eq. 9.5 in Eq. 9.4 yields
\[
\begin{equation*}
\mathbf{M} \ddot{\mathbf{q}}-\boldsymbol{\Phi}_{\mathrm{q}}^{T} \boldsymbol{\lambda}=\mathbf{g} \tag{9.6}
\end{equation*}
\]

Equation 9.6 and the constraint equations
\[
\begin{equation*}
\Phi=0 \tag{9.7}
\end{equation*}
\]
together represent the equations of motion for a system of constrained bodies.
In kinematic analysis, the number of degrees of freedom of a system must be equal to the number of driver constraint equations. This means that \(m\) kinematic constraint equations and \(k\) driver equations provide \(n\) equations in \(n\) unknowns and so will yield a unique solution. However, in dynamic analysis, in general, there are no driver equations to be specified. Since \(n>m\), there are more unknowns in the constraint equations of Eq. 9.7 than there are equations, and so there is no unique solution to these equations. In dynamic analysis, a unique solution is obtained when the constraint equations are considered simultaneously with the differential equations of motion, and a proper set of initial conditions is specified. These algebraic-differential equations are solved by numerical methods and will be discussed in Chap. 13.

\subsection*{9.2 VECTOR OF FORCES}

Vector \(\mathbf{g}\) in Eq. 9.6 contains the vectors of forces acting on all the bodies in the system; i.e.,
\[
\begin{equation*}
\mathbf{g}=\left[\mathbf{g}_{1}^{T}, \mathbf{g}_{2}^{T}, \ldots, \mathbf{g}_{b}^{T}\right]^{T} \tag{9.8}
\end{equation*}
\]

To construct vector \(\mathbf{g}\), the vector of force for each body must be determined. For a typical body \(i\), the vector of force \(\mathbf{g}_{i}\) contains all forces and moments acting on that body:
\[
\mathbf{g}_{i}=\left[f_{(x)}, f_{(y)}, n\right]_{i}^{T}
\]
where \(f_{\left(x_{i}\right)}, f_{\left(y_{i}\right)}\), and \(n_{i}\) are the sums of all forces in the x and y directions and the sum of all moments, respectively. In Secs. 9.2.1 to 9.2.7, a variety of external and internal forces that commonly appear in mechanical systems are discussed, and their contributions to the elements of \(\mathbf{g}_{i}\), and hence \(\mathbf{g}\), are shown.

\subsection*{9.2.1 Gravitational Force}

Figure 9.1 shows a body acted upon by a gravitational field in the negative \(y\) direction. The choice of the negative \(y\) direction as the direction of gravity is totally arbitrary. However, in this text the gravitational field will be considered to be acting in this direction in planar motion unless indicated otherwise.

If \(w_{i}\) is the weight of body \(i\) (mass of body \(i\) times the gravitational constant), then the contribution of this force to the vector of force of body \(i\) is
\[
\begin{equation*}
\mathbf{g}_{i}^{(\text {gravily })}=[0,-w, 0]_{i}^{T} \tag{9.9a}
\end{equation*}
\]

\subsection*{9.2.2 Single Force or Moment}

Consider a single force \(\vec{f}_{i}\) acting with known direction at point \(P_{i}\) on body \(i\) as shown in Fig. 9.2(a). This force has components \(f_{\left(x_{i}\right)}\) and \(f_{\left(y_{i}\right)}\). If the local coordinates of \(P_{i}\) are known as \(\mathbf{s}_{i}^{\prime P}=\left[\xi^{P}, \eta^{P}\right]_{i}^{T}\), then \(\mathbf{s}_{i}^{P}=\mathbf{A}_{i} \mathbf{s}_{i}^{P P}\). The moment of \(\vec{f}\) about the origin of the body is
\[
\begin{align*}
n_{i} & =\left(\tilde{\mathbf{s}}^{P} \mathbf{f}_{\mathbf{f}_{(z)}}\right. \\
& =s_{(y)}^{P} f_{\left.()_{i}\right)}+s_{(x)_{i}}^{P} f_{\left.()_{i}\right)} \\
& =-\left(\xi_{i}^{P} \sin \phi_{i}+\eta_{i}^{P} \cos \phi_{i}\right) f_{\left(x_{i}\right)}+\left(\xi_{i}^{P} \cos \phi_{i}-\eta_{i}^{p} \sin \phi_{i}\right) f_{()_{i}} \tag{9.10}
\end{align*}
\]


Figure 9.1 Gravitational field acting on a body.


Figure 9.2 A body acted upon by a constant (a) force, and (b) moment.
The contribution of this force to the vector of forces of body \(i\) is
\[
\begin{equation*}
\mathbf{g}_{i}^{\text {(single } \cap}=\left[f_{(x)}, f_{(y)}, n\right]_{i}^{T} \tag{9.11}
\end{equation*}
\]

When a pure moment with magnitude \(n_{i}\) acts on body \(i\) as shown in Fig. 9.2(b), its contribution to the vector of forces of body \(i\) is
\[
\begin{equation*}
\mathbf{g}_{i}^{(\text {single } n)}=[0,0, n]_{i}^{T} \tag{9.12}
\end{equation*}
\]

Equations 9.11 and 9.12 are valid for either constant or time-dependent forces or moments.

\section*{Example 9.1}

Body \(i\), with a mass of 2 , is acted upon by gravity, a constant force, and a pure moment, as shown in the illustration. The constant force has the components \(\mathbf{f}=[1.2,0.5]^{T}\), and the magnitude of the pure moment is 0.6 . Determine the vector of force for this body if \(\mathbf{s}_{i}^{\prime P}=[-0.2,0.3]^{T}, \phi_{i}=30^{\circ}\), and \(\mathbf{r}_{i}=[2.1,1.6]^{T}\).


Solution The weight of the body is \(w_{i}=2 \times 9.81=19.62\). The moment of the force is found from Eq. 9.10 to be \(n_{i}=-0.35\). Therefore, the vector of forces for
this body is
\[
\mathbf{g}_{i}=\left[\begin{array}{c}
1.2 \\
0.5-19.62 \\
-0.35-0.6
\end{array}\right]=\left[\begin{array}{c}
1.2 \\
-19.12 \\
-0.95
\end{array}\right]
\]

\subsection*{9.2.3 Translational Actuators}

Actuators provide a constant or a time-dependent pair of forces acting on two bodies without imposing any kinematic constraints. The forces making up the pair have a common line of action but are in opposite directions. As shown in Fig. 9.3(a) an actuator acts between bodies \(i\) and \(j\) at the attachment points \(P_{i}\) and \(P_{j}\). The equivalent representation for this system is shown in Fig. 9.3 (b) or (c), depending on the direction of the forces.

The sign convention for the pair of forces can be defined as positive when the forces pull on the bodies and negative when the forces push on the bodies. If the actuator force is denoted by \(f^{(a)}, f^{(a)}>0\) constitutes a pull and \(f^{(a)}<0\) constitutes a push. In order to find the forces being applied to bodies \(i\) and \(j\), i.e., \(\vec{f}_{i}^{(a)}\) and \(\vec{f}_{j}^{(a)}\), a unit vector on the line of action of the actuator must be defined.

A vector \(\vec{l}\) connecting points \(P_{i}\) and \(P_{j}\), as shown in Fig. 9.4, is defined as
\[
\begin{equation*}
\boldsymbol{l}=\mathbf{r}_{j}+\mathbf{A}_{j} \mathbf{s}_{j}^{\prime P}-\mathbf{r}_{j}-\mathbf{A}_{i} \mathbf{s}_{i}^{\prime p} \tag{9.13}
\end{equation*}
\]

The magnitude of this vector is
\[
\begin{equation*}
l=\left(l^{T} l\right)^{1 / 2} \tag{9.14}
\end{equation*}
\]

A unit vector \(\vec{u}\) is defined as
\[
\begin{equation*}
\mathbf{u}=\frac{l}{l} \tag{9.15}
\end{equation*}
\]

The unit vector \(\vec{u}\) has the same direction as \(\vec{f}_{i}^{(a)}\) in the case of a pull and \(\vec{f}_{i}^{(a)}\) in the case of a push. Therefore,
\[
\begin{equation*}
\mathbf{f}_{i}^{(a)}=f^{(a)} \mathbf{u} \tag{9.16}
\end{equation*}
\]


Figure 9.3 (a) An actuator acting between two bodies, and the equivalent representation; (b) pull; (c) push.


Figure 9.4 Defining a unit vector along the axis of the actuator forces.
and
\[
\begin{equation*}
\mathbf{f}_{j}^{(a)}=-f^{(a)} \mathbf{u} \tag{9.17}
\end{equation*}
\]

It is clear that since \(f^{(a)}\) can be either positive or negative, the sign convention in Eqs. 9.16 and 9.17 is automatically satisfied. The contribution of \(\mathbf{f}_{i}^{(a)}\) (or \(\mathbf{f}_{j}^{(\alpha)}\) ) to the vector of forces \(\mathbf{g}_{i}\left(\right.\) or \(\left.\mathbf{g}_{j}\right)\) can be found from Eqs. 9.10 and 9.11.

\subsection*{9.2.4 Translational Springs}

Translational (point-to-point) springs are the most commonly used force elements in mechanical systems. Figure 9.5(a) shows a spring attached between points \(P_{i}\) and \(P_{j}\) on bodies \(i\) and \(j\). The force of this spring can be found as
\[
\begin{equation*}
f^{(s)}=k\left(l-l^{0}\right) \tag{9.18}
\end{equation*}
\]


Figure 9.5 (a) A translational spring between two bodies with (b) linear characteristics or (c) nonlinear characteristics.
where \(k\) is the spring stiffness, \(l\) is the deformed length, and \(l^{0}\) is the undeformed length of the spring. The deformed length of the spring is found from Eq. 9.14.

The sign convention for the spring force is similar to that of the actuator force positive in tension (pull) and negative in compression (push). The forces of the spring acting on bodies \(i\) and \(j\) are
\[
\begin{equation*}
\mathbf{f}_{i}^{(s)}=f^{(s)} \mathbf{u} \tag{9.19}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{f}_{j}^{(s)}=-f^{(s)} \mathbf{u} \tag{9.20}
\end{equation*}
\]
where \(\mathbf{u}\) is a unit vector defined along \(l\) (Eq. 9.15). Equations 9.19 and 9.20 are valid in tension and compression-if \(l>l^{0}\) (for tension), \(f^{(s)}\) is positive; and if \(l<l^{0}\) (for compression), \(f^{(s)}\) is negative.

The contributions of \(\mathbf{f}_{i}^{(s)}\) (or \(\mathbf{f}_{j}^{(s)}\) ) to \(\mathbf{g}_{i}\) (or \(\mathbf{g}_{j}\) ) are found from Eqs. 9.10 and 9.11.
In Eq. 9.18 a linear characteristic is assumed for the spring (Fig. 9.5(b)). However, the spring may have nonlinear force-deformation characteristics, e.g., the curve shown in Fig. 9.5(c). In this case, the force-deformation curve can be used directly instead of Eq. 9.18. If the force-deformation data are available in discretized form, the linear or cubic spline function technique (Sec. 4.2.4) can be employed to compute \(f^{(s)}\) for a deformation \(l-l^{0}\).

\section*{Example 9.2}

Two bodies are connected by a translational spring, where \(\mathbf{s}_{1}^{\prime P}=[0.15,0]^{T}\) and \(\mathbf{s}_{2}^{\prime P}=[0,0.1]^{T}\) (see the illustration). Write the equations of motion when \(\mathbf{q}_{1}=\) \([-0.1,0.2,0.785]^{T}\) and \(\mathbf{q}_{2}=[0.1,0.1,0.262]^{T}\), and then calculate the accelerations.


Solution From Eq. 9.13, vector \(l\) is found to be equal to \([0.068,-0.109]^{T}\), and hence \(l=0.129\). The unit vector along \(l\) is \(\mathbf{u}=[0.528,-0.849]^{T}\). The spring force is \(f^{(s)}=50(0.129-0.2)=-3.558\). From Eqs. 9.19 and 9.20 it is found that \(\mathbf{f}_{1}^{(s)}=[-1.878,3.022]^{T}\) and \(\mathbf{f}_{2}^{(s)}=[1.878,-3.022]^{T}\). Equation 9.10 yields \(n_{1}^{(s)}=0.520\) and \(n_{2}^{(s)}=-0.103\). The vectors of forces for bodies 1 and 2 contain the contribution from the spring and from gravity and are found to be
\(\mathbf{g}_{1}=[-1.878,1.060,0.520]^{T}\) and \(\mathbf{g}_{2}=[1.878,-4.494,-0.103]^{T}\). The equations of motion are written from Eq. 9.3 as
\[
\left[\begin{array}{llllll}
0.2 & & & & & \\
& 0.2 & & & & \\
& & 0.03 & & & \\
& & & 0.15 & & \\
& & & & 0.15 & \\
& & & & & 0.02
\end{array}\right]\left[\begin{array}{l}
\ddot{x}_{1} \\
\ddot{y}_{1} \\
\ddot{\phi}_{1} \\
\ddot{x}_{2} \\
\ddot{y}_{2} \\
\ddot{\phi}_{2}
\end{array}\right]=\left[\begin{array}{r}
-1.878 \\
1.060 \\
0.520 \\
1.878 \\
-4.494 \\
-0.103
\end{array}\right]
\]

The accelerations are found easily to be
\[
\ddot{\mathbf{q}}=[-9.389,5.302,17.326,12.518,-29.959,-5.154]^{T}
\]

\subsection*{9.2.5 Translational Dampers}

A translational (point-to-point) damper between two bodies \(i\) and \(j\) is shown in Fig. 9.6. The damping force can be found to be
\[
\begin{equation*}
f^{(a)}=d \dot{l} \tag{9.21}
\end{equation*}
\]
where \(d\) is the damping coefficient and \(i\) is the time rate of change of the damper length. \(i\) is found by taking the time derivative of Eq. 9.14:
\[
\begin{equation*}
i=\frac{l^{T} \dot{l}}{l} \tag{9.22}
\end{equation*}
\]
where \(\boldsymbol{i}\), in turn, is found from Eq. 9.13:
\[
\begin{equation*}
\dot{i}=\dot{\mathbf{r}}_{j}+\dot{\phi}_{j} \mathbf{B}_{j} \mathbf{s}_{j}^{\prime P}-\dot{\mathbf{r}}_{i}-\dot{\phi}_{i} \mathbf{B}_{i} \mathbf{s}_{i}^{\prime P} \tag{9.23}
\end{equation*}
\]
and where
\[
\mathbf{B}_{k}=\left[\begin{array}{rr}
-\sin \phi & -\cos \phi \\
\cos \phi & -\sin \phi
\end{array}\right]_{k} \quad k=i, j
\]

The sign convention for the damping force is defined as positive for \(i>0\) and negative for \(i<0\). Since a damper opposes the relative motion of two bodies, when the two bodies move away from each other (when \(i>0\) ), the forces of the damper exhibit a pull, and when the bodies move toward each other (when \(i<0\) ), the forces of the damper exhibit a push.


Figure 9.6 A translational damper between two bodies.

By defining a unit vector \(\vec{u}\) along \(\vec{l}\), we express the forces \(\mathbf{f}_{i}^{(d)}\) and \(\mathbf{f}_{j}^{(d)}\) as
\[
\begin{equation*}
\mathbf{f}_{i}^{(d)}=f^{(d)} \mathbf{u} \tag{9.2.2}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{f}_{j}^{(d)}=-f^{(d)} \mathbf{u} \tag{9.25}
\end{equation*}
\]

Equations 9.24 and 9.25 are valid for both pull and push cases.
As was true of deformation in the case of springs, the relationship between force and deformation rate for a damper can be linear or nonlinear. Equation 9.21 assumes a linear characteristic for the damper. However, if the damper characteristic is nonlinear, a curve or a table of data describing the relationship between force and deformation rate can be used instead of Eq. 9.21.

\section*{Example 9.3}

A spring-damper element is connected between bodies 2 and 4 of a four-bar linkage, as shown in the illustration. The attachment points are \(\mathbf{s}_{2}^{\prime P}=[0.3,0.2]^{T}\) and \(\mathbf{s}_{4}^{\prime P}=[-0.1,0.1]^{T}\). If at a particular instant the vectors of coordinates and velocities are \(\mathbf{q}_{2}=[0.4,0.1,1.3]^{T}, \mathbf{q}_{4}=[-0.35,0.2,5.6]^{T}, \dot{\mathbf{q}}_{2}=[0.8,-0.6,-0.3]^{T}\), and \(\dot{\mathbf{q}}_{4}=[-0.5,0.45,-0.1]^{T}\), determine the vector of forces for the three moving bodies.


Solution From Eqs. 9.13 and 9.23 it can be found that \(l=[-0.652,-0.102,]^{T}\) and \(\dot{i}=[-1.389,1.018]^{T}\), which yield \(l=0.660\) and \(\dot{l}=1.215\). A unit vector along \(l\) is \(\mathbf{u}=[-0.988,-0.154]^{T}\). Equations 9.18 and 9.21 yield the spring and damper forces \(f^{(s)}=100(0.660-0.9)=-24.012\) and \(f^{(d)}=25 \times 1.215=\) 30.373. Since the spring and the damper have the same point of application on each body, their total force can be used as \(f^{(s+d)}=-24.012+30.373=6.361\). The components of this force acting on the two bodies are
\[
\mathbf{f}_{2}^{(s+d)}=\left[\begin{array}{l}
-6.285 \\
-0.982
\end{array}\right] \quad \mathbf{f}_{4}^{(s+d)}=\left[\begin{array}{l}
6.285 \\
0.982
\end{array}\right]
\]

Equation 9.10 can be used to determine that the moments of these forces are \(n_{2}^{(s+d)}=2.263\) and \(n_{4}^{(s+d)}=-0.898\).

The weights of the bodies are \(w_{2}=29.430, w_{3}=6.867\), and \(w_{4}=23.544\). The vectors of forces for the bodies are
\[
\mathbf{g}_{2}=\left[\begin{array}{r}
-6.285 \\
-30.412 \\
2.263
\end{array}\right] \quad \mathbf{g}_{3}=\left[\begin{array}{c}
0.0 \\
-6.867 \\
0.0
\end{array}\right] \quad \mathbf{g}_{4}=\left[\begin{array}{r}
6.285 \\
-22.562 \\
-0.898
\end{array}\right]
\]

\subsection*{9.2.6 Rotational Springs}

A rotational (torsional) spring acting between two bodies \(i\) and \(j\) is shown in Fig. 9.7(a). The two bodies are also assumed to be connected by a revolute joint whose axis is the same as the spring axis. A rotational spring applies pure moments on the bodies, equal in magnitude and opposite in direction.

The moment is found as
\[
\begin{equation*}
n^{(r-s)}=k\left(\theta-\theta^{0}\right) \tag{9.26}
\end{equation*}
\]
where \(k\) is the spring stiffness, \(\theta\) is the deformed angle of the spring, and \(\theta^{0}\) is the undeformed angle, as shown in Fig. 9.7(b). Vectors \(\vec{s}_{i}\) and \(\vec{s}_{j}\) are assumed to be attached to the spring in order to define the spring angle.


Figure 9.7 (a) A rotational spring acting between two bodies. (b) Free (undeformed) state of the spring.

When \(\theta>\theta^{0}\), the moment of the spring acts on body \(i\) in the positive rotational direction and on body \(j\) in the negative rotational direction, as shown in Fig. 9.8(a). When \(\theta<\theta^{0}\) the situation is reversed, as shown in Fig. 9.8(b). Therefore,
\[
\begin{equation*}
n_{i}^{(r-s)}=n^{(r-s)} \tag{9.27}
\end{equation*}
\]
and
\[
\begin{equation*}
n_{j}^{(r-s)}=-n^{(r-s)} \tag{9.28}
\end{equation*}
\]

(a)

(b)

Figure 9.8 The moments of a rotational spring acting on the bodies, (a) for \(\theta>\theta^{0}\) and (b) for \(\theta<\theta^{0}\).

\subsection*{9.2.7 Rotational Dampers}

The rotational element shown in Fig. 9.7 may also contain a damper, in addition to the spring. For a rotational damper the moment is found as
\[
\begin{equation*}
n^{(r-d)}=d \dot{\theta} \tag{9.29}
\end{equation*}
\]
where \(d\) is the damping coefficient and
\[
\begin{equation*}
\dot{\theta}=\dot{\phi}_{j}-\dot{\phi}_{i} \tag{9.30}
\end{equation*}
\]
is the time rate of change of the element angle.
When \(\dot{\theta}>0\), the moment of the damper acts on body \(i\) in the positive rotational direction and on body \(j\) in the negative rotational direction. When \(\dot{\theta}<0\), the situation is reversed. Therefore,
\[
\begin{equation*}
n_{i}^{(r-d)}=n^{(r-d)} \tag{9.31}
\end{equation*}
\]
and
\[
\begin{equation*}
n_{j}^{(r-d)}=-n^{(r-d)} \tag{9.32}
\end{equation*}
\]

\subsection*{9.3 CONSTRAINT REACTION FORCES}

The joint reaction forces can be expressed in terms of the Jacobian matrix of the constraint equations and a vector of Lagrange multipliers, as shown in Eq. 8.50, as
\[
\begin{equation*}
\mathbf{g}^{(c)}=\boldsymbol{\Phi}_{9}^{T} \boldsymbol{\lambda} \tag{9.33}
\end{equation*}
\]

This equation is studied for several commonly used constraints in Secs. 9.3.1 to 9.3.3.

\subsection*{9.3.1 Revolute Joint}

Consider two bodies \(i\) and \(j\) connected by a revolute joint, as shown in Fig. 9.9(a). The kinematic constraint equations for this joint are given by Eq. 4.7. The equations of motion for bodies \(i\) and \(j\) are
\[
\begin{equation*}
\mathbf{M}_{i} \ddot{\mathbf{q}}_{i}-\boldsymbol{\Phi}_{\mathbf{4} i}^{T} \boldsymbol{\lambda}=\mathbf{g}_{i} \tag{a}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{M}_{j} \ddot{\mathbf{q}}_{j}-\boldsymbol{\Phi}_{\mathbf{q}_{j}}^{T} \boldsymbol{\lambda}=\mathbf{g}_{j} \tag{b}
\end{equation*}
\]

Using the entries of the Jacobian matrix for a revolute joint from Table 4.2, we can write Eq. \(a\) in the expanded form
\[
\left[\begin{array}{ccc}
m & 0 & 0  \tag{c}\\
0 & m & 0 \\
0 & 0 & \mu
\end{array}\right]_{i}\left[\begin{array}{l}
\ddot{x} \\
\ddot{y} \\
\ddot{\phi}
\end{array}\right]_{i}-\left[\begin{array}{cc}
1 & 0 \\
0 & 1 \\
-\left(y_{i}^{P}-y_{i}\right) & \left(x_{i}^{P}-x_{i}\right)
\end{array}\right]^{2}\left[\begin{array}{l}
\lambda_{1} \\
\lambda_{2}
\end{array}\right]=\left[\begin{array}{c}
f_{(x)} \\
f_{(y)} \\
n
\end{array}\right]_{i}
\]

Since there are two algebraic equations in the constraint equations for a revolute joint, vector \(\boldsymbol{\lambda}\) is correspondingly a 2 -vector. Equation \(c\) can be written as the set of equations
\[
\begin{gather*}
m_{i} \ddot{x}_{i}=f_{\left(x_{i}\right)}+\lambda_{1}  \tag{9.34}\\
m_{i} \ddot{y}_{i}=f_{\left(y_{i}\right)}+\lambda_{2}  \tag{9.35}\\
\mu_{i} \ddot{\phi}_{i}=n_{i}-\left(y_{i}^{P}-y_{i}\right) \lambda_{1}+\left(x_{i}^{P}-x_{i}\right) \lambda_{2} \tag{9.36}
\end{gather*}
\]

(a)

(b)

Figure 9.9 (a) Two bodies connected by a revolute joint; (b) free-body diagrams for the bodies.

A free-body diagram for body \(i\) is shown in Fig. 9.9(b). Equation 9.34 indicates that besides \(f_{(x)}\), another force, \(\lambda_{1}\), acts in the \(x\) direction on body \(i\). Similarly, from Eq. 9.35 it is deduced that a force \(\lambda_{2}\) acts in the \(y\) direction on the same body. However, in order for Eq. 9.36 to be satisfied, forces \(\lambda_{1}\) and \(\lambda_{2}\) must act at point \(P_{i}\). The moment arm of \(\lambda_{1}\) is \(y_{i}^{p}-y_{i}\), and hence a moment \(\left(y_{i}^{P}-y_{i}\right) \lambda_{1}\) acts in the negative rotational direction. The moment arm of \(\lambda_{2}\) is \(x_{i}^{p}-x_{i}\), and so a moment \(\left(x_{i}^{P}-x_{i}\right) \lambda_{2}\) acts in the positive rotational direction.

Equations of motion for body \(j\), in the same form as Eq. \(c\) are written as follows:
\[
\left[\begin{array}{ccc}
m & 0 & 0  \tag{d}\\
0 & m & 0 \\
0 & 0 & \mu
\end{array}\right]_{j}\left[\begin{array}{l}
\ddot{x} \\
\ddot{y} \\
\ddot{\phi}
\end{array}\right]_{j}-\left[\begin{array}{cc}
-1 & 0 \\
0 & -1 \\
\left(y_{j}^{P}-y_{j}\right) & -\left(x_{j}^{P}-x_{j}\right)
\end{array}\right]^{2}\left[\begin{array}{l}
\lambda_{1} \\
\lambda_{2}
\end{array}\right]=\left[\begin{array}{c}
\mathbf{f}_{(x)} \\
\mathbf{f}_{(y)} \\
n
\end{array}\right]_{j}
\]
or
\[
\begin{gather*}
m_{j} \ddot{x}_{j}=f_{(x)_{j}}-\lambda_{1}  \tag{9.37}\\
m_{j} \ddot{y}_{\mathrm{j}}=f_{\left(y_{j}\right)}-\lambda_{2}  \tag{9.38}\\
\mu_{j} \ddot{\phi}_{j}=n_{j}+\left(y_{j}^{P}-y_{j}\right) \lambda_{1}-\left(x_{j}^{P}-x_{j}\right) \lambda_{2} \tag{9.39}
\end{gather*}
\]

It is shown in Fig. 9.9(b) that \(\lambda_{1}\) and \(\lambda_{2}\) are two forces acting at point \(P_{j}\) in the negative \(x\) and \(y\) directions, respectively. The moment arm for \(\lambda_{1}\) is \(y_{j}^{P}-y_{j}\) which yields a positive moment \(\left(y_{j}^{P}-y_{j}\right) \lambda_{1}\), and the moment arm for \(\lambda_{2}\) is \(x_{j}-x_{j}^{P}\), which yields a moment \(\left(x_{j}-x_{j}^{P}\right) \lambda_{2}\) or \(-\left(x_{j}^{P}-x_{j}\right) \lambda_{2}\).

The multipliers \(\lambda_{1}\) and \(\lambda_{2}\) can be positive or negative quantities. In any case, the reaction forces acting at the revolute joint on the connecting bodies are always equal in magnitude and opposite in direction.

\section*{Example 9.4}

Consider a system of two bodies connected by a revolute joint as shown in Fig. 9.9(a). The external forces acting on the system are gravity, a constant force of 10 N acting on body \(i\) in the negative \(x\) direction, and a constant force of 10 N acting on body \(j\) in the positive \(x\) direction. Calculate the joint reaction forces at the instant for which
\[
\begin{array}{cc}
\mathbf{q}_{i}=[1.58,1.59,0.6]^{T}, & \mathbf{q}_{j}=[3.4,1.96,0.2]^{T} \\
\dot{\mathbf{q}}_{i}=[1.1,0.2,-0.02]^{T}, & \dot{\mathbf{q}}_{j}=[1.14,0.24,0.03]^{T}
\end{array}
\]

The constant quantities for this system are: \(m_{i}=1.2, m_{j}=2, \mu_{i}=2.5, \mu_{j}=4\), \(\mathbf{s}_{i}^{\prime P}=[0.9,0.7]^{T}\), and \(\mathbf{s}_{j}^{\prime P}=[-1.3,1]^{T}\).
Solution The constraint equations for this revolute joint are
\[
\begin{aligned}
& x_{i}+0.9 \cos \phi_{i}-0.7 \sin \phi_{i}-x_{j}+1.3 \cos \phi_{j}+\sin \phi_{j}=0 \\
& y_{i}+0.9 \sin \phi_{i}+0.7 \cos \phi_{i}-y_{j}+1.3 \sin \phi_{j}-\cos \phi_{j}=0
\end{aligned}
\]

The Jacobian matrix for these constraints is
\[
\boldsymbol{\Phi}_{\mathbf{q}}=\left[\begin{array}{rrrrrr}
1 & 0 & -1.09 & -1 & 0 & 0.72 \\
0 & 1 & 0.35 & 0 & -1 & 1.47
\end{array}\right]
\]

From Eqs. 9.34 through 9.36, the equations of motion for body \(i\) are
\[
\begin{align*}
1.2 \ddot{x}_{i}-\lambda_{1} & =-10 \\
1.2 \ddot{y}_{i}-\lambda_{2} & =-11.77  \tag{1}\\
2.5 \ddot{\phi}_{i}+1.09 \lambda_{1}-0.35 \lambda_{2} & =0
\end{align*}
\]

Similarly, Eqs. 9.37 through 9.39 provide equations of motion for body \(j\) :
\[
\begin{align*}
2 \ddot{x}_{j}+\lambda_{1} & =10 \\
2 \ddot{y}_{j}+\lambda_{2} & =-19.62  \tag{2}\\
4 \ddot{\phi}_{j}-0.72 \lambda_{1}-1.47 \lambda_{2} & =0
\end{align*}
\]

Equations 1 and 2 are six equations in eight unknowns, and therefore two more equations are needed. These two additional equations are the kinematic acceleration equations. The second-time derivative of the constraint equations (refer to Table 4.3) can be used to obtain the acceleration equations for the revolute joint, as follows:
\[
\begin{align*}
& \ddot{x}_{i}-1.09 \ddot{\phi}_{i}-\ddot{x}_{j}+0.72 \ddot{\phi}_{j}=0 \\
& \ddot{y}_{i}+0.35 \ddot{\phi}_{i}-\ddot{y}_{j}+1.47 \ddot{\phi}_{j}=0 \tag{3}
\end{align*}
\]

The right side of the acceleration equations is approximately zero, i.e., \(\boldsymbol{\gamma}=[0.0017,-0.0002]^{T}\). Equations 1 through 3 can be solved to find
\[
\begin{aligned}
& \ddot{\mathbf{q}}_{i}=[-2.571,-10.154,-3.061]^{T}, \ddot{\mathbf{q}}_{j}=[1.543,-9.604,1.096]^{T}, \text { and } \\
& \lambda^{(c)}=[6.915,-0.413]^{T} . \text { Hence }, \mathbf{f}_{i}{ }^{(c)}=[6.915,-0.413]^{T} \text { and } \\
& \mathbf{f}_{j}^{(c)}=[-6.915,0.413]^{T} .
\end{aligned}
\]

\subsection*{9.3.2 Revolute-Revolute Joint}

Consider two bodies \(i\) and \(j\) connected by a revolute joint as shown in Fig. 9.10(a). The equations of motion for bodies \(i\) and \(j\), using the elements of the Jacobian matrix for a revolute-revolute joint from Table 4.2, are written as
\[
\begin{gather*}
m_{i} \ddot{x}_{i}=f_{\left(x_{i}\right.}+2\left(x_{i}^{P}-x_{j}^{P}\right) \lambda_{1}  \tag{9.40}\\
m_{i} \ddot{y}_{i}=f_{\left(y_{i}\right)}+2\left(y_{i}^{P}-y_{j}^{P}\right) \lambda_{1}  \tag{9.41}\\
\mu_{i} \ddot{\phi}_{i}=n_{i}-2\left[\left(x_{i}^{P}-x_{j}^{P}\right)\left(y_{i}^{P}-y_{i}\right)-\left(y_{i}^{P}-y_{j}^{P}\right)\left(x_{i}^{P}-x_{i}\right)\right] \lambda_{1} \tag{9.42}
\end{gather*}
\]

(b)

Figure 9.10 (a) Two bodies connected by a revolute-revolute joint. (b) Free-body diagrams for the bodies.
and
\[
\begin{gather*}
m_{j} \ddot{x}_{\mathrm{j}}=f_{\left(x_{j}\right)}-2\left(x_{i}^{p}-x_{j}^{p}\right) \lambda_{1}  \tag{9.43}\\
m_{j} \ddot{y}_{j}=f_{\left(y_{j}\right)}-2\left(y_{i}^{p}-y_{j}^{P}\right) \lambda_{1}  \tag{9.44}\\
\mu_{j} \ddot{\phi}_{j}=n_{j}+2\left[\left(x_{i}^{P}-x_{j}^{P}\right)\left(y_{j}^{P}-y_{j}\right)-\left(y_{i}^{P}-y_{j}^{P}\right)\left(x_{i}^{P}-x_{j}\right)\right] \lambda_{1} \tag{9.45}
\end{gather*}
\]

There is only one Lagrange multiplier in these equations; it corresponds to the one constraint equation describing the revolute-revolute joint.

From Eqs. 9.40 and 9.41 it is deduced that the terms \(2\left(x_{i}^{P}-x_{j}^{P}\right) \lambda_{1}\) and \(2\left(y_{i}^{P}-\right.\) \(\left.y_{j}^{\rho}\right) \lambda_{1}\) can be considered reaction forces acting on body \(i\) in the \(x\) and \(y\) directions, respectively. However, in order for Eq. 9.42 to be valid, these forces must act at point \(P_{i}\). Figure 9.10(b) shows the components of the reaction force and the moment arms at point \(P_{i}\). Similarly, Eqs. 9.43 and 9.44 show that the \(x\) and \(y\) components of the reaction force on body \(j\) are \(-2\left(x_{i}^{P}-x_{j}^{P}\right) \lambda_{1}\) and \(-2\left(y_{i}^{P}-y_{j}^{P}\right) \lambda_{1}\), and Eq. 9.45 indicates that these forces must act at point \(P_{j}\).

The reaction forces at points \(P_{i}\) and \(P_{j}\) are equal in magnitude and opposite in direction. These forces act along the revolute-revolute joint axis, i.e., a line passing through points \(P_{i}\) and \(P_{j}\).

\section*{Example 9.5}

For the two-body system of Example 9.2, assume that a revolute-revolute joint with a length \(l=0.175\) is connected between points \(Q_{1}\) and \(Q_{2}\), where \(\mathrm{s}_{1}^{\prime \theta}=\) \([-0.05,-0.05]^{T}\) and \(\mathbf{s}_{2}^{\prime Q}=[-0.03,0.0]^{T}\) (see the illustration). Write the equations of motion and calculate the reaction forces due to this added link if \(\dot{\mathbf{q}}_{1}=\) \([0.0,1.22,0.0]^{T}\) and \(\dot{\mathbf{q}}_{2}=[-0.71,-2.06,0.0]^{T}\).


Solution The global coordinates of \(Q_{1}\) and \(Q_{2}\) are found to be \(\mathbf{r}_{1}^{Q}=[-0.1\), \(0.129]^{T}\) and \(\mathbf{r}_{2}^{Q}=[0.071,0.092]^{T}\). The equations of motion of Example 9.2 are modified as follows:
\[
\left[\begin{array}{llllll}
0.2 & & & & & \\
& 0.2 & & & & \\
& & 0.03 & & & \\
& & & 0.15 & & \\
& & & & 0.15 & \\
& & & & & 0.02
\end{array}\right]\left[\begin{array}{r}
\ddot{x}_{1} \\
\ddot{y}_{1} \\
\ddot{\phi}_{1} \\
\ddot{x}_{2} \\
\ddot{y}_{2} \\
\ddot{\phi}_{2}
\end{array}\right]-\left[\begin{array}{r}
-0.342 \\
0.074 \\
0.024 \\
0.342 \\
-0.074 \\
-0.001
\end{array}\right] \lambda_{1}=\left[\begin{array}{r}
-1.878 \\
1.060 \\
0.520 \\
1.878 \\
-4.494 \\
-0.103
\end{array}\right]
\]

Since \(\boldsymbol{\Phi}_{\mathbf{q}}^{T}\) is already available and \(\boldsymbol{\gamma}\) can be found from Table 4.3, the kinematic acceleration equation for the revolute-revolute joint can now be found:
\[
-0.342 \ddot{x}_{1}+0.074 \ddot{y}_{1}+0.024 \ddot{\phi}_{1}+0.342 \ddot{x}_{2}-0.074 \ddot{y}_{2}-0.001 \ddot{\phi}_{2}=-22.525
\]

There are seven equations in seven unknowns that are solved to find
\[
\begin{aligned}
\lambda_{1} & =-22.828 \\
\ddot{\mathbf{q}} & =[29.646,-3.146,-0.929,-39.528,-18.698,-4.008]^{T}
\end{aligned}
\]

The reaction forces at \(Q_{1}\) and \(Q_{2}\) are \(\mathbf{f}_{1}^{(c)}=[7.807,-1.689]^{T}\) and \(\mathbf{f}_{2}^{(c)}=[-7.807\), \(1.689]^{T}\).

\subsection*{9.3.3 Translational Joint}

If a translational joint is considered between bodies \(i\) and \(j\) as shown in Fig. 9.11(a), the equations of motion for body \(i\) can be written as
\[
\begin{gather*}
m_{i} \ddot{x}_{i}=f_{(x)_{i}}+\left(y_{i}^{P}-y_{i}^{Q}\right) \lambda_{1}  \tag{9.46}\\
m_{i} \ddot{y}_{i}=f_{\left(y_{i}\right.}-\left(x_{i}^{P}-x_{i}^{Q}\right) \lambda_{1}  \tag{9.47}\\
\mu_{i} \ddot{\phi}_{i}=n_{i}-\left[\left(x_{j}^{P}-x_{i}\right)\left(x_{i}^{P}-x_{i}^{Q}\right)+\left(y_{j}^{P}-y_{i}\right)\left(y_{i}^{P}-y_{i}^{Q}\right)\right] \lambda_{1}+\lambda_{2} \tag{9.48}
\end{gather*}
\]

The free-body diagram for body \(i\) is shown in Fig. 9.11(b). In this diagram the force associated with \(\lambda_{1}\) is the reaction force caused by the first constraint equation. It is a simple matter to show that this force, \(\vec{k}_{1}\), is perpendicular to the line of translation. The contribution of the second constraint equation is a couple acting on body \(i\). Note that \(\lambda_{2}\) may be a positive or negative quantity.

In order to find a simpler physical description of the reaction force \(\vec{k}_{1}\), one should not locate the points \(P_{i}, Q_{i}\), and \(P_{j}\) arbitrarily on the line of translation. These points can be selected to coincide with the edges of the slider, as shown in Fig. 9.12(a). If \(P_{j}\) is allowed to slide with the slider, then the reaction force \(\vec{k}_{1}\) always acts at the edge of the slider, as shown in Fig. 9.12(b).

\subsection*{9.4 SYSTEM OF PLANAR EQUATIONS OF MOTION}

For an unconstrained mechanical system, the equations of motion are as given in Eq. 9.3:
\[
\begin{equation*}
\mathbf{M} \ddot{\mathbf{q}}=\mathbf{g} \tag{9.49}
\end{equation*}
\]

Vector \(\mathbf{g}\) is, in general, a function of \(\mathbf{q}, \dot{\mathbf{q}}\), and \(t\). If, at an instant, \({ }^{4} \mathbf{q}\) and \(\dot{\mathbf{q}}\) are known, \(\ddot{\mathbf{q}}\) can be found as follows:
\[
\begin{equation*}
\ddot{\mathbf{q}}=\mathbf{M}^{-1} \mathbf{g} \tag{9.50}
\end{equation*}
\]


Figure 9.11 (a) Two bodies connected by a translational joint and (b) the reaction forces acting on body \(i\) associated with a translational joint.


Figure 9.12 (a) A typical translational joint. (b) Forces acting on body \(i\) by the sliding body \(j\).

For a well-posed problem, every body in the system must have nonzero mass and moment of inertia. Therefore, \(\mathbf{M}\) is a diagonal nonsingular matrix, and \(\mathbf{M}^{-1}\) can be calculated easily.

For a constrained mechanical system with \(m\) independent constraints
\[
\begin{equation*}
\boldsymbol{\Phi}=0 \tag{9.51}
\end{equation*}
\]
the velocity and acceleration equations are
\[
\begin{equation*}
\Phi_{q} \dot{\mathbf{q}}=0 \tag{9.52}
\end{equation*}
\]
and
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{q}} \ddot{\mathbf{q}}-\gamma=0 \tag{9.53}
\end{equation*}
\]

The equations of motion for this constrained system are as given in Eq. 9.6:
\[
\begin{equation*}
\mathbf{M} \ddot{\mathbf{q}}-\boldsymbol{\Phi}_{\mathrm{q}}^{T} \boldsymbol{\lambda}=\mathbf{g} \tag{9.54}
\end{equation*}
\]

Equation 9.53 can be appended to Eq. 9.54 and the result can be written as
\[
\left[\begin{array}{ll}
\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{T}  \tag{9.55}\\
\boldsymbol{\Phi}_{\mathbf{q}} & 0
\end{array}\right]\left[\begin{array}{r}
\ddot{\mathbf{q}} \\
-\boldsymbol{\lambda}
\end{array}\right]=\left[\begin{array}{l}
\mathbf{g} \\
\gamma
\end{array}\right]
\]

The Jacobian matrix \(\Phi_{\mathbf{q}}\) is a function of \(\mathbf{q}\), and vectors \(\mathbf{g}\) and \(\boldsymbol{\gamma}\) are functions of \(\mathbf{q}, \dot{\mathbf{q}}\), and \(t\). Therefore, at any given instant, if \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) are known, Eq. 9.55 provides \(n+m\) linear algebraic equations in \(n+m\) unknowns that can be solved for \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\). For constrained mechanical systems, Eqs. 9.51 through 9.54 must be considered together as the system equations of motion.

A FORTRAN program for solving the planar equations of motion is presented in Chap. 10. Numerical methods for solving ordinary differential equations (for unconstrained systems) and mixed algebraic-differential equations (for constrained systems) are discussed in detail in Chaps. 12 and 13.

\subsection*{9.5 STATIC FORCES}

The subjects that are discussed in this section and in Secs. 9.6 and 9.7 are valid for both planar and spatial systems. However, because of the simplicity of illustrations for planar systems, these alone will be treated in this chapter.

A mechanical system becomes a structure (a nonmovable system) when the number of independent constraint equations is equal to the number of coordinates in the system. For example, the system shown in Fig. 9.13 contains 8 links and the ground, which yields \(n=(8+1) \times 3=27\) coordinates. There are 12 revolute joints in the system, resulting in 24 algebraic equations, and 3 algebraic equations for the ground constraints, totaling \(m=24+3=27\). This yields \(k=27-27=0\) degree of freedom. In general, for a structure with \(n\) coordinates \(\mathbf{q}, n\) constraint equations can be written as
\[
\boldsymbol{\Phi}(\mathbf{q})=0
\]

These equations can be solved to find the coordinates \(\mathbf{q}\). Since for a structure \(\dot{\mathbf{q}}=\ddot{\mathbf{q}}=\boldsymbol{0}\), Eq. 9.4 yields
\[
\begin{equation*}
\mathbf{g}^{(c)}=-\mathbf{g} \tag{9.56}
\end{equation*}
\]

This equation shows that the constraint reaction forces acting on each body of the system can be found directly from the vector of forces. In order to find the constraint reac-


Figure 9.13 A planar truss subject to an external force.
tion forces at each joint, the Lagrange multipliers can be determined as
\[
\begin{equation*}
\boldsymbol{\lambda}=-\left(\boldsymbol{\Phi}_{4}^{T}\right)^{-1} \mathbf{g} \tag{9.57}
\end{equation*}
\]

The inverse of \(\boldsymbol{\Phi}_{\mathrm{q}}^{T}\) exists, since it is assumed that the constraints are independent and, for a system with 0 degree of freedom, \(\boldsymbol{\Phi}_{\mathrm{q}}\) is a square matrix. After the determination of \(\boldsymbol{\lambda}\), a process similar to that of Secs. 9.3.1 to 9.3 .3 can be employed to find the reaction forces at each point.

\subsection*{9.6 STATIC BALANCE FORCES}

Consider the planar robot manipulator shown in Fig. 9.14. The motion of the robot is controlled by three electric motors (rotational actuators) acting about the axes of revolute joints \(A, B\), and \(C\). What moments must the motors apply on the bodies in order to keep the system in equilibrium, in the configuration shown? The moments (or forces, in other examples) are referred to as the static balance forces. If the number of unknown static balance forces is equal to the number of degrees of freedom, then the forces can be found by the following method.

The vector of forces is split into two vectors, as follows:
\[
\begin{equation*}
\mathbf{g}=\mathbf{g}^{(k)}+\mathbf{g}^{(k)} \tag{9.58}
\end{equation*}
\]
where \(\mathbf{g}^{(k)}\) contains the known forces acting on the system and \(\mathbf{g}^{(u)}\) is the vector of unknown forces, which can be the static balance forces. Hence, the equations of motion


Figure 9.14 A planar robot manipulator.
for the system are written as
\[
\mathbf{M} \ddot{\mathbf{q}}-\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}=\mathbf{g}^{(k)}+\mathbf{g}^{(u)}
\]
or
\[
\begin{equation*}
-\boldsymbol{\Phi}_{\mathrm{q}}^{T} \boldsymbol{\lambda}=\mathbf{g}^{(k)}+\mathbf{g}^{(u)} \tag{a}
\end{equation*}
\]
since in a static configuration \(\ddot{\mathbf{q}}=0\). The same mechanical system can be kept in equilibrium, in its given configuration, if the actuators with unknown forces (or moments) are replaced by artificial constraint equations, equal in number to the number of degrees of freedom. These artificial constraints are denoted here by \(k\) algebraic equations as
\[
\begin{equation*}
\Phi^{*}(\mathbf{q})=0 \tag{9.59}
\end{equation*}
\]

For example, for the robot manipulator of Fig. 9.14, three artificial constraints are defined as
\[
\begin{aligned}
& \Phi_{1}^{*} \equiv \phi_{2}-\phi_{1}-c_{1}=0 \\
& \Phi_{2}^{*} \equiv \phi_{3}-\phi_{2}-c_{2}=0 \\
& \Phi_{3}^{*} \equiv \phi_{4}-\phi_{3}-c_{3}=0
\end{aligned}
\]

These constraints keep the relative angles between bodies constant.
If these equations are appended to the original \(m\) kinematic constraint equations \(\boldsymbol{\Phi}(\mathbf{q})=\boldsymbol{0}\), then the equations of motion become
\[
\begin{equation*}
-\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}-\boldsymbol{\Phi}_{\mathbf{q}}^{* T} \boldsymbol{\lambda}^{*}=\mathbf{g}^{(k)} \tag{9.60}
\end{equation*}
\]
where \(\ddot{\mathbf{q}}=\mathbf{0}\). Comparing Eq. \(a\) and Eq. 9.60 results in
\[
\begin{equation*}
\mathbf{g}^{(u)}=\boldsymbol{\Phi}_{\mathbf{q}}^{* T} \boldsymbol{\lambda}^{*} \tag{9.61}
\end{equation*}
\]

Equation 9.60 represents \(n\) linear algebraic equations in \(n\) unknowns. The \(n\) unknowns are \(m\) multipliers \(\boldsymbol{\lambda}\) and \(k\) multipliers \(\boldsymbol{\lambda}^{*}\). If these equations are solved for \(\boldsymbol{\lambda}\) and \(\boldsymbol{\lambda}^{*}\), then Eq. 9.61 yields the unknown static balance forces \(\mathbf{g}^{(u)}\).

\section*{Example 9.6}

The motion of the five-bar linkage in Fig. 9.15 is controlled by two actuators as shown in Fig. 9.15(a). The number of actuators is the same as the number of degrees of freedom. In order to find what forces applied by the actuators will keep the system in equilibrium, the actuators are replaced by two revolute-revolute joints, as shown in Fig. 9.15(b). The two artificial constraints for the revolute-


Figure 9.15 A five-bar mechanism.
revolute joints lower the number of degrees of freedom from 2 to 0 . If the system of Fig. \(9.15(b)\) is solved for the reaction forces along the revolute-revolute joints, then Eq. 9.61 will yield the desired actuator forces in the equivalent system of Fig. 9.15(a).

\subsection*{9.7 KINETOSTATIC ANALYSIS}

If the forces acting on a mechanical system are known, then the equations of motion can be solved to obtain the motion of the system. This process is known as forward dynamic analysis. In some problems, a specified motion for a mechanical system is sought and the objective is to determine the forces that must act on the system to produce such a motion. This process is usually referred to as inverse dynamic or kinetostatic analysis.

As an example, consider the 3-degrees of freedom robot manipulators of Fig. 9.14. Assume that the end effector, point \(P\), must move along a known path, such as the straight line shown in Fig. 9.16. The range of interest is from \(E\) to \(F\), and it is further required that point \(P\) keep a constant velocity within this range. One additional requirement is that the angle of body 4 must remain unchanged with respect to the line \(E F\). The objective is to find the torque that actuators \(A, B\), and \(C\) must supply, as a function of time, in order to produce such motion.


Figure 9.16 The end-effector of the robot must move along a specified path, keeping a specified orientation.

This problem can be solved by specifying \(k\) driving constraint equations, equal to the number of degrees of freedom, to describe the required motion - such as
\[
\begin{equation*}
\Phi^{*}(\mathbf{q}, t)=0 \tag{9.62}
\end{equation*}
\]

For example, the driving constraints for the robot manipulator are
\[
\begin{aligned}
& \Phi_{1}^{*} \equiv x_{4}^{P}-a_{1}-v_{1} t=0 \\
& \Phi_{2}^{*} \equiv y_{4}^{P}-a_{2}-v_{2} t=0 \\
& \Phi_{3}^{*} \equiv \phi_{4}-c_{1}=0
\end{aligned}
\]
where \(a_{1}\) and \(a_{2}\) are the initial \(x\) and \(y\) coordinates of \(P\) at \(t=0, v_{1}\) and \(v_{2}\) are the constant velocities of \(P\) along the \(x\) and \(y\) axes, and \(c_{1}\) is the specified angle for body 4 .

The \(k\) driving constraints of Eq. 9.62 are appended to \(m\) kinematic constraints to yield \(n\) constraints in \(n\) unknowns. This is a kinematics problem that can be solved by the method described in Sec. 3.2.2. The time \(t\) is varied from \(t=0\) to \(t^{e}\) in order to move point \(P\) from \(E\) to \(F\). At every time step, position, velocity, and acceleration analyses are performed and the results, i.e., \(\mathbf{q}, \dot{\mathbf{q}}\), and \(\ddot{\mathbf{q}}\), are saved in numerical form.

The equations of motion can be written as
\[
\begin{equation*}
\mathbf{M} \ddot{\mathbf{q}}-\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\mathcal { L }}=\mathbf{g}^{(k)}+\mathbf{g}^{(u)} \tag{9.63}
\end{equation*}
\]
where \(\mathbf{g}^{(k)}\) contains the known forces, such as gravity, and \(\mathbf{g}^{(u)}\) contains the unknown forces of the actuators (the moments of the actuators \(A, B\), and \(C\) in the robot example). Since \(\mathbf{q}, \dot{\mathbf{q}}\), and \(\ddot{\mathbf{q}}\) are calculated kinematically, \(\mathbf{M} \ddot{\mathbf{q}}, \boldsymbol{\Phi}_{\mathbf{q}}\), and \(\mathbf{g}^{(k)}\) are known. Therefore,
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{q}}^{r} \boldsymbol{\lambda}+\mathbf{g}^{(u)}=\mathbf{M} \ddot{\mathbf{q}}-\mathbf{g}^{(k)} \tag{9.64}
\end{equation*}
\]
can be solved for \(\boldsymbol{\lambda}\) and \(\mathbf{g}^{(t)}\). Equation 9.64 represents \(n\) equations in \(m\) unknowns \(\boldsymbol{\lambda}\) and \(k\) unknowns embedded in \(\mathbf{g}^{(u)}\). These equations can be solved at every time step from \(t=0\) to \(t^{e}\), and the actuator forces can be found numerically as functions of time.

\section*{PROBLEMS}
9.1 A force \(\vec{f}\) acts at point \(P\) on body \(i\) as shown in Fig. P. 9.1. This force keeps a fixed angle \(\alpha\) with vector \(\vec{s}_{i}^{P}\). Find the component of this force and its corresponding moment for inclusion in vector \(\mathbf{g}_{i}\).
9.2 Repeat Prob. 9.1 and assume that the force keeps a constant angle \(\beta\) with the global \(x\) axis as shown in Fig. P. 9.2.


Figure P. 9.1


Figure P. 9.2
9.3 A multibody model of a vehicle is assembled in the configuration shown in Fig. P. 9.3(a), where the gravitational force is perpendicular to the road. If the vehicle is placed on a slope


Figure P. 9.3
as shown in Fig. P. 9.3(b), the gravitational force makes an angle \(\alpha\) with the normal to the road. Instead of changing the coordinate values from model (a) to model (b), devise a simple method to modify the vector of forces by rotating the direction of the gravitational force with respect to the global coordinate axes.
9.4 Derive the equations of motion for a body when the origin of the local coordinate system does not coincide with the body center of mass, as shown in Fig. P. 9.4.
9.5 For the single pendulum shown in Fig. P. 9.5, write the equations of motion in terms of Cartesian coordinates. Use the kinematic acceleration equations to eliminate the translational components of acceleration and Lagrange multipliers. The resultant equation should be a second-order differential equation in terms of \(\ddot{\phi}_{2}\).


Figure P. 9.4


Figure P. 9.5
9.6 The rod shown in Fig. P. 9.6 is attached to the ground by a spring. Write the equations of motion for the rod. What are the initial conditions on the coordinates? Assume \(m=4\), \(\mu=3, k=50\), and \(l^{0}=1\).
9.7 Two unconstrained bodies are connected to each other and the ground by springs and dampers as shown in Fig. P. 9.7. Let \(m_{1}=4, \mu_{1}=3, m_{2}=3, \mu_{2}=1, k=40, l^{0}=1.2\), and \(d=12\).
(a) Define local and global coordinate systems.
(b) Determine the initial condition for the vector of coordinates.


Figure P. 9.6
Figure P. 9.7
(c) If \(\dot{x}_{1}=-0.3\) and \(\dot{\phi}_{2}=0.05\), determine the initial condition for the vector of velocities.
(d) Write the equations of motion for the system.
9.8 Two rods are connected to each other by a revolute joint as shown in Fig. P. 9.8. Let \(m_{1}=m_{2}=6, \mu_{1}=\mu_{2}=12.5, k_{1}=20, l_{1}^{0}=5, k_{2}=30, l_{2}^{0}=4.5\), and \(d_{2}=6\).
(a) Define local and global coordinate systems.
(b) Determine the initial condition for the vector of coordinates.
(c) Test the constraint equations for any violations. In case of violation, correct the initial conditions.
(d) If rod 1 has a positive rotational velocity of \(\dot{\phi}_{1}=0.01 \mathrm{rad} / \mathrm{s}\), find a proper set of initial conditions for the vector of velocities consistent with the constraints.
(e) Write the equations of motion for the system.


Figure P. 9.8
9.9 Two masses \(m_{1}\) and \(m_{2}\) go through a one-dimensional motion in the \(x\) direction as shown in Fig. P. 9.9. Assume that \(m_{1}=1, m_{2}=2, k_{1}=10, k_{2}=15, l_{1}^{0}=1.25, l_{2}^{0}=1, d_{1}=5\), \(d_{2}=6, a=1\), and \(b=3\).
(a) Write the equations of motion for this system in terms of \(\ddot{x}_{1}\) and \(\ddot{x}_{2}\) (do not combine the two masses into a single mass).
(b) At the instant shown, \(x_{1}=1.2, x_{2}=2.2\), and \(\dot{x}_{1}=\dot{x}_{2}=0.3\). Solve the equations of motion for the accelerations.
(c) Draw the free-body diagram for each mass and show all the forces in their proper directions.


Figure P. 9.9
9.10 The radial deformation of an automobile wheel may be modeled \({ }^{4}\) by a translational springdamper combination as long as the wheel is in contact with the ground. Knowing the radius, position, and velocity of the wheel, find:
(a) The coordinates of the contact point (center of the contact patch)
(b) The spring force
(c) The damper force
(d) The components of the resultant force acting on the wheel

Repeat this process for the three cases shown in Fig. P. 9.10. Assume that complete geometrical data for the road are available.


Figure P. 9.10
9.11 Repeat Prob. 9.10 for the case where the wheel and the ground are in contact at two points (patches) as shown in Fig. P. 9.11. The resultant force acting on the wheel can be found as the sum of forces from two spring-damper elements.


Figure P. 9.11
9.12 Deformation of the cantilever beam shown in Fig. P. 9.12(a) may be modeled by a rigid body, a revolute joint, and a rotational spring, as shown in Fig. P. 9.12(b). For a beam with

(a)

(b)

(c)

Figure P. 9.12
length \(l\) under an external load \(f\), the free end yields a displacement \(d\). The equivalent rigidbody model yields the same displacement if the spring stiffness \(k_{1}\) is selected properly.
(a) If the beam is modeled by two rigid bodies, two revolute joints, and two rotational springs with stiffness \(k_{2}\), as shown in Fig. P. 9.12(c), find an approximate formula for \(k_{2}\) in terms of \(k_{1}\) (for small deformations \(d \ll l\) ).
(b) If the beam is modeled by \(n\) equal-length bodies, \(n\) revolute joints, and \(n\) rotational springs, find a formula for \(k_{n}\) in terms of \(k_{1}\).
9.13 For two bodies connected by a revolute-translational joint, show that the reaction force between the bodies can be found from the term \(\Phi_{\mathrm{q}}^{T}\) in the equations of motion. Show the forces on free-body diagrams of the two bodies.

\section*{10}

\section*{A FORTRAN Program} for Analysis of Planar Dynamics

In this chapter a FORTRAN program for planar dynamic analysis is presented. The program employs several subroutines from the kinematic analysis program (KAP) in Chap. 5 without any modifications. This program can model constant forces, gravity, and translational elements consisting of a spring, a damper, and/or an actuator. The program is organized in a form that allows it to be expanded to include other types of force elements. The problems at the end of this chapter provide a pattern to use for expanding the program.

Numerical methods for solving a system of mixed algebraic and differential equations, such as the equations of motion given in Sec. 9.4, are discussed in detail in Chaps. 12 and 13. However, in order to show how the dynamic analysis program (DAP) listed in this chapter solves the equations of motion, a brief discussion is provided in Sec. 10.1.

The listed program can solve the equations of motion for the dynamic response of constrained systems. In addition, this program can solve static problems as formulated in Secs. 9.5 and 9.6.

\subsection*{10.1 SOLVING THE EQUATIONS OF MOTION}

For an unconstrained mechanical system, the equations of motion are given by
\[
\begin{equation*}
\mathbf{M} \ddot{\boldsymbol{q}}=\mathbf{g} \tag{10.1}
\end{equation*}
\]
with initial conditions on the coordinates and velocities given as \(\mathbf{q}^{0}\) and \(\dot{\mathbf{q}}^{0}\). Since \(\mathbf{M}\) is a constant diagonal matrix and \(\mathbf{g}\) can be a function of \(\mathbf{q}\) and \(\dot{\mathbf{q}}\), Eq. 10.1 can be solved for the unknowns \(\ddot{\boldsymbol{q}}^{0}\) at the initial time.

The numerical integration algorithms that will be discussed in Chap. 12, can integrate the velocity and acceleration vectors at a given time and obtain the position and velocity vectors at a new time step. If the position and velocity vectors are appended together as the vector
\[
\mathbf{y}=\left[\begin{array}{c}
\mathbf{q}  \tag{10.2}\\
\dot{\mathbf{q}}
\end{array}\right]
\]
then the velocity and acceleration vectors are represented in the vector
\[
\dot{\mathbf{y}}=\left[\begin{array}{l}
\dot{\mathbf{q}}  \tag{10.3}\\
\ddot{\mathbf{q}}
\end{array}\right]
\]

At time \(t=t^{i}\), vector \(\dot{\mathbf{y}}^{(i)}\) can be integrated numerically to obtain \(\mathbf{y}^{(i+1)}\), where \(t^{i+1}=\) \(t^{i}+\Delta t\); i.e.,
\[
\begin{equation*}
\dot{\mathbf{y}}^{(i)} \xrightarrow{\text { (integrate) }} \mathbf{y}^{(i+1)} \tag{10.4}
\end{equation*}
\]

Initially, at \(i=0\), the initial conditions on \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) are required to start the integration process.

For a constrained mechanical system, the equations of motion are, from Eq. 9.55,
\[
\left[\begin{array}{cc}
\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{r}  \tag{10.5}\\
\boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0}
\end{array}\right]\left[\begin{array}{r}
\ddot{\mathbf{q}} \\
-\lambda
\end{array}\right]=\left[\begin{array}{c}
\mathbf{g} \\
\gamma
\end{array}\right]
\]
with initial conditions \(\mathbf{q}^{\mathbf{0}}\) and \(\dot{\mathbf{q}}^{0}\). The Jacobian \(\boldsymbol{\Phi}_{\mathbf{q}}\) is a function of \(\mathbf{q}\), and \(\mathbf{g}\) and \(\gamma\) are functions of \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) that can be evaluated at the initial time. Hence, Eq. 10.5 can be solved for the unknowns at the initial time, i.e., \(\dot{\mathbf{q}}^{0}\) and \(\boldsymbol{\lambda}^{0}\).

The initial conditions on \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) for a constrained system cannot be specified arbitrarily. The initial conditions \(\mathbf{q}^{0}\) and \(\dot{\mathbf{q}}^{0}\) must satisfy the constraint equations; i.e.,
\[
\begin{equation*}
\boldsymbol{\Phi}=\boldsymbol{0} \quad\left(\text { for } \mathbf{q}=\mathbf{q}^{0}\right) \tag{10.6}
\end{equation*}
\]
and
\[
\begin{equation*}
\Phi_{\mathbf{q}} \dot{\mathbf{q}}=0 \quad\left(\text { for } \mathbf{q}=\mathbf{q}^{0} \text { and } \dot{\mathbf{q}}=\dot{\mathbf{q}}^{0}\right) \tag{10.7}
\end{equation*}
\]

For the constrained equations of motion, vectors \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) are as defined in Eqs. 10.2 and 10.3, and a numerical integration algorithm is applied to process Eq. 10.4. This is a simple but crude method of solving the constrained equations of motion. The possible error accumulation associated with this method and the techniques for resolving the problem are discussed in Chap. 13.

\subsection*{10.2 DYNAMIC ANALYSIS PROGRAM (DAP)}

The main routine of the dynamic analysis program performs three major tasks:
1. Reads input data, either directly or by making calls to other input subroutines
2. Splits the working arrays A and IA into smaller subarrays biy defining pointers
3. Makes calls to subroutine RUNG4 for integrating the equations of motion, or subroutine STATIC for static analysis

A detailed explanation of these tasks follows.
- Input/Output: Same as for KAP, Sec. 5.1.
- Working Arrays: Same as for KAP, Sec. 5.1.
- Number of Elements: The first set of data the program requests is ENTER NB, NR, NT, NG, NS, NSP, NP
which are defined as follows:
NB Number of bodies in the system, including ground
NR Number of revolute joints in the system
NT Number of translational joints in the system
NG Number of bodies that are attached to (or considered to be) ground
NS Number of simple constraints in the system
NSP Number of translational spring, damper, or actuator elements
NP Number of points of interest
The program computes the number of coordinates N and the total number of constraint equations M from the above information. If M is greater than N , then an error message is given. Otherwise, the program continues.
- Subarrays: The working arrays A and IA are divided into smaller subarrays, according to the number of elements in the problem. The subarrays and their corresponding pointers and lengths are shown in Fig. 10.1. The function of the subarrays is explained in Secs. 10.2.1, 10.2.2, and 10.2.4.
- Input Data: The main program makes calls to other subroutines to read additional information for the problem at hand. These subroutines are discussed in Sec. 10.2.1.
- Time Parameters: Same as for KAP, Sec. 5.1.
- Static Analysis: If \(\mathrm{N}=\mathrm{M}\), then a call is made to subroutine STATIC. For static analysis, the time parameters are not used.
- Dynamic Analysis: If \(\mathrm{N}>\mathrm{M}\), then a call to subroutine RUNG4 is made to start the integration process for dynamic analysis.

The main routine for DAP is as follows:
```

C. . . . . . . . . . .DYNAMMIC ANALYSIS/STATIC ANALYSIS . . . . . . . .

```
```

    COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
    + N17,N18,N19,N2O,N21,N22,N23,N24
COMMON /ROMCOL/ IR,IC,M,N,NPM,NC2
OOMMDN /TIME / TO,TE,DT,T
DIMENSION A(3000),IA(500)
C.....If more storage space in A and IA arrays are needed increase the
C....dimension and update MAXA and MAXIA accordingly
MAXA =3000
MAXIA=500
C.....Read number of bodies, revolute joints, translational joints,
C....grounded bodies, simple constraints, spring-damper-actuators,
C.....points of interest
10 WRITE (1,200)
READ (1,* ) NB,NR,NT,NG,NS,NSP ,NP
C.....Determine number of coordinates N and number of constraint M
N=3*NB
M=2*(NR+NT)+3*NG+NS
NPM=N+M
NC2=N+N
C....N must be greater to M
IF (M.LE.N) GOTO 20
WRITE(1,210) N,M
GOTO 10
C....Define pointers and split A and IA into subarrays
C.....Refer to Figure 10.1
20 N1=1
N2=N1+4*NR
N3=N2+7*NT
N4=N3+3*NG
N5mN4+ NS
N6=N5+12*NSP
N7*N6+7*NB
M1=1
M2=M1+2*NR
M3=M2+2*NT
M4=M3+6*NG
M5-M4+2*NS
M7-M5+2*NSP
N10=N7+2*NP
N11=N10+N
N12=N11+N
N13=N12+N
N14=N13+M
N15=N14+N*M
N16=N15+M
N17=N16+NPM
N18=N17+N
N19 =N18+M
N20}=\textrm{N}19+NPM*NPM
N21=N20+NC2
N22=N21+NC2
N23=N22+NC2
NUSED=N23+NC2-1
M10=M7+NP
MUSED-M1 0+NPM-1
C.....Check for sufficient storage space in A and IA arrays
IF(NUSED.LE.MAXA .AND. MUSED.LE.MAXIA)GOTO }3
WRITE(1, 220) NUSED,MUSED
STOP
C....'Rigid body information
30 CALL INBODY (A(N10),A(N11),A(N6),NB)
C.....Read revolute joints data
IF (NR.GT.0) CALL INRVLT (A(N1),IA(M1),NR)

```

Figure 10.1 Subarrays of A and IA with their corresponding pointers and lengths.
```

C.....Read translational joints data
IF (NT.GT.0) CALL INTRAN (A(N2),IA(M2),NT,A(N10),NB)
C.....Read ground constraints data
NG3=3*NG
IF (NG.GT.0) CALL INGRND (A(N3),IA(M3),NG,A(N10),NG3,NB)
C.....Read simple constraints data
IF (NS.GT.0) CALL INSMPL (A(N4), IA(M4),NS,A(N10),NB)
C.....Read spring-damper-actuator elements data
IF (NSP.GT.0) CALL INSPRG (A(N5),IA(M5),NSP)
C....Read special points of interest
IF (NP.GT.0) CALL INPOIN (A(N7),IA(M7),NP)
C.....Read initial time, final time, and time increments
WRITE(1,230)
READ (1,* ) TO,TE,DT
C.....End of input data
EPSLU=0.00001
C.....Static analysis
IF (M.EQ.N) CALL STATIC (A,IA,MAXA,MAXIA)
C.....Start dynamic analysis
C....Transfer Q and QD to YS
CALL TRANSF (A(N22),A(N10),NC2)
C.....Start numerical integration
CALL RUNG4 (A,IA,A(N20),A(N21),A(N22),A(N23),MAXA,MAXIA)
STOP
200 FORMAT(5X, 'ENTER NB,NR,NT,NG,NS ,NSP,NP')
210 FORMAT (5X,'***INPUT ERROR*** N =',I 3,' M =',I3)
220 FORMAT(5X,'***DIMENSION ON A AND/OR IA ARRAYS NOT SUFFICIENT***',
+ /,10X,'MINIMMM DIMENSION ON A MUST BE',I5,
+ /,10X,'MINIMMM DIMENSION ON IA MUST BE',I5)
230 FORMAT(5X,'ENTER TSTART, TEND, AND STEP')
END

```

\subsection*{10.2.1 Model Description Subroutines}

The following subroutines are called by the main routine of DAP to read the description of the model.

Subroutine INBODY. This subroutine reads initial conditions on \(x_{i}, y_{i}\), and \(\phi_{i}\), initial velocities \(\dot{x}_{i}, \dot{y}_{i}\), and \(\phi_{i}\), mass \(m_{i}\), moment of inertia \(\mu_{i}\), constant external applied forces acting at the center of mass \(f_{(x)_{i}}\) and \(f_{(y)}\), and moments \(n_{i}\). The prompt from this subroutine is repeated for each body \(i\) as follows:

FOR BODY i ENTER INITIAL COND. ON X, Y, PHI
INITIAL CONDITIONS ON XD, YD, PHID
MASS, MOMENT OF INERTIA
CONSTANT FORCE-MOMENT FX, FY, N
The coordinates and velocities are stored in Q and QD (just as in KAP). The rest of the information is stored in array RB , dimensioned as \(\mathrm{RB}(\mathrm{NB}, 7)\). For example, for body \(i\),
\[
\begin{array}{lll}
\sin \phi_{i} & \rightarrow & \mathrm{RB}(\mathrm{I}, 1) \\
\cos \phi_{i} & \rightarrow & \mathrm{RB}(\mathrm{I}, 2) \\
m_{i} & \rightarrow & \mathrm{RB}(\mathrm{I}, 3) \\
\mu_{i} & \rightarrow & \mathrm{RB}(\mathrm{I}, 4) \\
f_{(x)_{i}} & \rightarrow & \mathrm{RB}(\mathrm{I}, 5) \\
f_{(y)_{i}}-w_{i} & \rightarrow & \mathrm{RB}(\mathrm{I}, 6) \\
n_{i} & & \rightarrow \\
\mathrm{RB}(\mathrm{I}, 7)
\end{array}
\]

The weight of each body is computed as \(w_{i}=9.81 m_{i}\) (where the mass is in SI units) and is added in the negative \(y\) direction to \(f_{\left(y_{2}\right)}\) (refer to Sec. 9.2.1).

Subroutine INBODY is as follows:
```

    SUBROUTINE INBODY ( \(\mathrm{Q}, \mathrm{QD}, \mathrm{RB}, \mathrm{NB}\) )
    DIMENS ION \(\mathrm{Q}(3, \mathrm{NB}), \mathrm{QD}(3, \mathrm{NB}), \mathrm{RB}(\mathrm{NB}, 7)\)
    DO \(10 \mathrm{I}=1\), NB
        \(\operatorname{WRITE}(1,200)\) I
        \(\operatorname{READ}(1, *)(Q(J, I), J=1,3),(Q D(J, I), J=1,3),(R B(I, J), J=3,7)\)
    $10 \quad \operatorname{RB}(\mathrm{I}, 6)=\mathrm{RB}(\mathrm{I}, 6)-\mathrm{RB}(\mathrm{I}, 3) * 9.81$
RETURN
200 FORMAT (5X,'FOR BODY',I4,' ENTER INITIAL OOND. ON X, Y, PHI',/,
$+\quad 10 \mathrm{X}$, 'INITIAL CONDITIONS ON XD, YD, PHID',/,

+ 10X,'MASS, MOMENT OF INERTIA',/,
$+\quad 10 \mathrm{X}$, 'CONSTANT FORCE-MOMENT FX, FY, N')
END

```

Subroutine INRVLT. Same as for KAP, Sec. 5.1.1.
Subroutine INTRAN. Same as for KAP, Sec. 5.1.1.
Subroutine INGRND. Same as for KAP, Sec. 5.1.1.
Subroutine INSMPL. Same as for KAP, Sec. 5.1.1.
Subroutine INSPRG. This subroutine is called if NSP \(>0\) to read information on spring, damper, and actuator elements (refer to Sec. 9.2.3, 9.2.4, and 9.2.5). An element can have one spring, one damper, and one actuator as long as the attachment points are shared. One possibility is that an element to contain only one spring and no damper or actuator. The prompt given by this subroutine is
```

FOR SPRING ELEMENT NO. k ENTER BODY NOS. I and J
XI-P-I, ETA-P-I, XI-P-J, ETA-P-J
SPRING CONST., DAMPING COEF., ACTUATOR FORCE, UNDEFORMED SPRING
LENGTH

```

This prompt is repeated for \(k=1, \ldots\), NSP. The body numbers of bodies \(i\) and \(j\), which are by definition those connected by the \(k\) th element, are stored in array ISP, dimensioned as \(\operatorname{ISP}(\) NSP, 2\()\). The local coordinates of points \(P_{i}\) and \(P_{j}\), the spring constant \(k\), the damping coefficient \(d\), the actuator force \(f^{(a)}\), and the undeformed spring length \(l^{0}\) are stored for each element in array SP, dimensioned as \(\operatorname{SP}(\mathrm{NSP}, 12)\). For the \(k\) th element, the entries of SP are,
\[
\begin{array}{rlrl}
\xi_{i}^{P} & \rightarrow \mathrm{SP}(\mathrm{~K}, 1) & k & \rightarrow \mathrm{SP}(\mathrm{~K}, 5) \\
\eta_{i}^{p} & \rightarrow \mathrm{SP}(\mathrm{~K}, 2) & d & \rightarrow \mathrm{SP}(\mathrm{~K}, 6) \\
\xi_{j}^{P} & \rightarrow \mathrm{SP}(\mathrm{~K}, 3) & f^{(a)} & \rightarrow \mathrm{SP}(\mathrm{~K}, 7) \\
\eta_{j}^{p} & \rightarrow \mathrm{SP}(\mathrm{~K}, 4) & l^{0} \rightarrow \mathrm{SP}(\mathrm{~K}, 8)
\end{array}
\]

The last four columns of SP, columns 9 through 12, are not used in this subroutine. They are used in subroutine SPRNG for storing the deformed length of the spring \(l\), the time rate of change in length \(l\), the spring force, and the damper force. This information will be reported to the user by subroutine REPORT.

Subroutine INSPRG is as follows:
```

    SUBROUTINE INSPRG (SP,ISP,NSP)
    DIMENSION SP(NSP,12),ISP(NSP,2)
    DO 10 K=1,NSP
        MRITE(1,200) K
    10 READ (1,* ) (ISP(K,L),L=1,2),(SP(K,L),L=1,8)
RETURN
200 FORMAT(5X,'FOR SPRING EL. NO.',I3,' ENTER BODY NOS. I AND J',/,
+ 10X,'XI-P-I,ETA-P-I,XI-P-J,ETA,P-J',',
+10X,'SPRING OONST., DAMPING ODEF., ACTUATOR FORCE, SPRING LENGTH')
END

```

Subroutine INPOIN. Same as for KAP, Sec. 5.1.1.

\subsection*{10.2.2 Dynamic Analysis}

The dynamic analysis program (DAP) performs dynamic analysis by employing the methodology of Sec. 13.3. The N second-order differential equations of motion are converted into \(2 * \mathrm{~N}\) first-order differential equations. A fourth-order Runge-Kutta algorithm is employed to solve the initial-value problem. A discussion on Runge-Kutta algorithms and other algorithms for solving initial value problems is provided in Chap. 12.

The Runge-Kutta algorithm uses the four arrays Y, YD, YS, and FTOT, each having a dimension of \(2 * \mathrm{~N}\). Vector y of Eq. 10.2 is stored in Y. At every time step, a copy of Y is saved in YS. Vector \(\dot{\mathbf{y}}\) of Eq. 10.3 is stored in YD, and FTOT stores the sum of the functions evaluated by the algorithm.

Following the process of data input, the main program calls subroutine TRANSF to transfer \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) from Y to YS. Then the main program calls subroutine RUNG4.

Subroutine TRANSF. Same as for KAP, Sec. 5.1.1.
Subroutine RUNG4. Subroutine RUNG4 evaluates \(\dot{\mathbf{y}}\) four times in every time step \(\Delta t\). This subroutine calls subroutine DIFEQN to evaluate \(\dot{\mathbf{y}}\). The time variable T is incremented from TS to TE. At the beginning of every time step, a call is made to subroutine REPORT for reporting the result.

Subroutine RUNG4 is as follows:
```

    SUBROUTINE RUNG4 (A,IA,Y,YD,YS,FTOT,MAXA,MAXIA)
    OCMMDN /MPNTR / M1 ,M2,M3,M4 ,M5,M6,M7,M8,M9,M10
    COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
    COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
    + N17,N18,N19,N20,N21,N22,N23,N24
OCMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
OOMMON /TIME / TO,TE,DT,T
DIMENSION A(MAXA),IA(MAXIA),Y(NC2),YD(NC2),YS(NC2),FTOT(NC2)
T=T0
DTH=0.5*DT
TWODT=2.0*DT
WRITE (1,200)
C......Step 1......
1 TS=T
DO 10 I=1,NC2
10 Y(I)=YS(I)
CALL DIFEQN (A,IA,MAXA,MAXIA)

```
```

    CALL REPORT (A,IA,A(N5),A(N6),A(N7),A(N10),A(N11),A(N12),
        + IA(M7),T,MAXA,MAXIA)
        IF (T.GT.TE) REIURN
        DO 11 I=1,NC2
        11 FTOT( I )=DT*YD(I )
    C.....Step 2......
T=TS+DTH
DO 20 I=1,NC2
20 Y(I)=YS(I )+DTH*YD(I)
CALL DIFEQN (A, IA,MAXA,MAXIA)
DO 21 I=1,NC2
21 FTOT( I ) =FTOT( I )+TWODT*YD( I )
C.....Step 3......
DO 30 I=1,NC2
30 Y(I )=YS(I )+DIH*YD(I )
CALL DIFEQN (A,IA,MAXA,MAXIA)
DO 31 I=1,NC2
31 FTOT(I )=FIOT( I )+TWODT*YD( I )
C....Step 4......
T=TS+DT
DO 40 I=1,NC2
40 Y(I )=YS (I )+DT*YD(I )
CALL DIFEQN (A,IA,MAXA,MAXIA)
DO 41 I=1,NC2
41 FTOT(I) =FTOT(I )+DT*YD(I )
C.....Determine new values for Q and QD
DO 50 I=1,NC2
50 YS(I )=YS(I )+FTOT(I )/6.0
GOTO 1
200 FORMAT(///,10X,'***** DYNAMIC ANALYSIS *****,,//)
END

```

Subroutine DIFEQN. This subroutine transfers the contents of \(Y\) to the arrays Q and QD prior to a call to subroutine DYNAM, by calling subroutine TRANSF. This transfer is necessary because subroutine RUNG4 modifies the contents of Y four times in every time step. Similarly, after the return from subroutine DYNAM, the contents of QD and QDD are transferred to YD.

Subroutine DIFEQN is as follows:
```

    SUBROUTINE DIFEQN (A,IA,MAXA,MAXIA)
    COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
    COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
    OCMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
    + N17,N18,N19,N20,N21,N22,N23,N24
OOMMON /RONOOL/ IR,IC,M,N,NPM,NC2
DIMENSION A(MAXA),IA(MAXIA)
C....Transfer Y to Q and QD
CALL TRANSF (A(N10),A(N20),NC2)
C.....Determine YD
CALL DYNAM (A,IA,A(N15),A(N18),MAXA,MAXIA)
C.....Transfer QD and QDD to YD
CALL TRANSF (A(N21),A(N11),NC2)
RETURN
END

```

Subroutine DYNAM. This subroutine can be considered the central station of DAP. The input to this subroutine is \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) and the output is \(\ddot{\mathbf{q}}\). This subroutine calls the subroutine FUNCT to evaluate the Jacobian matrix \(\Phi_{q}\) and vector \(\gamma\), whereupon they are stored in arrays FQ and RHS, respectively.

A call to subroutine FORCE evaluates the vector of forces \(\mathbf{g}\) which is stored in array FRC. Vectors \(\boldsymbol{g}\) and \(\boldsymbol{\gamma}\) are stored in FRC and RHS back to back, forming an \(\mathrm{N}+\mathrm{M}\) array in a form given by the right-hand side of Eq. 10.5 .

A call to subroutine MASS generates the matrix at the left in Eq. 10.5. This matrix contains the diagonal matrix \(\mathbf{M}\), along with the Jacobian matrix \(\boldsymbol{\Phi}_{\mathrm{q}}\) and its transpose \(\boldsymbol{\Phi}_{\mathrm{q}}^{r}\).

Finally, subroutine LINEAR is called to solve Eq. 10.5 for \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\). The results are transferred to arrays QDD and EL.

Subroutine DYNAM is as follows:
```

        SUBROUTINE DYNAM (A, IA, F, RHS ,MAXA,MAXIA)
        COMMON /ANALYS/ JACOB,IFNCT
        CCMMON /CONST / NRMAX,FEPS,EPSLU
        COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
        COMMDN /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
        COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
        + N17,N18,N19,N20,N21,N22,N23,N24
        COMMON /ROMODL/ IR,IC,M,N,NPM,NC2
        OOMMON /TIME / TO,TE,DT,T
        DIMENSION A(MAXA),IA(MAXIA),F(M),RHS(M)
    C
C....Calculate sine and cosine of rotational coordinates
CALL TRIG (A(N6),A(N10),NB)
C.....Evaluate right-hand-side of acceleration equations.
JACOB=1
IFNCT=3
CALL FUNCT (A,IA,MAXA,MAXIA,A(N10),A(N11),A(N14),A(N15),JACOB)
DO 20 I=1,M
20 RHS(I)=F(I)
C.....Evaluate forces
CALL FORCE (A, IA,MAXA,MAXIA,A(N17),N)
C.....Evaluate mass matrix, Jacobian, Jacobian transpose
CALL MASS (A(N6),A(N14),A(N19),NB,N,M,NYM)
C....Solve for accelerations and Lagrange multipliers
CALL LINEAR (A(N19),A(N17),A(N16),IA(M10),NPM,1,EPSLU)
C....Transfer accs. and Lag. mults. to QDD and EL
CALL TRANSF (A(N12),A(N17),NPM)
RETURN
END

```

Subroutine TRIG. Same as in KAP, Sec. 5.1.2 except that the dimension of RB must be changed to \(\mathrm{RB}(\mathrm{NB}, 7)\).

Subroutine MASS is as follows:
```

    SUBROUTINE MASS (RB,FQ,EM,NB,N,M,NPM)
    DIMENSION RB(NB,7),FQ(M,N),EM(NPM,NPM)
    C.....Initialize mass matrix
DO 10 I= 1,NPM
DO 10 J=1,NPM
10 EM(I, J )=0.0
C.....Add mass and moment of inertia to diagonal
DO 20 I=1,NB
J=3*(I-1)+1
EM(J ,J )= RB(I,3)
EM(J+1,J+1)= RB(I,3)
EM(J+2,J+2)= RB(I,4)

```
```

C....Include Jacobian and Jacobian transpose
DO 30 I=1,M
II =N+I
DO 30 J=1,N
EM(II,J )= FQ(I,J)
EM(J ,II )= FQ(I,J)
RETURN
END

```

Subroutines LINEAR and LU. See Sec. 3.3.5.

\subsection*{10.2.3 Function Evaluation}

\section*{- Subroutine FUNCT: Same as in KAP, Sec. 5.2.3, but the call to subroutine DRVR is deleted.}
- Subroutine RVLT: Same as in KAP, Sec. 5.2.3.
- Subroutine TRAN: Same as in KAP, Sec. 5.2.3.
- Subroutine SMPL: Same as in KAP, Sec. 5.2.3.

Note that in DAP the control flag IFNCT is never set to 1 or 2 (in contrast to the case for KAP ). Therefore, the constraint equations and the right side of the velocity equations are never evaluated in the three subroutines just named. Only vector \(\gamma\) is evaluated when IFNCT is set to 3 in subroutine DYNAM. Several methods are discussed in Chap. 13 for controlling the accumulation of numerical errors during the numerical integration process. If DAP is modified to incorporate such a technique as the constraint violation stabilization method (refer to Chap. 13), then IFNCT \(=1\) may be used in order to provide the violation in the constraints for the algorithm.

\subsection*{10.2.4 Force Evaluation}

Subroutine FORCE. The vector of forces is evaluated by a call to subroutine FORCE. All of the external and internal forces and moments acting on the bodies are stored in array FRC, dimensioned as \(\operatorname{FRC}(3, \mathrm{NB})\). For example, the forces acting on body \(i\) are stored as follows:
\[
\begin{aligned}
f_{(x)_{i}} & \rightarrow \operatorname{FRC}(1, \mathrm{I}) \\
f_{(y)_{i}} & \rightarrow \operatorname{FRC}(2, \mathrm{I}) \\
n_{i} & \rightarrow \operatorname{FRC}(3, \mathrm{I})
\end{aligned}
\]

This subroutine calls subroutines BODYF and SPRNG to evaluate these forces.
Subroutine FORCE is as follows:
```

    SUBROUTINE FORCE (A,IA,MAXA,MAXIA,FRC,N)
    COMMON /MPNTR / M1 ,M2 ,M3 ,M4,M5,M6,M7,M8,M9,M10
    COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
    COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
    + N17,N18,N19,N20,N21,N22,N23,N24
DIMENSION A(MAXA), IA(MAXIA), FRC(N)
CALL BODYF (A(N6), FRC,NB)
IF (NSP.GT.0) CALL SPRNG (A(N5),A(N6),A(N10),A(N11),A(N17),
+ RETURN
END

```

Subroutine BODYF. This subroutine transfers the constant external forces and moments, including the gravitational force, from RB to FRC, for all of the bodies.

Subroutine BODYF is as follows:
```

    SUBROUTINE BODYF (RB,FRC,NB)
    DIMENSION RB(NB,7),FRC(3,NB)
C.....Add constant forces and weights to FRC
DO 10 I=1,NB
FRC(1,I)= RB(I,5)
FRC(2,I)= RB(I,6)
10
FRC(3,I)= RB(I,7)
RETURN
END

```

Subroutine SPRNG. This subroutine computes the forces of spring, damper, and actuator elements between pairs of bodies. For each element, the spring constant \(k\), damping coefficient \(d\), actuator force \(f^{(a)}\), and undeformed spring length \(l^{0}\) are obtained from array SP. The body numbers connected by each element are found in array ISP. The element forces are calculated from the equations of Sec. 9.2.3, 9.2.4, and 9.2.5. The element forces and moments acting on each body are entered in array FRC.

This subroutine saves the values of \(l, l, f^{(s)}\), and \(f^{(d)}\) for each element in the last four entries of array SP. This information is reported by subroutine REPORT at each time step.

Subroutine SPRNG is as follows:
```

        SUBROUTINE SPRNG(SP,RB,Q,QD,FRC, I SP,NB,NSP)
        DIMENSION SP(NSP,12),ISP(NSP,2),RB(NB,7),FRC(3,NB),Q(3,NB),
        + QD(3,NB)
        DO 10 K=1,NSP
        I=ISP(K,1)
        J=ISP(K,2)
        XPIMXI=SP(K,1)*RB(I , 2)-SP(K, 2)*RB(I , 1)
        YPIMYI=SP(K,1)*RB(I ,1)+SP(K,2)*RB(I, 2)
        XPJMXJ=SP(K, 3)*RB(J, 2) -SP(K, 4)*RB(J,1)
        YP JMYJ =SP(K, 3)*RB(J , 1) +SP(K, 4) *RB(J , 2)
    C....Current spring length and change of length
ELX =Q(1,J )+XPJMXJ -Q(1,I )-XP IMXI
ELY =Q(2,J )+YPJMYJ -Q(2,I ) -YP IMYI
EL = SQRT(ELX**2 +ELY**2)
DELEL = EL-SP(K,8)
IF(ABS(EL).LT.1.E-10) EL=1.E-10
C......Unit vector
UX =ELX/EL
UY =ELY/EL
C....Spring velocity
ELXD=QD(1, J ) -YP JMYJ *QD( 3, J )-QD(1, I ) +YPIMYI *QD( 3, I )
ELYD=QD(2,J )+XP JMXJ *QD( 3, J ) -QD( 2,I ) -XPIMXI *QD( 3,I)
ELD = (ELX*ELXD + ELY*ELYD)/EL
C.....Element forces
FS=SP(K,5)*DELEL
FD=SP(K,6)*ELD
FF=FS+FD+SP(K,7)
FX= UX*FF
FY= UY*FF
C.....Save element length, vel., spr. force, damp. force for output
SP(K, 9)=EL
SP(K,10)=ELD
SP(K,11)=FS
SP(K, 12)=FD

```
```

C.....Add forces to the vector of forces
FRC(1,I )=FRC(1,I )+FX
FRC( 2,I)=FRC( 2,I )+FY
FRC( 3,I)=FRC( 3,I) - YP IMYI *FX+XP IMXI *FY
FRC(1,J )=FRC(1,J )-FX
FRC( 2, J )=FRC( 2, J ) -FY
10 FRC( 3 , J )=FRC( 3, J )+YP JMYJ * FX-XP JMXJ * FY
RETURN
END

```

\subsection*{10.2.5 Reporting}

For reporting the result of the dynamic analysis at the beginning of each time step, subroutine RUNG4 calls subroutine REPORT.

Subroutine REPORT. This subroutine is similar to subroutine REPORT in KAP. It reports the coordinates, velocities, and accelerations of the bodies and points of interest. In addition, if there are any spring, damper, or actuator elements in the system, the contents of the last four columns of the SP array are reported. This subroutine calls subroutine REACT to calculate and report the reaction forces at the kinematic joints.

Subroutine REPORT is as follows:

SUBROUTINE REPORT (A,IA,SP,RB, PI ,Q, OD, QDD, IPI ,T,MAXA,MAXIA)
COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
OOMMON /RONCOL/ IR,IC,M,N,NPM,NC2
DIMENS ION SP(NSP, 12), Q(3,NB) , \(\mathrm{QD}(3, \mathrm{NB}), \mathrm{QDD}(3, \mathrm{NB}), \mathrm{PI}(\mathrm{NP}, 2)\),
\(+\quad\) IPI (NP) , RB(NB 7 ) , A(MAXA), IA(MAXIA)
WRITE \((1,200)\) T
DO \(10 \mathrm{I}=1, \mathrm{NB}\)
\(10 \underset{\operatorname{HRITE}(1,210) I,(Q(J, I), J=1,3),(\operatorname{QD}(J, I), J=1,3),(\operatorname{QDD}(J, I), J=1,3)}{ }\)
IF (NP.EQ.0) GO TO 30
WRITE \((1,220)\)
DO \(20 \mathrm{~K}=1\), NP
\(\mathrm{I}=\mathrm{IPI}(\mathrm{K})\)
\(X P M X=P I(K, 1) * R B(I, 2)-P I(K, 2) * R B(I, 1)\) YPMY \(=\mathrm{PI}(\mathrm{K}, 1) * \mathrm{RB}(\mathrm{I}, 1)+\mathrm{PI}(\mathrm{K}, 2) * \mathrm{RB}(\mathrm{I}, 2)\)
\(\mathrm{XP}=\mathrm{Q}(1, \mathrm{I})+\mathrm{XPMX}\)
\(\mathrm{YP}=\mathrm{Q}(2, \mathrm{I})+\mathrm{YPMY}\)
\(X D P=Q D(1, I)-Y P M Y * Q D(3, I)\)
\(\mathrm{YDP}=\mathrm{QD}(2, \mathrm{I})+\mathrm{XPMX} * \mathrm{QD}(3, \mathrm{I})\)
\(\mathrm{XDDP}=\operatorname{ODD}(1, \mathrm{I})-\mathrm{XPMX} * \operatorname{QD}(3, \mathrm{I}) * * 2-\mathrm{YPMY} * \operatorname{QDD}(3, \mathrm{I})\)
\(\operatorname{YDDP}=Q D D(2, I)-Y P M Y * Q D(3, I) * * 2+X P M X * Q D D(3, I)\)
20 VRITE (1, 260) K, XP , YP ,XDP , YDP , XDDP , YDDP
30 IF(NSP.EQ.0) GO TO 50
WRITE \((1,240)\)
DO \(40 \mathrm{~K}=1\), NSP
40 WRITE \((1,250) K, \operatorname{SP}(K, 9), \operatorname{SP}(K, 10), \operatorname{SP}(K, 11), \operatorname{SP}(K, 12)\)
50 IF (M.GT.0) CALL REACT(A, IA,MAXA,MAXIA)
RETURN

\(+\quad\) BODY',5X,'X',7X,'Y',5X,'PHI',6X, 'XD' \(6 \mathrm{X},{ }^{\prime} \mathrm{YD}^{\prime}, 4 \mathrm{X},{ }^{\prime}\) PHID',6X,
\(+\quad\) 'XDD',6X,'YDD',4X,'PHIDD')
210 FORMAT (I3, 6F8.3,3F9.3)
220 FORMAT(' POINTS OF INTEREST', /,' NO.', 6X, 'X',7X, 'Y',6X,
\(+\quad\) ' XD ', 6X, 'YD',6X, 'XDD',6X, 'YDD')
240 FORMAT (', TRANSLATIONAL SPRING-DAMPER-ACIUATOR', \(/\),
\(+\quad, \quad\) NO. LENGTH \(d(E L) / d t ', 6 X, ' f(s) ', 6 X, ' f(d) \prime)\)
250 FORMAT ( 13 , 4F10.3)
260 FORMAT ( 3 , 4F8.3,2F9.3)
END

Subroutines REACT and RFORCE. Subroutine REACT calls subroutine RFORCE for computing the product \(\boldsymbol{\Phi}_{q}^{T} \boldsymbol{\lambda}\). This product is computed separately for each kinematic joint. Some of the variables in the argument list of subroutine RFORCE are dependent on the type of joint. For example, for a revolute joint, NEQ, which represents the number of equations, is set to 2 , and NBJ, which represents the number of bodies associated with the joint, is also set to 2 .

Subroutine RFORCE reports the reaction forces acting on each body that are due to that body's kinematic joints. A reaction force is reported in terms of its \(x\) and \(y\) components and its moment with respect to the centroid of the body. The reported reaction forces are labeled with REV. for a revolute joint, TRA. for a translational joint, and SMP. for a simple constraint.

Subroutines REACT and RFORCE are as follows:
```

        SUBROUTINE REACT(A, IA,MAXA,MAXIA)
        OCMMON /MPNTR / M1 ,M2,M3,M4,M5,M6,M7 ,M8,M9,M10
        OCMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
        COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,
        + N15,N16,N17,N18,N19,N20,N21,N22,N23,N24
        DIMENSION A(MAXA),IA(MAXIA)
        JR=0
        WRITE(1,200)
        IF(NR.GT.0) CALL RFORCE(A(N13),A(N14),IA(M1),NR,2,2,JR,'REV.')
        IF(NT.GT.0) CALL RFORCE(A(N13),A(N14),IA(M2),NT,2,2,JR,'TRA.')
        JR=JR+NG3
        IF(NS.GT.0) CALL RFORCE(A(N13),A(N14),IA(M4),NS,1,1,JR,'SMP.')
        RETURN
    200 FORMAT(' REACTION FORCES',/,2X,'JOINT NO. I ',5X,'FX-I',6X,
+ 'FY-I',7X,'N-I',3X,'J ',5X,'FX-J',6X,'FY-J',7X,'N-J'')
END
SUBROUTINE RFORCE(EL, FQ, I J ,NJ ,NEQ,NBJ , JR,NAME)
OOMMON /RONOOL/ IR,IC,M,N,NPM,NC2
CHARACTER*4 NAME
DIMENSION FQ(M,N),EL(M),IJ(NJ,NBJ),F(3,2)
DO 20 K=1,NJ
DO 10 L=1,NBJ
I =IJ(K,L)
IC=(I-1)*3
DO 10 J=1,3
F(J,L)=0.0
DO 10 MM=1,NEQ
F(J,L)=F(J,L ) -FQ(JR+MM, IC+J )*EL(JR+MM)
WRITE(1, 200) NAME,K,(IJ (K,L),(F(J,L), J=1, 3),L=1,NBJ)
JR=JR+NEQ
RETURN
200 FORMAT( 2X,A4,I3,I4,3F10.3,I4,3F10.3)
END

```

\subsection*{10.2.6 Static Analysis}

When the number of constraint equations is equal to the number of coordinates, the main program calls subroutine STATIC to perform static analysis. This subroutine calls subroutine FUNCT to evaluate the Jacobian matrix \(\boldsymbol{\Phi}_{\mathbf{q}}\), calls subroutine FORCE to evaluate the vector of forces \(\mathbf{g}\), and then calls subroutine JACTRN to transpose the Jacobian matrix. A call to subroutine LINEAR solves for the Lagrange multipliers according to Eq. 9.57. The constraint reaction forces are then reported by a call to subroutine REPORTS.

Subroutines STATIC, JACTRN, and REPORTS are as follows:
```

    SUBROUTINE STATIC (A,IA,MAXA,MAXIA)
    OOMMON /ANALYS/ JACOB,IFNCT
    COMMON /OONST / NRMAX,FEPS,EPSLU
    COMMON /MPNTR / M1 ,M2,M3,M4,M5,M6,M7 ,M8 ,M9 ,M10
    COMMDN /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
    COMMDN /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
    + N17,N18,N19,N20,N21,N22,N23,N24
COMMON /RONOOL/ IR,IC,M,N,NPM,NC2
DIMENSION A(MAXA),IA(MAXIA)
C....Calculate sine and cosine of rotational coordinates
CALL TRIG (A(N6),A(N10),NB)
C.....Evaluate Jacobian matrix
JACOB=1
I FNCT=0
CALL FUNCT (A,IA,MAXA,MAXIA,A(N10),A(N11),A(N14),A(N15),JACOB)
C.....Evaluate forces
CALL FORCE (A, IA,MAXA,MAXIA,A(N17),N)
C.....Jacobian transpose
CALL JACTRN(A(N14),A(N19),M)
C.....Solve for Lagrange multipliers
CALL LINEAR (A(N19),A(N17),A(N16),IA(M10),M,1,EPSLU)
C.....Transfer Lagrange multipliers to EL
CALL TRANSF (A(N13),A(N17),M)
C.....Output the result
CALL REPORTS (A,IA,A(N10),MAXA,MAXIA)
STOP
END
SUBROUTINE JACTRN (FQ,CM,M)
DIMENSION FQ(M,M),CM(M,M)
DO 10 I=1,M
DO 10 J=1,M
10 CM(I, J)=FQ(J , I )
RETURN
END
SUBROUTINE REPORTS (A,IA,Q,MAXA,MAXIA)
COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
OOMMON /RONCOL/ IR,IC,M,N,NPM,NC2
DIMENSION Q(3,NB),A(MAXA),IA(MAXIA)
WRITE (1,200)
DO 10 I=1,NB
10 WRITE(1,210)I,(Q(J , I ) , J=1,3)
CALL REACT(A,IA,MAXA,MAXIA)
REIURN
210 FORMAT(I3,3F8.3)
200 FORMAT(///,10X,'***** STATIC ANALYSIS *****',//,
    + ' BODY',5X,'X',7X,'Y',5X,'PHI')
END

```

\subsection*{10.2.7 Input Prompts}

A list of all the prompts given by the program DAP is given here. The prompts are labeled for easy reference, from (a) through (i). In the examples that follow, each prompt is referred to by its corresponding label. The parameter k in the prompts is problem-dependent.

The prompts given are as follows:
(a) ENTER NB, NR, NT, NG, NS, NSP, NP
(b) FOR BODY \(k\) ENTER INITIAL COND. ON \(X, Y\), PHI

INITIAL CONDITIONS ON XD, YD, PHID
MASS, MOMENT OF INERTIA
CONSTANT FORCE-MOMENT FX, FY, N
(c) FOR REV. JOINT NO. \(k\) ENTER BODY NOS. I AND J

XI-P-I, ETA-P-I, XI-P-J, ETA-P-J
(d) FOR TRAN. JOINT NO. k ENTER BODY NOS. I AND J

XI-P-I, ETA-P-I, XI-Q-I, ETA-Q-I, XI-P-J, ETA-P-J
(e) ENTER BODY NO. FOR NO. k GROUNDED BODY
(f) FOR SIMPLE CONSTRAINT NO. k ENTER BODY NO.

AND 1, 2, OR 3 FOR \(\mathrm{X}, \mathrm{Y}\), OR PHI CONSTRAINT DIRECTION
(g) FOR SPRING EL. NO. k ENTER BODY NOS. I AND J

XI-P-I, ETA-P-I, XI-P-J, ETA-P-J
SPRING CONST., DAMPING COEF., ACTUATOR FORCE, UNDEFORMED SPRING LENGTH
(h) FOR POINT OF INTEREST NO. k ENTER BODY NO.

XI-P AND ETA-P COORDINATES
(i) ENTER STARTING TIME, FINAL TIME, AND TIME INCREMENT

\subsection*{10.3 SIMIPLE EXAMPLES}

In Secs. 10.3.1 through 10.3 .3 several simple examples are presented. The steps needed to set up a model for each mechanical system are explained. Input data for the dynamic analysis program (DAP) are listed for each example. Similar steps can be followed to analyze many other mechanical systems by means of this program.

\subsection*{10.3.1 Four-Bar Linkage}

The four-bar linkage of Sec. 5.2 .1 is considered here. The mechanism is released from an initial position, where all of the initial velocities are zero, and falls under its own weight. The mass and moment of inertia for the moving bodies are:
\[
\begin{array}{ccc}
m_{2}=1, & m_{3}=2.25, & m_{4}=2 \\
\mu_{2}=0.3, & \mu_{3}=2, & \mu_{4}=1.35
\end{array}
\]

For body 1 , the nonmoving body, any arbitrary value can be assigned for the mass and moment of inertia.

When DAP is executed, the following sequence of prompts is given by the program; each prompt must be followed by input from the user to describe the model:
\begin{tabular}{|c|c|c|}
\hline Prompt & \(\underline{\text { k }}\) & Input \\
\hline (a) & & 4, 4, 0, 1, 0, 0, 1 \\
\hline (b) & 1 & \(0,0,0,0,0,0,0,0,0,0,0\) \\
\hline (b) & 2 & .5, .866, 1.047, 0, 0, 0, 1, .3, 0, 0, 0 \\
\hline (b) & 3 & \(2.824,2.553, .423,0,0,0,2.25,2,0,0,0\) \\
\hline (b) & 4 & \(3.574,1.687,1.004,0,0,0,2,1.35,0,0,0\) \\
\hline (c) & 1 & 1, 2, 0, -1, 0 \\
\hline (c) & 2 & 2, 3, 1, 0, -2, 0 \\
\hline (c) & 3 & 3, 4, 2, 0, 2, 0 \\
\hline (c) & 4 & \(4,1,-2,0,2.5,0\) \\
\hline (e) & 1 & 1 \\
\hline (b) & 1 & 3, 0.5, 1.5 \\
\hline (i) & & \(0.0,0.25,0.025\) \\
\hline
\end{tabular}

The initial conditions for the coordinates satisfy the constraint equations as stated by Eq. 10.6. This is assured, since these coordinates are taken from the output of KAP at \(t=0\) for the same linkage, as given in Sec. 5.2.1. The initial conditions for the velocities do not violate the velocity equations, since \(\dot{\mathbf{q}}=0\) automatically satisfies Eq. 10.7.

The output of DAP for the first time step is as follows:
```

***** DYNAMIC ANALYSIS *****

```


The output gives the coordinates, velocity, and acceleration of each body's centroid and of each point of interest. The reaction forces corresponding to each joint are also printed; these are shown in Fig. 10.2.

\subsection*{10.3.2 Horizontal Platform}

The horizontal platform shown in Fig. 10.3(a) is connected to ground by four slender legs through eight revolute joints. The mass and moment of inertia of the platform


Figure 10.2 Reaction forces acting on each body at \(t=0\).


Figure 10.3 (a) A horizontal platform and (b) its corresponding planar model.
(about the \(\zeta\)-axis out of the plane of motion) are 1.5 and 0.2 , respectively. The mass and moment of inertia of each leg are 0.3 and 0.05 , respectively. There are two parallel spring-damper elements in the system, as shown. For each element, \(k=350, l^{0}=0.6\), and \(d=25\). When the angle of each leg with the horizontal is \(\alpha=110^{\circ}\), the platform is released with an upward velocity of \(\dot{y}=0.12\).

A planar model for this system is shown in Fig. 10.3(b). In this model, the masses and moments of inertia are:
\[
\begin{aligned}
m_{2} & =m_{4}=0.6, & & m_{3}=1.5 \\
\mu_{2} & =\mu_{4}=0.1, & & \mu_{3}=0.2
\end{aligned}
\]

For the spring-damper element:
\[
k=700, \quad l^{0}=0.6, \quad d=50
\]

Since the initial conditions for \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) are not available, a simulation on KAP for \(t=0\) can be performed initially. In this simulation the known initial conditions are the vertical coordinate and velocity of the platform. From \(\alpha=110^{\circ}\), it is found that \(y_{3}=0.46895\), and it is given that \(\dot{y}_{3}=0.12\). Therefore, a driver constraint can be specified on \(y_{3}\). The input to KAP is
\begin{tabular}{|c|c|c|}
\hline Prompt (KAP) & \(\underline{k}\) & Input \\
\hline (a) & & 4, 4, 0, 1, 0, 1, 0 \\
\hline (b) & 1 & 0, 0, 0 \\
\hline (b) & 2 & -0.35, 0.2, 0.3 \\
\hline (b) & 3 & -0.1, 0.46895, 0 \\
\hline (b) & 4 & \(0.15,0.2,0.3\) \\
\hline (c) & 1 & 1, 2, -0.25, 0, 0, -0.25 \\
\hline (c) & 2 & 2, 3, 0, 0.25, -0.25, 0 \\
\hline (c) & 3 & 3, 4, 0.25, 0, 0, 0.25 \\
\hline (c) & 4 & \(4,1,0,-0.25,0.25,0\) \\
\hline (e) & 1 & 1 \\
\hline (g) & 1 & 3, 2, 0.46895, 0.12,0 \\
\hline (i) & & 0.0, 0.0, 0.1 \\
\hline
\end{tabular}

The output from the kinematic analysis at \(t=0\) is
***** KINEMATIC ANALYSIS *****
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|}
\hline \multicolumn{10}{|l|}{\(\mathrm{ME}=\quad .0000\)} \\
\hline BODY & X & Y & PHI & XD & YD & PHID & XDD & YDD & PHIDD \\
\hline 1 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 2 & -. 335 & . 235 & . 349 & . 165 & . 060 & -. 702 & . 360 & . 000 & -1.353 \\
\hline 3 & -. 171 & . 470 & . 000 & . 330 & . 120 & . 000 & . 720 & . 000 & . 000 \\
\hline 4 & . 165 & . 235 & . 349 & . 165 & . 060 & -. 702 & . 360 & . 000 & -1.353 \\
\hline
\end{tabular}

The initial conditions for \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) are taken from this output to generate the input to DAP:
\begin{tabular}{|c|c|c|}
\hline Prompt (DAP) & k & Input \\
\hline (a) & & 4, 4, 0, 1, 0, 1, 0 \\
\hline (b) & 1 & \(0,0,0,0,0,0,0,0,0,0,0\) \\
\hline (b) & 2 & -.335, .235, .349, .165, .060, -.702, .6, .1, 0, 0, 0 \\
\hline (b) & 3 & --.171, .46895,.000, .330,.120,.000, 1.5, .2, 0, 0, 0 \\
\hline (b) & 4 & .165, .235, .349, .165, .060, -.702, .6, .1, 0, 0, 0 \\
\hline (c) & 1 & 1, 2, -. 25, 0, 0, -. 25 \\
\hline (c) & 2 & 2, 3, 0, .25, -. 25,0 \\
\hline (c) & 3 & 3, 4, .25, 0, 0, . 25 \\
\hline (c) & 4 & 4, 1, 0, -. \(25, .25,0\) \\
\hline (e) & 1 & 1 \\
\hline (g) & 1 & \(1,3,0.25,0,-0.25,0,700,50,0,0.6\) \\
\hline (i) & & 0, 3, 0.01 \\
\hline
\end{tabular}

The result of this simulation for \(t=0\) and \(t=3 \mathrm{~s}\) is as follows:
***** DYNAMIC ANALYSIS *****

TIME = . 0000


TIME \(=3.0000\)
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|}
\hline BODY & X & Y & Y & PHI & XD & YD & PHID & XOD & YDD & PHIDD \\
\hline 1 & . 000 & . 000 & & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 2 & -. 173 & . 238 & & -. 311 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 3 & . 153 & . 476 & & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 4 & . 327 & . 238 & & -. 311 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline \multicolumn{11}{|l|}{TRANSLATIONAL SPRING-DAMPER-ACTUATOR} \\
\hline NO. & LENGTH & & f (s) & & f (d) & & & & & \\
\hline 1 & . 589 & & . 000 & & -7.805 & . 000 & & & & \\
\hline \multicolumn{11}{|l|}{REACTION FORCES} \\
\hline JOIN & NT NO. I & & FX-I & & FY-I & N-I & J & FX- J & FY- J & N-J \\
\hline REV & 11 & & 1.283 & & -6.933 & 1.733 & 2 & 1.283 & 6.933 & -. 225 \\
\hline REV & 22 & & 1.283 & & -1.047 & . 225 & 3 & 1.283 & 1.047 & -. 262 \\
\hline REV & 3 3 & & 3.311 & & 7.358 & 1.839 & 4 & -3.311 & -7.358 & . 225 \\
\hline REV & 44 & & 3.311 & & 13.244 & -. 225 & 1 & -3.311 & -13.244 & -3.311 \\
\hline
\end{tabular}

At \(t=3 \mathrm{~s}\), all of the velocities and accelerations are zero. This indicates that the system reaches the state of static equilibrium within 3 seconds as a result of the presence of dampers in the system. Figure 10.4 shows the plots of \(y_{3}, \dot{y}_{3}\), spring force, and damper force as a function of time.


Figure 10.4 The dynamic response for the horizontal platform. (a) Plots of \(y\) displacement and velocity of the platform, and (b) spring and damper forces versus time.

\subsection*{10.3.3 Dump Truck}

A hydraulic actuator controls the unloading process of a dump truck. For the configuration shown in Fig. \(10.5(\theta=0.38 \mathrm{rad})\), find the force that the actuator must apply be-


Figure 10.5 A dump truck in unloading configuration.
tween points \(A\) and \(D\) in order to keep the system in equilibrium. Masses and moments of inertia for the bodies are:
\[
\begin{array}{ccc}
m_{E F}=m_{A C}=0.4, & m_{C K}=m_{B E}=2, & m_{\text {load }}=100 \\
\mu_{E F}=\mu_{A C}=0.005, & \mu_{C K}=\mu_{B E}=0.7, & \mu_{\text {load }}=27
\end{array}
\]

Assume that the center of mass of each body is at its geometric center, and that the center of mass of the load is at \(G\).

First, as shown in Fig. 10.6(a), a model is set up for KAP to determine the correct initial conditions on the coordinates. This model is executed on KAP for \(t=0\), with the known value of \(\phi_{2}\), which is found from the specified \(\theta\). Since the system is not in motion, a constraint on \(\phi_{2}\) can be stated in the form of either a simple constraint or a driver constraint. The input to KAP is as follows:


Figure 10.6 A model for the dump truck for (a) kinematic analysis, (b) static analysis, and (c) dynamic analysis.
\begin{tabular}{lll} 
(b) & 2 & \(-.2, .08,-0.38\) \\
(b) & 3 & \(.6, .3,0.15\) \\
(b) & 4 & \(.6, .4,-0.2\) \\
(b) & 5 & \(-.15, .6,0.25\) \\
(b) & 6 & \(.9, .95,-0.15\) \\
(c) & 1 & \(1,2,0,0, .2,0\) \\
(c) & 2 & \(1,4,1.6, .2,1,0\) \\
(c) & 3 & \(2,3,-.2,0,-1,0\) \\
(c) & 4 & \(3,4, .3,0, .3,0\) \\
(c) & 5 & \(3,6,1,0, .8,-.4\) \\
(c) & 6 & \(4,5,-1,0,-.2,0\) \\
(c) & 7 & \(5,6, .2,0,-.8,-.4\) \\
(e) & 1 & 1 \\
(g) & 1 & \(2,3,-.38\) \\
(b) & 1 & \(3, .3,0\) \\
(i) & & \(0,0,0.1\)
\end{tabular}

In this model, there are six bodies (one grounded), and seven revolute joints, and point \(D\) is given as a special point of interest on body 3 (or body 4). The output from this simulation is:
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|}
\hline \multicolumn{10}{|l|}{TIME \(=\quad .0000\)} \\
\hline BODY & X & Y & PHI & XD & YD & PHID & XDD & YDD & PHIDD \\
\hline 1 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 2 & -. 186 & . 074 & -. 380 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 3 & . 618 & . 296 & . 148 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 4 & . 620 & . 401 & -. 202 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 5 & -. 166 & . 652 & . 256 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline 6 & . 882 & . 968 & -. 163 & . 000 & . 000 & . 000 & . 000 & . 000 & . 000 \\
\hline \multicolumn{10}{|l|}{POINTS OF INTEREST} \\
\hline NO. & X & Y & XD & YD & XDD & YD & & & \\
\hline 1 & . 914 & . 341 & . 000 & . 000 & . 000 & . 00 & & & \\
\hline
\end{tabular}

According to Sec. 9.6 , in order to determine the force required by the actuator for keeping the system in equilibrium, the actuator may be replaced by a revolute-revolute joint. The reaction forces exerted at \(A\) and \(D\) on this revolute-revolute joint give the required actuator force. However, since the formulation for the revolute-revolute joint is not included in the present version of DAP, the actuator can be replaced with a fictitious body and two revolute joints, one at \(A\) and one at \(D\). This model is shown in Fig. 10.6(b), and the input to DAP is as follows:

Prompt (DAP) \(\underline{\mathbf{k}}\) Input

\(7,9,0,1,0,0,0\)
(b)

1
\(0,0,0,0,0,0,0,0,0,0,0\)
\begin{tabular}{lll} 
(b) & 2 & \(-.1857, .0742,-0.380,0,0,0, .4, .005,0,0,0\) \\
(b) & 3 & \(.6175, .2962,0.148,0,0,0,2, .7,0,0,0\) \\
(b) & 4 & \(.6204, .4008,-0.202,0,0,0,2, .7,0,0,0\) \\
(b) & 5 & \(-.1658, .6522,0.256,0,0,0, .4, .005,0,0,0\) \\
(b) & 6 & \(.8818, .9682,-0.163,0,0,0,100,27 ., 0,0,0\) \\
(b) & 7 & \(.4571, .1703,0.357,0,0,0, .001, .001,0,0,0\) \\
(c) & 1 & \(1,2,0,0, .2,0\) \\
(c) & 2 & \(1,4,1.6, .2,1,0\) \\
(c) & 3 & \(2,3,-.2,0,-1,0\) \\
(c) & 4 & \(3,4, .3,0, .3,0\) \\
(c) & 5 & \(3,6,1,0, .8,-.4\) \\
(c) & 6 & \(4,5,-1,0,-.2,0\) \\
(c) & 7 & \(5,6, .2,0,-.8,-.4\) \\
(c) & 8 & \(1,7,0,0,-.4878,0\) \\
(c) & 9 & \(3,7, .3,0, .4878,0\) \\
(e) & 1 & 1 \\
(i) & & \(0,0,0.1\)
\end{tabular}

In this model, there are seven bodies and nine revolute joints. The coordinates of point \(D\) from the first simulation, with the known coordinates of \(A\), are used to calculate the correct coordinates of body 7 . Since this is a 0 -degree of freedom system, DAP performs static analysis and the output is as follows:
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|}
\hline & & *** & * STATIC & ANALYSIS & ***** & & & & \\
\hline BODY & X & & Y & PHI & & & & & \\
\hline 1 & . 000 & & . 000 & . 000 & & & & & \\
\hline 2 & -. 186 & & . 074 & . 380 & & & & & \\
\hline 3 & . 618 & & . 296 & . 148 & & & & & \\
\hline 4 & . 620 & & . 401 & . 202 & & & & & \\
\hline 5 & -. 166 & & . 652 & . 256 & & & & & \\
\hline 6 & . 882 & & . 968 & . 163 & & & & & \\
\hline 7 & . 457 & & . 170 & . 357 & & & & & \\
\hline \multicolumn{10}{|l|}{REACTION FORCES} \\
\hline JOINT & T NO. & I & FX-I & FY-I & N -I & J & FX-J & FY- J & N - J \\
\hline REV. & 1 & 1 & -835.818 & 331.874 & . 000 & 2 & 835.818 & -331.874 & . 365 \\
\hline REV. & 2 & 1 & 4505.031 & 8.582 & -887.275 & 4 & -4505.031 & -8.582 & -912.248 \\
\hline REV. & 3 & 2 & -835.817 & 335.798 & -. 364 & 3 & 835.817 . & -335.798 & 455.377 \\
\hline REV. & 4 & 3 & -5565.859 & -307.829 & 154.884 & 4 & 5565.859 & 307.829 & 425.473 \\
\hline REV. & 5 & 3 & 1060.831 & -705.297 & -854.017 & 6 & -1060.831 & 705.297 & -45.452 \\
\hline REV. & 6 & 4 & -1060.831 & -279.627 & 486.775 & 5 & 1060.831 & 279.627 & -. 380 \\
\hline REV. & 7 & 5 & -1060.831 & -275.703 & . 380 & 6 & 1060.831 & 275.703 & 45.452 \\
\hline REV. & 8 & 1 & -3669.213 & -1368.553 & . 000 & 7 & 3669.213 & 1368.553 & -. 002 \\
\hline REV. & 9 & 3 & 3669.213 & 1368.544 & 243.756 & 7 & -3669.213 & -1368.544 & . 002 \\
\hline
\end{tabular}

This output yields the reaction forces acting at \(A\) and \(D\) on the fictitious body 7. These force components indicate a compressive force on body 7 , having a magnitude
\[
f=\left(3.669 .2^{2}+1368.5^{2}\right)^{1 / 2}=3916
\]

If body 7 is replaced by an actuator, the force of the actuator must be \(f^{(a)}=-3916\) to keep the system in equilibrium. The negative sign is assigned to the actuator force, ac-
cording to the sign convention of Sec. 9.2.3. Note that the force that the actuator must apply on the system is in the direction opposite that of the force that the system applies on the actuator (or the fictitious body 7).

For dynamic analysis, another model can be set up by replacing the fictitious body with an actuator, as shown in Fig. 10.6(c). If the force of the actuator is specified as \(f^{(a)}=\) -3916, then the system remains in equilibrium. For simulating the unloading process, the actuator force is increased slightly to -3955 and the input to DAP is as follows:
\begin{tabular}{|c|c|c|}
\hline Prompt (DAP) & k & Input \\
\hline (a) & & 6, 7, 0, 1, 0, 1, 0 \\
\hline (b) & 1 & \(0,0,0,0,0,0,0,0,0,0,0\) \\
\hline (b) & 2 & -.1857, .0742, -0.380, 0, 0, 0, 4, .005, 0, 0, 0 \\
\hline (b) & 3 & .6175, .2962, 0.148, 0, 0, 0, 2, .7, 0, 0, 0 \\
\hline (b) & 4 & .6204, 4008, -0.202, 0, 0, 0, 2, .7, 0, 0, 0 \\
\hline (b) & 5 & -.1658, .6522, 0.256, 0, 0, 0, , 4, .005, 0, 0, 0 \\
\hline (b) & 6 & .8818, .9682, \(-0.163,0,0,0,100,27,0,0,0\) \\
\hline (c) & 1 & 1, 2, 0, 0, .2, 0 \\
\hline (c) & 2 & 1, 4, 1.6, 2, 1, 0 \\
\hline (c) & 3 & \(2,3,-2,0,-1,0\) \\
\hline (c) & 4 & 3, 4, .3, 0, .3, 0 \\
\hline (c) & 5 & 3, 6, 1, 0, .8, -. 4 \\
\hline (c) & 6 & \(4,5,-1,0,-2,0\) \\
\hline (c) & 7 & 5, 6, .2, 0, -.8,-. 4 \\
\hline (e) & 1 & 1 \\
\hline (g) & 1 & 1, 4, 0, 0, .3, 0, 0, 500, -3955., 0 \\
\hline (i) & & 0, 1.5, 0.05 \\
\hline
\end{tabular}

The result of this simulation is shown in Fig. 10.7 for several intermediate stages.
In the dynamic simulation of this system, or other systems, a problem may arise. In certain kinematic configurations, the Jacobian matrix \(\boldsymbol{\Phi}_{\mathrm{q}}\) may lose rank - one or more of the constraint equations may become redundant. In this case, the matrix at the left in Eq. 10.5 becomes singular. Therefore, the present version of subroutine LINEAR cannot solve Eq. 10.5 for the accelerations and Lagrange multipliers. For the dump truck model, this situation occurs when the linkage system is completely stretched open or when it is folded. In these two cases, some of the bodies align in a way that causes kinematic redundancy in the constraint equations.

\subsection*{10.4 TIME STEP SELECTION}

One of the most crucial problems in using a constant-step numerical integration algorithm, such as the Runge-Kutta algorithm, is the selection of a proper step size. A large step size may cause erroneous results. A step size too small may yield accurate results while increasing the computation time unreasonably. Therefore, it is important to choose a reasonably small step size to obtain accurate results without unnecessarily in-


Figure 10.7 Unloading process of the dump truck.
creasing the computation time. A thorough discussion on the subject of time-step selection is outside the scope of this book. The interested reader may refer to other textbooks on the subject of numerical solution of differential equations. In this section only two highly simple examples are presented to familiarize the reader with this important point. Since these examples deal with vibratory motion, which has not been covered in this book, the reader may refer to any textbook on the subject of mechanical vibration for more detail.

The simplest form of vibratory motion is a simple harmonic motion which is described by the differential equation
\[
\begin{equation*}
\ddot{y}+p^{2} y=0 \tag{10.8}
\end{equation*}
\]
where \(p\) is a real number. The frequency of oscillation of this single degree-of-freedom system is
\[
\begin{equation*}
f=\frac{p}{2 \pi} \tag{10.9}
\end{equation*}
\]

The period of the oscillation is
\[
\begin{equation*}
\tau=\frac{2 \pi}{p} \tag{10.10}
\end{equation*}
\]

If Eq. 10.8 is solved numerically, the step size \(\Delta t\) must be much smaller than the period \(\tau\).
As an example, consider the one-dimensional motion of the mass-spring system shown in Fig. 10.8(a). The only external force acting on the system is gravity. The equation of motion for this system, in the \(y\) direction, is given"by
\[
m \ddot{y}=-m g-k\left(y-l^{0}\right)
\]

(a)

(b)

Figure 10.8 (a) A one-dimensional vibrating mass-spring system, and (b) a full cycle of the response denoted by \(\tau\).
or
\[
\begin{equation*}
\ddot{y}+\frac{k}{m} y=-g+\frac{k}{m} l^{0} \tag{a}
\end{equation*}
\]

For a system in free vibration, if the left sides of Eq. 10.10 and Eq. a are compared, it is found that
\[
p^{2}=\frac{k}{m}
\]
or
\[
\begin{equation*}
\tau=2 \pi \sqrt{\frac{m}{k}} \tag{b}
\end{equation*}
\]

Given the values of \(m\) and \(k\), the time period \(\tau\) can be calculated. For numerical integration, a reasonable value for \(\Delta t\) can be \(\Delta t \simeq \tau / 20\).

As another example, consider the single pendulum shown in Fig. 10.9. A single equation of motion for this system in terms of the angular coordinate can be written

directly. However, here the equation is derived from the Cartesian equations of motion. Considering gravity to be the only external force, the equations of motion are
\[
\begin{align*}
& m \ddot{x}-\lambda_{1}=0 \\
& m \ddot{y}-\lambda_{2}=-m g  \tag{c}\\
& \mu \ddot{\phi}+d \cos \phi \lambda_{1}+d \sin \phi \lambda_{2}=0
\end{align*}
\]
where \(\lambda_{1}\) and \(\lambda_{2}\) are two Lagrange multipliers associated with the constraints for the revolute joint,
\[
\begin{aligned}
& x-d \sin \phi=0 \\
& y+d \cos \phi=0
\end{aligned}
\]

The kinematic acceleration equations are,
\[
\begin{align*}
& \ddot{x}-d \cos \phi \ddot{\phi}+d \sin \phi \dot{\phi}^{2}=0 \\
& \ddot{y}-d \sin \phi \ddot{\phi}-d \cos \phi \dot{\phi}^{2}=0 \tag{d}
\end{align*}
\]

Elimination of \(\lambda_{1}, \lambda_{2}, \ddot{x}\), and \(\ddot{y}\) in Eqs. \(c\) and \(d\) results in one equation:
\[
\left(\mu+m d^{2}\right) \ddot{\phi}+m d g \sin \phi=0
\]

For oscillations of small amplitude, \(\sin \phi\) can be replaced by \(\phi\); this last equation is then linearized as follows:
\[
\begin{equation*}
\left(\mu+m d^{2}\right) \ddot{\phi}+m d g \phi=0 \tag{e}
\end{equation*}
\]

Comparing Eq. 10.8 and Eq. \(e\) yields
\[
p^{2}=\frac{m d g}{\mu+m d^{2}}
\]
or
\[
\begin{equation*}
\tau=2 \pi \sqrt{\frac{\mu+m d^{2}}{m d g}} \tag{f}
\end{equation*}
\]

The preceding examples show how the time period of the oscillation can be calculated. For more complicated systems, the calculation of natural frequencies will not be that simple. For systems with several interconnected moving bodies, the linearization of the equations of motion in explicit form can be rather cumbersome. For systems with more than 1 degree of freedom, there will be more than one natural frequency. For such systems, the highest frequency must be found, and a step size much smaller than the period of the highest frequency must be selected. Since the equations of motion are generally nonlinear in terms of the coordinates, linearization of these equations yields a time step which is valid only in the individual configuration. In a different configuration for the same system, the linearization process may yield a different time period, and consequently a different time step size.

To avoid the difficulties associated with the selection of a proper time step, it is strongly suggested that a variable-step size algorithm be used for dynamic analysis. Many well-developed algorithms of this sort are available. Most of these algorithms determine a proper time step and will adjust the time step automatically during the simulation.

\section*{PROBLEMS}

The following problems provide examples that can be simulated by using a dynamic analysis program such as DAP. Many of the problems can be simulated on the existing listed version of DAP. Other problems may require some modifications or extensions to the program. Guidelines for improving the versatility and increasing the capability of DAP are included for some of those problems.
10.1 Refer to Probs. 5.8 through 5.11 and include similar modifications in DAP for input-output versatility.
10.2 Modify DAP to accept data in different but consistent units.
10.3 Formulate additional force elements such as the rotational spring-damper-actuator element in DAP.
10.4 Refer to Probs. 5.13 and 5.16 through 5.19. Include similar changes in DAP.
10.5 Refer to Prob. 9.5 and include this capability in DAP.
10.6 An external force acting on a body can be time-dependent. Modify DAP to accept time-dependent external forces in the following forms:
(a) A closed-form expression
(b) A table of data points
10.7 Provide the user with the option of including or excluding gravitational forces in a simulation.
10.8 Refer to Prob. 5.18. A dummy subroutine USRFRC, similar to the user-supplied subroutine USRCON, can be called from subroutine UFORCE before the RETURN statement. This subroutine may be used for nonstandard forces defined by the user.
10.9 The present version of DAP applies L-U factorization to the coefficient matrix in Eq. 10.5 to solve for \(\overline{\mathbf{q}}\) and \(\lambda\). This process can be made more efficient by the following modification:
(a) From Eq. 10.5 it is found that,
\[
\begin{gather*}
\mathbf{B} \boldsymbol{\lambda}=\boldsymbol{\gamma}-\mathbf{C g}  \tag{a.1}\\
\ddot{\mathbf{q}}=\mathbf{C}^{T} \boldsymbol{\lambda}+\mathbf{M}^{-1} \mathbf{g} \tag{a.2}
\end{gather*}
\]
where \(\mathbf{C}=\boldsymbol{\Phi}_{\mathbf{q}} \mathbf{M}^{-1}\) and \(\mathbf{B}=\mathbf{C} \boldsymbol{\Phi}_{\mathbf{q}}^{T}\). Matrix \(\mathbf{B}\) is an \(m \times m\) symmetric matrix.
(b) Perform L-U factorization on \(\mathbf{B}\), and then solve Eq. a. 1 for \(\boldsymbol{\lambda}\).
(c) Substitute the result for \(\boldsymbol{\lambda}\) in Eq. a. 2 to find \(\mathbf{q}\).

Note that this process requires \(\mathbf{M}^{-1}\) to be evaluated only once, since \(\mathbf{M}\) is a constant diagonal matrix. If the mass or the moment of inertia of a body is specified as zero, then the inversion cannot be performed. This is usually the case for nonmoving (grounded) bodies. For these bodies, the mass and moment of inertia can be set to any nonzero value; e.g., \(m=1\) and \(\mu=1\).
10.10 Matrix B in Prob. 10.9, for a well-posed problem, is symmetric and positive-definite. For these types of matrices, special-purpose matrix factorization algorithms are available. These algorithms are more efficient than the standard L-U factorization algorithms. Replace subroutine LU with a subroutine utilizing such an algorithm.
10.11 Evaluation of matrices \(\mathbf{C}\) and \(\mathbf{B}\) in Prob. 10.9 requires matrix multiplication involving \(\boldsymbol{\Phi}_{\mathbf{q}}\). Since \(\boldsymbol{\Phi}_{q}\) is a sparse matrix, a large number of multiplications by zero are performed in the process of evaluating \(\mathbf{C}\) and \(\mathbf{B}\). In order to eliminate an unnecessary multiplication by zero, store the row and column numbers of the nonzero elements of \(\boldsymbol{\Phi}_{\mathrm{q}}\) in two index arrays IRN
and ICN. The elements of these arrays, plus the fact that \(\mathbf{M}^{-1}\) is diagonal, can minimize the number of multiplications in the evaluation of \(\mathbf{C}\) and \(\mathbf{B}\).
10.12 For very small integration time steps and long simulation time periods, the output of DAP can be extensive. Modify DAP to report the result at every \(n\)th time step, where \(n\) is an integer to be specified by the user.
10.13 If the initial conditions on coordinates and velor ities violate the constraints and their time derivatives respectively, they must be corrected before the start of dynamic analysis. Perform a kinematic position and velocity analysis at \(t=0\) to correct the initial conditions. For this process, additional constraints equal to the number of degrees of freedom must be introduced. The corrected initial conditions are then used to start the dynamic analysis.
10.14 Replace the Runge-Kutta algorithm in DAP with a predictor-corrector algorithm in order to make the integration process more efficient (refer to Chap. 12). You may find such an algorithm in the library of your computer.
10.15 Use a variable step/order predictor-corrector algorithm in DAP instead of subrouting RUNG4. This can be the most important modification to DAP for minimizing the numerical error in the computation (refer to Prob. 12.7).
10.16 Refer to the constraint violation stabilization method in Sec. 13.3.1. Modify vector \(\gamma\), according to Eq. 13.18, to include the feedback terms \(-2 \alpha \dot{\boldsymbol{\Phi}}\) and \(-\beta^{2} \boldsymbol{\Phi}\). Before evaluating the forces in subroutine DYNAM, perform the following tasks:
(a) Call FUNCT, with IFNCT \(=1\) and \(\mathrm{JACOB}=0\), to obtain the constraint violations in F. Add \(-\beta^{2} \mathrm{~F}(\mathrm{I})\) to \(\mathrm{RHS}(\mathrm{I})\) for \(\mathrm{I}=1, \ldots, \mathrm{M}\).
(b) Since the Jacobian matrix is already available, add \(-2 \alpha \boldsymbol{\Phi}_{\mathrm{q}} \dot{\mathbf{q}}\) to \(\operatorname{RHS}(\mathrm{I})\) for \(\mathrm{I}=\) \(1, \ldots\), M. The user must specify the parameters \(\alpha\) and \(\beta\). Simulate different problems and experiment with different values of these parameters.
10.17 Modify DAP in order to handle both unconstrained and constrained systems. If \(\mathrm{M}=0\), there are no constraints. Therefore, the program should solve the equations of motion stated in Eq. 10.1 .
10.18 In the dynamic analysis of mechanical systems, it might be necessary to include aerodynamic forces in vector \(\mathbf{g}\). A simple formula for calculating aerodynamic forces is given as
\[
f^{\text {(aero) }}=\frac{1}{2} c_{d} \rho A \nu^{2}
\]
where
\[
\begin{aligned}
c_{d}= & \text { drag coefficient; e.g., } c_{d}=0.5 \\
\rho= & \text { air density } \\
v= & \text { velocity of the body } \\
A= & \text { cross-sectional area of the body perpendicular } \\
& \text { to the direction of velocity }
\end{aligned}
\]

Included this capability in DAP as another force element.
10.19 Simulate the dynamics of the chain shown in Fig. P.10.19. The links are connected by revolute joints and the external force is gravity. Start the dynamic analysis from two different initial positions:


Figure P. 10.19
(a) The links are open and the chain is stretched.
(b) The links are folded on top of one another.
10.20 The front landing gear of an aircraft is designed in such a way that it avoids hitting a protruding pod (this might be an extra fuel tank) while retracting. The opening or retracting of the landing gear is controlled by a hydraulic actuator between points \(P\) and \(Q\) as shown in Fig. P. 10.20. Assume \(A B=1.2, C D=1.62, B C=0.5, B O=1.32, A E=0.97\), \(D E=0.14\), and the wheel radius \(\rho=0.2\).
(a) Assign values to the mass and moment of inertia of each moving body.
(b) Simulate the retracting process by applying a constant force (or a variable force, if DAP is modified to accommodate one) to the actuator.
(c) Determine the path of the wheel with respect to the aircraft.


Figure P. 10.20
10.21 Simulate the bouncing of a rubber ball against the ground. Assume that the ground surface is flat at \(y=0\), as shown in Fig. P.10.21. Write a nonstandard force subroutine (refer to Prob. 10.8) to determine the reaction force between the ball and the ground during contact. Monitor the deformation \(\Delta l=\rho-y_{2}\). If \(\Delta l<0\), then there is no contact; hence, there is no reaction force. If \(\Delta l>0\), calculate the reaction force as \(f=f^{(s)}+f^{(t)}\)
where
\[
\begin{aligned}
f^{(s)} & =k \Delta l \\
f^{(d)} & =\left\{\begin{array}{cc}
-d \dot{y}_{2} & \text { for } \dot{y}_{2}<0 \\
0 & \text { for } \dot{y}_{2}>0
\end{array}\right.
\end{aligned}
\]

It is assumed that \(k\) and \(d\) are the stiffness and damping coefficient of the ball.


Figure P. 10.21
10.22 Repeat Prob. 10.21 and assume that the ground surface makes an angle with the horizontal direction.
10.23 Two rubber balls are constrained to move in the vertical direction inside a frictionless cylinder as shown in Fig. P.10.23. Write a subroutine to calculate the reaction forces between the balls and the ground. Simulate the motion of the balls using DAP.
10.24 The apparatus shown in Fig. P. 10.24 consists of five pendulums terminating in rubber balls which can be modeled as five moving bodies and five revolute joints. The interaction between the balls can be modeled by unilateral spring elements, as can that in Prob. 10.23. Move one ball (or two) from the equilibrium state and then release it (or them). Simulate the motion of the balls using DAP.

(1)

Figure P. 10.23


Figure P. 10.24
10.25 The motion of the articulated bulldozer shown in Fig. P. 10.25 is controlled by two hydraulic actuators. The centers of mass of the moving bodies are shown as \(G_{1}, \ldots, G_{4}\). Take measurements from the figure and assume \(m_{1}=190, m_{2}=52, m_{3}=12, m_{4}=3\), \(\mu_{1}=45, \mu_{2}=0.9, \mu_{3}=0.3\), and \(\mu_{4}=0.1\).
(a) Find a correct set of initial conditions for the coordinates.
(b) Determine the actuator forces in the equilibrium state.

10.26 A hydraulic excavator is shown in Fig. P.10.26. Set up a model for this vehicle by taking direct measurements from the figure and assuming reasonable values for the mass and moment of inertia of each body.
(a) For different orientations, find the correct set of initial conditions.
(b) For each set of initial conditions, find the necessary force for each actuator to keep the system in equilibrium.
(c) Perform dynamic analysis by controlling the force of each actuator.


Figure P. 10.26
10.27 Modify the revolute-revolute joint constraint formulation of Eq. 4.13 by assuming that the length of the link varies as a function of time; i.e.,
\[
l^{T} l-[d(t)]^{2}=0
\]

For this constraint, \(\dot{d}\) and \(\ddot{d}\) must be included in the velocity and acceleration equations.
(a) Include this formulation in DAP.
(b) Repeat Prob. 10.26 and employ this variable-length revolute-revolute joint constraint to represent the actuators. Specify a time function for each \(d(t)\) and monitor the reaction forces associated with each actuator.
This process is an inverse approach for determining the required force of each actuator as a function of time in order to generate a particular motion for the system.
10.28 A fork-lift mechanism is shown in Fig. P. 10.28. Rotation of the crank provides an upward or a downward motion of the fork. Set up a model for this vehicle by taking direct measurements from the figure and assigning values to the mass and moment of inertia of each body.
(a) For different orientations, find the correct set of initial conditions.
(b) For each set of initial conditions, find the torque on the crank that keeps the system in equilibrium.
(c) Perform a dynamic analysis by changing the applied torque. Monitor the path of the fork for a complete revolution of the crank.

10.29 The mechanism shown in Fig. P. 10.29 is a centrifugal brake system. The braking mechanism is designed so that when the drive shaft exceeds a certain angular velocity, the three pistons are forced against the hub wall. The contacting surfaces of the pistons are covered by brake pads, which provide the friction force opposing the motion, hence reducing the angular velocity. When the shaft is not rotating, the distance between the shaft axis and the contact surface of each pad is equal to the inner radius of the hub. The contact interface beween the pads and the hub wall can be modeled by unilateral spring elements. The spring force \(f^{(r)}=k_{2} \Delta l\) constitutes the reaction force, where \(k_{2}\) is the pad stiffness and \(\Delta l\) is the pad deformation. A friction force \(f^{(f)}=\mu_{d} f^{(r)}\) is applied to the pad at the contacting surface, where \(\mu_{d}\) is the kinetic coefficient of friction. Assume \(m_{2}=10, m_{3}=3, \mu_{2}=10\), \(\mu_{3}=3, k_{1}=150, d_{1}=10, k_{2}=10,000\), and \(\mu_{d}=0.5\).
(a) Write a user-supplied subroutine for the unilateral spring, reaction force, and friction force (repeat for each pad).
(b) For a specified initial rotational velocity of body 2, determine the initial velocity vector for the system that will satisfy the constraints.
(c) Apply a constant input torque to the shaft and simulate the response.
(d) Determine the equilibrium rotational velocity of the shaft.

Note: Since the three braking elements are identical, you may model only one element and multiply the resisting force (moment) by 3 .


Figure P. 10.29
10.30 The pressure-type altimeter shown in Fig. P. 10.30 utilizes the difference between the pressure at sea level and the ambient pressure to displace a pointer which indicates the altitude. This is accomplished by sealing air at sea-level pressure inside a bellows. When the outside pressure is different from the air pressure inside the bellows, the bellows expands or contracts until the pressure in the bellows is equal to the outside pressure.
(a) Write a user-supplied subroutine to model the bellows and convert the pressure difference to a force acting on the attached piston.
(b) Include some damping (friction) force acting on the piston.
(c) Simulate the dynamics of the system and show whether or not the displacement of the pointer is a linear function of the pressure change.
10.31 A simple model of a vehicle can be set up by assuming that it consists of three bodies and two revolute joints as shown in Fig. P. 10.31. Body 1 is the chassis, and bodies 2 and 3 are the wheels. Let \(m_{1}=600, m_{2}=m_{3}=20, \mu_{1}=1200\), and \(\mu_{2}=\mu_{3}=5\). The axial de-


Figure P. 10.30


Figure P. 10.31
formation of each tire can be modeled by unilateral spring-damper elements, where \(k=175,000\) and \(d=7000\).
(a) Use DAP to determine the static equilibrium state of the vehicle.
(b) Perform a dynamic analysis, starting from the static equilibrium state, by assuming that the wheels slide (i.e., do not rotate). Assign an initial velocity to the vehicle. Refer to Prob. 9.10 and write a user-supplied subroutine for different terrains and obstacles.
(c) For a more realistic dynamic simulation, the rotation of the wheels must be included in the analysis. Assume that the rear wheel (body 3) is the driving wheel. A moment \(n_{3}\) is applied to this wheel by the engine. If there is no slipping between the wheel and the ground, then a force \(f_{3}=-n_{3} / \rho_{3}\), where \(\rho_{3}\) is the deformed radius of the wheel, is applied to the wheel in the direction shown in Fig. P. 10.31(b).
(d) If the results of part (c) are studied, it is observed that the rotational and the translational velocity of body 3 do not satisfy the no-slip assumption; i.e., \(x_{3} \neq-\rho_{3} \dot{\phi}_{3}\). Therefore, a friction model must be included in the simulation. If \(v=\dot{x}_{3}+\rho_{3} \dot{\phi}_{3}\) is nonzero, then the wheel is slipping. A friction force can be calculated according to the friction curve shown in Fig. P.10.31(c). In this curve \(\beta\) is a coefficient based on the tire characteristics (for simulation, assume values of \(\beta\) between 1000 and 5000), and
the upper bound for the friction force is \(f^{(n}=\mu_{d} f^{(n)}\), wherc \(\mu_{d}\) is the kinetic coefficient of friction and \(f^{(r)}\) is the normal reaction force. \(f^{(r)}\) is available from the unilateral spring-damper element representing the tire deformation.
(e) Assume that the front wheel (body 2) is the driven wheel. Include a no-slip friction model for this wheel and perform a dynamic simulation.
10.32 Improve the vehicle model of Prob. 10.31 by adding a suspension system to the front and rear wheels as shown in Fig. P. 10.32. Assume that the front wheel is attached to an extra body (body 4) by a revolute joint, and that body 4 is attached to the chassis by a translational joint. Also, assume that the rear wheel is attached to an extra body (body 5) by a revolute joint, and that body 5 is attached to the chassis by another revolute joint. Let \(m_{4}=m_{5}=4, \mu_{4}=1\), and \(\mu_{5}=2\). The spring-damper characteristics for the front suspension system are \(k_{1}=90,000\) and \(d_{1}=5000\), and for the rear suspension system \(k_{2}=60,000\) and \(d_{2}=5000\). The tire characteristics are the same as in Prob. 10.31.
(a) Find the static equilibrium state for the vehicle. Adjust the attachment points or the undeformed lengths of the suspension springs in such a way that the main chassis and the axis of body 5 remain horizontal when the vehicle is in static equilibrium.
(b) Include friction models for a no-slip condition of the wheels.
(c) Perform a variety of simulations for different terrains.


Figure P. 10.32
10.33 Include an additional capability in DAP to perform a static equilibrium analysis prior to dynamic analysis when necessary. Refer to the algorithms in Chap. 14 (the iterative method of Eq. 14.3 requires a minimum amount of programming).

\section*{11}

\section*{Spatial Dynamics}

A general system of constrained equations of rigid-body spatial motion is formulated in this chapter on the basis of the principles of dynamics discussed in Chap. 8 and the spatial kinematics formulation from Chaps. 6 and 7. The equations of motion are formulated in terms of Euler parameters. The formulation developed here is identical in nature to that for planar systems in Chap. 9; the principal difference between the formulations for spatial and planar dynamics is in their dimensionality.

\subsection*{11.1 VECTOR OF FORCES}

The forces and moments acting on a body can be due to such force elements as springs, dampers, or gravity, among others. The derivation of equations to calculate the forces (or moments) of these force elements in spatial motion is identical to that shown in Secs. 9.2.1 through 9.2.7 for planar motion. If the resultant force and moment acting on body \(i\) are \(\vec{f}_{i}\) and \(\vec{n}_{i}\), they must be transformed into the coordinate system in which the equations of motion are derived. For the translational equations of motion shown in Eq. 8.31, the force \(\vec{f}_{i}\) must be defined in terms of its \(x y z\) components; i.e., \(\mathbf{f}_{i}\). If the rotational equations of motion given by Eq. 8.32 are used, then the moment \(\vec{n}_{i}\) must be defined in terms of its \(\xi \eta \zeta\) components; i.e., \(\mathbf{n}\). However, if Euler parameters are used and the equations of motion are expressed in terms of these coordinates, then the moment \(\vec{n}_{i}\) must be transformed to a coordinate system associated with the Euler parameters.

\subsection*{11.1.1 Conversion of Moments}

It is possible to convert the three rotational equations of motion represented by Eq. 8.32 to four rotational equations of motion associated with the Euler parameters (this will be
seen in Sec. 11.2.1). In this case the moment \(\vec{n}_{i}\) must be expressed in terms of four components denoted by \(\mathbf{n}_{i}^{*}\). The objective is to find the transformation between \(\mathbf{n}_{i}^{\prime}\) ( or \(\mathbf{n}_{i}\) ) and \(\mathbf{n}_{i}^{*}\). Two methods for deriving this transformation are shown here.

The first method is based on the scalar product of two vectors. As long as two vectors are described in the same coordinate system, their scalar product yields a scalar quantity independent of the coordinate system in which the vectors are expressed. In Eq. 8.32, \(\mathbf{n}_{i}^{\prime}\) is expressed in the same coordinate system as \(\boldsymbol{\omega}_{i}^{\prime}\). When Euler parameters are used, the moment \(\mathbf{n}_{i}^{*}\) must be expressed in the same coordinate system as \(\dot{\mathbf{p}}_{i}\). Hence,
\[
\begin{equation*}
\dot{\mathbf{p}}_{i}^{T} \mathbf{n}_{i}^{*}=\boldsymbol{\omega}_{i}^{\prime T} \mathbf{n}_{i}^{\prime} \tag{a}
\end{equation*}
\]

Then, Eq. 6.107 yields
\[
\begin{equation*}
\mathbf{n}_{i}^{*}=2 \mathbf{L}_{i}^{T} \mathbf{n}_{i}^{\prime} \tag{11.1}
\end{equation*}
\]

If the global components of these vectors are considered, then Eq. \(a\) is also equal to \(\boldsymbol{\omega}_{i}^{T} \mathbf{n}_{i}\), and therefore it can be found that
\[
\begin{equation*}
\mathbf{n}_{i}^{*}=2 \mathbf{G}_{i}^{T} \mathbf{n}_{i} \tag{11.2}
\end{equation*}
\]

The inverse transformations are
\[
\begin{equation*}
\mathbf{n}_{i}^{\prime}=\frac{1}{2} \mathbf{L}_{i} \mathbf{n}_{i}^{*} \tag{11.3}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{n}_{i}=\frac{1}{2} \mathbf{G}_{\boldsymbol{i}} \mathbf{n}_{i}^{*} \tag{11.4}
\end{equation*}
\]

The second method considers the virtual displacement of the point of application of a force on a body. In Fig. 11.1, \(\vec{f}_{i}\) acts on point \(P_{i}\) and the moment of the force is \(\mathbf{n}_{i}^{\prime}=\tilde{\mathbf{s}}_{i}^{\prime} \mathbf{f}_{i}^{\prime}\). The position of \(P_{i}\) is
\[
\begin{equation*}
\mathbf{r}_{i}^{P}=\mathbf{r}_{i}+\mathbf{A}_{i} \mathbf{s}_{i}^{\prime} \tag{11.5}
\end{equation*}
\]

The total differential of Eq. 11.5 is
\[
\begin{align*}
\delta \mathbf{r}_{i}^{P} & =\delta \mathbf{r}_{i}+\frac{\delta\left(\mathbf{A}_{i} \mathbf{s}_{i}^{\prime}\right)}{\delta \mathbf{p}_{i}} \delta \mathbf{p}_{i} \\
& =\delta \mathbf{r}_{i}+2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime} \delta \mathbf{p}_{i}+2 \mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T} \delta \mathbf{p}_{i} \tag{b}
\end{align*}
\]

Since the four Euler parameters are subject to the constraint \(\mathbf{p}_{i}^{T} \mathbf{p}_{i}^{-1}=0\), the total differential of this constraint yields
\[
\frac{\partial\left(\mathbf{p}_{i}^{T} \mathbf{p}_{i}-1\right)}{\partial \mathbf{p}_{i}} \delta \mathbf{p}_{i}=0
\]
or
\[
\begin{equation*}
\mathbf{p}_{i}^{T} \delta \mathbf{p}_{i}=0 \tag{11.6}
\end{equation*}
\]


Figure 11.1 Applied forces.

Hence, Eq. \(b\) is simplified as follows:
\[
\begin{aligned}
\delta \mathbf{r}_{i}^{P} & =\delta \mathbf{r}_{i}+2 \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime} \delta \mathbf{p}_{i} \\
& =\delta \mathbf{r}_{i}+2 \mathbf{A}_{i} \mathbf{L}_{i} \overline{\mathbf{s}}_{i}^{\prime} \delta \mathbf{p}_{i} \\
& =\delta \mathbf{r}_{i}+2 \mathbf{A}_{i}\left(-\tilde{\mathbf{s}}_{i}^{\prime} \mathbf{L}_{i}+\mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T}\right) \delta \mathbf{p}_{i} \\
& =\delta \mathbf{r}_{i}-2 \tilde{\mathbf{s}}_{i} \mathbf{G}_{i} \delta \mathbf{p}_{i}
\end{aligned}
\]
where Eqs. \(6.49,6.71\), and 6.88 have been employed.
The virtual work done by \(\mathbf{f}_{i}\) is
\[
\begin{align*}
\delta w_{i} & =\mathbf{f}_{i}^{T} \delta \mathbf{r}_{i}^{P} \\
& =\mathbf{f}_{i}^{T}\left(\delta \mathbf{r}_{i}-2 \tilde{\mathbf{s}}_{i} \mathbf{G}_{i} \delta \mathbf{p}_{i}\right) \\
& =\mathbf{f}_{i}^{T} \delta \mathbf{r}_{i}+2 \mathbf{n}_{i}^{T} \mathbf{G}_{i} \delta \mathbf{p}_{i} \tag{11.7}
\end{align*}
\]

This equation shows that the virtual work \(\delta w_{i}\) is the sum of the virtual work of the force \(\mathbf{f}_{i}\) causing a virtual translation \(\delta \mathbf{r}_{i}\) and the virtual work of a moment \(\mathbf{n}_{i}^{*}=2 \mathbf{G}_{i}^{T} \mathbf{n}_{i}\) causing a virtual rotation \(\delta \mathbf{p}_{i}\). This result agrees with Eq. 11.2.

\subsection*{11.2 EQUATIONS OF MOTION FOR AN UNCONSTRAINED BODY}

The translational equations of motion for an unconstrained body are given by Eq. 8.31 as
\[
\begin{equation*}
\mathbf{N}_{i} \ddot{\mathbf{r}}_{i}=\mathbf{f}_{i} \tag{11.8}
\end{equation*}
\]
where \(\mathbf{N}_{i}=\operatorname{diag}[m, m, m]_{i}\). The rotational equations of motion for an unconstrained body given by Eq. 8.32 are converted into three different forms in this section.

Formulation I. Substitution of Eqs. 6.107 and 6.111 into Eq. 8.32 yields
\[
\begin{equation*}
2 \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \ddot{\mathbf{p}}_{i}+4 \mathbf{L}_{i} \dot{\mathbf{L}}_{i}^{T} \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \dot{\mathbf{p}}_{i}=\mathbf{n}_{i}^{\prime} \tag{11.9}
\end{equation*}
\]

Premultiplication of this equation by \(2 \mathbf{L}_{i}^{T}\) gives
\[
\begin{equation*}
\mathbf{J}_{i}^{*} \ddot{\mathbf{p}}_{i}+2 \mathbf{L}_{i}^{T} \mathbf{L}_{i} \mathbf{H}_{i} \dot{\mathbf{p}}_{i}=\mathbf{n}_{i}^{*} \tag{11.10}
\end{equation*}
\]
where
\[
\begin{equation*}
\mathbf{J}_{i}^{*}=4 \mathbf{L}_{i}^{T} \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \tag{11.11}
\end{equation*}
\]
and
\[
\begin{equation*}
\mathbf{H}_{i}=4 \dot{\mathbf{L}}_{i}^{T} \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \tag{11.12}
\end{equation*}
\]

Using Eq. 6.46 and defining
\[
\begin{align*}
\sigma_{i} & =8 \dot{\mathbf{p}}_{i}^{T} \mathbf{L}_{i}^{T} \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \dot{\mathbf{p}}_{i} \\
& =-8 \mathbf{p}_{i}^{T} \dot{\mathbf{L}}_{i}^{T} \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \dot{\mathbf{p}}_{i} \tag{11.13}
\end{align*}
\]
we can write Eq. 11.10 as
\[
\begin{equation*}
\mathbf{J}_{i}^{*} \ddot{\mathbf{p}}_{i}+\sigma_{i} \mathbf{p}_{i}+2 \mathbf{H}_{i} \dot{\mathbf{p}}_{i}=\mathbf{n}_{i}^{*} \tag{11.14}
\end{equation*}
\]

This represents the rotational equations of motion in terms of \(\ddot{\mathbf{p}}_{i}\). However, since the four Euler parameters are not independent, Eq. 6.61, i.e,
\[
\begin{equation*}
\mathbf{p}_{i}^{T} \ddot{\mathbf{p}}+\dot{\mathbf{p}}_{i}^{T} \dot{\mathbf{p}}_{i}=0 \tag{11.15}
\end{equation*}
\]
must be considered with Eq. 11.14. Equations 11.14 and 11.15 in matrix form are
\[
\left[\begin{array}{cc}
\mathbf{J}^{*} & \mathbf{p}  \tag{11.16}\\
\mathbf{p}^{T} & 0
\end{array}\right]_{i}\left[\begin{array}{l}
\ddot{\mathbf{p}} \\
\sigma
\end{array}\right]_{i}+\left[\begin{array}{c}
2 \mathbf{H} \dot{\mathbf{p}} \\
\dot{\mathbf{p}}^{\dot{\mathbf{p}}}
\end{array}\right]_{i}=\left[\begin{array}{c}
\mathbf{n}^{*} \\
0
\end{array}\right]_{i}
\]

Equation 11.16 contains five equations that can be solved for \(\ddot{\mathbf{p}}_{i}\) and \(\sigma_{i}\) if \(\mathbf{n}_{i}^{\prime}, \mathbf{p}_{i}, \dot{\mathbf{p}}_{i}\), and \(\mathbf{J}_{i}^{\prime}\) are known. This, obviously, gives the same value for \(\sigma_{i}\) as given by Eq. 11.13! The artificial variable \(\sigma_{i}\) was defined in such a way as to have an exact inverse to the \(5 \times 5\) matrix at the left in Eq. 11.16. In Eq. 11.14, \(\sigma_{i}\) can be interpreted as a Lagrange multiplier associated with the constraint equation \(\mathbf{p}_{i}^{T} \mathbf{p}_{i}-1=0\).

Formulation II. Equation 11.15 can be appended to Eq. 11.9 and written in matrix form to yield
\[
\left[\begin{array}{c}
2 \mathbf{J}^{\prime} \mathbf{L}  \tag{11.17}\\
\mathbf{p}^{T}
\end{array}\right]_{i} \ddot{\mathbf{p}}_{i}+\left[\begin{array}{c}
\mathbf{L} \mathbf{H} \\
\dot{\mathbf{p}}^{T}
\end{array}\right]_{i} \dot{\mathbf{p}}_{i}=\left[\begin{array}{c}
\mathbf{n}^{\prime} \\
0
\end{array}\right]_{\mathrm{i}}
\]

If \(\mathbf{n}_{i}^{\prime}, \mathbf{p}_{i}, \dot{\mathbf{p}}_{i}\), and \(\mathbf{J}_{i}^{\prime}\) are known, then Eq. 11.17 can be solved exactly for \(\ddot{\mathbf{p}}_{i}\). Note that the matrix at the left in Eq. 11.17 is a \(4 \times 4\) matrix.

Formulation III. In the third formulation, the rotational equations of motion are left in their original form in terms of the angular velocities; i.e.,
\[
\begin{equation*}
\mathbf{J}_{i}^{\prime} \dot{\omega}_{i}^{\prime}+\tilde{\boldsymbol{\omega}}_{i}^{\prime} \mathbf{J}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime}=\mathbf{n}_{i}^{\prime} \tag{11.18}
\end{equation*}
\]

It is clear that \(\dot{\boldsymbol{\omega}}_{i}^{\prime}\) can be calculated from this equation if \(\mathbf{n}_{i}^{\prime}, \boldsymbol{\omega}_{i}^{\prime}\), and \(\mathbf{J}_{i}^{\prime}\) are known.
A comparison of these three formulations shows that Eq. 11.16 contains five equations in terms of \(\ddot{\mathbf{p}}_{i}\) and \(\sigma_{i}\), Eq. 11.17 contains four equations in terms of \(\ddot{\mathbf{p}}_{i}\), and Eq. 11.18 contains three equations in terms of \(\dot{\boldsymbol{\omega}}_{i}^{\prime}\).

\subsection*{11.3 EQUATIONS OF MOTION FOR A CONSTRAINED BODY}

For a constrained mechanical system containing body \(i\), it is assumed that there are \(m\) independent constraint equations,
\[
\begin{equation*}
\Phi \equiv \Phi(\mathbf{q})=0 \tag{11.19}
\end{equation*}
\]
where \(\mathbf{q}\) contains the coordinates of all of the bodies in the system, including the coordinates of body \(i\) :
\[
\mathbf{q}_{i}=\left[\begin{array}{c}
\mathbf{r}_{i} \\
\mathbf{p}_{i}
\end{array}\right]
\]

It was shown in Sec. 8.4.3 that the constraint reaction forces could be described in the form given by Eq. 8.50 in terms of the Jacobian matrix of the system and a vector of Lagrange multipliers:
\[
\begin{equation*}
\mathbf{g}^{(*)}=\boldsymbol{\Phi}_{9}^{T} \boldsymbol{\lambda} \tag{11.20}
\end{equation*}
\]

This equation was obtained with the assumption that the vectors of forces \(\mathbf{g}\) and \(\mathbf{g}^{(*)}\) were defined in a coordinate system consistent with \(\mathbf{q}\).

The constrained translational equations of motion for body \(i\) can be written, from Eq. 11.8, as
\[
\mathbf{N}_{i} \dot{\mathbf{r}}_{i}=\mathbf{f}_{i}+\mathbf{f}_{i}^{(c)}
\]

From Eq. 11.20 it is found that
\[
\mathbf{f}_{i}^{(c)}=\boldsymbol{\Phi}_{\mathbf{r}_{i}}^{T} \boldsymbol{\lambda}
\]

Therefore,
\[
\begin{equation*}
\mathbf{N}_{i} \ddot{\mathbf{r}}_{i}-\boldsymbol{\Phi}_{\mathbf{r}_{i}}^{T} \boldsymbol{\lambda}=\mathbf{f}_{i} \tag{11.21}
\end{equation*}
\]

This represents the translational equations of motion for constrained body \(i\). The rotational equations of motion for this body are derived in three forms corresponding to the formulations of Sec. 11.2.

Formulation I. The rotational equations of motion from Eq. 11.14 for constrained body \(i\) are written as
\[
\mathbf{J}_{i}^{*} \ddot{\mathbf{p}}_{i}+\sigma_{i} \mathbf{p}_{i}+2 \mathbf{H}_{i} \dot{\mathbf{p}}_{i}=\mathbf{n}_{i}^{*}+\mathbf{n}_{i}^{*(c)}
\]

Since \(\mathbf{n}_{i}^{*}\) and \(\mathbf{n}_{i}^{*(c)}\) are described in the same coordinate system as \(\mathbf{p}_{i}\), Eq. 11.20 yields
\[
\mathbf{n}_{i}^{*(c)}=\boldsymbol{\Phi}_{\mathbf{p}_{i}}^{T} \boldsymbol{\lambda}
\]

Therefore,
\[
\begin{equation*}
\mathbf{J}_{i}^{*} \ddot{\mathbf{p}}_{i}+\sigma_{i} \mathbf{p}_{i}+2 \mathbf{H}_{i} \dot{\mathbf{p}}_{i}-\boldsymbol{\Phi}_{\mathbf{p}_{i}}^{T} \boldsymbol{\lambda}=\mathbf{n}_{i}^{*} \tag{11.22}
\end{equation*}
\]

Equations 11.22 and 11.15 are the rotational equations of motion for a constrained body.
Formulation II. Equation 11.9 is written for a constrained body as
\[
2 \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \ddot{\mathbf{p}}_{i}+\mathbf{L}_{i} \mathbf{H}_{i} \dot{\mathbf{p}}_{i}=\mathbf{n}_{i}^{\prime}+\mathbf{n}_{i}^{\prime(c)}
\]

The transformation of moments is given by Eq. 11.3 as
\[
\begin{aligned}
\mathbf{n}_{i}^{\prime(c)} & =\frac{1}{2} \mathbf{L}_{i} \mathbf{n}_{i}^{*(c)} \\
& =\frac{1}{2} \mathbf{L}_{i} \boldsymbol{\Phi}_{\mathbf{p}_{i}}^{T} \boldsymbol{\lambda}
\end{aligned}
\]

Therefore,
\[
\begin{equation*}
2 \mathbf{J}_{i}^{\prime} \mathbf{L}_{i} \ddot{\mathbf{p}}_{i}+\mathbf{L}_{i} \mathbf{H}_{i} \dot{\mathbf{p}}_{i}-\frac{1}{2} \mathbf{L}_{i} \boldsymbol{\Phi}_{\mathbf{p}_{i}}^{T} \boldsymbol{\lambda}=\mathbf{n}_{i}^{\prime} \tag{11.23}
\end{equation*}
\]

Equations 11.23 and 11.15 together can be used as the rotational equations of motion for a constrained body.

Formulation III. Equation 11.18 can be written for a constrained body as
\[
\mathbf{J}_{i}^{\prime} \dot{\boldsymbol{\omega}}_{i}^{\prime}+\tilde{\boldsymbol{\omega}}_{i}^{\prime} \mathbf{J}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime}=\mathbf{n}_{i}^{\prime}+\mathbf{n}_{i}^{\prime(c)}
\]
or
\[
\begin{equation*}
\mathbf{J}_{i}^{\prime} \dot{\boldsymbol{\omega}}_{i}^{\prime}+\tilde{\boldsymbol{\omega}}_{i}^{\prime} \mathbf{J}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime}-\frac{1}{2} \mathbf{L}_{i} \boldsymbol{\Phi}_{\mathbf{p}_{i}}^{T} \boldsymbol{\lambda}=\mathbf{n}_{i}^{\prime} \tag{11.24}
\end{equation*}
\]

In this equation the constraint equations, and hence the Jacobian matrix, are expressed in terms of Euler parameters. However, the joint reaction moments are converted to a set of coordinates consistent with \(\boldsymbol{\omega}_{i}^{\prime}\) and \(\mathbf{n}_{i}^{\prime}\).

\subsection*{11.4 SYSTEM OF SPATIAL EQUATIONS OF MOTION}

In the preceding sections, the equations of motion for a single body were derived. Three formulations for the rotational equations of motion were shown. For a system of \(b\) bodies, constrained or unconstrained, these equations can be repeated \(b\) times in any of the three forms to find the system equations.

\subsection*{11.4.1 Unconstrained Bodies}

For a system of \(b\) unconstrained bodies, three formulations are given.
Formulation 1. Equations \(11.8,11.14\), and 11.15 , with a slight rearrangement, are written for all \(b\) bodies as
\[
\left[\begin{array}{cc}
\mathbf{M}^{*} & \mathbf{p}^{T}  \tag{11.25}\\
\mathbf{P} & \mathbf{O}
\end{array}\right]\left[\begin{array}{l}
\ddot{\mathbf{q}} \\
\boldsymbol{\sigma}
\end{array}\right]+\left[\begin{array}{c}
\mathbf{b}^{*} \\
\mathbf{c}
\end{array}\right]=\left[\begin{array}{c}
\mathbf{g}^{*} \\
\mathbf{0}
\end{array}\right]
\]
where
\[
\begin{align*}
& \mathbf{M}^{*}=\left[\begin{array}{lllllll}
\mathbf{N}_{1} & & & & & \\
& \mathbf{J}_{1}^{*} & & \mathbf{0} & & \\
& & \ddots & & & \\
& \mathbf{0} & & & \mathbf{N}_{b} & \\
& & & & & \mathbf{J}_{b}^{*}
\end{array}\right]  \tag{11.26}\\
& \mathbf{P}=\left[\begin{array}{llllll}
\mathbf{0}^{T} & \mathbf{p}_{1}^{T} & & & & \\
& & & \mathbf{0} & & \\
& & \ddots & & & \\
& \mathbf{0} & & & \mathbf{0}^{T} & \mathbf{p}_{b}^{T}
\end{array}\right]  \tag{11.27}\\
& \ddot{\mathbf{q}}=\left[\begin{array}{c}
\ddot{\mathbf{r}}_{1} \\
\ddot{\mathbf{p}}_{1} \\
\vdots \\
\ddot{\mathbf{r}}_{b} \\
\ddot{\mathbf{p}}_{b}
\end{array}\right]  \tag{11.28}\\
& \boldsymbol{\sigma}=\left[\begin{array}{c}
\sigma_{1} \\
\vdots \\
\sigma_{b}
\end{array}\right]  \tag{11.29}\\
& \mathbf{b}^{*}=\left[\begin{array}{c}
0 \\
2 \mathbf{H}_{i} \dot{\mathbf{p}}_{1} \\
\vdots \\
0 \\
2 \mathbf{H}_{b} \dot{\mathbf{p}}_{b}
\end{array}\right]  \tag{11.30}\\
& \mathbf{c}=\left[\begin{array}{c}
\dot{\mathbf{p}}_{1}^{T} \dot{\mathbf{p}}_{1} \\
\vdots \\
\vdots \\
\dot{\mathbf{p}}_{b}^{T} \dot{\mathbf{p}}_{b}
\end{array}\right] \tag{11.31}
\end{align*}
\]
\[
\mathbf{g}^{*}=\left[\begin{array}{c}
\mathbf{f}_{1}  \tag{11.32}\\
\mathbf{n}_{1}^{*} \\
\vdots \\
\mathbf{f}_{b} \\
\mathbf{n}_{b}^{*}
\end{array}\right]
\]

Formulation II. Equations 11.8 and 11.17 , with a slight rearrangement, are written for all \(b\) bodies as
\[
\left[\begin{array}{c}
M  \tag{11.33}\\
\mathbf{P}
\end{array}\right] \ddot{\mathbf{q}}+\left[\begin{array}{l}
\boldsymbol{b} \\
\mathbf{c}
\end{array}\right]=\left[\begin{array}{l}
\mathbf{g} \\
\theta
\end{array}\right]
\]
where
\[
\begin{gather*}
\boldsymbol{M}=\left[\begin{array}{ccccc}
\mathbf{N}_{\mathbf{1}} & & & & \\
& 2_{\mathbf{J}_{1}^{\prime} \mathbf{L}_{\mathbf{1}}} & & \mathbf{0} & \\
& & \ddots & & \\
& \mathbf{0} & & \mathbf{N}_{b} & \\
& \\
\boldsymbol{b}=\left[\begin{array}{c}
0 \\
\\
\\
\\
\mathbf{L}_{1} \mathbf{H}_{1} \dot{\mathbf{p}}_{1} \\
\vdots \\
\mathbf{0} \\
\mathbf{L}_{b} \mathbf{H}_{b} \dot{\mathbf{p}}_{b}
\end{array}\right] \\
{\left[\begin{array}{c}
\mathbf{f}_{1} \\
\mathbf{n}_{1}^{\prime} \\
\vdots \\
\vdots \\
\mathbf{f}_{b} \\
\mathbf{n}_{b}^{\prime}
\end{array}\right]}
\end{array}\right. \tag{11.34}
\end{gather*}
\]

Formulation III. Equations 11.8 and 11.18 are written for all \(b\) bodies to obtain a set of equations identical to Eq. 8.40:
\[
\begin{equation*}
\mathbf{M} \dot{\mathbf{h}}+\mathbf{b}=\mathbf{g} \tag{11.37}
\end{equation*}
\]
where
\[
\mathbf{M}=\left[\begin{array}{lllll}
\mathbf{N}_{1} & & & &  \tag{11.38}\\
& \mathbf{J}_{1}^{\prime} & & \mathbf{0} & \\
& & \ddots & & \\
& \mathbf{0} & & \mathbf{N}_{b} & \\
& & & & \mathbf{J}_{b}^{\prime}
\end{array}\right]
\]
\[
\begin{gather*}
\dot{\mathbf{h}}=\left[\begin{array}{c}
\ddot{\mathbf{r}}_{1} \\
\dot{\boldsymbol{\omega}}_{1}^{\prime} \\
\vdots \\
\vdots \\
\ddot{\mathbf{r}}_{b} \\
\dot{\boldsymbol{\omega}}_{b}^{\prime}
\end{array}\right]  \tag{11.39}\\
\mathbf{b}=\left[\begin{array}{c}
0 \\
\tilde{\boldsymbol{\omega}}_{1}^{\prime} \mathbf{J}_{1}^{\prime} \boldsymbol{\omega}_{i}^{\prime} \\
\vdots \\
\vdots \\
0 \\
\tilde{\boldsymbol{\omega}}_{b}^{\prime} \mathbf{J}_{b}^{\prime} \boldsymbol{\omega}_{b}^{\prime}
\end{array}\right] \tag{11.40}
\end{gather*}
\]

\subsection*{11.4.2 Constrained Bodies}

For a system of \(b\) constrained bodies with the \(m\) independent constraint equations Eq. 11.19, three different formulations are obtained. The second-time derivative of the constraint equations, i.e.,
\[
\begin{equation*}
\Phi_{q} \ddot{\mathbf{q}}=\gamma \tag{11.41}
\end{equation*}
\]
is appended to the equations of motion. The total number of equations becomes equal to the total number of accelerations and Lagrange multipliers.

Formulation I. Equations \(11.21,11.22\), and 11.15 are written for all \(b\) bodies and then Eq. 11.41 is appended to them to obtain \(8 \times b+m\) equations, as follows:
\[
\left[\begin{array}{lll}
\mathbf{M}^{*} & \mathbf{P}^{T} & \mathbf{B}^{T}  \tag{11.42}\\
\mathbf{P} & \mathbf{0} & \mathbf{0} \\
\mathbf{B} & \mathbf{0} & \mathbf{0}
\end{array}\right]\left[\begin{array}{r}
\ddot{\mathbf{q}} \\
\boldsymbol{\sigma} \\
-\boldsymbol{\lambda}
\end{array}\right]+\left[\begin{array}{l}
\mathbf{b}^{*} \\
\mathbf{c} \\
\mathbf{0}
\end{array}\right]=\left[\begin{array}{l}
\mathbf{g}^{*} \\
\mathbf{0} \\
\gamma
\end{array}\right]
\]
where
\[
\begin{align*}
\mathbf{B} & =\boldsymbol{\Phi}_{\mathbf{q}} \\
& =\left[\Phi_{\mathrm{r}_{1}}, \boldsymbol{\Phi}_{\mathbf{p}_{\mathbf{l}}}, \ldots, \boldsymbol{\Phi}_{\mathrm{r}_{b}}, \boldsymbol{\Phi}_{\mathbf{p}_{b}}\right] \tag{11.43}
\end{align*}
\]
and
\[
\begin{equation*}
\boldsymbol{\lambda}=\left[\lambda_{1}, \lambda_{2}, \ldots, \lambda_{m}\right]^{T} \tag{11.44}
\end{equation*}
\]

Note that the square matrix at the left in Eq. 11.42 is symmetric.
Formulation II. Equations I1.21, 11.23, and 11.15 are written for all \(b\) bodies and then Eq. 11.41 is appended to them to obtain \(7 \times b+m\) equations, as follows:
\[
\left[\begin{array}{ll}
\boldsymbol{M} & \boldsymbol{B}^{T}  \tag{11.45}\\
\mathbf{P} & \mathbf{0} \\
\mathbf{B} & \mathbf{0}
\end{array}\right]\left[\begin{array}{r}
\ddot{\mathbf{q}} \\
-\lambda
\end{array}\right]+\left[\begin{array}{l}
\boldsymbol{b} \\
\mathbf{c} \\
\mathbf{0}
\end{array}\right]=\left[\begin{array}{l}
\mathbf{g} \\
\boldsymbol{0} \\
\gamma
\end{array}\right]
\]
where
\[
\begin{equation*}
\boldsymbol{B}=\left[\boldsymbol{\Phi}_{\mathrm{r}_{1}}, \frac{1}{2} \boldsymbol{\Phi}_{\mathbf{p}_{1}} \mathbf{L}_{1}^{T}, \ldots, \boldsymbol{\Phi}_{\mathrm{r}_{b}}, \frac{1}{2} \boldsymbol{\Phi}_{\mathbf{p}_{b}} \mathbf{L}_{b}^{T}\right] \tag{11.46}
\end{equation*}
\]

Note that the square matrix at the left in Eq. 11.45 is nonsymmetric.
Formulation III. In the last formulation, the acceleration equation as given by Eq. 11.41 is written in terms of the angular acceleration of each body, \(\dot{\boldsymbol{\omega}}_{i}^{\prime}\), instead of \(\ddot{\mathbf{p}}_{i}\), This conversion is performed first by writing Eq. 11.41 as
\[
\left[\boldsymbol{\Phi}_{\mathbf{r}_{1}}, \Phi_{\mathbf{p}_{1}}, \ldots, \Phi_{\mathbf{r}_{b}}, \Phi_{\mathbf{p}_{b}}\right]\left[\begin{array}{c}
\ddot{\mathbf{r}}_{1}  \tag{11.47}\\
\ddot{\mathbf{p}}_{1} \\
\vdots \\
\vdots \\
\ddot{\mathbf{r}}_{b} \\
\ddot{\mathbf{p}}_{b}
\end{array}\right]=\gamma
\]

From the identity \(\ddot{\mathbf{p}}_{i}=\frac{1}{2} \mathbf{L}_{i}^{T} \dot{\boldsymbol{\omega}}_{i}^{\prime}-\frac{1}{4} \omega^{2} \mathbf{p}_{i}\), a typical term \(\boldsymbol{\Phi}_{\mathbf{p}_{i}} \ddot{\mathbf{p}}_{i}\) in Eq. 11.47 can be written as
\[
\begin{aligned}
\boldsymbol{\Phi}_{\mathbf{p}_{i}} \ddot{\mathbf{p}}_{i} & =\boldsymbol{\Phi}_{\mathbf{p}_{i}}\left(\frac{1}{2} \mathbf{L}_{i}^{T} \dot{\boldsymbol{\omega}}_{i}^{\prime}-\frac{1}{4} \omega_{i}^{2} \mathbf{p}_{i}\right) \\
& =\frac{1}{2} \boldsymbol{\Phi}_{\mathbf{p}_{i}} \mathbf{L}_{i}^{T} \dot{\boldsymbol{\omega}}_{i}^{\prime}-\frac{1}{4} \omega_{i}^{2} \boldsymbol{\Phi}_{\mathbf{p}_{i}} \mathbf{p}_{i}
\end{aligned}
\]

Hence, Eq. 11.47 is rewritten as
\[
\left[\boldsymbol{\Phi}_{\mathbf{r}_{1}}, \frac{1}{2} \boldsymbol{\Phi}_{\mathbf{p}_{1}} \mathbf{L}_{1}^{T}, \ldots, \boldsymbol{\Phi}_{\mathbf{r}_{b}}, \frac{1}{2} \boldsymbol{\Phi}_{\mathbf{p}_{b}} \mathbf{L}_{b}^{T}\right]\left[\begin{array}{c}
\dot{\mathbf{r}}_{1}  \tag{11.48}\\
\dot{\boldsymbol{\omega}}_{1}^{\prime} \\
\vdots \\
\ddot{\mathbf{r}}_{b} \\
\dot{\boldsymbol{\omega}}_{b}^{\prime}
\end{array}\right]=\boldsymbol{\gamma}^{\#}
\]
where the terms \(-\frac{1}{4} \omega_{i}^{2} \boldsymbol{\Phi}_{p_{i}} \mathbf{p}_{i}, i=1, \ldots, b\), have been moved to the right side of the equation. A detailed explanation of this new form of the Jacobian matrix is given in the next section. Appending Eq. 11.48 to Eqs. 11.21 and 11.24 for all \(b\) bodies yields
\[
\left[\begin{array}{cc}
\mathbf{M} & B^{\tau}  \tag{11.49}\\
B & 0
\end{array}\right]\left[\begin{array}{c}
\dot{\mathbf{h}} \\
-\lambda
\end{array}\right]+\left[\begin{array}{l}
\mathbf{b} \\
0
\end{array}\right]=\left[\begin{array}{c}
\mathbf{g} \\
\gamma^{\#}
\end{array}\right]
\]

Note that the square matrix at the left in this equation is symmetric.

\subsection*{11.5 CONVERSION OF KINEMATIC EQUATIONS}

Although the Euler parameters are ideal for representing the angular orientation of a body in space, they yield too many equations when their time derivatives are used explicitly in the equations of motion, as was shown in Eq. 11.16. Equation 11.18 shows that only three rotational equations are needed if \(\dot{\boldsymbol{\omega}}_{i}^{\prime}\) is used instead of \(\ddot{\mathbf{p}}_{i}\). For a constrained body, the equations of motion given in Eq. 11.24 contain only three equations and also take advantage of the Euler parameters (the constraint equations and hence the Jacobian matrix are described in terms of Euler parameters). This advantage becomes
apparent when Eqs. 11.42 and 11.49 are compared. Equation 11.49 contains \(2 \times b\) fewer equations than Eq. 11.42.

In Eq. 11.49, the kinematic constraints are kept in terms of Euler parameters, as follows:
\[
\begin{align*}
\boldsymbol{\Phi} & \equiv \boldsymbol{\Phi}(\mathbf{q}) \\
& =\boldsymbol{\Phi}\left(\mathbf{r}_{1}, \mathbf{p}_{1}, \ldots, \mathbf{r}_{b}, \mathbf{p}_{b}\right)=\mathbf{0} \tag{11.50}
\end{align*}
\]

The velocity equations are written as
\[
\begin{align*}
\dot{\mathbf{\Phi}} & \equiv \mathbf{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} \\
& =\left[\mathbf{\Phi}_{\mathbf{r}_{1}}, \mathbf{\Phi}_{\mathbf{p}_{1}}, \ldots, \mathbf{\Phi}_{\mathbf{r}_{b}}, \mathbf{\Phi}_{\mathbf{p}_{b}}\right]\left[\begin{array}{c}
\dot{\mathbf{r}}_{1} \\
\dot{\mathbf{p}}_{1} \\
\vdots \\
\vdots \\
\dot{\mathbf{r}}_{b} \\
\dot{\mathbf{p}}_{b}
\end{array}\right] \\
& =\left[\mathbf{\Phi}_{\mathbf{r}_{1}}, \frac{1}{2} \mathbf{\Phi}_{\mathbf{p}_{1}} \mathbf{L}_{1}^{T}, \ldots, \boldsymbol{\Phi}_{\mathbf{r}_{b}}, \frac{1}{2} \mathbf{\Phi}_{\mathbf{p}_{b}} \mathbf{L}_{b}^{T}\right]\left[\begin{array}{c}
\dot{\mathbf{r}}_{1} \\
\boldsymbol{\omega}_{1}^{\prime} \\
\vdots \\
\vdots \\
\dot{\mathbf{r}}_{b} \\
\boldsymbol{\omega}_{b}^{\prime}
\end{array}\right]=0 \tag{11.51}
\end{align*}
\]

The modified Jacobian matrix of Eq. 11.51 is the same as that of Eq. 11.48. The modified Jacobian matrix and the modified vector \(\boldsymbol{\gamma}^{\#}\) can be obtained in explicit forms for the constraint equations of Chap. 7.

\section*{Example 11.1}

The modified velocity and acceleration equations and hence the modified Jacobian matrix and vector \(\boldsymbol{\gamma}^{\#}\) for the constraint equation of Eq. 7.3 are derived here. The velocity equation is
\[
\begin{align*}
\dot{\Phi}^{(n 1,1)} & \equiv \mathbf{s}_{i}^{T} \dot{\mathbf{j}}_{j}+\mathbf{s}^{T} \dot{\mathbf{s}}_{i} \\
& =\mathbf{s}_{i}^{T} \dot{\mathbf{A}}_{j} \mathbf{s}_{j}^{\prime}+\mathbf{s}_{\mathbf{j}}^{T} \dot{\mathbf{A}}_{i} \mathbf{s}_{i}^{\prime} \\
& =\mathbf{s}_{i}^{T} \mathbf{A}_{j} \tilde{\boldsymbol{\omega}}_{j}^{\prime} \mathbf{s}_{j}^{\prime}+\mathbf{s}_{j}^{T} \mathbf{A}_{i} \tilde{\boldsymbol{\omega}}_{i}^{\prime} \mathbf{s}_{i}^{\prime} \\
& =-\mathbf{s}_{i}^{T} \mathbf{A}_{j} \tilde{\mathbf{s}}_{j}^{\prime} \boldsymbol{\omega}_{j}^{\prime}-\mathbf{s}_{j}^{T} \mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime} \\
& =\left[-\mathbf{s}_{j}^{T} \mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime},-\mathbf{s}_{i}^{T} \mathbf{A}_{j} \tilde{\mathbf{s}}_{j}^{\prime}\right]\left[\begin{array}{c}
\boldsymbol{\omega}_{i}^{\prime} \\
\boldsymbol{\omega}_{j}^{\prime}
\end{array}\right]=0 \tag{1}
\end{align*}
\]

The entries of the modified transformation matrix could have been found directly from Table 7.2:
\[
\begin{align*}
\frac{1}{2} \boldsymbol{\Phi}_{\mathbf{p}_{i}}^{(m)} \mathbf{L}_{i}^{T} & =\frac{1}{2}\left(2 \mathbf{s}_{j}^{T} \mathbf{G}_{i} \overline{\mathbf{s}}_{i}^{\prime}\right) \mathbf{L}_{i}^{T} \\
& =-\mathbf{s}_{j}^{T} \mathbf{G}_{i}\left(-\tilde{\mathbf{s}}_{i}^{\prime} \mathbf{L}_{i}+\mathbf{s}_{i}^{\prime} \mathbf{p}_{i}^{T}\right)^{T} \\
& =-\mathbf{s}_{j}^{T} \mathbf{G}_{i} \mathbf{L}_{i}^{T} \mathbf{s}_{i}^{\prime} \\
& =-\mathbf{s}_{j}^{T} \mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime} \tag{2}
\end{align*}
\]

This result agrees with the coefficient of \(\boldsymbol{\omega}_{i}^{\prime}\) in Eq. 1. The time derivative of Eq. 1 yields
\[
\begin{align*}
\ddot{\boldsymbol{\Phi}}^{(n 1,1)} \equiv\left[-\mathbf{s}_{j}^{T} \mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime},-\mathbf{s}_{i}^{T} \mathbf{A}_{j} \tilde{\mathbf{s}}_{j}^{\prime}\right]\left[\begin{array}{c}
\dot{\boldsymbol{\omega}}_{i}^{\prime} \\
\dot{\boldsymbol{\omega}}_{j}^{\prime}
\end{array}\right]+ & \left(\dot{\mathbf{s}}_{j}^{T} \mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime}-\mathbf{s}_{j}^{T} \dot{\mathbf{A}}_{i} \tilde{\mathbf{s}}_{i}^{\prime}\right) \boldsymbol{\omega}_{i}^{\prime} \\
& +\left(-\dot{\mathbf{s}}_{i}^{T} \mathbf{A}_{j} \tilde{\mathbf{s}}_{j}^{\prime}-\mathbf{s}_{i}^{T} \dot{\mathbf{A}}_{j} \tilde{\mathbf{s}}_{j}^{\prime}\right) \boldsymbol{\omega}_{j}^{\prime}=0 \tag{3}
\end{align*}
\]

The last two terms of Eq. 3 can be simplified to obtain
\[
\begin{align*}
\boldsymbol{\gamma}^{\#(n 1,1)} & \equiv \dot{\mathbf{s}}_{j}^{T} \mathbf{A}_{i} \tilde{\mathbf{s}}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime}+\mathbf{s}_{j}^{T} \dot{\mathbf{A}}_{i} \tilde{\mathbf{s}}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime}+\dot{\mathbf{s}}_{i}^{T} \mathbf{A}_{j} \tilde{\mathbf{s}}_{j}^{\prime} \boldsymbol{\omega}_{j}^{\prime}+\mathbf{s}_{i}^{T} \dot{\mathbf{A}}_{j}^{T} \tilde{\mathbf{s}}_{j}^{\prime} \boldsymbol{\omega}_{j} \\
& =\dot{\mathbf{s}}_{j}^{T} \tilde{\mathbf{s}}_{i} \boldsymbol{\omega}_{j}+\mathbf{s}_{j}^{T} \tilde{\boldsymbol{\omega}}_{i} \tilde{\mathbf{s}}_{i} \boldsymbol{\omega}_{i}+\dot{\mathbf{s}}_{i}^{T} \tilde{\mathbf{s}}_{j} \boldsymbol{\omega}_{j}+\mathbf{s}_{i}^{T} \tilde{\boldsymbol{\omega}}_{j} \tilde{\mathbf{s}}_{j} \boldsymbol{\omega}_{j} \\
& =-2 \dot{\mathbf{s}}_{j}^{T} \dot{\mathbf{s}}_{i}-\mathbf{s}_{j}^{T} \tilde{\boldsymbol{\omega}}_{i} \dot{\mathbf{s}}_{i}-\mathbf{s}_{i}^{T} \tilde{\boldsymbol{\omega}}_{i} \mathbf{s}_{j} \\
& =-2 \dot{\mathbf{s}}_{i}^{T} \dot{\mathbf{s}}_{j}+\dot{\mathbf{s}}_{i}^{T} \tilde{\boldsymbol{\omega}}_{i} \mathbf{s}_{j}+\dot{\mathbf{s}}_{j}^{T} \tilde{\boldsymbol{\omega}}_{j} \mathbf{s}_{i} \tag{4}
\end{align*}
\]

This example illustrates how the modified Jacobian matrix and vector \(\boldsymbol{\gamma}^{*}\) can be calculated. Table 11.1 shows the components of the Jacobian matrix and vector \(\gamma^{*}\) for some of the most common constraints. This table provides sufficient information to assemble in the form of Eq. 11.49 the complete set of equations of motion for mechanical systems with the more commonly used constraints. Numerical methods for solving these equations are discussed in Chap. 13.

TABLE 11.1 Components in the Expansion of the Most Common Constraints \({ }^{13}\)
\begin{tabular}{|c|c|c|c|c|c|}
\hline Ф & \(\boldsymbol{\Phi}_{\mathbf{r}_{i}}^{(m)}\) & \({ }^{\frac{1}{2}} \boldsymbol{\Phi}_{\mathbf{p}_{i}}^{(m)} \mathbf{L}_{i}^{T}\) & \(\boldsymbol{\Phi}_{\mathrm{r}_{\mathrm{j}}}^{(m)}\) & \({ }^{\frac{1}{2}} \boldsymbol{\Phi}_{\mathbf{p}_{j}^{(m)} \mathbf{L}_{j}^{T}}\) & \(\gamma^{*}\) \\
\hline \(\Phi^{(n 1,1)}\) & \(0^{T}\) & \(-\mathbf{s}_{j}^{T} \tilde{S}_{i} \mathbf{A}_{i}\) & \(\boldsymbol{0}^{T}\) & \(-s_{1} \bar{S}_{j} \mathbf{A}_{j}\) & \(-2 \mathbf{s}^{T} \dot{\mathbf{j}}_{j}+\dot{\mathbf{s}}_{i}^{T} \tilde{\omega}_{i} \mathbf{s}_{j}+\dot{\mathbf{s}}_{j}^{T} \tilde{\omega}_{j} \mathbf{s}_{i}\) \\
\hline \(\Phi^{(n 2,1)}\) & \(-\mathbf{s}_{i}^{T}\) & \(-\left(\mathbf{d}+\mathbf{s}_{l}^{B}\right)^{T_{\overline{\mathbf{S}}_{i}}} \mathbf{A}_{i}\) & \(\mathbf{s}_{i}^{T}\) & \(-\mathbf{s}_{i}^{T} \stackrel{\Sigma}{j}_{j}^{B} \mathbf{A}_{j}\) & \(-2 \dot{\mathbf{d}}^{T} \dot{\mathbf{s}}_{i}-\mathbf{d}^{T} \tilde{\omega}_{i} \dot{\mathbf{s}}_{i}+\mathbf{s}_{i}^{T}\left(\tilde{\omega}_{i} \dot{\mathbf{s}}_{i}^{B}-\tilde{\boldsymbol{\omega}}_{j} \dot{\mathbf{s}}_{j}^{B}\right)\) \\
\hline \(\Phi^{(p 1,2)}\) & 0 & \(\overline{\mathbf{s}}_{j} \bar{s}_{i} \mathbf{A}_{i}\) & 0 & \(-\overline{\mathbf{s}}_{i} \tilde{\mathbf{S}}_{j} \mathbf{A}_{j}\) & \(-2 \tilde{\dot{s}}_{i} \dot{\mathbf{s}}_{j}+\tilde{\mathbf{s}}_{j} \tilde{\omega}_{i} \dot{\mathbf{s}}_{i}-\tilde{\mathbf{s}}_{i} \tilde{\boldsymbol{\omega}}_{j} \dot{s}_{j}\) \\
\hline \(\boldsymbol{\Phi}^{(p 2,2)}\) & \(-\tilde{\mathbf{s}}_{i}\) & \(\left(\overline{\mathbf{s}}_{i} \tilde{i}_{i}^{B}+\tilde{\mathbf{d}} \tilde{\mathbf{s}}_{i}\right) \mathbf{A}_{i}\) & \(\overline{\mathbf{s}}_{i}\) & \(-\overline{\mathbf{s}}_{\bar{i}} \bar{S}_{j}^{\mathbf{B}} \mathbf{A}_{j}\) & \(-2 \tilde{\mathbf{s}}_{i} \dot{\mathbf{d}}+\tilde{\mathbf{s}}_{i}\left(\bar{\omega}_{i} \dot{\mathbf{s}}_{i}^{B}-\tilde{\omega}_{j} \dot{s}_{j}^{B}\right)+\tilde{\mathbf{d}} \bar{\omega}_{i} \dot{s}_{i}\) \\
\hline \(\boldsymbol{\Phi}^{(5,3)}\) & I & \(-\overline{\mathbf{S}}_{\boldsymbol{i}} \mathrm{A}_{i} \mathrm{~A}_{i}\) & -I & \(\bar{s}_{j}^{P} \mathbf{A}_{j}\) & \(-\bar{\omega}_{i} \dot{\mathbf{s}}_{i}^{P}+\bar{\omega}_{j} \dot{\mathbf{s}}_{j}^{P}\) \\
\hline \(\Phi^{(s-s, 1)}\) & \(-2 \mathbf{d}^{T}\) & \(2 \mathbf{d}^{T} \overline{\mathbf{s}}_{i} \mathbf{A}_{i}\) & \(2 \mathrm{~d}^{T}\) & \(-2 \mathbf{d}^{T} \mathbf{S}_{j}^{P} \mathbf{A}_{j}\) & \(-2 \mathbf{d}^{T} \mathbf{d}+2 \mathbf{d}^{T}\left(\tilde{\omega}_{i} \mathbf{s}_{i}^{P}-\tilde{\omega}_{j} \mathbf{S}_{j}^{P}\right)\) \\
\hline
\end{tabular}

\section*{PROBLEMS}
11.1 Show that matrix \(J_{i}^{*}\) is singular.
11.2 Express matrix \(\mathbf{J}_{i}^{*}\) in terms of \(\mathbf{J}_{i}\) and \(\mathbf{G}_{i}\).
11.3 From the rotational equations of motion given in Eq. 11.6,
(a) Determine the inverse of the coefficient matrix \(\left[\begin{array}{ll}\mathbf{J}^{*} & \mathbf{p} \\ \mathbf{p}^{T} & 0\end{array}\right]\) i
(b) Solve Eq. 11.16 for vector \(\left[\dot{\mathbf{p}}^{T}, \sigma\right]_{i}^{T}\)
(c) Show that \(\sigma_{i}\) obtained in part (b) is the same as that shown in Eq. 11.13.
11.4 Solve Eq. 11.17 for \(\ddot{\mathbf{p}}_{i}\).
11.5 The kinetic energy of a body is defined as
\[
T_{i}=\frac{1}{2} \dot{\mathbf{r}}_{i}^{T} \mathbf{N}_{i} \dot{\mathbf{r}}_{i}+\frac{1}{2} \boldsymbol{\omega}_{i}^{\prime T} \mathbf{J}_{i}^{\prime} \boldsymbol{\omega}_{i}^{\prime}
\]

Express the kinetic energy in terms of:
(a) \(\boldsymbol{\omega}_{i}\) and \(\mathbf{J}_{i}\)
(b) \(\dot{\mathbf{p}}_{i}\) and \(\mathbf{J}_{i}^{*}\)
11.6 The angular orientation of a body is given as \(\mathbf{p}_{i}=[0.5,0.7,-0.5,0.1]^{T}\). Point \(P\) on this body has the local coordinates \(\mathbf{s}_{i}^{P}=[1,-1,2]^{T}\). A force \(\vec{f}\) acts on this body and has local components \(\mathbf{f}_{i}^{\prime}=[3,-2,-1]^{T}\). Find the components of the moment of this force in the following forms:;
(a) \(\mathbf{n}_{i}^{\prime}\)
(b) \(\mathbf{n}_{i}\)
(c) \(\mathrm{n}_{i}^{*}\)
11.7 Two bodies are connected to each other by a spherical joint as shown in Fig. P.11.7. In addition to the gravitational force, two external moments, \(\vec{n}_{1}\) and \(\vec{n}_{2}\), and one external force, \(\vec{f}_{\mathrm{i}}\), act on the bodies, where \(\vec{f}_{1}\) is parallel to the \(y\) axis.
(a) Write the equations of motion for the bodies in terms of angular accelerations.
(b) Show the elements of the vector of forces in terms of the applied loads.
(c) From the equations of motion, show the components of the reaction forces acting at \(P\) on body \(i\) and body \(j\).


Figare P. 11.7
11.8 Verify the entries of the modified Jacobian matrix and vector \(\gamma^{\#}\) listed in Table 11.1.

\section*{12}

\section*{Numerical Methods}
for Ordinary

\section*{Differential Equations}

The dynamic analysis of a mechanical system requires the solution of the equations of motion. The equations of motion, planar and spatial, are either a set of differential equations or a set of mixed differential and algebraic equations. In general, the equations of motion must be solved numerically, although it might be possible to obtain a closedform solution to the equations of motion for highly simplified systems.

This chapter provides a brief review of numerical methods for solving ordinary differential equations. It is assumed that the reader has some background in the area of numerical methods.

\subsection*{12.1 INITIAL-VALUE PROBLEMS}

A first-order differential equation may be written as
\[
\begin{equation*}
\dot{y}=f(y, t) \tag{12.1}
\end{equation*}
\]

This equation has a family of solutions \(y(t)\). The choice of an initial value \(y^{0}\) serves to determine one of the solutions of the family. The initial value \(y^{0}\) could be defined for any value of \(t^{0}\), although it is often assumed that a transformation has been made so that \(t^{0}=0\). This does not affect the solution or methods used to approximate the solution.

If more than one dependent variable is involved, the problem then is to solve a system of first-order equations; e.g.,
\[
\begin{align*}
& \dot{y}_{1}=f_{1}\left(y_{1}, y_{2}, t\right) \\
& \dot{y}_{2}=f_{2}\left(y_{1}, y_{2}, t\right) \tag{12.2}
\end{align*}
\]

Given the initial values \(y_{1}^{0}\) and \(y_{2}^{0}\), if the functions \(f_{1}\) and \(f_{2}\) are regular; i.e., continuously differentiable, then there will be a unique solution of the system.

Any \(n\) th-order ordinary differential equation that can be written with the highestorder derivatives on the left-hand side may be written as a system of \(n\) first-order equations by defining \(n-1\) new variables. For example, the second-order equation
\[
\begin{equation*}
\ddot{y}_{1}=f\left(y_{1}, \dot{y}_{1}, t\right) \tag{12.3}
\end{equation*}
\]
can be written as the system
\[
\begin{align*}
& \dot{y}_{1}=y_{2} \\
& \dot{y}_{2}=f\left(y_{1}, y_{2}, t\right) \tag{12.4}
\end{align*}
\]

In discussing methods of solving initial-value problems, it is convenient to think of a single equation in the form of Eq. 12.1, although the same methods can also be applied to a system of equations. These methods involve a step-by-step process in which a sequence of discrete points \(t^{0}, t^{1}, t^{2}, \ldots\) is generated. The discrete points may have either constant or variable spacing \(h^{i}=t^{i+1}-t^{i}\), where \(h^{i}\) is the step size for any discrete point \(t^{i}\). At each point \(t^{i}\), the solution \(y\left(t^{i}\right)\) is approximated by a number \(y^{i}\).

Since no numerical method is capable of finding \(y\left(t^{i}\right)\) exactly, the quantity
\[
\begin{equation*}
\left.\boldsymbol{\varepsilon}^{i}=\mid y\left(t^{i}\right)-y^{i}\right) \mid \tag{12.5}
\end{equation*}
\]
represents the total error at \(t=t^{i}\). The total error consists of two components: a truncation error and a round-off error. The truncation error depends on the nature of the numerical algorithm used in computing \(y^{i}\). The round-off error is due to the finite word length in a computer. In the rest of the text, when the term numerical error is used, we mean truncation error unless stated otherwise.

Although there exist many algorithms for solving initial-value problems, most of them are based on two basic approaches: Taylor series expansion and polynomial approximation. The objective of Sec. 12.2 and 12.3 is to present general formulas for some of these methods, without proof. Interested readers may refer to numerous textbooks for the derivation of details and for error analysis for these methods. Unless otherwise stated, in order to simplify matters, a uniform step size \(h^{i}=h\) is assumed in formulation of the algorithms.

\subsection*{12.2 TAYLOR SERIES ALGORITHMS \({ }^{2}\)}

Assume that \(y(t)\) is the exact solution to the initial-value problem
\[
\begin{equation*}
\dot{y}=f(y, t) \tag{12.6}
\end{equation*}
\]
where
\[
\begin{equation*}
y^{0}=y\left(t^{0}\right) \tag{1.7}
\end{equation*}
\]

If \(y(t)\) is expanded in a Taylor series about \(t=t^{i}\) and the expansion is evaluated at \(t=t^{i+1}\), it is found that
\[
\begin{equation*}
y\left(t^{i+1}\right)=y\left(t^{i}\right)+\frac{\dot{y}\left(t^{i}\right)}{1!} h+\frac{\ddot{y}\left(t^{i}\right)}{2!} h^{2}+\text { higher-order terms } \tag{12.8}
\end{equation*}
\]

Substituting Eq. 12.6 into Eq. 12.8 yields
\[
\begin{equation*}
y\left(t^{i+1}\right)=y\left(t^{i}\right)+h f\left(y^{i}, t^{i}\right)+\frac{h^{2}}{2!} \dot{f}\left(y^{i}, t^{i}\right)+\text { higher-order terms } \tag{12.9}
\end{equation*}
\]

The first-order Taylor algorithm, also known as the forward Euler algorithm, is obtained by eliminating \(f\left(y^{i}, t^{i}\right)\) and the higher-order terms in Eq. 12.9:
\[
\begin{equation*}
y^{i+1}=y^{i}+h f\left(y^{i}, t^{i}\right) \tag{12.10}
\end{equation*}
\]

Truncating Eq. 12.9 of all but the first two terms makes Eq. 12.10 an approximate solution to the initial-value problem of Eqs. 12.6 and 12.7.

The second-order Taylor algorithm is obtained by truncating Eq. 12.9 of only the higher-order terms, to obtain
\[
\begin{equation*}
y^{i+1}=y^{i}+h f\left(y^{i}, t^{i}\right)+\frac{h^{2}}{2!} \dot{f}\left(y^{i}, t^{i}\right) \tag{12.11}
\end{equation*}
\]

The term \(\dot{f}\left(y^{i}, t^{i}\right)\) can be expressed as follows:
\[
\begin{align*}
\dot{f}\left(y^{i}, t^{i}\right) & =\frac{\partial f}{\partial y} \dot{y}+\frac{\partial f}{\partial t} & \text { at } t=t^{i} \\
& =\frac{\partial f}{\partial y} f+\frac{\partial f}{\partial t} & \text { at } t=t^{i} \tag{12.12}
\end{align*}
\]

Similarly, higher-order Taylor algorithms may be derived. However, the higher the order, the more derivative terms of \(f(y, t)\) with respect to \(t\) must be evaluated. This is a major drawback of the Taylor algorithms. Consequently, they are seldom used except for the lower-order algorithms.

\subsection*{12.2.1 Runge-Kutta Algorithms \({ }^{2}\)}

The Runge-Kutta algorithms obviate the need for evaluating the partial derivatives and still retain the same order of accuracy as the Taylor algorithms. A second-order RungeKutta algorithm is stated as
\[
\begin{equation*}
y^{i+1}=y^{i}+h g \tag{12.13}
\end{equation*}
\]
where
\[
\begin{equation*}
g=(1-a) f\left(y^{i}, t^{i}\right)+a f\left(y^{i}+\frac{h}{2 a} f\left(y^{i}, t^{i}\right), t^{i}+\frac{h}{2 a}\right) \tag{12.14}
\end{equation*}
\]

Note that \(a \neq 0\) is a free parameter. Consequently, an entire family of second-order Runge-Kutta algorithms can be derived by assigning different values to \(a\). One common choice is the modified trapezoidal algorithm, in which \(a=\frac{1}{2}\); then,
\[
\begin{equation*}
y^{i+1}=y^{i}+\frac{h}{2} f\left(y^{i}, t^{i}\right)+\frac{h}{2} f\left(y^{i}+h f\left(y^{i}, t^{i}\right), t^{i}+h\right) \tag{12.15}
\end{equation*}
\]

Another common choice is the modified Euler-Cauchy algorithm, in which \(a=1\) :
\[
\begin{equation*}
y^{i+1}=y^{i}+h f\left(y^{i}+\frac{h}{2} f\left(y^{i}, t^{i}\right), t^{i}+\frac{h}{2}\right) \tag{12.16}
\end{equation*}
\]

For a larger step size and for greater accuracy, the fourth-order Runge-Kutta algorithm is most widely used. This algorithm is given by
\[
\begin{equation*}
y^{i+1}=y^{i}+h g \tag{12.17}
\end{equation*}
\]
where
\[
\begin{aligned}
& g=\frac{1}{6}\left(f_{1}+2 f_{2}+2 f_{3}+f_{4}\right) \\
& f_{1}=f\left(y^{i}, t^{i}\right) \\
& f_{2}=f\left(y^{i}+\frac{h}{2} f_{1}, t^{i}+\frac{h}{2}\right) \\
& f_{3}=f\left(y^{i}+\frac{h}{2} f_{2}, t^{i}+\frac{h}{2}\right) \\
& f_{4}=f\left(y^{i}+h f_{3}, t^{i}+h\right)
\end{aligned}
\]

Since the algorithm is a fourth-order one, the truncation error will remain relatively small even for a relatively large step size. The major disadvantage of this algorithm is that the function \(f(y, t)\) must be evaluated four times at each time step. In addition, the values of the function are not used in any subsequent computations. Hence, this algorithm is not as computationally efficient as some of the multistep algorithms presented in Secs. 12.3 through 12.3.3.

\subsection*{12.2.2 A Subroutine for a Runge-Kutta Algorithm}

The fourth-order Runge-Kutta algorithm in the preceding section is represented here in the form of a subroutine that can be embedded in a program to solve a set of ordinary differential equations. This subroutine is written in its simplest form and can be modified easily. \({ }^{\dagger}\)

Subroutine RUNGK4. The argument parameters in this subroutine are:
\begin{tabular}{ll} 
H & Time step \\
NSTEP & Number of time steps \\
N & \begin{tabular}{l} 
Number of dependent variables (same as the number of differen- \\
tial equations)
\end{tabular} \\
Y & An N-vector of dependent variables \(\mathbf{y}\) \\
F & An N-vector which upon return will contain \(\dot{\mathbf{y}}=\mathbf{f}(\mathbf{y}, t)\) \\
F 1, F2, F3, & \\
F 4, YY & N-vectors of working arrays
\end{tabular}

Subroutine RUNGK4 is as follows:
```

SUBROUTINE RUNGK4 (T,H,NSTEP,N,Y,F,F1,F2,F3,F4,YY)
DIMENSION Y(N),F(N),F1(N),F2(N),F3(N),F4(N),YY(N)
HH=0.5*H
TS=T
WRITE (1,200)
DO 100 I=1,NSTEP
WRITE (1,210) T, (Y(J), J=1,N)
CALL DIFEQN (T,N,Y,F)
DO 10 J=1,N

```

\footnotetext{
\({ }^{\dagger}\) The subroutine RUNG4 in the program DAP of Chap. 10 is a slightly modified version of this subroutine.
}
```

10 F1(J)=H*F(J)
TT=T+HH
DO 20 J=1,N
YY(J)=Y(J)+0.5*F1(J)
CALL DIFEQN (TT,N,YY,F)
DO 30 J=1,N
F2(J)=H*F(J)
YY(J)=Y(J)+0.5*F2(J)
CALL DIFEQN (TT,N,YY,F)
Tr=T+H
DO 40 J=1,N
F3(J)=H*F(J)
YY(J)=Y(J)+F3(J)
CALL DIFEQN (TT,N,YY,F)
T=TS+H*FLOAT( I )
DO 50 J=1,N
F4(J)=H*F(J)
Y(J)=Y(J)+(F1(J)+2.0*F2(J)+2.0*F3(J)+F4(J))/6.0
5 0
100 CONTINUE
200 FORMAT (5X,' TIME Y')
210 FORMAT (5X,4F10.6)
RETURN
END

```

This subroutine is written in a form that will handle one or more first-order differential equations. It calls subroutine DIFEQN for evaluating \(\dot{\mathbf{y}}=\mathbf{f}(\mathbf{y}, t)\).

\section*{Example 12.1}

Write a computer program, making use of subroutine RUNGK4, to solve \(\dot{y}=-y^{2}\) with the initial condition \(y^{0}=1\).
Solution A main program and two subroutines INITL and DIFEQN for this problem are:
```

C*****MAIN PROGRAM**
DIMENSION A(80)
C.....Data
N=1
T=0.0
H=0.1
NSTEP=50
C.....Pointers
N1=1
N}2=\textrm{N}1+\textrm{N
N3=N2+N
N4=N3+N
NS=N4+N
N6=N5+N
N7=N6+N
C.....Initial Conditions
CALL INITL (N,A(N1))
C....Integration
CALL RUNGK4 (T,H,NSTEP,N,A(N1),A(N2),A(N3),A(N4),A(N5),
+ A(N6),A(N7))
STOP
END

```
        SUBROUTINE INITL (N,Y)
    DIMENSION \(\mathrm{Y}(\mathrm{N})\)
    \(\mathrm{Y}(1)=1.0\)
        RETURN
    END
```

SUBROUTINE DIFEQN (T,N,Y,F)
DIMENSION Y(N),F(N)
F(1)=-Y(1)**2
RETURN
END

```

Subroutine INITL is used to specify the initial conditions. The result of this numerical computation can be compared against the exact solution \(y=1 /(1+t)\).

\section*{Example 12.2}

The equations of motion for the spring-mass system shown in the schematic diagram are:

\[
\begin{aligned}
& m_{1} \ddot{x}_{1}=-k_{1}\left(x_{1}-l_{1}^{0}\right)-d_{1} \dot{x}_{1}+k_{2}\left(x_{2}-x_{1}-l_{2}^{0}\right) \\
& m_{2} \ddot{x}_{2}=-k_{2}\left(x_{2}-x_{1}-l_{2}^{0}\right)+k_{3}\left(d-x_{2}-l_{3}^{0}\right)-d_{3} \dot{x}_{2}
\end{aligned}
\]

Solve these equations numerically for
\[
\begin{gathered}
d=3, \quad l_{1}^{0}=l_{2}^{0}=l_{3}^{0}=1, \quad k_{1}=k_{2}=k_{3}=100 \\
m_{1}=m_{2}=4, \quad d_{1}=d_{3}=40
\end{gathered}
\]
and the initial conditions
\[
x_{1}^{0}=1.0, \quad \dot{x}_{1}^{0}=0, \quad x_{2}^{0}=1.9, \quad \dot{x}_{2}^{0}=0
\]

Solution The second-order differential equations can be converted to first-order equations by defining four new variables:
\[
\begin{aligned}
& \dot{y}_{1}=y_{2} \\
& \dot{y}_{2}=-\frac{k_{1}}{m_{1}}\left(y_{1}-l_{1}^{0}\right)-\frac{d_{1}}{m_{1}} y_{2}+\frac{k_{2}}{m_{1}}\left(y_{3}-y_{1}-l_{2}^{0}\right) \\
& \dot{y}_{3}=y_{4} \\
& \dot{y}_{4}=-\frac{k_{2}}{m_{2}}\left(y_{3}-y_{1}-l_{2}^{0}\right)+\frac{k_{3}}{m_{2}}\left(d-y_{3}-l_{3}^{0}\right)-\frac{d_{3}}{m_{2}} y_{4}
\end{aligned}
\]

In the main program, \(N\) is set to \(N=4\), and the INITL and DIFEQN subroutines are written as follows:

> SUBROUTINE INITL (N,Y)
> DIMENSION \(\mathrm{Y}(\mathrm{N})\)
> \(\mathrm{Y}(1)=1.0\)
> \(\mathrm{Y}(2)=0.0\)
> \(\mathrm{Y}(3)=1.9\)
> \(\mathrm{Y}(4)=0.0\)
> RETURN
> END
```

SUBROUTINE DIFEQN (T,N,Y,F)
DIMENSION Y(N),F(N)
DATA A1/25.0/,A2/10.0/,A3/25.0/,A4/25.0/,A5/10.0/
DATA EL01/1.0/,EL02/1.0/,ELO3/1.0/,D/3.0/
F(1) = Y(2)
F(2) = -A1*(Y(1)-EL01)-A2*Y(2)+A3*(Y(3)-Y(1)-ELO2)
F(3)=Y(4)
F(4)= -A3*(Y(3)-Y(1)-EL02)+A4*(D-Y(3)-EL03)-A5*Y(4)
RETURN
END

```

The result of this computation is shown in the accompanying graph for \(x_{1}\) and \(d-x_{2}\) plotted versus time.


\subsection*{12.3 POLYNOMIAL APPROXIMATION \({ }^{2}\)}

Any algorithm that is capable of calculating the exact value \(y\left(t^{i+1}\right)\) for an initial-value problem that has an exact solution in the form of a \(k\) th-degree polynomial is called a \(n u\) merical integration formula of order \(k\). Of course, if the exact solution is not a polynomial, a numerical integration formula will generally give only an approximate value \(y^{i+1}\) and not the exact value \(y\left(t^{i+1}\right)\). In view of a classical theorem that asserts that any continuous function can be approximated arbitrarily within any closed interval by a polynomial of sufficiently high degree, it is clear that even if the solution is not a polynomial, a numerical integration formula of sufficiently high order can, in principle, be used to calculate \(y\left(t^{i+1}\right)\) to any desired accuracy. In practice, however, the amount of computation and round-off error increases with the order of the integration formula and only orders of \(k<10\) are of practical value.

In contrast to the procedure in the Taylor and Runge-Kutta algorithms, information from previous time steps is utilized in most numerical integration formulas to compute \(y^{i+1}\). A numerical integration formula is generally of the following form:
\[
\begin{align*}
y^{i+1}= & a_{0} y^{i}+a_{1} y^{i-1}+\cdots+a_{p} y^{i-p} \\
& +h\left[b_{-1} f\left(y^{i+1}, t^{i+1}\right)+b_{0} f\left(y^{i}, t^{i}\right)+\cdots+b_{p} f\left(y^{i-p}, t^{i-p}\right)\right] \tag{12.18}
\end{align*}
\]
where \(a_{0}, a_{1}, \ldots, a_{p}, b_{-1}, b_{0}, b_{1}, \ldots, b_{p}\) are \(2 p+3\) coefficients that are to be determined such that, if the exact solution is a polynomial and if the previously calculated values \(y^{i}, y^{i-1}, \ldots, y^{i-p}\) and \(f\left(y^{i}, t^{i}\right), f\left(y^{i-1}, t^{i-1}\right), \ldots, f\left(y^{i-p}, t^{i-p}\right)\) are assumed to be exact, then Eq. 12.18 gives the exact value of \(y^{i+1}\). A numerical integration algorithm with \(p \geq 1\) is called a multistep algorithm, in contrast to the Taylor and Runge-Kutta algorithms, which are single-step algorithms.

Note that Eq. 12.18 defines \(y^{i+1}\) only implicitly, since the unknown \(y^{i+1}\) appears on both sides of the equation. Thus, algorithms with \(b_{-1} \neq 0\) are called implicit algorithms. If \(b_{-1}=0\), the algorithm is an explicit algorithm, since the unknown \(y^{i+1}\) does not appear on the right side of the formula. Taylor and Runge-Kutta algorithms can be classified as explicit algorithms.

\subsection*{12.3.1 Explicit Multistep Algorithms \({ }^{2}\)}

Explicit multistep algorithms known as Adams-Bashforth algorithms are obtained by setting
\[
\begin{align*}
p & =k-1 \\
a_{1} & =a_{2}=\ldots=a_{k-1}=0  \tag{12.19}\\
b_{-1} & =0
\end{align*}
\]
in Eq. 12.18, where \(k\) is the degree of the polynomial. Table 12.1 shows four formulas for first- through fourth-order Adams-Bashforth algorithms. Examination of Table 12.1 shows that the \(k\) th-order Adams-Bashforth algorithm requires \(k\) starting values \(y^{i}, y^{i-1}, \ldots, y^{i-k+1}\).

TABLE 12.1 Adams-Bashforth Algorithms
\begin{tabular}{cl}
\hline Order & \multicolumn{1}{c}{\(y^{i+1}=\)} \\
\hline 1st & \(y^{i}+h f^{i}\) \\
2nd & \(y^{i}+h\left(\frac{3}{2} f^{i}-\frac{1}{2} f^{i-1}\right)\) \\
3rd & \(y^{i}+h\left(\frac{23}{12} f^{i}-\frac{16}{12} f^{i-1}+\frac{5}{12} f^{i-2}\right)\) \\
4th & \(y^{i}+h\left(\frac{54}{24} f^{i}-\frac{59}{24} f^{i-1}+\frac{37}{24} f^{i-2}-\frac{9}{24} f^{i-3}\right)\) \\
where & \(f^{i \equiv f\left(y^{i}, t^{i}\right)}\) \\
& \(f^{i-j}=f\left(y^{i-j}, t^{i-j}\right), \quad j=1,2,3\) \\
\hline
\end{tabular}

\subsection*{12.3.2 Implicit Multistep Algorithms \({ }^{2}\)}

Implicit multistep algorithms known as Adams-Moulton algorithms are obtained by setting
\[
\begin{align*}
& p=k-2  \tag{12.20}\\
& a_{1}=a_{2}=\ldots=a_{k-2}=0
\end{align*}
\]
in Eq. 12.18, where \(k\) is the degree of the polynomial. Table 12.2 shows four formulas for first- through fourth-order Adams-Mouton algorithms. Examination of Table 12.2

TABLE 12.2 Adams-Moulton Algorithms
\begin{tabular}{cl}
\hline Order & \multicolumn{1}{c}{\(y^{i+1}=\)} \\
\hline 1st & \(y^{i}+h f^{i+1}\) \\
2nd & \(y^{i}+h\left(\frac{1}{2} f^{i+1}+\frac{1}{2} f^{i}\right)\) \\
3rd & \(y^{i}+h\left(\frac{5}{12} f^{i+1}+\frac{8}{12} f^{i}-\frac{1}{12} f^{i-1}\right)\) \\
4th & \(y^{i}+h\left(\frac{9}{24} f^{i+1}+\frac{1}{24} f^{i}-\frac{5}{24} f^{i-1}+\frac{1}{24} f^{i-2}\right)\) \\
where & \(f^{i+1}=f\left(y^{i+1}, t^{i+1}\right)\) \\
& \(f^{i}=f\left(y^{i}, t^{i}\right)\) \\
& \(f^{i-j=f\left(y^{i-j}, t^{i-j}\right), \quad j=1,2}\) \\
\hline
\end{tabular}
shows that the \(k\) th-order Adams-Moulton algorithm requires only \(k-1\) starting values \(y^{i}, y^{i-1}, \ldots, y^{i-k+2}\). These formulas, if employed by themselves, are solved iteratively. In every time step, an initial estimate is given for \(y^{i+1}\), and then \(f\left(y^{i+1}, t^{i+1}\right)\) is evaluated. These values for \(y\) and \(f\) substituted in the formula yields an improved value for \(y^{i+1}\), and the sequence is repeated until very little change in \(y^{i+1}\) is observed. The number of iterations to achieve convergence on \(y^{i+1}\) depends on the estimated value of \(y^{i+1}\).

\subsection*{12.3.3 Predictor-Corrector Algorithms}

Consider the implicit numerical integration algorithms in Sec. 12.3.2. In these algorithms, an estimate of \(y^{i+1}\) is required to start the iteration of the formula. In order to obtain a relatively good estimate for \(y^{i+1}\), an explicit formula can be used. For example, consider the fourth-order Adams-Moulton formula of Table 12.2 and the third-order Adams-Bashforth formula of Table 12.1. Both formulas require values of \(f(y, t)\) at \(t^{i}\), \(t^{i-1}\), and \(t^{i-2}\). If the third-order explicit formula is employed, a good approximation on \(y^{i+1}\) can be obtained. This step is known as a predictor step. Then, the fourth-order implicit formula is employed to correct the predicted value of \(y^{i+1}\). This step is known as a corrector step. Sometimes an algorithm may iterate on the corrector step.

In most predictor-corrector algorithms, if a \(k\) th-order implicit formula is used as the corrector, a \(k-1\) st-order explicit formula is used as the predictor. Although it is possible to employ lower-order Taylor or Runge-Kutta formulas as predictors, numerically it is more efficient to stay with Admas-Bashforth formulas for prediction.

\subsection*{12.3.4 Methods for Starting Multistep Algorithms \({ }^{2}\)}

In contrast to single-step algorithms, multistep numerical integration algorithms are not self-starting, since initially only \(y^{0}\) and \(t^{0}\) are given. For example, it suffices to consider the simpler case in which \(b_{-1}=0\) in Eq. 12.18 and write out \(y^{1}\) explicitly as follows:
\[
\begin{align*}
y^{1}=a_{0} y^{0}+a_{1} y^{-1} & +\ldots+a_{p} y^{-p} \\
& +h\left[b_{0} f\left(y^{0}, t^{0}\right)+b_{1} f\left(y^{-1}, t^{-1}+\ldots+b_{p} f\left(y^{-p}, t^{-p}\right)\right]\right. \tag{12.21}
\end{align*}
\]

Equation 12.21 shows that the values \(y^{-1}, y^{-2}, \ldots, y^{-p}\) must be given, in addition to \(y^{0}\) and \(t^{0}\), in order to compute \(y^{1}\). In general, to compute \(y^{i+1}\), the \(p+1\) preceding values of \(y\) are needed, assuming a uniform step size \(h\). To obtain these values, a single-step algorithm must be used at least \(p+1\) times before a multistep algorithm can be ini-
tiated. Because of its high degree of accuracy, the fourth-order Runge-Kutta algorithm is frequently used to provide these initiating values.

Efficient and accurate numerical algorithms for solving initial-value problems are almost always a combination of single-step and multistep algorithms, with the former used only to obtain the starting values for initiating the latter. Multistep algorithms are used to compute the remaining points, because they are often computationally more efficient and with them the propagation of both truncation and round-off errors can be more easily controlled.

\subsection*{12.4 ALGORITHMS FOR STIFF SYSTEMS \({ }^{2}\)}

A stiff system is referred to as any initial-value problem in which the complete solution consists of fast and slow components. Technically, when the eigenvalues are widely spread, the system is said to be stiff. If a numerical solution is to display the entire transient response, integration must be performed over a relatively long time interval in order to cover the slow component(s) of the response. Furthermore, in order to capture the fast component(s) of the response and keep the numerical error within bounds, the selected step size must be relatively small. It is clear that carrying integration with small t:me steps over a long time interval can make the computer time, even for a small problem, prohibitive or unrealistic. A family of formulas that allow relatively large time steps and that guarantee stability and bounded numerical error is available. These multistep formulas are known as Gear algorithms.

The \(k\) th-order Gear algorithm is an implicit formula of the form
\[
\begin{equation*}
y^{i+1}=a_{0}(k) y^{i}+a_{1}(k) y^{i-1}+\cdots+a_{k-1}(k) y^{i-k+1}+h b_{-1}(k) f\left(y^{i+1}, t^{i+1}\right) \tag{12.22}
\end{equation*}
\]
where the designation \(a_{j}(k)\) emphasizes each coefficient's dependence on the order \(k\). The \(k+1\) coefficients \(a_{0}(k), \ldots, a_{k-1}(k)\), and \(b_{-1}(k)\) are to be determined so that Eq. 12.22 is exact for all polynomial solutions of degree \(k\). Table 12.3 shows four formulas for first- through fourth-order Gear algorithms. Examination of Table 12.3 shows that the \(k\) th-order Gear algorithm requires \(k\) starting values \(y^{i}, y^{i-1}, \ldots, y^{i-k+1}\).

TABLE 12.3 Gear Algorithms
\begin{tabular}{cl}
\hline Order & \(y^{i+1}=\) \\
\hline 1st & \(y^{i}+h f^{i+1}\) \\
2nd & \(\frac{4}{3} y^{i}-\frac{1}{3} y^{i-1}+\frac{2}{3} h f^{i+1}\) \\
3rd & \(\frac{18}{11} y^{i}-\frac{9}{11} y^{i-1}+\frac{2}{11} y^{i-2}+\frac{6}{11} h^{i+1}\) \\
4th & \(\frac{48}{25} y^{i}-\frac{36}{25} y^{i-1}+\frac{16}{25} y^{i-2}-\frac{3}{25} y^{i-3}+\frac{12}{25} h f^{i+1}\) \\
where & \(f^{i+1}=f\left(y^{i+1}, i^{i+1}\right)\) \\
\hline
\end{tabular}

Since the \(k\) th-order Gear algorithm is an implicit multistep algorithm, it is necessary to solve an implicit equation in each time step. The \(k\) th-order formula of Eq. 12.22 , for a system of equations
\[
\dot{\mathbf{y}}=\mathbf{f}(\mathbf{y}, t)
\]
can be recast into the form
\[
\begin{equation*}
\mathbf{y}^{i+1}-h b_{-1} \mathbf{f}\left(\mathbf{y}^{i+1}, t^{i+1}\right)-\sum_{j=0}^{k-1}\left(a_{j} \mathbf{y}^{i-j}\right)=\mathbf{0} \tag{12.23}
\end{equation*}
\]

Applying the Newton-Rhapson algorithm to Eq. 12.23 yields
\[
\begin{equation*}
\Delta \mathbf{y}^{i+1}=\left(\mathbf{I}-b_{-1} \frac{\partial \mathbf{f}^{i+1}}{\partial \mathbf{y}^{i+1}}\right)^{-1}\left[\mathbf{y}^{i+1}-h b_{-1} \mathbf{f}^{i+1}-\sum_{j=1}^{k-1}\left(a_{j} \mathbf{y}^{i-j}\right)\right] \tag{12.24}
\end{equation*}
\]
where \(\mathbf{I}\) is an identity matrix. Equation 12.24 is the Newton-Raphson corrector for implementing the Gear algorithm.

\subsection*{12.5 ALGORITHMS FOR VARIABLE ORDER AND STEP SIZE}

So far, it has been implicitly assumed that, given an initial-value problem, a numerical integration algorithm of certain "order" is selected and the order remains fixed during the entire integration process. Under this assumption, the step size for each time step may be optimized by choosing the largest possible value of \(h\) for which the truncation error remains bounded below the user-specified maximum allowable error, and for which the algorithm remains numerically stable. For large systems of equations, the amount of computation does not increase substantially when the order of the algorithm is increased. Consequently, it often turns out to be more efficient to vary both the order and the step size during each time step.

From a programming point of view, changing the order requires only selecting a set of coefficients that define the multistep algorithm of the desired order. Increasing (decreasing) the order would require an increase (decrease) in the number of coefficients, with a corresponding increase (decrease) in storage space. In most cases of practical interest, the order may vary from \(k=1\) to \(k=10\). Thus, enough "past" values must be stored that the highest-order algorithm can be implemented whenever called for. However, stored "past" values may not be needed if a lower-order algorithm is used. In any event, the unused values cost very little, since they require only a modest amount of storage space.

Unlike change of order, which requires little extra programming and computational effort, changing the step size could entail considerable additional computation time. Often, the previously stored "past" values corresponding to step size \(h\) must be interpolated to yield a set of transformed "past" values corresponding to the new step size \(h\).

\section*{PROBLEMS}
12.1 Solve the following types of problem with subroutine RUNGK4:
(a) A first-order differential equation
(b) A second-order differential equation
(c) A system of second-order differential equations.

Repeat each problem for different values of integration time steps and compare the results. If the exact solution to the problem is available, compare that against the numerical solutions.
12.2 Refer the Table 12.1 and develop subroutines for the following integration algorithms:
(a) The third-order Adams-Bashforth
(b) The fourth-order Adams-Bashforth
12.3 Refer to Table 12.2 and develop subroutines for the following integration algorithms:
(a) The third-order Adams-Moulton
(b) The fourth-order Adams-Moulton
12.4 Develop a predictor-corrector integration subroutine by employing a third-order AdamsBashforth algorithm and a fourth-order Adams-Moulton algorithm.
12.5 Compare the subroutines developed in Probs. 12.2 to 12.4 in terms of accuracy and computational efficiency by following a process similar to that stated in Prob. 12.1.
12.6 Refer to the software library of your computer and experiment with the available integration subroutines.
12.7 Find a computer program for a variable step/order predictor-corrector numerical integration algorithm. \({ }^{\dagger}\)
(a) Implement this program on your computer.
(b) Compare this algorithm with the subroutines developed in Probs. 12.2 to 12.4.
12.8 Experiment with numerical integration programs based on Gear algorithms. \({ }^{\ddagger}\) Compare these algorithms and those developed in Probs. 12.2 to 12.4 .

\footnotetext{
\({ }^{\dagger}\) An excellent variable step/order predictor-corrector algorithm can be found in L. F. Shampine, and M. K. Gordon, Computer Solution of Ordinary Differential Equations: The Initial Value Problem, W. H. Freeman, San Francisco, 1975.
\({ }^{\ddagger}\) Most scientific software libraries furnish numerical integration packages based on Gear algorithms. For more detailed discussion on these algorithms, refer to C. W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, N.J., 1971.
}

\section*{13}

\section*{Numerical Methods}
in

\section*{Dynamics}

In this chapter several algorithms for the numerical solution of the equations of motion are presented. These algorithms utilize the numerical methods given in Chap. 12 for solving ordinary differential equations. If a mechanical system does not have any kinematic joints, i.e., if it is an unconstrained system, then these algorithms can be employed directly. However, if a mechanical system contains kinematic joints, and if Cartesian coordinates are employed in deriving the equations of motion, then these numerical integration algorithms must be modified.

The techniques and algorithms that are discussed in this chapter can be applied to solve the equations of motion when they are derived either in Cartesian coordinates or in other coordinate systems, such as Lagrangian coordinates. If the Lagrangian coordinates describing the configuration of a system are the generalized coordinates (i.e., if the number of Lagrangian coordinates is equal to the number of degrees of freedom), then the equations of motion are ordinary differential equations with no algebraic constraints, regardless of the presence or the absence of any kinematic joints. If the number of Lagrangian coordinates is greater than the number of degrees of freedom, then the equations of motion are mixed algebraic-differential equations. This is the same type of equation as in the case of Cartesian coordinates for systems containing kinematic joints.

\subsection*{13.1 INTEGRATION ARRAYS}

A numerical solution to the equations of motion may be obtained by utilizing any commonly used numerical integration algorithm. These algorithms are useful in solving firstorder differential equations that take the form
\[
\begin{equation*}
\dot{\mathbf{y}}=\mathbf{f}(\mathbf{y}, t) \tag{13.1}
\end{equation*}
\]

If there are \(n\) second-order differential equations of motion, they can be converted to \(2 n\) first-order equations by defining the \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) arrays as follows:
\[
\mathbf{y}=\left[\begin{array}{c}
\text { Position coordinates }  \tag{13.2}\\
\text { Velocities }
\end{array}\right], \quad \dot{\mathbf{y}}=\left[\begin{array}{c}
\text { Velocities } \\
\text { Accelerations }
\end{array}\right]
\]

Although the arrangement of the elements in \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) is quite arbitrary, the two arrays must follow a similar order. For example, if the \(j\) th element of \(\mathbf{y}\) contains \(x_{i}\), then the \(j\) th element of \(\dot{\mathbf{y}}\) must contain \(\dot{x}_{i}\).

The process of numerical integration at time \(t=t^{i}\) can be interpreted by the following diagram:
\[
\begin{equation*}
\dot{\mathbf{y}}\left(t^{i}\right) \xrightarrow{\text { (integration) }} \mathbf{y}\left(t^{i}+\Delta t\right) \tag{13.3}
\end{equation*}
\]

In other words, velocities and accelerations at \(t=t^{i}\) yield coordinates and velocities at \(t=t^{i}+\Delta t\).

\subsection*{13.2 KINEMATICALLY UNCONSTRAINED SYSTEMS}

The equations of motion for \(b\) unconstrained bodies containing \(n\) coordinates are represented by
\[
\begin{equation*}
\mathbf{M} \ddot{q}=\mathbf{g} \tag{13.4}
\end{equation*}
\]

If there are no mathematical constraints \({ }^{\dagger}\) on the coordinates, then the number of degrees of freedom is also \(n\). A numerical solution to Eq. 13.4 can be found in the same manner as that shown in Example 12.2. In the following algorithm, called the direct integration algorithm (DI), arrays \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) are defined as follows:
\[
\mathbf{y}=\left[\begin{array}{c}
\mathbf{q} \\
\dot{\mathbf{q}}
\end{array}\right] \quad \dot{\mathbf{y}}=\left[\begin{array}{l}
\dot{\mathbf{q}} \\
\ddot{\mathbf{q}}
\end{array}\right]
\]

\section*{ALGORITHM DI-1}
(a) Main routine
(a.1) Specify initial conditions for \(\mathbf{q}\) and \(\dot{\mathbf{q}}\).
(a.2) Transfer the contents of \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) to vector \(\mathbf{y} \equiv\left[\mathbf{q}^{T}, \dot{\mathbf{q}}^{T}\right]^{T}\).
(a.3) Enter the numerical integration routine (NI).
(b) Numerical integration routine
(This routine solves initial-value problems of the form \(\dot{\mathbf{y}}=\mathbf{f}(\mathbf{y}, t)\) from an initial time \(t^{\circ}\) to a final time \(t^{e}\) )

\footnotetext{
\({ }^{\dagger}\) For an unconstrained system of bodies in spatial motion, there are no kinematic constraints; however, there is one mathematical constraint for each set of Euler parameters.
}
(b.1) In the process of numerical integration, \(\mathbf{f}(\mathbf{y}, t)\) must be evaluated. For this purpose enter a DIFEQN routine with known \(\mathbf{y}^{i}\) and \(t^{i}\) to determine \(\mathbf{f}\left(\mathbf{y}^{i}, t^{i}\right)\).
(c) DIFEQN routine
(c.1) Transfer the contents of \(\mathbf{y}\) to \(\mathbf{q}\) and \(\dot{\mathbf{q}}\).
(c.2) Evaluate \(\mathbf{M}\) (since \(\mathbf{M}\) is constant, it needs to be evaluated only once) and \(\mathbf{g}\).
(c.3) Solve Eq. 13.4 for \(\ddot{\mathbf{q}}\).
(c.4) Transfer the contents of \(\dot{\mathbf{q}}\) and \(\ddot{\mathbf{q}}\) to \(\dot{\mathbf{y}}\).
(c.5) Return.

During an integration time step, the routine DIFEQN is called several times. The contents of \(\mathbf{y}\) are changed automatically by the integration routine, according to the algorithm. For example, in the Runge-Kutta subroutine of Sec. 12.2.2, the subroutine DIFEQN is called four times in every integration time step. The arrays \(\mathbf{y}\) and \(\dot{y}\) are named Y and F , respectively, in that subroutine.

\subsection*{13.2.1 Mathematical Constraints}

A kinematically unconstrained system may be represented by a set of dependent coordinates. This situation exists when Euler parameters are employed as rotational coordinates. The complete set of equations of motion is written, from Eq. 11.25, as
\[
\begin{array}{rl}
\mathbf{p}_{i}^{T} \mathbf{p}_{i}-1=0 & i=1, \ldots, b \\
\dot{\mathbf{p}}_{i}^{T} \mathbf{p}_{i}=0 & i=1, \ldots, b \\
{\left[\begin{array}{cc}
\mathbf{M}^{*} & \mathbf{P}^{T} \\
\mathbf{P} & \mathbf{0}
\end{array}\right]\left[\begin{array}{l}
\ddot{\mathbf{q}} \\
\boldsymbol{\sigma}
\end{array}\right]=} & {\left[\begin{array}{l}
\mathbf{g}^{*}-\mathbf{b}^{*} \\
-\mathbf{c}
\end{array}\right]} \tag{13.7}
\end{array}
\]

An algorithm for solving Eqs. 13.5 through 13.7 can be developed by a slight modification to algorithm DI-1:

\section*{ALGORITHM DI-2}
(a) and (b) the same as for DI-1.
(c) DIFEQN routine
(c.1) Transfer \(\mathbf{y}\) to \(\mathbf{q}\) and \(\dot{\mathbf{q}}\).
(c.2) Evaluate \(\mathbf{M}^{*}, \mathbf{P}, \mathbf{g}^{*}, \mathbf{b}^{*}\), and \(\mathbf{c}\).
(c.3) Solve Eq. 13.7 for \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\sigma}\).
(c.4) Transfer \(\dot{\mathbf{q}}\) and \(\ddot{\mathbf{q}}\) to \(\dot{\mathbf{y}}\).
(c.5) Return.

In this algorithm, the artificial Lagrange multipliers \(\boldsymbol{\sigma}\) are evaluated as a byproduct when Eq. 13.7 is solved. This algorithm requires correct initial conditions on \(\mathbf{p}_{i}\)
and \(\dot{\mathbf{p}}_{i}, i=1, \ldots, b\). Since a numerical integration algorithm yields only an approximate solution to the exact response, the computed values for \(\mathbf{p}_{i}\) and \(\dot{\mathbf{p}}_{i}\) may contain some numerical error after several time steps. Therefore, Eqs. 13.5 and 13.6 may no longer be satisfied. If the accumulation of the error is not corrected or controlled, erroneous results may be obtained. Two methods for correcting the numerical error are discussed in the following.

Method 1. The numerically integrated values for the Euler parameters of body \(i\) at any time step are denoted by \(\mathbf{p}_{i}^{*}\), which may not satisfy Eq. 13.5 ; i.e., it may be that
\[
\begin{equation*}
\mathbf{p}_{i}^{* T} \mathbf{p}_{i}^{*}-1=\delta \tag{13.8}
\end{equation*}
\]
where \(\delta\) represents the violation in the constraint. In this case, the transformation matrix \(\mathbf{A}_{i}^{*}\), calculated in terms of \(\mathbf{p}_{i}^{*}\), will lose the orthogonality condition; i.e., the result will be that
\[
\mathbf{A}_{i}^{* T} \mathbf{A}_{i}^{*} \neq \mathbf{I}
\]

A correction in \(\mathbf{p}_{i}^{*}\) by \(\boldsymbol{\varepsilon}\) can be found to yield a corrected set of Euler parameters, as follows:
\[
\mathbf{p}_{i}=\mathbf{p}_{i}^{*}+\boldsymbol{\varepsilon}
\]
which will satisfy Eq. 13.5. An infinite number of \(\boldsymbol{\varepsilon}\) vectors can be found for this purpose.

A popular method for evaluating the best set of \(\boldsymbol{\varepsilon}\) vectors is to minimize the sum of squares of the elements of \(\varepsilon\) as follows:

Minimize \(f_{1}=\boldsymbol{\varepsilon}^{T} \boldsymbol{\varepsilon}\)
(a)
subject to the constraints of Eq. 13.5
or \({ }^{\dagger}\)
Minimize \(f_{2}=\boldsymbol{\varepsilon}^{T} \boldsymbol{\varepsilon}+\left(\mathbf{p}_{i}^{T} \mathbf{p}_{i}-1\right) \lambda\)
This yields:
\[
\left[\frac{\partial f_{2}}{\partial \mathbf{p}_{i}}\right]^{T} \equiv 2\left(\mathbf{p}_{i}-\mathbf{p}_{i}^{*}\right)+2 \mathbf{p}_{i} \lambda=0
\]
or
\[
\mathbf{p}^{*}=(1+\lambda) \mathbf{p}
\]

Substitution of this equation in Eq. 13.8 gives \((1+\lambda)^{2}=1+\delta\) which results in
\[
\begin{equation*}
\mathbf{p}=\frac{1}{\sqrt{1+\delta}} \mathbf{p}^{*} \tag{13.9}
\end{equation*}
\]

This is the correction formula for the Euler parameters. \({ }^{\ddagger}\) All four parameters are normalized by the same quantity, and in such a way that only the angle of rotation \(\phi\) is affected, not the direction \(\vec{u}\) of the orientational axis of rotation.
\({ }^{\dagger}\) In constrained optimization techniques, the constraint equation(s) can be included in the objective function by the use of Lagrange multipliers.
\({ }^{\ddagger}\) The selection of Eq. \(a\) as the objective function for the optimization process is rather arbitrary. If other objective functions are selected, different correction formulas are obtained. Somewhat different formula can be fround in Ref \(3 n\)

The numerical integration error may also yield numerical values for the violation of Eq. 13.6 by the elements of \(\dot{\mathbf{p}}_{i}\) :
\[
\begin{equation*}
\dot{\mathbf{p}}_{i}^{* T} \mathbf{p}_{i}=\sigma \tag{13.10}
\end{equation*}
\]

A process similar to the preceding minimization process gives a correction formula for the velocities:
\[
\begin{equation*}
\dot{\mathbf{p}}_{i}=\dot{\mathbf{p}}_{i}^{*}-\sigma \mathbf{p}_{i} \tag{13.11}
\end{equation*}
\]

The correction formulas of Eqs. 13.9 and 13.11 can be included in the algorithm DI-2, step c.1.

Method II. The constraints of Eqs. 13.5 and 13.6 can be treated in much tie same way as the kinematic constraint equations. This subject is discussed later in this chapter.

\subsection*{13.2.2 Using Angular Velocities}

If the equations of motion are taken in the form given by Eq. 11.37, a considerable amount of computational efficiency can be gained. In this case, vectors \(\mathbf{y}\) and \(\dot{y}\) are defined as follows:
\[
\mathbf{y}=\left[\begin{array}{l}
\mathbf{q} \\
\mathbf{h}
\end{array}\right] \quad \dot{\mathbf{y}}=\left[\begin{array}{c}
\dot{\mathbf{q}} \\
\dot{\mathbf{h}}
\end{array}\right]
\]

The dimension of \(\mathbf{y}\) or \(\dot{\mathbf{y}}\) is \(13 \times b\), so that each contains \(b\) fewer elements than the arrays of Sec. 13.2. The integration of the velocities and accelerations of a typical body \(i\) is performed according to the following diagram:
\[
\left[\begin{array}{c}
\dot{\mathbf{r}}_{i} \\
\dot{\mathbf{p}}_{i} \\
\dot{\mathbf{r}}_{i} \\
\dot{\boldsymbol{\omega}}_{i}^{\prime}
\end{array}\right] \xrightarrow{\text { (integration) }}\left[\begin{array}{c}
\mathbf{r}_{i} \\
\mathbf{p}_{i} \\
\dot{\mathbf{r}}_{i} \\
\boldsymbol{\omega}_{i}^{\prime}
\end{array}\right]
\]

The computed values for \(\mathbf{p}_{i}\) and \(\boldsymbol{\omega}_{i}^{\prime}\) are employed in Eq. 6.109 to find \(\dot{\mathbf{p}}_{i}\).
An algorithm for dynamic analysis using Eq. 11.37 can be stated by a slight modification to algorithm DI-1:

\section*{ALGORITHM DI-3}
(a) Main routine
(a.1) Specify initial condition for \(\mathbf{q}\) and \(\mathbf{h}\).
(a.2) Define vector \(\mathbf{y}\) as \(\mathbf{y}=\left[\mathbf{q}^{T}, \mathbf{h}^{T}\right]^{T}\).
(a.3) Enter the numerical integration routine (NI).
(b) Numerical integration routine
(Same as DI-1)
(c) DIFEQN routine
(c.1) Transfer the translational coordinates and velocities \(\mathbf{r}_{i}\) and \(\dot{\mathbf{r}}_{i}\), \(i=1, \ldots, b\), from \(\mathbf{y}\) to \(\mathbf{q}\) and \(\dot{\mathbf{q}}\). Transfer \(\mathbf{p}_{i}, i=1, \ldots, b\), from \(\mathbf{y}\) to \(\mathbf{q}\) after correcting for the numerical error. Obtain \(\boldsymbol{\omega}_{i}^{\prime}, i=1, \ldots, b\), from \(\mathbf{y}\), use Eq. 6.109 to calculate \(\dot{\mathbf{p}}_{i}\), and then transfer to \(\dot{\mathbf{q}}\).
(c.2) Evaluate \(\mathbf{M}\) (since \(\mathbf{M}\) is constant in Eq. 11.37, it can be evaluated only once), \(\mathbf{b}\), and \(\mathbf{g}\).
(c.3) Solve Eq. 11.37 for \(\dot{\mathbf{h}}\).
(c.4) Transfer \(\dot{\mathbf{q}}\) and \(\dot{\mathbf{h}}\) to \(\dot{\mathbf{y}}\).
(c.5) Return.

\subsection*{13.3 KINEMATICALLY CONSTRAINED SYSTEMS}

The complete set of equations of motion for a kinematically constrained mechanical system is given as
\[
\begin{array}{r}
\Phi \equiv \boldsymbol{\Phi}(\mathbf{q})=0 \\
\dot{\Phi} \equiv \Phi_{q} \dot{\mathbf{q}}=0 \\
\ddot{\boldsymbol{\Phi}} \equiv \boldsymbol{\Phi}_{\mathbf{q}} \ddot{\mathbf{q}}-\gamma=0 \\
\mathbf{M} \ddot{\mathbf{q}}-\boldsymbol{\Phi}_{\mathbf{q}}^{\tau} \lambda=\mathbf{g} \tag{13.15}
\end{array}
\]

These equations may represent the planar equations of motion given in Eq. 9.6, or the spatial equations of motion given in Eq. 11.42. In the case of spatial motion, it is assumed that the constraints of Eq. 13.12 contain both kinematic constraints and mathematical constraints. Therefore, the Jacobian matrix \(\boldsymbol{\Phi}_{\mathbf{q}}\) in Eqs. 13.13 to 13.15 contains the \(\mathbf{P}\) and \(\mathbf{B}\) matrices of Eq. 11.42. If Eqs. 13.14 and 13.15 are appended together, a set of algebraic equations, linear in \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\), is obtained:
\[
\left[\begin{array}{cc}
\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{\tau}  \tag{13.16}\\
\boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0}
\end{array}\right]\left[\begin{array}{r}
\ddot{\mathbf{q}} \\
-\lambda
\end{array}\right]=\left[\begin{array}{c}
\mathbf{g} \\
\gamma
\end{array}\right]
\]

It should be clear that \(\mathbf{g}, \boldsymbol{\lambda}\), and \(\boldsymbol{\gamma}\) in Eq. 13.16 represent \(\mathbf{g}^{*}-\mathbf{b}^{*}, \boldsymbol{\sigma}\) and \(\boldsymbol{\lambda}\), and \(\mathbf{c}\) and \(\gamma\), respectively, in Eq. 11.42.

A simple but crude method for obtaining the dynamic response of a system represented by Eqs. 13.12 to 13.15 is to employ the direct integration algorithm DI-1 with some minor modifications:

\section*{ALGORITHM DI-4}
(a) and (b) the same as in DI-1.
(c) DIFEQN routine
(c.1) Transfer \(\mathbf{y}\) to \(\mathbf{q}\) and \(\dot{\mathbf{q}}\).
(c.2) Evaluate \(\mathbf{M}\) (M is constant in Eq. 9.6, but a function of \(\mathbf{p}_{i}, i=1, \ldots, b\), in Eq. 11.42), \(\boldsymbol{\Phi}_{\mathbf{q}}, \mathbf{g}\), and \(\gamma\).
(c.3) Solve Eq. 13.16 for \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\).
(c.4) Transfer \(\dot{\mathbf{q}}\) and \(\ddot{\mathbf{q}}\) to \(\dot{\mathbf{y}}\).
(c.5) Return.

The initial conditions on \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) must satisfy Eqs. 13.12 and 13.13. However, on account of the numerical integration error, these equations may be violated. In the preceding sections, several methods for circumventing this problem were presented.

\subsection*{13.3.1 Constraint Violation Stabilization Method}

The constraint violation stabilization method \({ }^{1}\) is an extension of feedback control theory applied to the dynamic analysis of mechanical systems. One of the goals in designing a feedback controller is to suppress the growth of error and achieve a stable response.

In control systems, it is well known that circuits described by second-order differential equations such as
\[
\begin{equation*}
\ddot{y}=0 \tag{a}
\end{equation*}
\]
are unstable, since outside disturbances such as noise (or numerical error, in the case of a numerical integration process) can be amplified. In contrast to Eq. \(a\), which is said to be an open-loop system, a closed-loop system, such as
\[
\ddot{y}+2 \alpha \dot{y}+\beta^{2} y=0
\]
is stable if \(\alpha\) and \(\beta\) are positive constants. The terms \(2 \alpha \dot{y}\) and \(\beta^{2} y\) are the feedback control terms that achieve stability for the differential equation.

The violations in the constraints of Eqs. 13.12 and 13.13 are denoted as
\[
\begin{equation*}
\boldsymbol{\Phi} \equiv \boldsymbol{\Phi}\left(\mathbf{q}^{*}\right)=\boldsymbol{\varepsilon} \tag{b}
\end{equation*}
\]
and
\[
\begin{equation*}
\dot{\mathbf{\Phi}} \equiv \boldsymbol{\Phi}_{\mathrm{q}} \dot{\mathbf{q}}^{*}=\boldsymbol{\sigma} \tag{c}
\end{equation*}
\]
where \(\mathbf{q}^{*}\) and \(\dot{\mathbf{q}}^{*}\) are the computed values of \(\mathbf{q}\) and \(\dot{\mathbf{q}}\). Knowing \(\mathbf{q}^{*}\) and \(\dot{\mathbf{q}}^{*}\), we can find the acceleration vector \(\ddot{\mathbf{q}}\) from Eq. 13.16. For these computed vectors, Eq. 13.14 finds the form
\[
\begin{equation*}
\ddot{\boldsymbol{\Phi}} \equiv \boldsymbol{\Phi}_{\mathrm{q}} \ddot{\mathbf{q}}^{*}-\boldsymbol{\gamma}^{*}=0 \tag{13.17}
\end{equation*}
\]

Vector \(\ddot{\mathbf{q}}^{*}\) is different from the correct acceleration vector \(\ddot{\mathbf{q}}\). The errors in the three vectors are
\[
\begin{aligned}
\mathbf{q}^{*}-\mathbf{q} & =\Delta \mathbf{q} \\
\dot{\mathbf{q}}^{*}-\dot{\mathbf{q}} & =\Delta \dot{\mathbf{q}} \\
\ddot{\mathbf{q}}^{*}-\ddot{\mathbf{q}} & =\Delta \ddot{\mathbf{q}}
\end{aligned}
\]

Since \(\ddot{\mathbf{q}}^{*}\) is integrated to obtain \(\dot{\mathbf{q}}^{*}\) in the next step, any error \(\Delta \ddot{\mathbf{q}}\) subsequently adds to any existing error in the velocity vector. It is ideal to have \(\Delta \ddot{\mathbf{q}}=\boldsymbol{0}\). But since this is an open-loop system, it can be replaced, for the integration process, by the closedloop system
\[
\begin{equation*}
\Delta \ddot{\mathbf{q}}+2 \alpha \Delta \dot{\mathbf{q}}+\beta^{2} \Delta \mathbf{q}=\mathbf{0} \tag{d}
\end{equation*}
\]

Equation \(b\) is expanded about \(\mathbf{q}\) and the second- and higher-order terms are eliminated, to find
\[
\Phi_{\mathbf{q}} \Delta \mathbf{q}=\varepsilon
\]

From Eq. \(c\), it is found that
\[
\Phi_{\mathbf{q}} \Delta \dot{\mathbf{q}}=\sigma
\]

Premultiplying Eq. \(d\) by \(\mathbf{\Phi}_{\mathbf{q}}\) yields
\[
\boldsymbol{\Phi}_{\mathbf{q}}\left(\ddot{\mathbf{q}}^{*}-\ddot{\mathbf{q}}\right)+2 \alpha \boldsymbol{\Phi}_{\mathbf{q}} \Delta \dot{\mathbf{q}}+\beta^{2} \boldsymbol{\Phi}_{\mathbf{q}} \Delta \mathbf{q}=0
\]
or
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathrm{q}} \ddot{\boldsymbol{q}}^{*}-\boldsymbol{\gamma}+2 \alpha \boldsymbol{\sigma}+\beta^{2} \boldsymbol{\varepsilon}=\boldsymbol{0} \tag{e}
\end{equation*}
\]

If the constraint violations \(\boldsymbol{\varepsilon}\) and \(\boldsymbol{\sigma}\) are replaced by constraint symbols \(\boldsymbol{\Phi}\) and \(\dot{\boldsymbol{\Phi}}\) respectively, then Eq. 13.17 and Eq. \(e\) yield
\[
\boldsymbol{\gamma}^{*}=\gamma-2 \alpha \dot{\mathbf{\Phi}}-\beta^{2} \boldsymbol{\Phi}
\]

Appending Eq. 13.17 to Eq. 13.15 yields the stabilized form of Eq. 13.16:
\[
\left[\begin{array}{ll}
\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{T}  \tag{13.18}\\
\boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0}
\end{array}\right]\left[\begin{array}{c}
\ddot{\mathbf{q}} \\
-\boldsymbol{\lambda}
\end{array}\right]=\left[\begin{array}{c}
\mathbf{g} \\
\gamma-2 \alpha \dot{\boldsymbol{\Phi}}-\beta^{2} \boldsymbol{\Phi}
\end{array}\right]
\]
where \(\ddot{\mathbf{q}}\) represents the computed accelerations. When there is no violation in the constraints, Eq. 13.18 becomes identical to Eq. 13.16.

An algorithm CS for the constraint stabilization method can be stated by a slight modification to the algorithm DI-4:

\section*{ALGORITHM CS-1}
(a) and (b) are as in DI-1, but in (a) values are assigned to \(\alpha\) and \(\beta\).
(c) DIFEQN routine
(c.1) Transfer \(\mathbf{y}\) to \(\mathbf{q}\) and \(\dot{\mathbf{q}}\).
(c.2) Evaluate \(\mathbf{M}, \boldsymbol{\Phi}_{\mathbf{q}}, \mathbf{g}\), and \(\gamma\).
(c.3) Evaluate \(\boldsymbol{\Phi}\) and calulate \(\dot{\boldsymbol{\Phi}} \equiv \boldsymbol{\Phi}_{\mathbf{q}} \dot{\mathbf{q}}\).
(c.4) Solve Eq. 13.18 for \(\ddot{q}\) and \(\boldsymbol{\lambda}\).
(c.5) Transfer \(\dot{\mathbf{q}}\) and \(\ddot{\mathbf{q}}\) to \(\dot{\mathbf{y}}\).
(c.6) Return.

The effect of introducing the feedback terms in Eq. 13.18 is illustrated in Fig. 13.1, with some exaggeration, for a typical response. When both \(\alpha\) and \(\beta\) are given zero values, which is exactly the method of algorithm DI-4, the numerical result may diverge from the exact solution. For nonzero values of \(\alpha\) and \(\beta\), the solution oscillates about the exact solution. The amplitude and the frequency of the oscillation due to the stabilization terms depend upon the values of \(\alpha\) and \(\beta\). Experience has shown that for most practical problems, a range of values between 1 and 10 for \(\alpha\) and \(\beta\) is adequate. When \(\alpha=\beta\), critical damping is achieved, which usually stabilizes the response more quickly. \({ }^{19}\)


Figure 13.1 Schematic representation of the exact and numerical solutions to a typical dynamic response.

\subsection*{13.3.2 Coordinate Partitioning Method}

The coordinate partitioning method \({ }^{\dagger}\) controls the accumulation of the numerical error quite differently from the constraint stabilization method. This method makes use of the fact that the \(n\) coordinates \(\mathbf{q}\) are not independent. If the \(n\) coordinates are partitioned into \(m\) dependent coordinates \(\mathbf{u}\) and \(k\) independent coordinates \(\mathbf{v}\), then the velocity vector \(\dot{\mathbf{q}}\) can be partitioned accordingly into \(\dot{\mathbf{u}}\) and \(\dot{\mathbf{v}}\). The integration arrays \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) are defined in terms of the independent variables:
\[
\mathbf{y}=\left[\begin{array}{c}
\mathbf{v} \\
\dot{\mathbf{v}}
\end{array}\right], \quad \dot{\mathbf{y}}=\left[\begin{array}{c}
\dot{\mathbf{v}} \\
\ddot{\mathbf{v}}
\end{array}\right]
\]
where \(\ddot{v}\) is the vector of independent accelerations. The two arrays \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) each have a dimension of \(2 k\).

The kinematic constraints and velocity equations of Eqs. 13.12 and 13.13 can be expressed as
\[
\begin{equation*}
\Phi(\mathbf{u}, \mathbf{v})=0 \tag{13.19}
\end{equation*}
\]
and
\[
\begin{equation*}
\boldsymbol{\Phi}_{\mathbf{u}} \dot{\mathbf{u}}=-\boldsymbol{\Phi}_{\mathbf{v}} \dot{\mathbf{v}} \tag{13.20}
\end{equation*}
\]

Equations 13.19 and 13.20 each represent \(m\) independent equations in terms of \(\mathbf{u}\) and \(\dot{\mathbf{u}}\) respectively. Having \(\mathbf{v}\) and \(\dot{\mathbf{v}}\) from \(\mathbf{y}\), we can solve Eqs. 13.19 and 13.20 for \(\mathbf{u}\) and \(\dot{\mathbf{u}}\); then vectors \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) are completely known. At this point Eq. 13.16 is solved for \(\ddot{\mathbf{q}}\) and \(\lambda\).

An algorithm for the coordinate partitioning method (CP) can be stated, in its simplest form, as follows:

\section*{ALGORITHM CP-1}
(a) Main routine
(a.1) Specify initial conditions on \(\mathbf{q}\) and \(\dot{\mathbf{q}}\).
(a.2) Specify the independent variables \(\mathbf{v}\) (and \(\dot{\mathbf{v}}\) ).
(a.3) Define vector \(\mathbf{y}\) as \(\mathbf{y}=\left[\mathbf{v}^{T}, \dot{\mathbf{v}}^{T}\right]^{T}\).
\((a, 4)\) Enter the numerical integration routine (NI).
(b) Numerical integration routine
(same as for DI-1)
(c) DIFEQN routine
(c.1) Obtain \(\mathbf{v}\) and \(\dot{\mathbf{v}}\) from \(\mathbf{y}\).
(c.2) Solve Eq. 13.19 for \(\mathbf{u}\) using the Newton-Raphson method; \(\mathbf{q}\) is found.
(c.3) Solve Eq. 13.20 for \(\dot{\mathbf{u}} ; \dot{\mathbf{q}}\) is found.
(c.4) Solve Eq. 13.16 for \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\).
(c.5) Transfer \(\dot{\mathbf{v}}\) and \(\ddot{\mathbf{v}}\) (from \(\ddot{\mathbf{q}}\) ) to \(\dot{\mathbf{y}}\).
(c.6) Return.
\({ }^{\dagger}\) The coordinate partitioning method was first developed in a planar-motion computer program called DADS-2D (dynamic analysis and design system) by Wehage and Haug in \(1982 .{ }^{18}\) The three-dimensional motion version of this program for DADS-3D was first developed by Nikravesh and Chung, 1982. \({ }^{12}\)

The most troublesome part in this algorithm is step c.2. In this step, independent coordinates \(\mathbf{v}^{i}\) are known and the constraint equations are solved for the dependent coordinates \(\mathbf{u}^{i}\). Since the constraints are nonlinear algebraic equations, iterative methods must be employed. These require an estimate for \(\mathbf{u}^{i}\) in every time step. The estimate cannot be too far from the correct solution, since if it is it may cause divergence. An estimate for \(\mathbf{u}^{i}\), at time \(t^{i}\), can be found by using the information from the previous time \(t^{i-1}\) :
\[
\mathbf{u}^{i} \simeq \mathbf{u}^{i-1}+h \dot{\mathbf{u}}^{i-1}+0.5 h^{2} \ddot{\mathbf{u}}^{i-1}
\]
where \(h\) is the time step from \(t^{i-1}\) to \(t^{i}\).
Proper partitioning of the coordinates \(\mathbf{q}\) into \(\mathbf{u}\) and \(\mathbf{v}\) is critical in controlling the accumulation of the numerical error. In order to keep this error under control, it might be necessary to switch from one set of independent coordinates to a different set during the integration process. For example, consider the single pendulum shown in Fig. 13.2. Since this is a 1 -degree of freedom system, the dimension of \(\mathbf{v}\) is 1 . For the pendulum (the moving body), with coordinates \(\mathbf{q}=[x, y, \phi]^{T}\), two equations can be written:
\[
\begin{aligned}
& x=d \cos \phi \\
& y=d \sin \phi
\end{aligned}
\]

If the numerical error in the coordinates is denoted by \(\delta x, \delta y\), and \(\delta \phi\), then
\[
\begin{aligned}
& \delta x=-d \sin \phi \delta \phi \\
& \delta y=d \cos \phi \delta \phi
\end{aligned}
\]

In the selection of the independent coordinates, three cases may arise:
1. \(\mathbf{v}=[x], \mathbf{u}=[y, \phi]^{T}\). An error \(\delta x\) causes errors in \(y\) and \(\phi\), as follows:
\[
\begin{aligned}
& \delta \phi=-\frac{1}{d \sin \phi} \delta x \\
& \delta y=-\frac{\cos \phi}{\sin \phi} \delta x
\end{aligned}
\]


Figure 13.2 A single pendulum.
2. \(\mathbf{v}=[y], \mathbf{u}=[x, \phi]^{T}\). An error \(\delta y\) causes errors in \(x\) and \(\phi\), as follows:
\[
\begin{aligned}
\delta \phi & =\frac{1}{d \cos \phi} \delta y \\
\delta x & =-\frac{\sin \phi}{\cos \phi} \delta y
\end{aligned}
\]
3. \(\mathbf{v}=[\phi], \mathbf{u}=[x, y]^{T}\). An error in \(\delta \phi\) causes errors in \(x\) and \(y\), as follows:
\[
\begin{aligned}
& \delta x=-d \sin \phi \delta \phi \\
& \delta y=d \cos \phi \delta \phi
\end{aligned}
\]

A comparison of the three cases reveals that for \(\phi=0\) or \(\phi=\pi\), case 1 yields large errors in \(\phi\) and \(y\) for even a small error in \(x\). However, the errors of the other two cases are bounded. Therefore, for these values of \(\phi\), or any value of \(\phi\) in these neighborhoods, the selection of \(x\) as the independent coordinate is the worst case. Similarly, in the neighborhood of \(\phi= \pm \pi / 2\), the \(y\) coordinate is the worst choice for the independent coordinate. If the pendulum starts from the initial condition \(\phi=0\) and the \(y\) coordinate is selected as the independent coordinate, then around \(\phi= \pm \pi / 4\) the independent coordinate must be switched from \(y\) to \(x\) in order to keep the error under control. The third case shows that if \(\phi\) is selected as the independent coordinate, the error remains bounded regardless of the orientation of the pendulum, and therefore there is no need to switch to another coordinate at any time.

An automatic technique for partitioning the coordinates into the dependent and independent sets is shown in Sec. 13.3.3. During the integration process, some criteria must be used to indicate whether the independent coordinates must be redefined. Such criteria can be based upon the following observations:
1. The number of iterations in the corrector step of a predictor-corrector integration algorithm keeps increasing from one time step to the next.
2. The number of iterations in the Newton-Raphson process of step c. 2 keeps increasing from one time step to the next, i.e., the estimated values for \(\mathbf{u}\) are getting too far from the solution.

A conservative but safe process is to automatically redefine vector \(\mathbf{v}\) once every few time steps.

A modified version of algorithm CP-1 can be stated that allows for redefining the independent and dependent coordinates.

\section*{ALGORITHM CP-2}
(a) and (b) are the same as in CP-1.
(c) DIFEQN routine
(c.1) Obtain \(\mathbf{v}\) and \(\dot{\mathbf{v}}\) from \(\mathbf{y}\).
(c.2) Solve Eq. 13.19 for \(\mathbf{u}\).
(c.3) Is it necessary to redefine the independent coordinates?

If yes, then return to step a.2.
If no, then continue.
(c.4) Solve Eq. 13.20 for \(\mathbf{u}\).
(c.5) Solve Eq. 13.16 for \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\).
(c.6) Transfer \(\dot{\mathbf{v}}\) and \(\ddot{\mathbf{v}}\) to \(\dot{\mathbf{y}}\).
(c.7) Return.

\subsection*{13.3.3 Automatic Partitioning of the Coordinates}

In step a. 2 of algorithm CP-2, an automatic process can be employed to partition the coordinates into dependent and independent sets. A matrix factorization technique, such as the Gaussian elimination with full or partial (column) pivoting, can be performed on the Jacobian matrix for this process. For a mechanical system with \(m\) constraints and \(n\) coordinates, the Jacobian is an \(m \times n\) matrix. The order of the columns of the matrix corresponds to the order of the elements in vector \(\mathbf{q}\). After pivoting, the order of the columns determines the reordering of the elements of \(\mathbf{q}\). The first \(m\) elements of the reordered \(\mathbf{q}\) can be used as the dependent coordinates \(\mathbf{u}\), and the remaining \(k\) elements represent the independent coordinates \(\mathbf{v}\).

\section*{Example 13.1}

Consider the single pendulum with the oscillating mass shown in Fig. 13.3. Constraint equations for the ground and for the revolute and translational joints are written as
\[
\begin{aligned}
& x_{1}=0 \\
& y_{1}=0 \\
& \phi_{1}=0 \\
& x_{1}-x_{2}+0.5 \sin \phi_{2}=0 \\
& y_{1}-y_{2}-0.5 \cos \phi_{2}=0 \\
& \sin \phi_{2}\left(y_{3}-y_{2}\right)+\cos \phi_{2}\left(x_{3}-x_{2}\right)=0 \\
& \phi_{2}-\phi_{3}=0
\end{aligned}
\]

The first three equations are the ground constraints on body 1 , the fourth and fifth equations are the revolute joint constraints from Eq. 4.9, the sixth and seventh equations are the translational joint constraints from Eq. 4.12. The first translational constraint is obtained by defining three points on the line of translation having local coordinates \(\xi_{2}^{P}=0, \eta_{2}^{P}=0, \xi_{2}^{Q}=0, \eta_{2}^{Q}=1, \xi_{3}^{P}=0, \eta_{3}^{P}=0\).

If the vector of coordinates is defined as
\[
\mathbf{q}=\left[x_{1}, y_{1}, \phi_{1}, x_{2}, y_{2}, \phi_{2}, x_{3}, y_{3}, \phi_{3}\right]^{T}
\]
then the Jacobian matrix is written as
\[
\left[\begin{array}{ccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0.5 \cos \phi_{2} & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0.5 \sin \phi_{2} & 0 & 0 & 0 \\
0 & 0 & 0 & -\cos \phi_{2} & -\sin \phi_{2} & 1 & \cos \phi_{2} & \sin \phi_{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1
\end{array}\right]
\]

(a)

(b)

Figure 13.3 A single pendulum with an oscillating mass in two different orientations.
where
\[
(1) \equiv-\sin \phi_{2}\left(x_{3}-x_{2}\right)+\cos \phi_{2}\left(y_{3}-y_{2}\right)
\]

For the configuration in Fig. 13.3(a), the coordinates of the moving bodies are:
\[
\begin{array}{lll}
x_{2}=0.43, & y_{2}=-0.25, & \phi_{2}=60^{\circ} \\
x_{3}=0.69, & y_{3}=-0.40, & \phi_{3}=60^{\circ}
\end{array}
\]

With these coordinates, the Jacobian matrix becomes
\[
\left[\begin{array}{ccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0.25 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0.43 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.5 & -0.87 & -0.3 & 0.5 & 0.87 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1
\end{array}\right]
\]

The column indices corresponding to the order of elements in \(\mathbf{q}\) are shown at the top of the matrix. A Gaussian elimination with partial (column) pivoting yields
\[
\left.\begin{array}{ccccccccc}
1 & 2 & 3 & 4 & 5 & 8 & 6 & 7 & 9 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -0.25 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & -0.43 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -0.92 & 0.58 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1
\end{array}\right]
\]

The order of the column indices shows the reordering of the elements of \(\mathbf{q}\) as
\[
\mathbf{q} \equiv\left[x_{1}, y_{1}, \phi_{1}, x_{2}, y_{2}, y_{3}, \phi_{2}, x_{3}, \phi_{3}\right]^{T}
\]

Since \(m=7\) and \(k=2\), vectors \(\mathbf{u}\) and \(\mathbf{v}\) are defined as
\[
\begin{aligned}
\mathbf{u} & \equiv\left[x_{1}, y_{1}, \phi_{1}, x_{2}, y_{2}, y_{3}, \phi_{2}\right]^{T} \\
\mathbf{v} & \equiv\left[x_{3}, \phi_{3}\right]^{T}
\end{aligned}
\]

For the configuration shown in Fig. 13.3(b), the moving bodies have the coordinates
\[
\begin{array}{lll}
x_{2}=-0.21, & y_{2}=-0.45, & \phi_{2}=-25^{\circ} \\
x_{3}=-0.42, & y_{3}=-0.91, & \phi_{3}=-25^{\circ}
\end{array}
\]

With these coordinates, the Jacobian matrix becomes
\[
\begin{aligned}
& \begin{array}{lllllllll}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array} \\
& {\left[\begin{array}{ccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0.45 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & -0.21 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.91 & 0.42 & -0.5 & 0.91 & -0.42 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1
\end{array}\right]}
\end{aligned}
\]

A Gaussian elimination on this matrix yields
\[
\begin{gathered}
1 \\
1
\end{gathered} 2
\]

In this configuration, the appropriate sets of dependent and independent coordinates are
\[
\begin{aligned}
\mathbf{u} & \equiv\left[x_{1}, y_{1}, \phi_{1}, x_{2}, y_{2}, \phi_{2}, \phi_{3}\right]^{T} \\
\mathbf{v} & \equiv\left[y_{3}, x_{3}\right]^{T}
\end{aligned}
\]

The Gaussian elimination with pivoting suggests that for Fig. 13.3(a) the appropriate independent coordinates are \(x_{3}\) and \(\phi_{3}\) but for Fig. 13.3(b) the appropriate independent coordinates are \(x_{3}\) and \(y_{3}\). Note that the coordinates of the nonmoving body always become part of the set of dependent coordinates.

Several factors influence the selection of the independent coordinates. One such factor is the choice of the unit system. Since not all of the elements of the Jacobian matrix have the same physical dimension, their numerical values may form different ratios when different systems of unit are employed (e.g., SI units versus the U.S. customary FPS units). This in effect yields a different pivoting process, and hence a different set of dependent and independent coordinates.

The second factor is the type of pivoting. A partial (column) pivoting may yield a different result from that given by a full pivoting. Matrix factorization with full pivoting may have some advantage, in terms of the numerical error, over the partial pivoting. However, it cannot be said that the partitioning of the coordinates through a full pivoting process yields a better (in the physical sense) set of independent coordinates.

Possibly the most influential factor in an automatic partitioning of coordinates is the method of matrix factorization. If an L-U factorization process is employed on the constraint Jacobian matrix, instead of the standard Gaussian elimination, different sets of dependent and independent coordinates may be obtained. The original coordinate partitioning algorithm with L-U factorization was suggested by Wehage. \({ }^{18}\) A brief discussion on coordinate partitioning with L-U factorization can be found in Appendix C. In recent years, several other matrix factorization techniques have been employed by other researchers, such as the singular-value decomposition, the QR decomposition, and the Gram-Schmidt process. These techniques offer some advantages over L-U factorization, although the main idea remains basically the same. \({ }^{8,10,11}\)

\subsection*{13.3.4 Stiff Differential Equation Method}

The method of solving a mixed system of algebraic and differential equations of motion presented in this section is completely different in principle from the methods discussed in the preceding sections. This method considers the algebraic constraint equations to be a special form of differential equation in which the time derivatives of the variables do not appear. This assumption has proved to cause the system equations to become numerically stiff. Therefore, a stiff numerical integration method must be applied to solve the equations. \({ }^{\dagger}\)

\footnotetext{
\({ }^{\dagger}\) This algorithm has served as a forerunner in the development of the numerical methods in the area of mechanical systems. \({ }^{16}\) The algorithm has been formulated into a computer program for three-dimensional motion known as ADAMS.
}

In Sec. 13.1, it was stated that the standard numerical integration algorithms are designed to solve systems of differential equations of the form
\[
\begin{equation*}
\dot{\mathbf{y}}=\mathbf{f}(\mathbf{y}, t) \tag{13.21}
\end{equation*}
\]

The modified approach taken here allows for the simultaneous solution of mixed algebraic and differential equations of the form
\[
\begin{equation*}
\mathbf{g}(\mathbf{y}, \dot{\mathbf{y}}, t)=0 \tag{13.22}
\end{equation*}
\]
where some components of \(\dot{\mathbf{y}}\) may not appear in some of the equations. When none of the components of \(\dot{\mathbf{y}}\) appear in an equation, that equation is an algebraic equation; otherwise it is a differential equation.

The \(k\) th order Gear algorithm and its corresponding corrector formula are given by Eqs. 12.23 and 12.24. In order to modify these formulas to solve a mixed system of algebraic and differential equations, Eq. 13.22 is written as
\[
\begin{equation*}
\mathbf{g}(\mathbf{z}, t)=0 \tag{13.23}
\end{equation*}
\]
where \(\mathbf{z}=\left[\mathbf{y}^{T}, \dot{\mathbf{y}}^{T}\right]^{T}\). The Newton-Raphson formula for this equation is
\[
\begin{equation*}
\mathbf{g}_{\mathbf{z}}^{(l)} \Delta \mathbf{z}^{(l)}=-\mathbf{g}^{(l)} \tag{13.24}
\end{equation*}
\]
where \(l\) is the iteration number. When the substitution \(\mathbf{z}=\left[\mathbf{y}^{T}, \dot{\mathbf{y}}^{T}\right]^{T}\) is made in Eq. 13.24, it is found that
\[
\begin{equation*}
\mathbf{g}_{\mathrm{y}}^{(l)} \Delta \mathbf{y}^{(l)}+\mathbf{g}_{\mathrm{y}}^{(l)} \Delta \dot{\mathbf{y}}^{(l)}=-\mathbf{g}^{(l)} \tag{13.25}
\end{equation*}
\]

For the \(l\) th and \(l+1\) st Newton-Raphson iterations, Eq. 12.23 can be rewritten as
\[
\begin{gather*}
\left(\mathbf{y}^{i+1}\right)^{(l)}-h b_{-1}\left(\dot{\mathbf{y}}^{i+1}\right)^{(l)}-\sum_{j=0}^{k-1}\left(a_{j} \mathbf{y}^{i-j}\right)=\mathbf{0}  \tag{13.26}\\
\left(\mathbf{y}^{i+1}\right)^{(l+1)}-h b_{-1}\left(\dot{\mathbf{y}}^{i+1}\right)^{(l+1)}-\sum_{j=0}^{k-1}\left(a_{j} \mathbf{y}^{i-j}\right)=\mathbf{0} \tag{13.27}
\end{gather*}
\]

The summation terms in these equations are not a function of the iteration numberthey are a function of the information from the \(i\) th and previous time steps, so they remain constant at each iteration. Subtracting Eq. 13.26 from Eq. 13.27 yields
\[
\left(\mathbf{y}^{i+1}\right)^{(i+1)}-\left(\mathbf{y}^{i+1}\right)^{(l)}-h b_{-1}\left(\dot{\mathbf{y}}^{i+1}\right)^{(l+1)}+h b_{-1}\left(\dot{\mathbf{y}}^{i+1}\right)^{(l)}=0
\]
or
\[
\begin{equation*}
\Delta \dot{\mathbf{y}}^{(l)}=\frac{1}{h b_{-1}} \Delta \mathbf{y}^{(i)} \tag{13.28}
\end{equation*}
\]
which holds true for the \(i+1\) st or any other time step. Substitution of Eq. 13.28 into Eq. 13.25 results in the corrector formula
\[
\begin{equation*}
\left(\mathbf{g}_{\mathrm{y}}^{(l)}+\frac{1}{h b_{-1}} \mathbf{g}_{\mathrm{y}}^{(l)}\right) \Delta \mathbf{y}^{(l)}=-\mathbf{g}^{(l)} \tag{13.29}
\end{equation*}
\]

If Eq. 13.22 is of the form
\[
\begin{equation*}
\mathbf{g}(\mathbf{y}, \dot{\mathbf{y}}, t)=\mathbf{P} \dot{\mathbf{y}}+\mathbf{p}(\mathbf{y}, t)=0 \tag{13.30}
\end{equation*}
\]
where \(\mathbf{P}\) is a constant matrix or a time-dependent matrix, then Eq. 13.29 can be written in a simpler form as
\[
\begin{equation*}
\left(\mathbf{p}_{y}^{(l)}+\frac{1}{h b_{-1}} \mathbf{P}\right) \Delta \mathbf{y}^{(l)}=-\mathbf{g}^{(l)} \tag{13.31}
\end{equation*}
\]

At each time step, the iterative corrector process of Eq. 13.29 or Eq. 13.31 is continued until all of the Newton differences \(\Delta \mathbf{y}^{(l)}\) are below a specified tolerance level. At each Newton-Raphson iteration, arrays \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) are updated:
\[
\begin{align*}
\mathbf{y}^{(l+1)} & =\mathbf{y}^{(l)}+\Delta \mathbf{y}^{(l)} \\
\dot{\mathbf{y}}^{(l+1)} & =\dot{\mathbf{y}}^{(l)}+\frac{1}{h b_{-1}} \Delta \mathbf{y}^{(l)} \tag{13.32}
\end{align*}
\]

The total-system equations of motion of Eqs. 13.15 and 13.12 are written as
\[
\begin{gather*}
\mathbf{M} \dot{\mathbf{s}}-\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}=\mathbf{g}  \tag{13.33}\\
\dot{\mathbf{q}}=\mathbf{s}  \tag{13.34}\\
\boldsymbol{\Phi}(\mathbf{q})=0 \tag{13.35}
\end{gather*}
\]

Equations 13.33 to 13.35 may be expressed in the form of Eq. 13.30 , where
\[
\begin{aligned}
& \mathbf{P}=\left[\begin{array}{lll}
\mathbf{M} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{I} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0}
\end{array}\right] \\
& \mathbf{p}=\left[\begin{array}{c}
-\mathbf{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}-\mathbf{g} \\
-\mathbf{s} \\
\mathbf{\Phi}
\end{array}\right] \\
& \mathbf{y}=\left[\mathbf{s}^{T}, \mathbf{q}^{T}, \boldsymbol{\lambda}^{T}\right]^{T}
\end{aligned}
\]
and
\[
\dot{\mathbf{y}}=\left[\dot{\mathbf{s}}^{T}, \dot{\mathbf{q}}^{T}, \dot{\boldsymbol{\lambda}}^{T}\right]^{T}
\]

The corrector formula of Eq. 13.31 can be employed to solve for the unknown \(y\) at every time step. In this case, Eq. 13.31 provides \(2 n+m\) equations in \(2 n+m\) unknowns.

It must be expected that the iterative solution of \(2 n+m\) equations, using the Newton-Raphson method, may not be successful for every problem. For large systems of equations, this method may not be considered efficient. One major drawback of this algorithm is the initial estimate on the variables at time \(t=t^{0}\). At the starting time, the initial condition on \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) might be available. However, at \(t=t^{0}\), for almost every practical problem, no information on the Lagrange multipliers \(\boldsymbol{\lambda}\) can be found. Therefore, starting the Newton-Raphson iteration at \(t=t^{0}\) for an arbitrary estimate on \(\boldsymbol{\lambda}\) may cause divergence.

It was noted at the beginning of this section that treating algebraic equations as special forms of differential equations yields numerically stiff systems. This causes artificially high-frequency components in the solution. The high-frequency components of the response do not represent the physical system--they are introduced into the solution numerically. Because of the presence of the high-frequency components in the re-
sponse, the time increment \(h\) must be chosen relatively small. For small values of \(h\), the term \(1 / h\) in the algorithm can become substantially large. Experience has shown that this algorithm cannot be implemented on machines with small word length (4 bytes or single precision). Either a machine with a word length of 8 bytes is needed, or double precision must be employed.

\subsection*{13.4 JOINT COORDINATE METHOD}

In Table 1.1 of Sec. 1.4.1, a comparison was made among three different coordinate systems in terms of various aspects of formulating the equations of motion. The table showed, in a relative sense, that if a set of generalized coordinates (in which the coordinates are equal in number to the number of degrees of freedom) is employed, derivation of the equations of motion can be quite difficult. However, computational efficiency in solving these equations is gained. In contrast, employing Cartesian coordinates yields easy derivation of the equations of motion, but computational efficiency is lost. The joint coordinate method takes advantage of Cartesian coordinates for easy formulation, and Lagrangian coordinates for computational efficiency. This is done by numerically combining the two schemes of formulation. This method is based on the velocity transformations developed by Jerkovsky. \({ }^{7}\)

Consider two bodies \(i\) and \(j\) connected by a revolute joint as shown in Fig. 13.4(a). If the relative angle between the two bodies about the joint axis is denoted by \(\theta\), then, for known coordinates of body \(i\) and known \(\theta\), the coordinates of body \(j\) can be found; i.e.,
\[
\mathbf{q}_{j}=\mathbf{f}^{(r)}\left(\mathbf{q}_{i}, \theta\right)
\]

If the two bodies are connected by a translational joint, as in Fig. 13.4(b), a similar formula can be found:
\[
\mathbf{q}_{j}=\mathbf{f}^{(t)}\left(\mathbf{q}_{i}, \theta\right)
\]
where \(\theta\) indicates the relative distance between the two bodies. If there are 2 degrees of freedom between the two bodies, as there are for the universal joint shown in Fig. \(13.4(\mathrm{c})\), then there are two relative angles, \(\theta_{1}\) and \(\theta_{2}\), denoted by \(\boldsymbol{\theta}_{i j}=\left[\theta_{1}, \theta_{2}\right]^{T}\) :
\[
\mathbf{q}_{j}=\mathbf{f}^{(u)}\left(\mathbf{q}_{i}, \boldsymbol{\theta}_{i j}\right)
\]

(a)

(b)

(c)

Figure 13.4 Two bodies connected by (a) a revolute joint, (b) a translational joint, and (c) a universal joint.

In general, if two bodies \(i\) and \(j\) are connected by a kinematic joint having relative (joint) coordinates \(\boldsymbol{\theta}_{i j}\), then the coordinates \(\mathbf{q}_{j}\) can be expressed as follows:
\[
\begin{equation*}
\mathbf{q}_{j}=\mathbf{f}^{(c)}\left(\mathbf{q}_{i}, \boldsymbol{\theta}_{i j}\right) \tag{13.36}
\end{equation*}
\]

The number of relative coordinates in \(\boldsymbol{\theta}_{i j}\) is equal to the number of relative degrees of freedom between the two bodies, which is dependent only on the kinematic joint connecting the bodies.

A similar expression can be stated for velocity calculation. If the relative velocity between bodies \(i\) and \(j\) is described as \(\dot{\boldsymbol{\theta}}_{i j}\), then \(\dot{\mathbf{q}}_{j}\) can be expressed as follows:
\[
\begin{equation*}
\dot{\mathbf{q}}_{j}=\mathbf{f}^{(v)}\left(\dot{\mathbf{q}}_{i}, \dot{\theta}_{i j}\right) \tag{13.37}
\end{equation*}
\]

A similar but inverse expression can be stated for the accelerations. If it is assumed that the absolute accelerations \(\ddot{\mathbf{q}}_{i}\) and \(\ddot{\mathbf{q}}_{j}\) are known, then the relative acceleration can be found as
\[
\begin{equation*}
\ddot{\boldsymbol{\theta}}_{i j}=\mathbf{f}^{(a)}\left(\ddot{\mathbf{q}}_{i}, \ddot{\mathbf{q}}_{j}\right) \tag{13.38}
\end{equation*}
\]

Explicit formulas for coordinate, velocity, and acceleration computations (Eqs. 13.36 through 13.38 ) can be derived for a variety of kinematic joints. \({ }^{9}\) This is left as an exercise to the interested reader.

\subsection*{13.4.1 Open-Chain Systems}

Consider the open-chain system shown in Fig. 13.5(a), containing one branch and one grounded body (called the base body). Consecutive bodies are connected by kinematic joints. The system may contain force elements that are not shown in the figure. If the bodies are numbered from 1 to \(b\), in any desired order, then relative coordinates \(\boldsymbol{\theta}_{i j}\) are defined between every two adjacent bodies. In this system, the coordinates of the base body \(\mathbf{q}_{1}\) are constants. A vector of relative coordinates \(\boldsymbol{\theta}\) is defined as follows:
\[
\boldsymbol{\theta} \equiv\left[\boldsymbol{\theta}_{12}^{T}, \boldsymbol{\theta}_{23}^{T}, \ldots, \boldsymbol{\theta}_{(b-1) b}^{T}\right]^{T}
\]

For numerical integration, the two vectors \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) are then defined:
\[
\mathbf{y}=\left[\begin{array}{l}
\boldsymbol{\theta}  \tag{13.39}\\
\dot{\boldsymbol{\theta}}
\end{array}\right], \quad \dot{\mathbf{y}}=\left[\begin{array}{c}
\dot{\boldsymbol{\theta}} \\
\ddot{\boldsymbol{\theta}}
\end{array}\right]
\]


Figure 13.5 A single-branch open-chain system with (a) a fixed base body, and (b) a floating base body.

If, like the system shown in Fig. 13.5(b), the mechanical system does not have a fixed (grounded) base body, then a base body has to be chosen for the system; the resulting base body is called a floating base body. If, for example, body 1 is chosen to be the floating base body, then vectors \(\mathbf{y}\) and \(\dot{\mathbf{y}}\) are defined as follows:
\[
\mathbf{y}=\left[\begin{array}{c}
\mathbf{q}_{1}  \tag{13.40}\\
\boldsymbol{\theta} \\
\dot{\mathbf{q}}_{1} \\
\dot{\boldsymbol{\theta}}
\end{array}\right] \quad \dot{\mathbf{y}}=\left[\begin{array}{c}
\dot{\mathbf{q}}_{1} \\
\dot{\boldsymbol{\theta}} \\
\ddot{\mathbf{q}}_{1} \\
\ddot{\boldsymbol{\theta}}^{\prime}
\end{array}\right]
\]

The dimension of \(\mathbf{y}\) or \(\dot{y}\) in the form of either Eq. 13.39 or Eq. 13.40, is twice the number of degrees of freedom of the system.

For systems with no grounded body, there is at least one floating base body. Although the selection of the floating base body is not unique, one body may be a better candidate for the floating base body than another. If the mass of one body is substantially greater than the mass of any of the other bodies, then that body should be selected as the floating base body. It should be noted that the floating base body is not necessarily one of the end bodies in the chain. If no one body has a mass substantially greater than the others, a floating base body can be selected by employing a simple procedure. Each joint is given a number, called the distance, which represents the number of its relative degrees of freedom. For some commonly used three-dimensional kinematic joints, the following data can be found:
\begin{tabular}{lcc}
\multicolumn{1}{c}{ Joint } & Symbol & Distance \\
\hline Spherical (globular) & \(G\) & 3 \\
Revolute & \(R\) & 1 \\
Universal & \(U\) & 2 \\
Cylindrical & \(C\) & 2 \\
Translational (prismatic) & \(P\) & 1 \\
Screw & \(S\) & 1 \\
\hline
\end{tabular}

Each body in the system is treated momentarily as a candidate for the floating base. The sum of distances from branch to branch starting from the body under consideration is calculated and recorded. When the next neighboring body becomes the candidate, the sum of distances decreases in the direction of the move and increases in the other direction. For example, consider the system shown in Fig. 13.6. The following table can be found for the sums of distances for all the bodies:
\begin{tabular}{ccc}
\begin{tabular}{c} 
Floating base \\
candidate
\end{tabular} & \begin{tabular}{c} 
Sum of distances \\
to left
\end{tabular} & \begin{tabular}{c} 
Sum of distances \\
to right
\end{tabular} \\
\hline 1 & 0 & 11 \\
2 & 1 & 10 \\
3 & 2 & 9 \\
4 & 3 & 8 \\
5 & 5 & 6 \\
6 & 8 & 3 \\
7 & 11 & 0 \\
\hline
\end{tabular}


Figure 13.6 An open-chain system.
This table shows that the difference between the left and right distances is smallest for body 5 , and therefore body 5 is the best candidate for the floating base body. This selection minimizes the propagation of numerical error in the computation. For example, if body 5 is selected as the floating base body, then the coordinates of body 1 contain the numerical error accumulated from five relative coordinates. But if body 6 is selected as the floating base body, then body 1 contains the numerical error from eight relative coordinates. Applying this observation to both left and right subbranches for each of the bodies shows how the total error can be minimized by this process.

Some systems may have multiple branches, such as the system shown in Fig. 13.7, which has two branches. For position and velocity computation, the process starts from the base body and moves toward the last body in each branch. For multibranch systems, after the process is completed for the first branch, the process can start on the second branch from the branching body (in Fig. 13.7, body 3).

The order of connectivity between the bodies of a system is called the system topology. The topology of a system either can be defined by inspection or can be done automatically through graph theory. The topology of the system can be set up in the for of a table showing the direction to move in calculating the coordinates (or velocities) of body \(j\), once \(\boldsymbol{\theta}_{i j}\) and coordinates (or velocities) of body \(i\) are known. For the system of Fig. 13.7, this table can have the following entries:
\(\left.\begin{array}{cc}\text { Body } i & \text { Body } j \\ \hline 1 \text { (base) } & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & \ldots \\ \ldots & j\end{array}\right\}\) branch 1

This table also serves as a directive for calculating \(\ddot{\boldsymbol{\theta}}_{i j}\) from \(\ddot{\mathbf{q}}_{i}\) and \(\ddot{\boldsymbol{q}}_{j}\).
At this point, an algorithm can be stated for the joint coordinate method. At the beginning of each time step, the numerical values for \(\mathbf{q}_{\text {base }}, \dot{\mathbf{q}}_{\text {base }}, \boldsymbol{\theta}\), and \(\dot{\boldsymbol{\theta}}\) are known. Equations 13.36 and 13.37 yield \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) for all of the bodies in the system. The coefficient matrix and the right-side vector of Eq. 13.16 can be evaluated, since they are functions of \(\mathbf{q}\) and \(\dot{\mathbf{q}}\). Then the solution of Eq. 13.16 yields \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\). Since \(\ddot{\mathbf{q}}\) is known for all


Figure 13.7 A multibranch open-chain system.
of the bodies, Eq. 13.38 yields \(\boldsymbol{\theta}\). Then the numerical integration routine moves the process to the next time step.

\section*{ALGORITHM JC-1}
(a) Main routine
(a.1) Specify initial conditions for \(\mathbf{q}\) and \(\dot{\mathbf{q}}\).
(a.2) Specify (or automatically determine) the topology of the system.
(a.3) Compute initial conditions for \(\boldsymbol{\theta}\) and \(\ddot{\boldsymbol{\theta}}\).
(a.4) Transfer the initial values to \(y\) (Eq. 13.39 or 13.40).
(b) Numerical integration routing
(Same as DI-1)
(c) DIFEQN routine
(c.1) Transfer the contents of \(\mathbf{y}\) to \(\mathbf{q}_{\text {base }}, \dot{\mathbf{q}}_{\text {base }}\) (if there is a floating base body), \(\boldsymbol{\theta}\), and \(\dot{\boldsymbol{\theta}}\).
(c.2) Compute \(\mathbf{q}\) and \(\dot{\mathbf{q}}\) for all of the bodies (Eqs. 13.36 and 13.37).
(c.3) Evaluate \(\mathbf{M}, \Phi_{\mathbf{q}}, \mathbf{g}\), and \(\gamma\).
(c.4) Solve Eq. 13.16 for \(\ddot{q}\) and \(\boldsymbol{\lambda}\).
(c.5) Compute \(\ddot{\boldsymbol{\theta}}\) (Eq. 13.38).
(c.6) Transfer \(\dot{\mathbf{q}}_{\text {basc }}, \dot{\mathbf{q}}_{\text {basc }}\) (if there is a floating base body), \(\dot{\boldsymbol{\theta}}\), and \(\ddot{\boldsymbol{\theta}}\) to \(\dot{\boldsymbol{y}}\).
(c.7) Return.

In this algorithm, since the coordinates of a body are found from the coordinates of the adjacent body and the interconnected joint coordinates, the constraint equations in terms of Cartesian coordinates are never violated. The same argument is also true for the velocity equations.

\subsection*{13.4.2 Closed-Loop Systems}

A mechanical system may contain one or more independent closed kinematic loops. For example, the system shown in Fig. 13.8(a) contains one closed loop. If the closed loop is cut at one of its joints, as in Fig. 13.8(b), the system becomes equivalent to an openchain system. If the topology of the equivalent open-chain system is defined, then \(\mathbf{q}\) and


Figure 13.8 A system containing a closed loop.
\(\dot{\mathbf{q}}\) for all of the bodies can be computed from \(\mathbf{q}_{\text {base }}, \dot{\mathbf{q}}_{\text {base }}, \boldsymbol{\theta}\), and \(\dot{\boldsymbol{\theta}}\). In this process the joint coordinates and velocities of the cut edge(s) are not needed, e.g., the joint coordinates between bodies 3 and 4 of Fig. 13.8(b) denoted by \(\boldsymbol{\theta}_{34}^{*}\). If algorithm JC-1 is applied to this system, it is likely that the kinematic constraints describing the joint at the cut edge(s) will be violated. This may happen because the coordinates of \(\mathbf{q}_{4}\) are not found from \(\mathbf{q}_{3}\) and \(\boldsymbol{\theta}_{34}\). In order to eliminate the constraint violation at the cut edge(s), feedback terms for constraint stabilization can be introduced in Eq. 13.16.

The constraint equations can be divided into those for uncut edges and for cut edges, as follows:
\[
\begin{aligned}
\Phi(\mathbf{q}) & =0 \\
\Phi^{*}(\mathbf{q}) & =0
\end{aligned}
\]
where the asterisk denotes the cut edges. Then, stabilization terms are included in Eq. 13.16, as follows:
\[
\left[\begin{array}{lll}
\mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{T} & \boldsymbol{\Phi}_{\mathbf{q}}^{* 7}  \tag{13.41}\\
\boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} & \mathbf{0} \\
\boldsymbol{\Phi}_{\mathbf{q}}^{*} & \mathbf{0} & \mathbf{0}
\end{array}\right]\left[\begin{array}{c}
\ddot{\mathbf{q}} \\
-\boldsymbol{\lambda} \\
-\boldsymbol{\lambda}^{*}
\end{array}\right]=\left[\begin{array}{c}
\mathbf{g} \\
\gamma \\
\gamma^{*}-2 \alpha \dot{\Phi}^{*}-\beta^{2} \boldsymbol{\Phi}^{*}
\end{array}\right]
\]

Graph theory provides schemes for determining which joints should be cut. It should be noted that the number of joint coordinates for a closed-loop system is larger than the number of degrees of freedom of that system. Therefore, for the equivalent open-chain representation of a closed-loop system, vector \(\mathbf{y}\) contains more elements than twice the number of degrees of freedom of the actual system.

\section*{PROBLEMS}
13.1 Use algorithm DI-1 in conjunction with subroutine RUNGK4. Assuming zero initial velocities, solve the equations of motion from the following problems:
(a) Prob. 9.6
(b) Prob. 9.7
13.2 For the unconstrained body shown in Fig. P. 13.2, the translational coordinate vector has components \(\mathbf{r}=[0,5,4]^{T}\), and the local and the global coordinate systems are initially parallel. Let \(a=0.6, b=0.4\), and \(c=0.2\), and assume a mass of 50 . A force \(\vec{f}\) with a constant magnitude of 12 acts at point \(A\) and remains perpendicular to plane \(A B C\).
(a) Write the translation equations of motion (Eq. 11.8).
(b) Write the rotational equations of motion in terms of Euler parameters (Eq. 11.16).
(c) Use algorithm DI-2 in conjunction with subroutine RUNGK4 to solve these equations for a specified period of time. Assume initial velocities of zero.
(d) Monitor (plot) the constraint violations for \(\mathbf{p}^{T} \mathbf{p}-1=0\) and \(\mathbf{p}^{T} \dot{\mathbf{p}}=0\).
(e) Repeat the process for different values of \(\Delta t\).


Figure P. 13.2
13.3 Repeat Prob. 13.2 and correct the computed values of \(\mathbf{p}\) and \(\dot{\mathbf{p}}\) according to Eqs. 13.9 and 13.11.
13.4 Repeat Prob. 13.2 and correct the Euler parameters for any constraint violations using the constraint violation stabilization technique.
13.5 Employ the coordinate partitioning method to Prob. 13.2. Assume \(\mathbf{v}=\left[\mathbf{r}^{T}, \mathbf{e}^{T}\right]^{T}\) and \(\mathbf{u}=\left[e_{0}\right]\). For values of \(e_{0}\) close to zero, how would you determine the sign of \(e_{0}\) ?
13.6 Repeat Prob. 13.2 and instead of Eq. 11.16, use Eq. 11.18. Then:
(a) use algorithm DI-3 to solve the equations.
(b) Monitor the constraint violation for \(\mathbf{p}^{T} \mathbf{p}-1=0\).
(c) Modify the program to eliminate or control any constraint violation.
13.7 Why is there no need to correct the computed values of \(\mathbf{p}\) in algorithm DI-3?
13.8 State the reasons why algorithm DI-3 is more efficient than algorithm DI-2.
13.9 Solve the constrained equations of motion from Prob. 9.8 by employing subroutine RUNGK4 in the following algorithms:
(a) Algorithm DI-4
(b) Algorithm CS-1
(c) Algorithm CP-1, assuming \(\mathbf{v}=\left[x_{1}, y_{1}, \phi_{1}, \phi_{2}\right]^{T}\)
(d) Algorithm JC-1, assuming \(\theta=\left[x_{1}, y_{1}, \phi_{1}, \theta_{12}\right]^{T}\)
13.10 In order to develop a planar dynamic analysis program using the joint coordinate method, the transformation formulas of Eqs. 13.36 through 13.38 must be derived explicitly for some of the standard kinematic joints. Derive these formulations for the three kinematic joints shown in Fig. P.13.10 and assume:
(a) \(\theta_{i j}=\alpha\) for the revolute joint.
(b) \(\theta_{i j}=d\) for the translational joint.
(c) \(\theta_{i j}=\left[\alpha_{1}, \alpha_{2}\right]^{T}\) for the revolute-revolute joint.


Figure P. 13.10
13.11 Modify the dynamic analysis program in Chap. 10 from the direct integration method (DI) and devise the following algorithms:
(a) Coordinate partitioning method
(b) Joint coordinate method
13.12 Derive the joint coordinate transformation formulas for a spatial revolute joint in a general case where the joint axis is not parallel to any of the local coordinate axes. Hint: Assume a second local coordinate system \(\xi_{j^{\prime}} \eta_{j^{\prime}} \zeta_{j^{\prime}}\) attached to body \(j\) and initially parallel to \(\xi_{i} \eta_{i} \zeta_{i}\), as shown in Fig. P.13.12. The transformation matrix between \(\xi_{j} \eta_{j} \zeta_{j}\) and \(\xi_{j^{\prime}} \eta_{j^{\prime}} \zeta_{j}\) is a constant matrix. The joint axis becomes the relative orientational and instantaneous axis of rotation between \(\xi_{i} \eta_{i} \zeta_{i}\) and \(\xi_{j} \eta_{j^{\prime}} \zeta_{j^{\prime}}\). Therefore, \(\mathbf{A}_{i j^{\prime}}\) can be expressed as a function of \(\theta_{i j}\) and hence of \(\theta_{i j}\).


Figure P. 13.12
13.13 Repeat Prob. 13.12 for special cases where the joint axis is parallel to one of the coordinate axes of each body; e.g., where \(\xi_{i}\) is parallel to \(\xi_{j}\). Estimate the percentage of computational efficiency that is gained in this formulation as compared with the general-case formulation.
13.14 Derive the joint coordinate transformation formulas for a spatial translational joint in a general case where the joint axis is not parallel to any of the coordinate axes.
13.15 Repeat Prob. 13.14 for special cases where the joint axis is parallel to one of the coordinate axes of each body; e.g., for \(\xi_{i}\) parallel to \(\xi_{j}\). Estimate the percentage of computational efficiency that is gained in this formulation as compared with the general-case formulation.
13.16 Derive the joint coordinate transformation formulas for the following spatial kinematic joints (each joint allows two relative degrees of freedom):
(a) A universal joint
(b) A cylindrical joint

Derive the formulas for general and special cases. Hint: You may assume a third, fictitious body between the two bodies that has one relative DOF with each body.
13.17 Derive the joint coordinate transformation formulas for a spherical joint. Hint: Use a relative set of Euler parameters.
13.18 In order to develop a spatial dynamic analysis program, an approach similar to that of the planar program in Chap. 10 can be followed. For developing the first spatial analysis program, the following formulation and algorithm are suggested:
(a) Use the formulation of Eq. 11.49 for the equations of motion. Use the elements of Table 11.1 for the entries of the Jacobian matrix and vector \(\boldsymbol{\gamma}^{\#}\).
(b) Employ the direct integration algorithm DI-4. Do not be concerned initially with the constraint violation. After the initial version of the program is developed, an additional modification for constraint violation can be implemented,
(c) Employ a well-developed variable step/order predictor-corrector numerical integration package.
The constraint equations listed in Table 11.1 can be combined to model a variety of commonly used kinematic joints. The elements of this table can be easily programmed by employing elementary vector and matrix operations dealing with 3 -vectors and \(3 \times 3\) matrices. Rearrangement of the equations may yield an elementary operation on \(3 \times 4\) matrices (such as \(\mathbf{L}\) and \(\mathbf{G}\) matrices). A careful organization of the program and the use of these elementary operations can easily yield a spatial dynamic analysis program.

\section*{14}

\section*{Static Equilibrium}

\section*{Analysis}

Transient dynamic analysis of complex mechanical systems is often initiated from a position of static equilibrium. Assigning correct values to the coordinates that describe a state of static equilibrium can be a complicated (almost impossible) task for large-scale interconnected systems of bodies. Therefore, static equilibrium analysis is often performed to find the correct set of coordinates prior to the dynamic analysis. As an example, consider a spatial multibody system representing a particular vehicle. The model contains elements representing the main chassis, the wheels, and the suspension and steering systems. The springs of the suspension system may have nonlinear characteristics, and so may the stiffness of the tires. Finding the coordinates describing the static equilibrium configuration of this system from the available data and figures, or even from the actual vehicle, is not a trivial task.

In this chapter several methods for static equilibrium analysis are presented. These methods are based on the general formulation of the governing equations of motion given in Eqs. 13.12 through 13.15. The static equilibrium equations presented in this chapter can easily be modified to fit any particular formulation or any set of coordinates.

\subsection*{14.1 AN ITERATIVE METHOD}

Since velocities and accelerations are zero for static equilibrium, Eqs. 13.12 to 13.15 yield the equilibrium equations as
\[
\begin{array}{r}
\boldsymbol{\Phi}(\mathbf{q})=0 \\
\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}+\mathbf{g}=0 \tag{14.2}
\end{array}
\]

This is a set of \(m+n\) nonlinear algebraic equations in \(m+n\) unknowns \(\boldsymbol{\lambda}\) and \(\mathbf{q}\). An iterative technique such as Newton-Raphson can be employed to solve Eqs. 14.1 and 14.2. The corresponding iterative equation is
\[
\left[\begin{array}{cc}
\boldsymbol{\Phi}_{\mathbf{q}} & 0  \tag{14.3}\\
\left(\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}+\mathbf{g}\right)_{\mathbf{q}} & \boldsymbol{\Phi}_{\mathbf{q}}^{T}
\end{array}\right]\left[\begin{array}{c}
\Delta \mathbf{q} \\
\Delta \boldsymbol{\lambda}
\end{array}\right]=-\left[\begin{array}{c}
\boldsymbol{\Phi} \\
\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}+\mathbf{g}
\end{array}\right]
\]

This formula requires proper initial estimates for \(\mathbf{q}\) and \(\boldsymbol{\lambda}\). Determination of a reasonable set of initial estimates for \(\mathbf{q}\) is possible. However, a proper set of initial estimates for \(\boldsymbol{\lambda}\) is rather difficult. A poor estimate for \(\boldsymbol{\lambda}\) may lead to a badly conditioned matrix and divergence of the algorithm. This is an undesirable feature of this method.

It is possible to obtain an estimate for \(\boldsymbol{\lambda}\) by solving Eq. 13.16 for \(t=0\). The initial estimates for \(\mathbf{q}\), with \(\dot{\mathbf{q}}=0\), are used to solve Eq. 13.16 for \(\ddot{\mathbf{q}}\) and \(\boldsymbol{\lambda}\). Then, vector \(\boldsymbol{\lambda}\) obtained from this solution might be a reasonable estimate to start the iterative process of Eq. 14.3.

\subsection*{14.1.1 Coordinate Partitioning}

In order to circumvent the difficulty of finding a reasonable initial estimate for \(\boldsymbol{\lambda}\) in Eq. 14.3, the coordinate partitioning method can eliminate the presence of Lagrange multipliers in the equilibrium equations. \({ }^{18}\) If \(\mathbf{q}\) is partitioned into \(m\) dependent coordinates \(\mathbf{u}\) and \(k\) independent coordinates \(\mathbf{v}\), then \(\boldsymbol{\Phi}_{\mathbf{q}}\) and \(\mathbf{g}\) can be partitioned as follows:
\[
\begin{aligned}
\Phi_{\mathbf{q}} & \equiv\left[\Phi_{\mathbf{u}}, \Phi_{\mathbf{v}}\right] \\
\mathbf{g} & =\left[\begin{array}{l}
\mathbf{g}_{(u)} \\
\mathbf{g}_{(v)}
\end{array}\right]
\end{aligned}
\]

Hence, Eq. 14.2 is written as
\[
\begin{align*}
& \boldsymbol{\Phi}_{\mathbf{u}}^{T} \boldsymbol{\lambda}+\mathbf{g}_{(u)}=\mathbf{0}  \tag{a}\\
& \boldsymbol{\Phi}_{\mathbf{v}}^{T} \boldsymbol{\lambda}+\mathbf{g}_{(v)}=\mathbf{0} \tag{b}
\end{align*}
\]

If Eq. \(a\) is solved for \(\boldsymbol{\lambda}\) and the result is substituted in Eq. \(b\), it is found that
\[
\begin{equation*}
\mathbf{g}_{(v)}=-\mathbf{H}^{T} \mathbf{g}_{(u)} \tag{14.4}
\end{equation*}
\]
where \(\mathbf{H}\) is as defined in Appendix \(\mathbf{C}\) :
\[
\mathbf{H}=-\boldsymbol{\Phi}_{\mathbf{u}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}}
\]

Equation 14.4 can be written as
\[
\begin{equation*}
\mathbf{f} \equiv \mathbf{H}^{T} \mathbf{g}_{(u)}+\mathbf{g}_{(v)}=0 \tag{14.5}
\end{equation*}
\]
which represents \(k\) equations in \(k\) unknowns \(\mathbf{v}\). Equation 14.5 can be solved by a NewtonRaphson algorithm provided an initial estimate is given for \(\mathbf{v}\). The iterative formula in this case is
\[
\begin{equation*}
\left[\frac{d \mathbf{f}}{d \mathbf{v}}\right] \Delta \mathbf{v}=-\mathbf{f} \tag{14.6}
\end{equation*}
\]

Finding explicit expressions for the elements of \(d \mathbf{f} / d \mathbf{v}\) can be too complicated, since \(\mathbf{H}\) and \(\mathbf{g}\) are implicitly functions of \(\mathbf{v}\). Therefore, this matrix is evaluated by a numerical
differencing method. This can be done by perturbing one element of \(v\) at a time to determine its corresponding column of \(d \mathbf{f} / d \mathbf{v}\). If \(\mathbf{v}\) is defined prior to and after perturbation of its \(i\) th element by the equations
\[
\mathbf{v}=\left[\begin{array}{c}
v_{1} \\
\vdots \\
v_{i} \\
\vdots \\
v_{k}
\end{array}\right] \quad \mathbf{v}^{i}=\left[\begin{array}{c}
v_{1} \\
\vdots \\
v_{i}+\Delta v_{i} \\
\vdots \\
v_{k}
\end{array}\right]
\]
then
\[
\begin{equation*}
\frac{d \mathbf{f}}{d v_{i}}=\frac{1}{\Delta v_{i}}\left[\mathbf{f}\left(\mathbf{v}^{i}\right)-\mathbf{f}(\mathbf{v})\right] \tag{14.7}
\end{equation*}
\]

This process is repeated for \(i=1, \ldots, k\) to obtain all \(k\) columns of \(d \mathbf{f} / d \mathbf{v}\).

\subsection*{14.2 POTENTIAL ENERGY FUNCTION}

For the purpose of static equilibrium analysis, the definition of potential energy is stated here without proof. For a conservative system, the vector of forces \(\mathbf{g}\) can be derived from a potential energy function \(V\) such that \({ }^{5}\)
\[
\begin{equation*}
\mathbf{g}=-V_{\mathrm{q}}^{T} \tag{14.8}
\end{equation*}
\]

The potential energy \(V\) depends only on position coordinates \(\mathbf{q}\) and not on time or velocities. The most common factors contributing to the system potential energy are gravitational field, externally applied forces, and springs.

Assume that the gravitational field is acting on a system in the negative \(z\) direction. If the gravitational constant is \(g\) and the mass of body \(i\) is \(m_{i}\), then Eq. 14.8 is written as
\[
\begin{equation*}
-m_{i} g=-\frac{\partial V_{i}^{(,)}}{\partial z} \tag{a}
\end{equation*}
\]
where \(V_{i}^{(8)}\) denotes the potential energy of body \(i\) due to gravity. Eq. \(a\) yields
\[
\begin{equation*}
V_{i}^{(8)}=m_{i} g\left(z_{i}-z^{0}\right) \tag{14.9}
\end{equation*}
\]
where \(z^{0}\) is a constant. For simplification, the potential energy \(V^{(8)}\) may be taken as zero at the \(x y\) plane. Then Eq. 14.9 becomes
\[
\begin{equation*}
V_{i}^{(g)}=m_{i} z_{i} g \tag{14.10}
\end{equation*}
\]

If a constant external force \(\vec{f}_{k}\) acts on body \(i\), the potential energy corresponding to \(\vec{f}_{k}\) can be written as
\[
\begin{equation*}
V_{i}^{(f)}=-\left(f_{(x)} x_{i}+f_{(x)} y_{i}+f_{(z)} z_{i}\right) \tag{14.11}
\end{equation*}
\]
where, as was done with the gravitational force, the \(x y z\) coordinate system is considered the reference frame in which the potential energy due to \(\vec{f}_{k}\) is zero in the coordinate planes.

Potential (strain) energy stored in the \(k\) th translational spring, which connects bodies \(i\) and \(j\), may be written as
\[
\begin{equation*}
V_{i}^{(s)}=\int_{l^{0}}^{l} f^{(s)} d l \tag{b}
\end{equation*}
\]

If the spring characteristic is linear, then \(f^{(s)}=k\left(l-l^{0}\right)\) yields
\[
\begin{equation*}
V_{i}^{(s)}=\frac{1}{2} k\left(l-l^{0}\right)^{2} \tag{14.12}
\end{equation*}
\]

If the spring has nonlinear characteristics and the force \(f^{(s)}\) is available as a nonlinear function of \(\Delta l\), then Eq. \(b\) may be integrated numerically to obtain the potential energy.

Contributions to the potential energy of a system from other sources, such as rotational springs or external moments, can be determined as in the preceding derivations. The total potential energy of a system may be defined as the sum of the potential energies of the springs, and other externally applied forces. If there are \(b\) bodies in the system, the total potential energy due to the gravitational field is
\[
\begin{equation*}
V^{(g)}=\sum_{i=1}^{b} V_{i}^{(g)} \tag{14.13}
\end{equation*}
\]

For \(c\) constant externally applied forces, Eq. 14.11 yields
\[
\begin{equation*}
V^{(n)}=\sum_{k=1}^{c} V_{k}^{(n} \tag{14.14}
\end{equation*}
\]

Similarly, if there are \(d\) translational springs in the system, the total potential energy of the springs is
\[
\begin{equation*}
V^{(s)}=\sum_{k=1}^{d} V_{k}^{(s)} \tag{14.15}
\end{equation*}
\]

The system potential energy is thus
\[
\begin{equation*}
V=V^{(g)}+V^{(n)}+V^{(s)} \tag{14.16}
\end{equation*}
\]

If there are any other force elements in the system that are not discussed here, their contribution to the total potential energy must be added to Eq. 14.16.

\subsection*{14.2.1 Minimization of Potential Energy}

The static equilibrium configuration of a mechanical system may be determined by evaluating the position for which the potential energy function is at its minimum. \({ }^{5}\) For the potential energy function \(V=V(\mathbf{q})\) of Eq. 14.16, the minimization problem can be stated as
\[
\begin{align*}
& \text { Minimize } V=V(\mathbf{q}) \\
& \text { subject to constraints } \boldsymbol{\Phi}(\mathbf{q})=\mathbf{0} \tag{14.17}
\end{align*}
\]

This equation represents a constrained minimization problem; i.e., the kinematic constraints \(\boldsymbol{\Phi}(\mathbf{q})=\mathbf{0}\) must be satisfied for all feasible \(\mathbf{q}\). Many constrạined optimization algorithms are available and can be applied to this problem. However, Eq. 14.17 can be transformed into an unconstrained minimization problem. \({ }^{14}\) The method is based on the coordinate partitioning method.

The total differential of the potential energy function may be written as
\[
\begin{aligned}
d V & =V_{\mathbf{q}} d \mathbf{q} \\
& =V_{\mathbf{u}} d \mathbf{u}+V_{\mathbf{v}} d \mathbf{v} \\
& =\left(V_{\mathbf{u}} \mathbf{H}+V_{\mathbf{v}}\right) d \mathbf{v}
\end{aligned}
\]
where Eq. C. 5 of Appendix C has been employed. This equation can be written as
\[
\begin{equation*}
\frac{d V}{d \mathbf{v}}=V_{\mathbf{u}} \mathbf{H}+V_{\mathbf{v}} \tag{14.18}
\end{equation*}
\]

Hence, the minimization problem can be restated as
Minimize \(V=V(\mathbf{v})\)
\[
\begin{equation*}
\text { where } \frac{d V}{d \mathbf{v}}=V_{\mathbf{u}} \mathbf{H}+V_{\mathbf{v}} \tag{14.19}
\end{equation*}
\]

This is an unconstrained optimization problem in \(k\) variables. The gradient vector \(d V / d \mathbf{v}\) is needed for most of the commonly used optimization algorithms, e.g., steepest descent or conjugate gradient.

The gradient vector \(d V / d \mathbf{v}\) can be determined easily at a feasible position \(\mathbf{q}\), i.e., where \(\boldsymbol{\Phi}(\mathbf{q})=0\) is satisfied. The partial derivative \(V_{\mathbf{q}}\) is the negative of the force vector \(\mathbf{g}^{T}\) at position \(\mathbf{q}\); hence,
or
\[
\begin{gather*}
V_{\mathbf{u}}=-\mathbf{g}_{(u)}^{T} \\
V_{\mathbf{v}}=-\mathbf{g}_{(v)}^{T} \\
\frac{d V}{d \mathbf{v}}=-\mathbf{g}_{(u)}^{T} \mathbf{H}-\mathbf{g}_{(v)}^{T} \tag{14.20}
\end{gather*}
\]

The following steps outline an algorithm, based on the minimization of potential energy, for finding a stable equilibrium configuration. Any well-developed unconstrained minimization algorithm can be used.

\section*{ALGORITHM SE-1}
(a) Main routine
(a.1) Specify initial estimates for \(\mathbf{q}\).
(a.2) Specify (or determine automatically) the independent variables \(\mathbf{v}\).
(a.3) Enter the minimization routine.
(b) Minimization routine
(b.1) In the process of minimization, the function under consideration and its vector of gradient (in this case \(V\) and \(d V / d \mathbf{v}\) ) must be evaluated for any particular \(\mathbf{v}\). This is accomplished by a call to FUNCTION routine.
(b.2) If Flag \(=0\), then continue the minimization process. If Flag \(=1\), then a feasible solution for the kinematic constraint equations for the assigned values of \(\mathbf{v}\) does not exist.
One of the following two conditions may exist:
(1) The step taken by the minimization routine is too large. Make the step smaller; e.g., cut it by half, update \(\mathbf{v}\), and return to step b. 1 to repeat the process.
(2) The step taken by the minimization routine is too small. This may be an indication that the set of coordinates considered as independent coordinates is not an adequate set. In order to define a new set of independent coordinates return to step a. 2 .
(c) FUNCTION routine
(c.1) Knowing \(\mathbf{v}\), solve the constraint equations for \(\mathbf{u}\). If \(\mathbf{u}\) cannot be found, then set Flag \(=1\) and return to the minimization routine. Otherwise set Flag \(=0\) and continue to step c.2.
(c.2) Knowing \(\mathbf{u}\) and \(\mathbf{v}\); i.e., \(\mathbf{q}\), evaluate the potential energy function from Eq. 14.16.
(c.3) Evaluate matrices \(\boldsymbol{\Phi}_{u}\) and \(\boldsymbol{\Phi}_{\mathbf{v}}\), and then \(\mathbf{H}\).
(c.4) Evaluate the force vector \(\mathbf{g}\) and partition it into \(\mathbf{g}_{(u)}\) and \(\mathbf{g}_{(v)}\).
(c.5) Compute the gradient vector \(d V / d v\) from Eq. 14.20 .
(c.6) Return.

It is stated in step c. 1 of this algorithm that for some predicted values of \(\mathbf{v}\) a solution to the constraint equations \(\Phi(\mathbf{u}, \mathbf{v})=0\) may not exist. This can be because the unconstrained minimization algorithm is not aware of the presence of the constraint equations. The algorithm predicts the largest possible step toward a minimum along a predicted direction. The predicted point may be outside the feasible region of the constraint equations. In cases such as this, the step must be made smaller.

If the initial estimate for the coordinates is too far from the equilibrium state, then, as was true of the coordinate partitioning method in dynamic analysis, the initial set of independent coordinates may not be valid in other points. Therefore, the preceding algorithm requires occasional checking or possibly switching to a different set of independent coordinates.

\subsection*{14.3 FICTITIOUS DAMPING METHOD}

A mechanical system in motion with no damping elements can oscillate about its static equilibrium state. If several dampers, which are energy-dissipating elements, are added to the system, the total energy of the system will decrease as time passes. The oscillation will be slowed, and finally the mechanical system will reach its static equilibrium state. Therefore, if a mechanical system contains some damping elements, then its static equilibrium state may be determined by performing dynamic analysis, using any of the algorithms stated in Chap. 13.

For systems containing no damping elements, or not having a sufficient number of energy-dissipating elements, fictitious damping terms can be included in the equations of motion:
\[
\begin{equation*}
\mathbf{M} \ddot{\mathbf{q}}-\boldsymbol{\Phi}_{\mathbf{q}}^{T} \boldsymbol{\lambda}=\mathbf{g}-\mathbf{D} \dot{\mathbf{q}} \tag{14.21}
\end{equation*}
\]

Matrix \(\mathbf{D}\) is a positive-definite matrix containing the fictitious damping coefficients. For simplification, \(\mathbf{D}\) can be defined as a diagonal matrix. The values of the fictitious damping coefficients do not change the static equilibrium state, but influence the speed of reaching that state.

\subsection*{14.4 JOINT COORDINATE METHOD}

In Sec. 13.4 and subsections 13.4.1 and 13.4.2, a method based on the joint coordinates was introduced for the transient dynamic analysis of mechanical systems. The joint coordinates in conjunction with the potential energy function can be used for determining the static equilibrium configuration of a system.

For a system containing an open chain (single-branch or multibranch), such as the one shown in Fig. 14.1(a), a vector \(s\) is defined in terms of the coordinates of the base body and the joint coordinates:
\[
\mathbf{s}=\left[\begin{array}{c}
\mathbf{q}_{\text {base }}  \tag{14.22}\\
\boldsymbol{\theta}
\end{array}\right]
\]

For a known vector \(\mathbf{s}\), vector \(\mathbf{q}\) representing the configuration of the system can be computed as stated in Sec. 13.4. Then the potential energy of the system in that configuration can be computed. The static equilibrium configuration may be found by minimizing a function \(f\) :
\[
\begin{equation*}
\text { Minimize } f=V(\mathbf{s}) \tag{14.23}
\end{equation*}
\]
where the gradient vector is
\[
\begin{align*}
\frac{d f}{d \mathbf{s}} & =\frac{d V}{d \mathbf{s}} \\
& =V_{\mathrm{q}}\left[\frac{d \mathbf{q}}{d \mathbf{s}}\right] \\
& =-\mathbf{g}^{T}\left[\frac{d \mathbf{q}}{d \mathbf{s}}\right] \tag{14.24}
\end{align*}
\]

The elements of the matrix \([d \mathbf{q} / d \mathbf{s}]\) can be computed by a numerical differencing method similar to that shown in Eq. 14.7. Note that since \(\mathbf{q}\) is computed directly from \(\mathbf{q}_{\text {base }}\) and \(\boldsymbol{\theta}\), there is no constraint violation to consider.

For systems with closed loops, such as the one shown in Fig. 14.1(b), each closed loop can be cut at one of the joints to achieve an equivalent open-chain system. For the


Figure 14.1 Schematic presentation of a mechanical system with (a) an open chain and (b) a closed loop.
equivalent system, vector \(\boldsymbol{s}\) is defined as in Eq. 14.22, where \(\boldsymbol{\theta}\) does not include the joint coordinates at the cut edges; such as \(\theta_{i j}^{*}\) in Figure 14.1(b). If the kinematic joints at the cut edges are defined by the constraint equations \(\boldsymbol{\Phi}^{*}(\mathbf{q})=\boldsymbol{0}\), then a minimization problem can be stated as
\[
\begin{equation*}
\text { Minimize } f=V(\mathbf{s})+\frac{1}{2} w \boldsymbol{\Phi}^{* T} \boldsymbol{\Phi}^{*} \tag{14.25}
\end{equation*}
\]
where \(w\) is a weighting coefficient. Note that in this minimization problem, the constraint violations at the cut edges are introduced in the objective function. The weighting coefficient \(w\) is introduced to scale the two terms in Eq. 14.25. Depending on the unit system used, and also the reference frame for zero potential energy, the magnitude of the potential energy \(V\) is normally several orders of magnitude larger than the sum of the squares of the constraint violations \(\boldsymbol{\Phi}{ }^{*} \boldsymbol{\Phi}^{*}\). Therefore, the minimization algorithm is not sensitive to the constraint violation as much as to the changes in the potential energy. For this reason, a large value for \(w\) can bring the magnitude of both terms within the same range. The magnitude of \(w\) may be redefined several times during the minimization process.

The gradient vector for the function of Eq. 14.25 is
\[
\begin{align*}
\frac{d f}{d \mathbf{s}} & =\frac{d V}{d \mathbf{s}}+w \boldsymbol{\Phi}^{* T}\left[\frac{d \boldsymbol{\Phi}^{*}}{d \mathbf{s}}\right] \\
& =\left[-\mathbf{g}^{T}+w \mathbf{\Phi}^{* T} \boldsymbol{\Phi}_{\mathbf{q}}^{*}\right]\left[\frac{d \mathbf{q}}{d \mathbf{s}}\right] \tag{14.26}
\end{align*}
\]

\section*{Euler Angles and}

\section*{Bryant Angles}

Among the most common parameters used to describe the angular orientation of a body in space are Euler angles. The angular orientation of a given body-fixed coordinate system \(\xi \eta \zeta\) can be envisioned to be the result of three successive rotations. The three angles of rotation corresponding to the three successive rotations are defined as Euler angles. The sequence of rotations used to define the final orientation of the coordinate system is to some extent arbitrary. A total of twelve conventions is possible in a righthand coordinate system. For the Euler angles described here, a particular sequence of rotations known as the \(x\) convention is considered. Another convention, known as the xyz convention, is also discussed here; the parameters associated with this convention are often referred to as Bryant angles.

\section*{A. 1 Euler Angles}

Euler angles provide a set of three coordinates without any constraint equations. The sequence of rotations employed in the \(x\) convention starts by rotating the initial system of \(x y z\) axes counterclockwise about the \(z\) axis by an angle \(\psi\), as shown in Fig. A.1. The resulting coordinate system is labeled \(\xi^{\prime \prime} \eta^{\prime \prime} \zeta^{\prime \prime}\). In the second step the intermediate \(\xi^{\prime \prime} \eta^{\prime \prime} \zeta^{\prime \prime}\) axes are rotated about \(\xi^{\prime \prime}\) counterclockwise by an angle \(\theta\) to produce another intermediate set, the \(\xi^{\prime} \eta^{\prime} \zeta^{\prime}\) axes. Finally, the \(\xi^{\prime} \eta^{\prime} \zeta^{\prime}\) axes are rotated counterclockwise about \(\zeta^{\prime}\) by an angle \(\sigma\) to produce the desired \(\xi \eta \zeta\) system of axes. \({ }^{\dagger}\) The angles \(\psi, \theta\), and \(\sigma\), which are the Euler angles, completely specify the orientation of the \(\xi \eta \zeta\) system relative to the \(x y z\) system and can therefore act as a set of three independent coordinates.

\footnotetext{
\({ }^{\dagger}\) In most textbooks, the third Euler angle is denoted by \(\phi\). Since, in this text, \(\phi\) is used to describe the angle of rotation about the orientational axis of rotation, \(\boldsymbol{\sigma}\) is used here for the third Euler angle instead of \(\phi\).
}


Figure A. 1 The rotations defining the Euler Angles.

The elements of the complete transformation matrix \(\mathbf{A}\) can be obtained as the triple product of the matrices that define the separate rotations, i.e., the matrices
\[
\mathbf{D}=\left[\begin{array}{ccc}
\mathrm{c} \psi & -\mathrm{s} \psi & 0 \\
\mathrm{~s} \psi & \mathrm{c} \psi & 0 \\
0 & 0 & 1
\end{array}\right] \quad \mathbf{C}=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \mathrm{c} \theta & -\mathrm{s} \theta \\
0 & \mathrm{~s} \theta & \mathrm{c} \theta
\end{array}\right] \quad \mathbf{B}=\left[\begin{array}{ccc}
\mathrm{c} \sigma & -\mathrm{s} \sigma & 0 \\
\mathrm{~s} \sigma & \mathrm{c} \sigma & 0 \\
0 & 0 & 1
\end{array}\right]
\]
where \(\mathrm{c} \equiv \cos\) and \(\mathrm{s} \equiv \sin\). Hence, \(\mathbf{A}=\mathbf{D C B}\) is found to be
\[
\mathbf{A}=\left[\begin{array}{ccc}
\mathrm{c} \psi \mathrm{c} \sigma-\mathrm{s} \psi \mathrm{c} \theta \mathrm{~s} \sigma & -\mathrm{c} \psi \mathrm{~s} \sigma-\mathrm{s} \psi \mathrm{c} \theta \mathrm{c} \sigma & \mathrm{~s} \psi \mathrm{~s} \theta  \tag{A.1}\\
\mathrm{~s} \psi \mathrm{c} \sigma+\mathrm{c} \psi \mathrm{c} \theta \mathrm{~s} \sigma & -\mathrm{s} \psi \mathrm{~s} \sigma+\mathrm{c} \psi \mathrm{c} \theta \mathrm{c} \sigma & -\mathrm{c} \psi \mathrm{~s} \theta \\
\mathrm{~s} \theta \mathrm{~s} \sigma & \mathrm{~s} \theta \mathrm{c} \sigma & \mathrm{c} \theta
\end{array}\right]
\]

It can be verified that matrix \(\mathbf{A}\) is orthonormal, i.e., that \(\mathbf{A}^{T}=\mathbf{A}^{-1}\).
The advantage of having three independent rotational coordinates, instead of nine dependent direction cosines, is offset by the disadvantage that the elements of \(\mathbf{A}\) in terms of the Euler angles are complicated trigonometric functions. Still, a more severe problem exists. Figure A. 2 shows that if \(\theta=n \pi, n=0, \pm 1, \pm 2, \ldots\), the axes of the first and third rotations coincide, so that \(\psi\) and \(\sigma\) cannot be distinguished. This fact is illustrated by setting \(\theta=0\) in \(\mathbf{A}\) to obtain
\[
\mathbf{A}=\left[\begin{array}{ccc}
\mathrm{c} \alpha & -\mathrm{s} \alpha & 0 \\
\mathrm{~s} \alpha & -\mathrm{c} \alpha & 0 \\
0 & 0 & 1
\end{array}\right] \quad \theta=0
\]
where \(\alpha=\psi+\sigma\).


Figure A. 2 Euler angles for the case \(\theta=n \pi(n=0, \pm 1, \pm 2, \ldots)\).

It may be necessary to calculate Euler angles that correspond to a known transformation matrix. For this purpose, the following formulas are deduced from Eq. A.1:
\[
\begin{array}{ll}
\cos \theta=a_{33} & \sin \theta= \pm \sqrt{1-\cos ^{2} \theta} \\
\cos \psi=\frac{a_{23}}{\sin \theta} & \sin \psi=\frac{a_{13}}{\sin \theta}  \tag{A.2}\\
\cos \sigma=\frac{a_{32}}{\sin \theta} & \sin \sigma=\frac{a_{31}}{\sin \theta}
\end{array}
\]

These formulas show that numerical difficulties are to be expected for values of \(\theta\) that are close to the critical values \(n \pi, n=0, \pm 1, \pm 2, \ldots\).

\section*{A.1.1 Time Derivatives of Euler Angles}

The general rotation associated with \(\vec{\omega}\) can be considered equivalent to three successive rotations with angular velocities \(\omega_{(\psi)}=\dot{\psi}, \omega_{(\theta)}=\dot{\theta}\), and \(\omega_{(\sigma)}=\dot{\sigma}\). Hence, the vector \(\vec{\omega}\) can be obtained as the sum of three separate angular velocity vectors. This vector sum cannot be obtained easily, since the directions \(\vec{\omega}_{(\psi)}, \vec{\omega}_{(\theta)}\), and \(\vec{\omega}_{(\sigma)}\) are not orthogonally placed: \(\vec{\omega}_{(\psi)}\) is along the global \(z\) axis and, \(\vec{\omega}_{(\theta)}\) is along the line of nodes, while \(\vec{\omega}_{(\sigma)}\) is along the body \(\zeta\) axis. However, the orthonormal transformation matrices B, C, and D may be used to determine the components of these vectors along any desired set of axes.

Figure A. 3 can be used to obtain the components of the velocity vector \(\vec{\omega}\) in the \(\xi \eta \zeta\) axes in terms of Euler angles and rates. Since \(\dot{\psi}\) is parallel to the \(z\) axis, its components along the body axes are given by applying the orthonormal transformation \(\mathbf{B}^{T} \mathbf{C}^{T}\).
\[
\begin{aligned}
& \dot{\psi}_{(\xi)}=\dot{\psi} \sin \theta \sin \sigma \\
& \dot{\psi}_{(n)}=\dot{\psi} \sin \theta \cos \sigma \\
& \dot{\psi}_{(b)}=\dot{\psi} \cos \theta
\end{aligned}
\]


Figure A. 3 Euler angle rates.
The line of nodes, which is the direction of \(\overrightarrow{\dot{\theta}}\), coincides with the \(\xi^{\prime}\) axis, and so the components of \(\overrightarrow{\boldsymbol{\theta}}\) with respect to the body axes are fumished by applying only the final orthonormal transformation \(\mathbf{B}^{T}\) :
\[
\begin{aligned}
& \dot{\theta}_{(5)}=\dot{\theta} \cos \sigma \\
& \dot{\theta}_{(\eta)}=-\dot{\theta} \sin \sigma \\
& \dot{\theta}_{(5)}=0
\end{aligned}
\]

No transformation is necessary for the component of \(\overrightarrow{\dot{\sigma}}\), which lies along the \(\zeta\) axis. When these components of the separate angular velocities are added, the components of \(\vec{\omega}\) with respect to the body axes are
\[
\begin{aligned}
& \omega_{(\xi)}=\dot{\psi} \sin \theta \sin \sigma+\dot{\theta} \cos \sigma \\
& \omega_{(\eta)}=\dot{\psi} \sin \theta \cos \sigma-\dot{\theta} \sin \sigma \\
& \omega_{(\zeta)}=\dot{\psi} \cos \theta+\dot{\sigma}
\end{aligned}
\]
or, in matrix form,
\[
\left[\begin{array}{c}
\omega_{(g)}  \tag{A.3}\\
\omega_{(\eta)} \\
\omega_{(t)}
\end{array}\right]=\left[\begin{array}{ccc}
\sin \theta \sin \sigma & \cos \sigma & 0 \\
\sin \theta \cos \sigma & -\sin \sigma & 0 \\
\cos \theta & 0 & 1
\end{array}\right]\left[\begin{array}{l}
\dot{\psi} \\
\dot{\theta} \\
\dot{\sigma}
\end{array}\right]
\]

In addition, the Euler angle rates can be expressed in terms of \(\omega_{(\xi)}, \omega_{(\eta)}\), and \(\omega_{(\xi)}\). Since Euler angle rates are not orthogonal, the inverse of the matrix of Eq. A. 3 yields
\[
\left[\begin{array}{c}
\dot{\psi}  \tag{A.4}\\
\dot{\theta} \\
\dot{\sigma}
\end{array}\right]=\frac{1}{\sin \theta}\left[\begin{array}{ccc}
\sin \sigma & \cos \sigma & 0 \\
\cos \sigma \sin \theta & -\sin \sigma \sin \theta & 0 \\
-\sin \sigma \cos \theta & -\cos \sigma \cos \theta & \sin \theta
\end{array}\right]\left[\begin{array}{l}
\omega_{(\xi)} \\
\omega_{(7)} \\
\omega_{(\xi)}
\end{array}\right]
\]

Similar techniques may be applied to express the components of \(\overrightarrow{\dot{\omega}}\) along the \(x y z\) axes, in terms of Euler angles and rates. Equation A. 4 shows, again, that numerical problems will arise if \(\theta\) is close to the critical values \(n \pi, n=0, \pm 1, \ldots\).

\section*{A. 2 BRYANT ANGLES \({ }^{19}\)}

The Bryant angle convention considers rotations about axes other than those for the Euler angles. The first rotation may be carried out counterclockwise about the \(x\) axis through an angle \(\phi_{1}\); the resultant coordinate system will be labeled \(\xi^{\prime \prime} \eta^{\prime \prime} \zeta^{\prime \prime}\), as shown in Fig. A.4. The second rotation, through an angle \(\phi_{2}\) counterclockwise about the \(\eta^{\prime \prime}\) axis, produces the coordinate system \(\xi^{\prime} \eta^{\prime} \zeta^{\prime}\). Finally, the third rotation, counterclockwise about the \(\zeta^{\prime}\) axis through an angle \(\phi_{3}\), results in the \(\xi \eta \zeta\) coordinate system. The transformation matrices for the individual rotations are
\[
\mathbf{D}=\left[\begin{array}{ccc}
1 & 0 & 0 \\
0 & \mathrm{c} \phi_{1} & -\mathrm{s} \phi_{1} \\
0 & \mathrm{~s} \boldsymbol{\phi}_{1} & \mathrm{c} \phi_{1}
\end{array}\right] \quad \mathbf{C}=\left[\begin{array}{ccc}
\mathrm{c} \phi_{2} & 0 & \mathrm{~s} \phi_{2} \\
0 & 1 & 0 \\
-\mathrm{s} \phi_{2} & 0 & \mathrm{c} \phi_{2}
\end{array}\right] \quad \mathbf{B}=\left[\begin{array}{ccc}
\mathrm{c} \phi_{3} & -\mathrm{s} \phi_{3} & 0 \\
\mathrm{~s} \phi_{3} & \mathrm{c} \phi_{3} & 0 \\
0 & 0 & 1
\end{array}\right]
\]

Hence, the matrix of the complete transformation, \(\mathbf{A}=\mathbf{D C B}\), is
\[
\mathbf{A}=\left[\begin{array}{ccc}
\mathrm{c} \phi_{2} \mathrm{c} \phi_{3} & -\mathrm{c} \phi_{2} \mathrm{~s} \phi_{3} & \mathrm{~s} \phi_{2}  \tag{A.5}\\
\mathrm{c} \phi_{1} \mathrm{~s} \phi_{3}+\mathrm{s} \phi_{1} \mathrm{~s} \phi_{2} \mathrm{c} \phi_{3} & \mathrm{c} \phi_{1} \mathrm{c} \phi_{3}-\mathrm{s} \phi_{1} \mathrm{~s} \phi_{2} s \phi_{3} & -\mathrm{s} \phi_{1} \mathrm{c} \phi_{2} \\
\mathrm{~s} \phi_{1} \mathrm{~s} \phi_{3}-\mathrm{c} \phi_{1} \mathrm{~s} \phi_{2} \mathrm{c} \phi_{3} & \mathrm{~s} \phi_{1} \mathrm{c} \phi_{3}+\mathrm{c} \phi_{1} \mathrm{~s} \phi_{2} \mathrm{~s} \phi_{3} & \mathrm{c} \phi_{1} \mathrm{c} \phi_{2}
\end{array}\right]
\]

Again, it may be necessary to calculate Bryant angles that correspond to a known transformation matrix. This can be done, with the help of formulas derived from


Figure A. 4 Rotations defining Bryant angles.

Eq. A.5, to obtain
\[
\begin{array}{ll}
\sin \phi_{2}=a_{13} & \cos \phi_{2}= \pm \sqrt{1} \\
\sin \phi_{1}=-\frac{a_{23}}{\cos \phi_{2}} & \cos \phi_{1}=\frac{a_{33}}{\cos \phi_{2}}  \tag{A.6}\\
\sin \phi_{3}=-\frac{a_{12}}{\cos \phi_{2}} & \cos \phi_{3}=\frac{a_{11}}{\cos \phi_{2}}
\end{array}
\]

It can be observed again that there exists a critical case, namely, when \(\phi_{2}=\pi / 2+n \pi\), \(n=0, \pm 1, \pm 2, \ldots\), in which the axes of the first and third rotations coincide, so that the rotation angles \(\phi_{1}\) and \(\phi_{3}\) become indistinguishable.

\section*{A.2.1 Time Derivative of Bryant Angles}

The relationship between angular velocity \(\vec{\omega}\) and Bryant angles and rates can be found in a similar fashion to that for the Euler rates. The transformation matrix for the velocity components is
\[
\left[\begin{array}{l}
\omega_{(\xi)}  \tag{A.7}\\
\omega_{(\eta)} \\
\omega_{(\xi)}
\end{array}\right]=\left[\begin{array}{ccc}
\cos \phi_{1} \cos \phi_{3} & \sin \phi_{3} & 0 \\
-\cos \phi_{2} \sin \phi_{3} & \cos \phi_{3} & 0 \\
\sin \phi_{2} & 0 & 1
\end{array}\right]\left[\begin{array}{l}
\dot{\phi}_{1} \\
\dot{\phi}_{2} \\
\dot{\phi}_{3}
\end{array}\right]
\]

The inverse transformation can be found to be
\[
\left[\begin{array}{c}
\dot{\phi}_{1}  \tag{A.8}\\
\dot{\phi}_{2} \\
\dot{\phi}_{3}
\end{array}\right]=\frac{1}{\cos \phi_{2}}\left[\begin{array}{ccc}
\cos \phi_{3} & -\sin \phi_{3} & 0 \\
\sin \phi_{3} \cos \phi_{2} & \cos \phi_{3} \cos \phi_{2} & 0 \\
-\cos \phi_{3} \sin \phi_{2} & \sin \phi_{3} \sin \phi_{2} & \cos \phi_{2}
\end{array}\right]\left[\begin{array}{c}
\omega_{(\xi)} \\
\omega_{(\eta)} \\
\omega_{(\xi)}
\end{array}\right]
\]

It can be seen that Eq. A. 8 fails numerically in the vicinity of the critical values \(\phi_{2}=\pi / 2+n \pi, n=0,+1, \ldots\).

\section*{Appendix B}

\section*{Relationship between}

\section*{Euler Parameters}

\section*{and Euler Angles}

In some kinematics problems, the angular orientation of a body with respect to the global coordinate system is described in terms of Euler angles and it is desired to determine the corresponding set of Euler parameters, or vice versa. There are simple formulas that can be used directly to find one set of variables if the other set is known.

\section*{B. 1 EULER PARAMETERS IN TERMS OF EULER ANGLES}

If the angular orientation of a local coordinate system is described in terms of three Euler angles \(\psi, \theta\), and \(\sigma\), it is possible to find the corresponding Euler parameters. The trace of matrix \(\mathbf{A}\) in terms of Euler angles, from Eq. A.1, is
\[
\operatorname{tr} \mathbf{A}=4 \cos ^{2} \frac{\theta}{2} \cos ^{2} \frac{\psi+\sigma}{2}-1
\]

Then, Eq. 6.25 yields
\[
\begin{equation*}
e_{0}=\cos \frac{\theta}{2} \cos \frac{\psi+\sigma}{2} \tag{B.1}
\end{equation*}
\]

From Eqs. 6.26a-c with \(a_{11}, a_{22}\), and \(a_{33}\) taken from the transformation matrix of Eq. A.1, it is found that
\[
\begin{align*}
& e_{1}=\sin \frac{\theta}{2} \cos \frac{\psi-\sigma}{2}  \tag{B.2}\\
& e_{2}=\sin \frac{\theta}{2} \sin \frac{\psi-\sigma}{2}  \tag{B.3}\\
& e_{3}=\cos \frac{\theta}{2} \sin \frac{\psi+\sigma}{2} \tag{B.4}
\end{align*}
\]

Note that the four Euler parameters can always be determined if the three Euler angles are known.

\section*{B. 2 EULER ANGLES IN TERMS OF EULER PARAMETERS}

The Euler angles can be determined from the Euler parameters by comparing the transformation matrices in terms of Euler angles and Euler parameters: i.e., Eqs. A. 1 and 6.19. Equating the \(a_{33}\) elements of the two matrices yields
\[
\begin{equation*}
\cos \theta=2\left(e_{0}^{2}+e_{3}^{2}\right)-1 \tag{B.5}
\end{equation*}
\]

Equating \(a_{32}\) and \(a_{23}\), we get
\[
\begin{align*}
& \cos \sigma=\frac{2\left(e_{2} e_{3}+e_{0} e_{1}\right)}{\sin \theta}  \tag{B.6}\\
& \cos \psi=-\frac{2\left(e_{2} e_{3}-e_{0} e_{1}\right)}{\sin \theta} \tag{B.7}
\end{align*}
\]

It is clear that for \(\sin \theta=0, \cos \sigma\) and \(\cos \psi\) cannot be evaluated. In this case, from Eq. B.5, \(\cos \theta=1\) yields
\[
\begin{equation*}
e_{0}^{2}+e_{3}^{2}=1 \tag{B.8}
\end{equation*}
\]

Then, from the constraints between Euler parameters, i.e., Eq. 6.21, it is found that
\[
\begin{equation*}
e_{1}^{2}+e_{2}^{2}=0 \tag{B.9}
\end{equation*}
\]
which can be true only if
\[
\begin{equation*}
e_{1}=e_{2}=0 \tag{B.10}
\end{equation*}
\]

Since \(\mathbf{e}=\left[e_{1}, e_{2}, e_{3}\right]^{T}\) consists of the components of \(\vec{e}\) along both the \(x y z\) and \(\xi \eta \zeta\) coordinate axes, Eq. B. 10 indicates that the orientational axis of rotation, denoted by \(\vec{e}\), is along the \(z\) or the \(\zeta\) axis. The ambiguity for \(\theta=k \pi, k=0,1, \ldots\), is discussed in Appendix A in more detail. However, if the \(a_{21}\) elements of the two transformation matrices are used when \(\cos \theta=1\), it is found that
\[
\begin{equation*}
\sin (\psi+\sigma)=2 e_{0} e_{3} \tag{B.11}
\end{equation*}
\]

Now, if either \(\psi\) or \(\sigma\) is given an arbitrary value, the value of the other can be determined.

\section*{Appendix C}

\section*{Coordinate Partitioning}

\section*{with L-U Factorization}

Crout's algorithm LU-I from Sec. 3.3.3 can easily be modified to perform L-U factorization on nonsquare matrices. If L-U factorization with full pivoting is performed on an \(m \times n\) matrix \(\mathbf{A}\), it may result in the following partitioned form:


It is assumed that there are \(s\) redundant rows in the matrix that have ended up as the bottom \(s\) rows after factorization as a result of full pivoting. The rank of this matrix is \(m-s\). The \(\mathbf{L}\) and \(\mathbf{U}\) matrices occupy the \((m-s) \times(m-s)\) top left elements, and \(\mathbf{D}\) is a submatrix all of whose elements begin at approximately zero (i.e., smaller than a specified tolerance). The left \(m-s\) columns of the factored matrix are called the basic columns, and the remaining \(n-m+s\) columns are the nonbasic columns. If all of the rows of \(\mathbf{A}\) are independent, i.e., if \(s=0\), then L-U factorization with full or partial (column) pivoting partitions \(\mathbf{A}\) as follows:


Without any loss of generality, it can be assumed that \(\mathbf{A}\) represents the Jacobian matrix \(\boldsymbol{\Phi}_{\mathbf{q}}\), where all of the \(m\) constraints are independent. Since the elements of \(\mathbf{q}\) cor-
respond to the column indices of \(\boldsymbol{\Phi}_{\mathbf{q}}\), the indices of the columns of \(\mathbf{L}\) (or \(\mathbf{U}\) ) define the dependent (basic) coordinates \(\mathbf{u}\), and indices of the columns of \(\mathbf{R}\) define the independent (nonbasic) coordinates \(\mathbf{v}{ }^{18}\)

Partitioning of \(\mathbf{q}\) into \(\mathbf{u}\) and \(\mathbf{v}\) also corresponds to the partitioning of \(\Phi_{q}\) into \(\boldsymbol{\Phi}_{\mathbf{u}}\) and \(\boldsymbol{\Phi}_{\mathbf{v}}\). In terms of the \(\mathbf{L}, \mathbf{U}\), and \(\mathbf{R}\) matrices,
\[
\begin{align*}
& \boldsymbol{\Phi}_{\mathbf{u}}=\mathbf{L U}  \tag{C.1}\\
& \boldsymbol{\Phi}_{\mathbf{v}}=\mathbf{L R} \tag{C.2}
\end{align*}
\]

In some well-developed L-U factorization subroutines, matrix \(\mathbf{R}\) is replaced by a matrix \(\mathbf{H}\), as follows:
where
\[
\begin{equation*}
\mathbf{H}=-\mathbf{U}^{-1} \mathbf{R} \tag{C.3}
\end{equation*}
\]

This yields
\[
\begin{aligned}
\boldsymbol{\Phi}_{v} & =-\mathbf{L U H} \\
& =-\boldsymbol{\Phi}_{u} \mathbf{H}
\end{aligned}
\]
or
\[
\begin{equation*}
\mathbf{H}=-\boldsymbol{\Phi}_{\mathbf{u}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}} \tag{C.4}
\end{equation*}
\]

The matrix \(\mathbf{H}\) is called the influence coefficient matrix. This matrix relates variations of \(\mathbf{u}\) to variations of \(\mathbf{v}\). This is obtained by taking the differential of the constraint equations \(\boldsymbol{\Phi}=0\) :
\[
\delta \boldsymbol{\Phi}=\boldsymbol{\Phi}_{\mathbf{q}} \delta \mathbf{q}=0
\]
or
\[
\boldsymbol{\Phi}_{\mathbf{u}} \delta \mathbf{u}+\boldsymbol{\Phi}_{\mathbf{v}} \delta_{\mathbf{v}}=0
\]
which yields
\[
\begin{align*}
\delta \mathbf{u} & =-\boldsymbol{\Phi}_{\mathbf{u}}^{-1} \boldsymbol{\Phi}_{\mathbf{v}} \delta \mathbf{v} \\
& =\mathbf{H} \delta \mathbf{v} \tag{C.5}
\end{align*}
\]

The kinematic velocity equations also yield
\[
\begin{equation*}
\dot{\mathbf{u}}=\mathbf{H} \dot{\mathbf{v}} \tag{C.6}
\end{equation*}
\]

\section*{REFERENCES}
1. Baumgarte, J., "Stabilization of Constraints and Integrals of Motion," Concept. Methods Appl. Mech. Eng., Vol. 1, 1972.
2. Chua, L. O., and P. M. Lin, Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques. Prentice-Hall, Englewood Cliffs, N.J., 1975.
3. Goffman, C., Calculus of General Variables. Harper and Row, New York, 1965.
4. Goldstein, H., Classical Mechanics. 2nd. ed., Addison-Wesley, Reading, Mass., 1980.
5. Greenwood, D. T., Principles of Dynamics. Prentice-Hall, Englewood Cliffs, N.J. 1965.
6. Hiller, M. H., "Analytical-Numerical Description of the Kinematical and Dynamical Behavior of Series-Connected Spatial Four-Bar Mechanisms and its Applications," Proc. 6th World Cong. Theory of Machines and Mechanisms, 1983.
7. Jerkovsky, W., "The Structure of Multibody Dynamics Equations," J. Guidance and Control, Vol. 1, No. 3, pp. 173-182, May-June 1978.
8. Kim, S. S., and M. J. Vanderploeg, "QR Decomposition for State Space Representation of Constrained Mechanical Dynamic Systems," ASME J. Mech., Trans., and Auto. in Design, Vol. 108, No. 2, pp. 183-188, June 1986.
9. Kim, S. S., and M. J. Vanderploeg, "A General and Efficient Mcthod for Dynamic Analysis of Mechanical Systems Using Velocity Transformations," ASME J. Mech., Trans., and Auto. in Design, Vol. 108, No. 2, pp. 176-182, Junc 1986.
10. Liang, C. G., and G. M. Lance, "A Differentiable Null Space Method for Constrained Dynamic Analysis," ASME Paper 85-DET-86.
11. Mani, N. K., E. J. Haug, and K. E. Atkinson, "Application of Singular Value Decomposition for Analysis of Mechanical System Dynamics," ASME J. Mech., Trans., and Auto. in Design, Vol. 107, No. 1, pp. 82-87, March 1986.
12. Nikravesh, P. E., and I. S. Chung, "Application of Euler Parameters to the Dynamic Analysis of Three Dimensional Constrained Mechanical Systems," ASME J. Mech. Design, Vol. 104, No. 4, pp. 785-791, October 1982.
13. Nikravesh, P. E., O. K. Kwon, and R. A. Wehage, "Euler Parameters in Computational Kinematics and Dynamics, Part 2," ASME J. Mech., Trans., and Auto. in Design, Vol. 107, No. 3, pp. 366-369, September 1985.
14. Nikravesh, P. E., and M. Srinivasan, "Generalized Coordinate Partitioning in Static Equilibrium Analysis of Large-Scale Mechanical Systems," Int. J. Numerical Methods in Engineering, Vol. 21, pp. 451-464, 1985.
15. Nikravesh, P. E., R. A. Wehave, and O. K. Kwon, "Euler Parameters in Computational Kinematics and Dynamics, Part 1," ASME J. Mech., Trans., and Auto. in Design, Vol. 107, No. 3, pp. 358-365, September 1985.
16. Orlandea, N., M. A. Chace, and D. A. Calahan, "A Sparsity-Oriented Approach to Dynamic Analysis and Design of Mechanical Systems," ASME J. Engineering for Industry, Vol. 99, pp. 773-784, August 1977.
17. Paul, B., Kinematics and Dynamics of Planar Machinery. Prentice-Hall, Englewood Cliffs, N.J., 1979.
18. Wehage, R. A., and E. J. Haug, "Generalized Coordinate Partitioning of Dimension Reduction in Analysis of Constrained Dynamic Systems," ASME J. Mech. Design, Vol. 104, pp. 247-255, January 1982.
19. Wittenburg, J., Dynamics of Systems of Rigid Bodies. Teubner, Stuttgart, 1977.
20. Wittenburg, J., "A New Correction Formula for Euler-Rodriguez Parameters," ZAMM, Vol. 62, pp. 495-497, 1982.

\section*{BIBLIOGRAPHY}

Andrews, G. C., and H. K. Kesavan, "The Vector Network Model: A New Approach to Vector Dynamics," Mechanism and Machine Theory, Vol. 10, pp. 509-519, 1975.
Barman, N. C., "Design Sensitivity Analysis and Optimization of Constrained Dynamic Systems." Ph.D. Thesis, The University of Iowa, 1979.
Baumgarte, J., "Stabilisierumg Von Bindungen im Lagraneshen Formalismo," ZAMM 58, T360T361, 1978.
Bottema, O., and B. Roth, Theoretical Kinematics. North-Holland, 1979.
Bryson, A. E., and W. F. Denham, "A Steepest-Ascent Method for Solving Optimum Programming Problems," ASME J. Applied Mechanics, pp. 247-257, 1962.
Calahan, D. A., Computer-Aided Network Design. New York, McGraw-Hill, 1962.
Chace, M. A., and P. M. Sheth, "Adaptation of Computer Techniques to the Design of Mechanical Dynamic Machinery," ASME Paper 73-DET-58, 1973.
Chace, M. A. and D. A. Smith, "DAMN-A Digital Computer Program for the Dynamic Analysis of Generalized Mechanical Systems," SAE paper 710244, January 1971.
Chang, C. O., and P. E. Nikravesh, "An Adaptive Constraint Violation Stabilization Method for Dynamic Analysis of Mechanical Systems," ASME J. Mech., Trans., and Auto. in Design, Vol. 107, No. 4, pp. 488-492, 1985.
Chang, C. O., and P. E. Nikravesh, "Optimal Design of Mechanical Systems with Constraint Violation Stabilization Method," ASME J. Mech., Trans., and Auto. in Design, Vol. 107, No. 4, pp. 493-498, 1985.
Clifford, W, "Preliminary Sketch of Biquaternions," Proc. London Math. Soc., Vol. IV, 1873.
Conte, S. D., and C. D. Boor, Elementary Numerical Analysis, 3rd. ed., New York: McGrawHill, 1980.
Denavit, J., Displacement Analysis of Mechanisms Based on \((2 \times 2)\)-Matrices of Dual Numbers. VDI-Berichte, 1958.
Deo, N., Graph Theory with Applications in Engineering and Computer Science. Englewood Cliffs, N.J.: Prentice-Hall, 1974.
Dijkstra, E. W., "A Note on Two Problems in Connection with Graphs," Numerische Mathematik, Vol. 1, 1959.
Dimentberg, The Screw Calculus and its Applications in Mechanics. Moscow: Nauka, 1965.
Duffy, J., Analysis of Mechanisms and Robot Manipulator. London: Arnold, 1980.
Gear, C. W., "Numerical Solution of Differential-Algebraic Equations," IEEE Transaction on Circuit Theory, Vol. CT-18, pp. 89-95, January 1981.
Gupta, G. K., R. Sacks-Davis, and P. E. Tischer, "A Review of Recent Developments in Solving ODEs," Computing Surveys, Vol. 17, No. 1, pp. 5-47, March 1985.
Hamad, B. M., "Optimal Design of Vibratory Systems." Ph.D. Thesis, The University of Wisconsin, 1968.
Haug, E. J., V. N. Sohoni, S.S. Kim, and H. G. Seong, "Vehicle Suspension Dynamics Optimization," Proc. Int. Conf. on Vehicle Design Analysis, London, June 21-24, 1983.

Haug, E. J., R. A. Wehage, and N. C. Barman, "Design Sensitivity Analysis of Planar Mechanisms and Machine Dynamics," ASME J. Mechanical Design, Vol. 103, pp. 560-570, 1981.
Hemami, H., "Some Aspects of Euler-Newton Equation of Motion," Ingenieur-Archiv, Vol. 52, pp. 167-176, 1982.
Hiller, M., "The Dynamics of Spatial Multi-Bar Mechanisms," German-Japanese Seminar on Nonlinear Problems in Dynamical Systems - Theory and Applications, ed. M. Hiller and H. Sorg, Stuttgart, Germany, 1984.
Hiller, M., and C. Woernle, "A Unified Representation of Spatial Displacements," J. Mech. and Machine Theory, Vol. 19, No. 6, pp. 477-486, 1984.
Huston, R. L., and J. W. Kaman, A Discussion on Constraint Equations in Multibody Dynamics," Mechanics Research Communications, Vol. 9(4), pp. 251-256, 1982.
Huston, R. L., and C. E. Passerello, "On Multi-Rigid Body System Dynamics," Computers and Structures, Vol. 19, pp. 439-446, 1979.
Huston, R. L., and C. E. Passerello, "Multibody Structural Dynamics Including Translation Between the Bodies," Computers and Structures, Vol. 11, pp. 715-720, 1980.
Huston, R. L., C. E. Passerello, and M. W. Harlow, "Dynamics of Multirigid Body Systems," J. Applied Mechanics, Vol. 45, No. 4, pp. 889-894, 1978.
Kamman, J. W., and R. L. Huston, "Dynamics of Constrained Multibody Systems," J. Applied Mechanics, Vol. 51, No. 4, pp. 387-904, 1984.
Kamman, J. W., and R. L. Huston, "Constrained Multibody System Dynamics," Computers and Sructures, Vol. 18, pp. 999-1003, 1984.
Kane, T. R., "Solution of Kinematical Differential Equations for a Rigid Body," ASME J. Appl. Mech., Vol. 40, pp. 109-113, March 1973.
Kane, T. R., and D. A. Levinson, Dynamics Theory and Applications. New York: McGrawHill, 1985.
Kane, T. R., and C. F. Wang, "On The Derivation of Equations of Motion," J. Soc. Indust. Appl. Math., Vol. 13, No. 2, June 1965.
Keat, J., "Dynamics Equations of Multibody Systems with Applications to Space Structure Development." Ph.D. Dissertation, Massachusetts Institute of Technology, May, 1983.
Khulief Y. A., and A. A. Shabana, "Dynamics of Multibody Systems With Variable Kinematic Structure," ASME Paper 85-DET-83, 1985.
Khulief Y. A., and A. A. Shabana, "Impact Responses of Multibody Systems with Consistent and Lumped Masses," J. Sound and Vibration, Vol. 104 No. 2, pp. 187-208, 1986.
Kim, S. S., "State Space Formulation for Multibody Dynamic Systems." Ph.D. Dissertation, The University of Iowa, 1984.
Krishnaswami, P., and M. A. Bhatti, "A General Approach for Design Sensitivity Analysis of Constrained Dynamic Systems," ASME Paper 84-DET-132, 1984.
Kruskal, J. B. Jr., "On the Shortest Spanning Subtree of Graph and the Traveling Salesman Problems," Proc. Am. Math. Soc., Vol. 7, 1956.
Liang, C. G., and G. M. Lance, "A Differentiable Nüll Space Method for Constrained Dynamic Analysis," ASME Paper 85-DET-86, 1985.
Orlandea, N. V., "Node Analogous Sparsity-Oriented Methods for Simultation of Mechanical Dynamic Systems." Ph.D. Thesis, University of Michigan, 1973.
Ortega, J. M., and W. G. Poole, Jr., An Introduction to Numerical Methods for Differential Equations. Pitman, Marshfield, M. A., 1981.

Park, T., and E. J. HAUG, "A Hybrid Numerical Integration Method for Machine Dynamic Simulation," ASME Paper 85-DET-59, 1985.
Passerello, C.E., and R.L. Huston, "An Analysis of General Chain Systems," NASA CR-127924, Report No. N72-30532, 1972.
Paul, B., "Analytical Dynamics of Mechanisms: A Computer Oriented Overview," Mechanism and Machine Theory, Vol. 10, pp. 481-507, 1975.
Paul, B., and D. Krajcinovic, "Computer Analysis of Machines with Planar MotionPart I: Kinematics; Part II: Dynamics," J. Applied Mechanics, Ser. E, Vol. 37, pp. 697-712, 1970.

Ralston, A., and P. Rabinowitz, A First Course in Numerical Analysis, 2nd ed., New York: McGraw-Hill, 1978.
Schielen, W.O., "Dynamics of Complex Multibody Systems," SM Archives, Vol. 9, pp. 297308, 1972.
Schielen, W. O., "Computer Generation of Equations of Motion," Computer Aided Analysis and Optimization of Mechanical Systems, ed. E. J. Haug, Heidelburg: Springer-Verlag, 1984.
Schielen, W. O. and E.J. Dreuzer, "Symbolic Computerized Derivation of Equations of Motion," Dynamics of Multibody Systems, ed. K. Magnus. pp. 290-305, Heidelburg: SpringerVerlag, 1978.
Shabana, A. A., "Automated Analysis of Constrained Systems of Rigid and Flexible Bodies," ASME Paper 85-DET-29, 1985.
Shabana, A. A., and R. A. Wehage, "Variable Degree of Freedom Component Mode Analysis of Inertial Variant Flexible Mechanical Systems," ASME J. Mechanical Design, Vol. 105, no. 3, pp. 1193-1205, September, 1983.
SHETH, P. N., "A Digital Computer Based Simulation Procedure for Multiple Degree of Freedom Mechanical Systems with Geometric Constraints," Ph.D. Thesis, University of Wisconsin, 1972.

Sheth, P. N., and J. J. Uicker Jr., "IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkages," J. Engrg. For Industry, Vol. 94, p. 454, 1972.
Smith, D. H., "Reaction Forces and Impact in Generalized Two-Dimensional Mechanical Dynamic Systems." Ph.D. Dissertation, The University of Michigan, 1973.
Song, J. O., and E.J. HaUG, "Dynamic Analysis of Planar Flexible Mechanisms," Computer Methods in Applied Mechanics and Engineering, Vol. 24, pp. 359-381, 1976.
Srinivasan, M., "Multi-Rate Numerical Integration in Design and Analysis of Flexible Mechanical Systems." Ph.D. Thesis, The University of Iowa, 1984.
Sunada, A. A., and S. Dubowsky, "The Application of Finite Element Methods to the Dynamic Analysis of Flexible Spatial and Co-Planar Linkage Systems," J. Mechanical Design, Vol. 103, pp. 643-651, July 1981.
'Tong, K. N., Theory of Mechanical Vibration. New York: John Wiley, 1960.
Tse, F. S., I. E. Morse, and R.T. Hinkle, Mechanical Vibration. 2nd ed., Boston: Allyn and Bacon, 1978.
Uicker, J. J., Jr., "Dynamic Behavior of Spatial Linkages," J. Engrg. Indust., Trans. ASME, Ser. B, Vol. 91, pp. 251-265, 1969.
Wehage, R. A., "Generalized Coordinate Partitioning in Dynamic Analysis of Mechanical Systems." Ph.D. Dissertation, The University of Iowa, 1980.

Wehage, R. A., "Quaterions and Euler Parameters-A Brief Exposition," Comp. Aided Anal. and Opt. of Mech. Systems, ed. E. J. Haug, Heidelburg: Springer-Verlag, 1984.
Willmert, K. D., and R.L. Fox, "Optimal Design of A Linear Multi-Degree of Freedom Shock Isolation System," J. Engrg. For Indus., Vol. 96, pp. 465-471, 1972.
Winfrey, R. C., "Elastic Link Mechanism Dynamics," ASME J. Engrg. for Industry, Ser. B, Vol. 93, No. 1, pp. 268-272, February 1971.
Winfrey, R. C., "Dynamic Analysis of Elastic Link Mechanisms by Reduction of Coordinates," ASME J. Engineering for Industry, Ser. B, Vol. 94, No. 2, pp. 577-582, May, 1972.
Wittenburg, J., "Dual Quaternions in the Kinematics of Spatial Mechanisms," Comp. Aided Anal. and Opt. of Mech. Systems, ed. E. J. Haug, Heidelburg: Springer-Verlag, 1984.
Yang, A. T., Calculus of Screws: Basic Questions of Design Theory, ed. W. R. Spiller, pp. 265281, New York: American Elsevier Publishing Co., 1974.
Yang, A. T., and Freudenstein, F., "Application of Dual-Number Quaternion Algebra to the Analysis of Spatial Mechanisms," J. Appl. Mech. 86, pp. 300-308, 1964.
Yuan, M. S. C., and F. Freudenstein, "Kinematic Analysis of Spatial Mechanisms by Means of Screw Coordinates," J. Engrg. in Industry, pp. 61-73, 1971.

\section*{Index}

\section*{A}

Acceleration analysis, 43, 101, 105, 128, 200, 202
Acceleration:
angular, 175
identities with Euler parameters, 175
independent, 46
vector of, 46
Acceleration equations, 46, 49, 102, 201-2
right side, 49, 104-5, (table) 109, 202
modified, (table) 203, 298, (table) 299
Active point of view, 157
Actuator, 222, 231, 255, 273, 275
subroutine, 259, 264
ADAMS, 327
Adams-Bashforth algorithms, 308
Adams-Moulton algorithms, 308
Algebraic-differential equations, 228, 244, 253, 313, 328-35
Algebraic equations:
linear, 50-66
nonlinear, 66-72, 101
Algorithms:
dynamic analysis:
constrained systems:
CP-1, 321
CP-2, 323

CS-1, 320
DI-4, 318
JC-1, 334
unconstrained systems:
DI-1, 314
DI-2, 315
DI-3, 317
kinematic analysis:
K-I, 46
K-II, 50
linear algebraic equations:
LU-I, 58
nonlinear algebraic equations:
NR-I, 68
static equilibrium analysis:
SE-1, 343
Analysis, 6, 16
methods, 6-9
Angular:
acceleration, 175
identities with Euler parameters, 175
orientation, 39, 153
velocity, 172, 174
identities with Euler parameters, 174-75
numerical integration algorithm, 317
Animation, computer, 17, 18
Appended driving constraint method, 48
Algorithm K-II, 50

Artificial constraint, 246
Automated factory, 2
Axis of rotation:
instantaneous, 157
intermediate, 180
orientational, 157, 158
relative, 177

\section*{B}

Base body:
fixed, 332
floating, 332
Bodies, collection 3
Body: (see also Link)
outline, 5
shape, 5
Bryant angles, 157, 162, 347, 351
critical values, 352
time derivatives, 352

\section*{C}

Centroid, 210
Centroidal coordinate system, 215
Chain (see Kinematic, chain)
Chasles' theorem, 157
Clockwise rotation, 157
Closed-form solution, 9
Closed-loop, 334 (see also Mechanisms) cut, 334, 345
Computational techniques, 3
Computer-aided:
analysis (CAA), 1, 2
design (CAD), 1
drafting, 1
engineering (CAE), 1
manufacturing (CAM), 1
manufacturing design, 1
product design, 1
Constraint:
artificial, 246
driving, 48, 49, 100, 101
subroutine, 126, 133
equality, 41
equations, \(40,41,44,49\)
conversion, 297
number of, 13
force, 222, 223, 228, 244, 266, 269, 276, 292
revolute joint, 237
revolute-revolute joint, 240
translational joint, 242
ground, 98
subroutine, 125
holonomic, 41, 189
independent, 42
inequality, 41
Jacobian (see Jacobian matrix)
nonholonomic, 41
not integrable, 41
redundant, 41
simple, 98, 199
subroutine, 126, 133
stabilization method, 319, 335
Algorithm CS-1, 320
violation, 316, 319
maximum error allowed, 122
stabilization, 319
Constraints between:
parallel vectors, 188
perpendicular vectors, 188
two bodies, 189
two vectors, 186
Coordinate:
body-fixed, \(38,77,154\)
Cartesian, 12, 13, 38, 77, 153, 313, 330
dependent, 4I, 44, 321, 324, 340, 356
driven, 44
driving, 44
generalized, 10, 13, 313
global, 77, 153
independent, 41, 44, 321, 324, 340, 356
joint, 330
La Grangian, 38, 77, 313
local (see Base body, fixed)
number of, 13, 39, 101
partitioning method:
automatic process, 324
dynamic analysis, 321
Algorithm CP-1, 321
Algorithm CP-2, 323
kinematic analysis, 43
Algorithm K-I, 46
static equilibrium, 340, 342
with LU factorization, 355
relative, 11,13
rotational, 39, 154 ;
translational, 39, 154
Coordinates, vector of, 38, 39, 222
Counterclockwise rotation, 77, 157
Couple, 213

Cramer's rule, 51
Critical damping, 320
Crout's method, 57, 63, 355
Algorithm LU-I, 58-93
Curve representation, 89

\section*{D}

DADS, 321
Damper, 15, 222, 234, 237, 255, 271
subroutine, 259, 264
Damping:
coefficient, 234, 237
critical, 320
DAP, 253, 254
Deformable body, 36
Degrees of freedom, 3, 40, 42
Design, 6
sensitivity analysis, 1
Differential-algebraic equations (see Alge-braic-differential equations)
Differential equations (see Ordinary differential equations)
Digitizer tablet, 15
Direction cosines, 20, 155, 158, 164, 348
Distance, kinematic, 332
Divergence, 68
Double parallel-crank mechanism, 41, 42
Driving constraint method, appended, 48
Algorithm K-II, 50
Dump truck, 273
Dynamic analysis, 228, 255
forward, 247
inverse, 247
numerical methods, 313
program (DAP), 253-77
Dynamics, 6
basic concepts, 208-26
planar motion, 227
spatial motion, 289

\section*{E}

Efficiency, numerical, 10, 13
Eigenvalues, 310
End-effector, 247
Equations of motion:
a constrained body, 292
an unconstrained body, 219, 227, 291
constrained bodies, 222, 228
planar, 244, 254
spatial, 296-97
rotational, 215, 217, 291-93, 297
translational, 211
unconstrained bodies, 221, 228
planar, 242, 253
spatial, 294-96
Equilibrium (see also Static, equilibrium), 275
equations, 339
Error:
numerical, 14, 302
Euler parameters, 316
joint coordinates, 333
round-off, 63, 64, 302
sum of squares, 316
tolerance, \(68,70,122\)
total, 302
truncation, 302, 310
Euler angles, 153, 157, 162, 164, 347
conventions, 347
critical values, 348,349
in terms of Euler parameters, 354
time derivatives, 349
Euler parameters, 153 158, 160, 289
in terms of Euler angles, 353
identities, 166-72
partial derivative, 171
time derivatives, \(168,169,171\)
time derivatives, 174
Euler's theorem, 157, 158, 160, 176
Exact solution, 9
Explicit algorithm, 308

\section*{F}

Factory automation, 1
Feedback control theory, 319
Fictitious damping, 344
Finite rotation, 180
Five-bar mechanism, 9
Force:
gravitational, 229, 341
vector of:
body, 221, 229
system, 222, 228
Force-deformation, 233
Forward:
dynamic analysis, 247
Euler algorithm, 303
Four-bar mechanism, 3, 11-13, 40, 43, 48, 135, 268
Frictionless constraints, 223

\section*{G}

Gaussian:
elimination, 51, 55, 324
back substitution, 51, 52
forward elimination, 51, 52, 54
methods, 51
Gauss-Jordan reduction, 52
Gear algorithms, 310,328
General-purpose program, 9, 13, 14
Gram-Schmidt process, 327
Graphics, computer, 15-18
Graph theory, 333
Gravitational force, 229, 341

\section*{H}

Harmonic motion, 278
HARWELL, Library of Subroutines, 144

\section*{I}

Implicit:
algorithms, 308
function theorem, 45
Indices:
column, 56, 63
row, 56
Inertia:
principal axes, 219
tensor, 218
global, 217
Infinitesimal:
displacement, 223
mass, 216
Influence coefficient matrix, 356
Initial conditions (values), 254, 301, 314, 315
Initial-value problems, 301, 307
Input/Output, 15, 17, 210, 255
Input prompts:
DAP, 255, 258-59, 267
KAP, 120, 123-27, 134
Integration:
arrays, 254, 313
numerical, 254, 301-11
Inverse dynamics, 247

\section*{J}

Jacobian matrix, 45, 103, (table) 106, 200, (table) 201
modified, 201, (table) 203, (table) 299
rank reduction, 189
Joint (see Kinematic, joints):
coordinate method:
dynamic analysis, 330
static equilibrium analysis, 345
reaction force (see Constraint, force)

\section*{K}

KAP, 119, 253,
Kinematic:
acceleration equations (see Acceleration equations)
analysis, 36, 42, 127
numerical methods, 35
program (KAP), 119-40
chain, 36,331
constraints, 80 (see also Constraints and Kinematic, joints)
joints (see also Kinematic pairs), 36
ball (see Spherical)
cam-follower, 37, 38, 90, 93-96
composite, 84, 196
cylindrical, 189, 192
gears, 37, 38, 86, 87
globular (see also spherical), 332
Hook, (see universal)
point-follower, 97
prismatic (see translational)
rack and pinion, 86,88
revolute, 38, 81, 120, 191, 255, 330
subroutine, 124,131
revolute-cylindrical, 198
revolute-revolute, 84,197
revolute-translational, 85
screw, 38, 193
sliding (see Translational)
spherical, 38, 190
spherical-spherical, 196
spur gears, (see gears)
translational, \(37,38,83,120,193,255\), 330
subroutine, 125, 132
universal, 190, 330
modeling, 105
pairs (see also Kinematic, joints), 37, 38
lower, 37, 80, 189
higher, 37, 80
Kinematics, 6
basic concepts, 35-42
planar motion, 77
spatial motion, 186
synthesis, 36
Kinetics, 6
Kinetostatic analysis, 247

\section*{L}

LaGrange multiplier, 224, 228, 237-42, 244-47, 254, 292, 296, 316
artificial, 292, 315
Linear algebraic equations, 50-66
Line:
of nodes, 348
of translation, 83
Link (see also Body), 36
coupler, 41
driving, 100
fixed, 36
frame, 36
ground, 36
L-U factorization, 56, 327, 355
Algorithm LU-I, 58-93

\section*{M}

MAIN computer program for:
dynamic analysis, DAP, 255
kinematic analysis, KAP, 122
linear algebraic equations, 65
nonlinear algebraic equations, 72
Mass:
center, 210
matrix:
body, 220
system, 221, 294, 295
Mathematical constraint, 314, 315
Matrix:
column, 21
diagonal, 22
difference, 22
differentiation, 28
dimension, 21
identity, 22
permuted, 49
inverse, 23
linearly (in)dependent rows, 23
lower triangular, 57
multiplication by a scalar, 22
null, 22
orthogonal, 24, 156
orthonormal, 24, 348
product, 22
quasi-diagonal, 22
rank, 23
row, 21
singular, 23
skew-symmetric, 23, 25
sparse, 110, 144
square, 22
symmetric, 23
time derivative, 29
trace, 161, 353
transpose, 21
unit (see identity matrix)
upper triangular, 51, 57
zero (see null matrix)
MAXA, 120
MAXIA, 120
Mechanics, branches, 6
Mechanism, 36
closed-loop, 36, 37
multiloop, 37
open-loop (chain), 36, 37
planar, 36, 77
single-loop, 37
spatial, 36
Mechanisms, kinematically equivalent, 41, 42
Minimization, 342, 345
Moment, 212, 229
conversion, 289
Moments:
of inertia, 218
polar, 227
principle, 212
Motion:
planar, 3, 77
conditions, 224
spatial, 3
Multibody system, 3
schematic presentation, 5, 6
Multistep algorithm, 308, 309

\section*{N}

Newton defference, 68
Newton-Euler equations, 220
Newton-Raphson, 66-67, 311, 323, 328, 340
iteration number, 67
iterations, maximum number, 70, 122
Algorithm NR-I, 68
residuals, 68

Vector, cont.
scalar (or dot) product, 20, 23, 24
time derivative, 28
vector (or cross) product, 20
collinear, 19
coplanar, 19
differentiation, 28
dimension, 27
sliding, 214
sum, 20, 14
unit, 19, 154
zero (see null vector)
Velocities, vector of, 45, 220
Velocity:
analysis, \(43,101,128,200,202\)
angular, 172, 174
identities with Euler parameters, 174-75
numerical integration algorithm, 317
dependent, 45
equations, 42, 45, 49, 102, 298
modified, 298
independent, 45
transformation, 330
Vibration, free, 279
Virtual:
displacement, 223, 290
work, 291

\section*{W}

Working arrays, 120, 255
Work of a force, 223

\section*{CONIPUTVR-AIDED ANAMYSSS \({ }^{0}\) MIDCHANCAL SYSTVINS \\ Parviz E.Nikravesh}```


[^0]:    ${ }^{\dagger}$ or degree

[^1]:    ${ }^{\dagger}$ The abbreviation CAD is commonly used for both computer-aided drafting and computer-aided design. Most of the CAD systems available today are intelligent computerized drafting systems with limited design capability.

[^2]:    ${ }^{\dagger}$ In most textbooks the three unit vectors along the $x, y$, and $z$ axes are denoted by $\vec{i}, \vec{j}$, and $\vec{k}$. In this text, since $i$ and $j$ are used to denote indices of bodies, to avoid any confusion, unit vectors are denoted by $\vec{u}_{(\lambda)}$, $\vec{u}_{(y)}$, and $\vec{u}_{(z)}$.

[^3]:    ${ }^{\dagger}$ It will be shown in the forthcoming sections how redundant equations can be found by such techniques as Gaussian elimination or L-U factorization.

