

**COMPUTER-AIDED KINEMATICS AND DYNAMICS  
OF MECHANICAL SYSTEMS  
Volume II: Modern Methods**

Edward J. Haug  
Carver Distinguished Professor Emeritus  
Department of Mechanical and Industrial Engineering  
The University of Iowa  
[echaug@gmail.com](mailto:echaug@gmail.com)

Third Edition  
May 2022

Copyright 2022  
All rights reserved

## PREFACE

This electronic text uses modern methods of differential geometry to (1) obtain ordinary differential equations (ODE) of motion for mechanical system dynamics, (2) construct numerical algorithms for their solution, and (3) implement equation construction and numerical solution in the form of MATLAB computer code. Mechanical systems treated include rigid multibody systems that are subject to holonomic and nonholonomic kinematic constraints, including the effects of friction and input-output influence of nonredundant and redundant manipulators. A first volume of this series (Haug, 1989), Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume I: Basic Methods, was based on the Lagrange multiplier form of differential-algebraic equations (DAE) of dynamics that was state-of-the-art in the late 1900s.

### DAE vs ODE of Dynamics

While the DAE formulation of mechanical system dynamics is easily derived, the resulting equations are extraordinarily difficult to solve. A landmark paper by Petzold (1982), entitled roughly “DAE” are not ODE”, identified severe challenges in solving the Index 3 DAE of multibody dynamics. Nevertheless, the literature has continued for half a century to focus on methods for numerical solution of DAE. In stark contrast with the DAE approach, ODE formulations are more difficult to derive, but once obtained can be reliably and accurately solved using well established numerical integration methods.

A manifold theoretic foundation for ODE formulation of mechanical system dynamics is presented, based on the mathematical theory of differential geometry. A fascinating insight provided by this approach is that Lagrange multipliers and the associated DAE of dynamics could have been avoided for two and one half centuries since landmark dynamics contributions by d'Alembert (1743) and Lagrange (1788) in the 18<sup>th</sup> century. Lagrange multipliers are introduced in Chapter 4 of the text only to represent constraint reaction forces and are not used in formulating ODE of mechanical system dynamics for holonomic systems in Chapter 5, nonholonomic systems in Chapter 6, and manipulators in Chapter 9. It is interesting to ruminante on what might have been a world without Lagrange multipliers and DAE. In hindsight, one might conclude that *the only good DAE is one with an equivalent ODE*.

### The Modern Focus of Computational Kinematics and Dynamics

This text presents foundations for computational kinematics and dynamics of planar and spatial mechanical systems. The approach taken emphasizes systematic methods for formulation and numerical solution of the governing equations of kinematics and dynamics. Equally important is the creation of numerical algorithms and their implementation in MATLAB computer code that reliably simulates the kinematics and dynamics of a broad spectrum of mechanical systems.

The focus of the text is on simulation of mechanical systems that are encountered in diverse fields of application. Emphasis is placed on modeling mechanical systems and implementing equations of kinematics and dynamics in MATLAB computer code that is made available with the text for use in analyzing the motion of examples presented and creating simulations of mechanical systems of interest to the reader. While emphasis is on spatial systems, less complicated planar systems that make concepts easier to grasp are studied. In

addition to code for specific examples, general-purpose computer codes are provided for simulation of planar and spatial mechanical system kinematics and dynamics.

The Author's goal is an electronic text that is based on a minimal set of fundamental laws of mechanics, basic mathematics, and numerical methods that provide the foundation for computational treatment of mechanical system kinematics and dynamics. Rather than publishing another printed book on the subject, the intention is to create an electronic document that can be used and evolved by faculty teaching courses that introduce students to the field and by practicing engineers. Computer implementation and demonstration of methods are presented, in the form of MATLAB programs that can be used by students and practicing engineers in the learning process and in applications.

## Organization of the Text

Basic elements of mechanical system dynamics are introduced in Chapter 1, using examples to identify challenges and opportunities to be encountered. The nonlinear character of mechanical system performance is emphasized, to set the stage for developments that follow.

The body of the text begins with Chapter 2 on representation of position and orientation of bodies, with primary focus on spatial motion. Basics of vector analysis, matrix algebra, multivariable calculus, and differential geometry are summarized, as the foundation for a rigorous but practical formulation of mechanical system kinematics and dynamics.

Libraries of kinematic joints between pairs of bodies are presented in Chapter 3, including associated velocity and acceleration equations. Methods of differential geometry are used to define regular kinematic configuration space and to show that it is a differentiable manifold that is naturally partitioned into maximal, singularity free, path connected components. Methods for kinematic analysis on a time grid are presented and singularities that can arise in improperly designed or poorly modeled mechanisms are analyzed, with the help of examples. General-purpose MATLAB computer codes are provided for kinematic analysis of planar and spatial systems. Results using the codes for a spectrum of examples are presented

A variational formulation of the equations of motion for rigid bodies and multibody systems is presented in Chapter 4. Equations of motion are derived for systems of particles, planar rigid bodies, and spatial rigid bodies, based on only Newton's three laws of dynamics of particles and d'Alembert's variational principle for constrained systems of rigid bodies. Runge-Kutta numerical integration methods for both first and second order ODE are presented and implemented, in the form of MATLAB computer code that is provided. Numerical solutions of ODE for examples treated in the chapter are presented and analyzed. Constraint reaction forces associated with joints between bodies are represented using Lagrange multipliers and computed using results of ODE-based dynamic simulation.

A broadly applicable ODE formulation of multibody dynamics for holonomically constrained systems is presented in Chapter 5, using a constraint tangent space differential geometry formulation. Conditions for existence, uniqueness, and differentiability of solutions of the resulting equations of motion with respect to design parameters are presented. Solution algorithms are presented that enable application of explicit and implicit ODE numerical integration methods, with error control. Examples are presented to test methods and provide the basis for experimentation and analysis. Index 0 DAE that are algebraically equivalent to ODE are derived, using the tangent space formulation, and implemented with both explicit and implicit numerical integration algorithms. General-purpose MATLAB computer codes are

provided for simulation of planar and spatial multibody system dynamics. They are used in applications of moderate complexity that would be oppressive if treated with ad-hoc derivation and programming of the highly nonlinear equations of motion.

While nonholonomic systems are not common in mechanical system dynamics, significant applications motivate their treatment in Chapter 6. The tangent space ODE formulation of Chapter 5 is extended to provide a broadly applicable formulation and numerical solution methods for dynamics of nonholonomic systems. MATLAB computer codes for treatment of examples and studies of excursions are provided.

A basic theory of the DAE of holonomic multibody dynamics is presented in Chapter 7, based on existence and uniqueness of solutions and continuous dependence of solutions on model parameters presented in Chapter 5. Methods for obtaining approximate solutions of the DAE are presented and implemented in general-purpose MATLAB computer codes for simulation of planar and spatial systems. Examples are studied to compare properties of DAE solution methods with those of ODE methods presented in Chapter 5.

Friction effects in multibody systems are introduced in Chapter 8. Kinematic formulations are presented that account for contact geometry in joints and determine constraint contact forces, as functions of Lagrange multipliers in the Index 0 DAE of dynamics. Continuous models of friction forces are introduced as realistic models of multibody dynamics with friction, without complexities associated with the physically dubious discontinuous Coulomb friction model. Situations that lead to static friction lock, or stiction, that occur in applications are studied. Numerical solution methods based on the Index 0 DAE formulation and implicit numerical solvers are presented and shown to yield realistic solutions of problems that include friction and stiction effects in planar and spatial mechanical systems. General-purpose MATLAB computer codes are provided, for dynamic simulation of planar and spatial mechanical systems with friction.

A manipulator kinematics formulation that includes input coordinates, mechanism generalized coordinates, and output coordinates is presented in Chapter 9 that naturally partitions the spectrum of nonredundant manipulators into four disjoint categories that are rigorously treated using analytical and numerical input-output mappings. Criteria that assure local existence and uniqueness of forward kinematic mappings that are needed in manipulator control and inverse kinematic mappings that are needed in manipulator programming are presented. It is shown that these mappings define a differentiable manifold structure for the manipulator configuration space. Differential geometry results of Chapter 3 show that the regular manipulator configuration space is composed of maximal, disjoint, path connected, singularity free domains of functionality on which the manipulator can be reliably programmed and controlled. ODE of manipulator dynamics are embedded in singularity free domains of functionality, without the need for ad-hoc derivation. Finally, the formulation is extended to redundant manipulators with more input coordinates than output coordinates. It is shown that forward kinematics and dynamics of redundant manipulators have the same properties as nonredundant manipulators. The existence of set-valued inverse kinematics for redundant manipulators gives rise to unique control opportunities, but analytical complexity that is dealt with using methods of differential geometry.

This organization has been selected to permit the student and practicing engineer to master basic concepts and gain experience in modeling and analysis, prior to building upon the foundation for large scale dynamic system simulation.

## **Availability of the Text and Computer Code**

While the text and associated computer code are copyrighted, they are made available from the author's account at *researchgate.net*, in electronic form, at no cost. The author encourages feedback and suggestions for improvements in the text and computer code in future releases at [echaug@gmail.com](mailto:echaug@gmail.com).

## **Graduate Teaching Experience with the Text**

With Professor Bahram Ravani of the University of California at Davis, the author taught a graduate course during the fall of 2020, based on the first edition of the text. The content of the text, excepting chapters 6 and 8 on nonholonomic systems and systems with friction was covered, including use of MATLAB computer code for class projects. As expected, many previously undetected errors in the text and computer codes were found and corrected. Based on experience gained in the UC Davis course, the second edition was published in January, 2021.

With Professor Corina Sandu of Virginia Tech University, the author taught a two-semester graduate sequence using the second edition of the text during the 2021-22 academic year. The entire second edition, which had the same structure as the present third edition, was covered in these courses, including substantial team projects carried out by students. Experience gained in this adventure has provided the foundation for this, hopefully improved, third edition of the text.

The author thanks Professors Ravani and Sandu for the opportunity to interact with them and their students in presenting and refining the contents of the present text.

## CONTENTS

1 ELEMENTS OF MECHANICAL SYSTEM DYNAMICS	1
1.0 Introduction	1
1.1 Scope and Typical Applications	2
1.1.1 Slider-Crank Mechanism	3
1.1.2 Fly-Ball Governor	4
1.1.3 Windshield Wiper	5
1.1.4 Automobile Suspension	5
1.1.5 Serial Robotic Manipulator	7
1.1.6 Front-End Loader	8
1.2 Objectives and Foundations of Computational Dynamics	10
1.2.1 Emergence of the Discipline of Computational Dynamics	10
1.2.2 Basic Elements of Mechanical System Dynamics	10
1.2.3 Foundations for Computational Mechanical System Dynamics	15
1.3 Organization of the Text	17
2 POSITION AND ORIENTATION OF BODIES	20
2.0 Introduction	20
2.1 Vector Analysis	22
2.1.1 Geometric Vectors	22
2.1.2 Algebraic Vectors	27
2.1.3 Vector Identities	30
2.1.4 Time Derivatives of Vectors	32
2.2 Matrix Algebra and Multivariable Calculus	36
2.2.1 Matrix Algebra	36
2.2.2 Lagrange Multiplier Theorem of Linear Algebra	41
2.2.3 Multivariable Calculus	42
2.2.4 Euclidean Space, Continuity of Functions, and Topologies	43
2.2.5 Implicit Function Theorem	45
2.2.6 Newton-Raphson Method for Solution of Nonlinear Equations	47
2.2.7 Matrix Condition Number	48
2.3 Position and Orientation of a Body in a Plane	51
2.3.1 Reference Frames for Location and Orientation in a Plane	51
2.3.2 Orientation Degrees of Freedom	53
2.3.3 Orientation Identities	55
2.3.4 Time Derivatives and Variations in Rotation and Displacement	57
2.4 Position and Orientation of a Body in Space	59
2.4.1 Reference Frames for Location and Orientation in Space	59
2.4.2 Transformation of Coordinates	61
2.4.3 Orientation Degrees of Freedom	63
2.4.4 Identities Involving the Orientation Transformation Matrix	67
2.4.5 Velocity, Acceleration, Angular Velocity, and Angular Acceleration	69
2.4.6 Infinitesimal Rotation and Displacement	72
2.5 Euler Parameter Orientation Coordinates for a Body in Space	75
2.5.1 Invariants of Orientation Transformation	75
2.5.2 Orientation of Reference Frames	76
2.5.3 Euler Parameters	78
2.5.4 Mapping from the Orientation Matrix to Euler Parameters	79
2.5.5 Point Definition of an Orthogonal Reference Frame	83
2.5.6 Code 2.5, Definition of $A$ and Solution for $p$ with Given $A$	84
2.6 Euler Parameter Identities and Derivatives	87
2.6.1 Euler Parameter Identities	87
2.6.2 Euler Parameter Derivatives	90
2.6.3 Time Derivatives and Variations of Euler Parameters	93
2.6.4 Euler Parameter Differentials and Infinitesimal Rotation	95

2.6.5 Nonintegrability of Angular Velocity	95
2.6.6 Angular Acceleration as a Function of Euler Parameter Derivatives	96
2.6.7 Special Identities	97
2.6.8 Derivatives with Respect to Model Parameters	98
Appendix 2.A Euler Angle Orientation Coordinates	101
2.A.1 Definition of Euler Angles	101
2.A.2 Euler Angle Singularity	103
2.A.3 Complexity of Computation with Euler Angles	105
Appendix 2.B Position and Orientation Code	108
2.B.1 Code 2.2 Newton Raphson Solution of $k$ Equations $F(x) = 0$ for $k$ Variables $x$	108
2.B.2 Code 2.4 Computation of Relative Rotation About Common $z\phi$ -Axes	109
2.B.3 Code 2.5 Definition of $A$ and Solution for $p$ with Given $A$	110
2.B.4 Code 2.6 Euler Parameter Identities and Derivative Operators	112
Appendix 2.C Key Formulas, Chapter 2	114
 3 KINEMATICS OF MULTIBODY SYSTEMS	118
3.0 Introduction	118
3.1 Holonomic Constraints	119
3.1.1 Constraints Between Pairs of Bodies	119
3.1.2 Constraint, Velocity, and Acceleration Equations	122
3.2 Planar Joints, Constraint Equations, and Drivers	126
3.2.1 Constraints	126
3.2.1.1 Distance Constraint	126
3.2.1.2 Revolute Constraint	127
3.2.1.3 Translational Constraint	128
3.2.2 Kinematic Drivers	130
3.2.3 Planar System Examples	131
3.2.3.1 Double Pendulum	131
3.2.3.2 Slider-Crank, 3-Body Model	132
3.2.3.3 Slider-Crank, 2-Body Model with Rotation Driver	134
3.2.3.4 Windshield Wiper	135
3.3 Spatial Joints, Constraint Equations, and Drivers	137
3.3.1 Structure of Constraint Equations	137
3.3.2 Building Block Constraints	138
3.3.2.1 Distance Constraint	138
3.3.2.2 Spherical Constraint	140
3.3.2.3 Dot 1 Constraint	141
3.3.2.4 Dot 2 Constraint	142
3.3.3 Cylindrical, Revolute, and Translational Joints	143
3.3.3.1 Cylindrical Constraint	143
3.3.3.2 Revolute Constraint	144
3.3.3.3 Translational Constraint	145
3.3.4 Composite Joints	145
3.3.4.1 Universal Constraint	145
3.3.4.2 Strut Constraint	149
3.3.4.3 Revolute-Spherical Constraint	150
3.3.5 Kinematic Drivers	150
3.3.6 Euler Parameter Normalization Constraints	153
3.3.7 Spatial System Examples	153
3.3.7.1 Two Body Spatial Pendulum	153
3.3.7.2 Two-Bar Spatial Mechanism with Two Drivers	155
3.3.7.3 Spatial Slider-Crank	156
3.3.7.4 Fly-Ball Governor	157
3.3.7.5 Automobile Suspension	158
3.4 Configuration Spaces	161

3.5 Constraint Manifolds	168
3.5.1 Parameterization of a Particle on a Unit Sphere	168
3.5.2 Tangent Space Parameterization of the Regular Configuration Space	172
3.5.3 The Regular Configuration Space is a Differentiable Manifold	175
3.5.4 Computation of $\mathbf{B}(\mathbf{q})$ and $\mathbf{h}(\mathbf{v})$	176
3.6 Differential Geometry of Constraint Manifolds	179
3.6.1 Components of Constraint Manifolds	179
3.6.2 Dynamics of a Particle on a Unit Sphere	181
3.6.2.1 Tangent Space Generalized Coordinates	182
3.6.2.2 ODE of Motion	184
3.6.2.3 Numerical Solution	185
3.6.2.4 A Roadmap for Development of Methods of Multibody Dynamics	186
3.6.3 Differentiable Manifolds of a Triple Slider-Crank in R4	187
3.7 Singular Configurations	191
3.7.1 Kinematic Singularities	191
3.7.2 Singularities Due to Time Dependent Drivers	192
3.7.3 Criteria for Singular Behavior in Kinematic Analysis	193
3.7.4 Differentiable Manifold Components and Mechanism Singularities	195
3.8 Kinematic Position, Velocity, and Acceleration Analysis	197
3.8.1 Generalized Coordinates and Constraint Equations	197
3.8.2 System Assembly	199
3.8.3 Position Analysis	199
3.8.4 Velocity and Acceleration Analysis	200
3.8.5 Criteria for Singular Behavior	200
3.9 Code 3.9 for Planar System Kinematic Analysis	202
3.9.1 User Components of Code	202
3.9.2 Computational Components of Code	205
3.9.3 Code Output	210
3.10 Planar System Kinematic Analysis Using Code 3.9	211
3.10.1 Planar Slider-Crank Mechanisms	211
3.10.1.1 Two Body Slider-Crank	211
3.10.1.2 Three Body Double Slider-Crank	213
3.10.2 Quick Return Mechanism	215
3.10.3 Windshield Wiper	217
3.11 Code 3.11 for Spatial System Kinematic Analysis	220
3.11.1 User Components of Code	220
3.11.2 Computational Components of Code	226
3.11.3 Code Output	229
3.12 Spatial System Kinematic Analysis Using Code 3.11	231
3.12.1 Two Bar Mechanism with Two Rotation Drivers	235
3.12.2 Four Bar Mechanism with Internal Rotation Driver	235
3.12.3 Spatial Slider-Crank	238
3.12.4 Fly-Ball Governor	242
Appendix 3.A Kinematics Code	246
Appendix 3.B Key Formulas, Chapter 3	247
 4 EQUATIONS OF MOTION FOR BODIES AND MULTIBODY SYSTEMS	249
4.0 Introduction	249
4.1 Equations of Motion for Particles	250
4.1.1 Newton's Laws	250
4.1.2 Virtual Displacements	251
4.1.3 d'Alembert's Principle	253
4.1.4 Variational Equations of Motion for Systems of Particles	254
4.2 Equations of Motion for Planar Rigid Bodies	260
4.3 Equations of Motion for Spatial Rigid Bodies	266
4.3.1 Angular Velocity Form of Equations of Motion	266
4.3.2 Euler Parameter Form of Equations of Motion	270

4.4 Centroid and Moments and Products of Inertia	273
4.4.1 <i>Location of Centroid</i>	273
4.4.2 <i>Transformation of Inertia Matrices</i>	275
4.4.3 <i>Principal Axes</i>	276
4.4.4 <i>Transformation of Planar Polar Moment of Inertia</i>	277
4.4.5 <i>Inertia Properties of Complex Bodies</i>	278
4.5 Internal Generalized Forces	283
4.5.1 <i>Internal Forces in Spatial Systems</i>	283
4.5.2 <i>Internal Forces in Planar Systems</i>	286
4.6 Variational Equations of Motion for Multibody Systems	288
4.6.1 <i>System Constraints and Virtual Displacements</i>	288
4.6.2 <i>Variational Equations of Motion</i>	288
4.6.3 <i>Positive Definiteness of System Mass Matrix</i>	290
4.6.4 <i>Initial Conditions</i>	291
4.7 ODE of Motion in Independent Generalized Coordinates	294
4.7.1 <i>Examples of Systems with Independent Generalized Coordinates</i>	294
4.7.2 <i>General Form of ODE of Multibody Dynamics</i>	300
4.7.3 <i>Theory of ODE</i>	302
4.8 Runge-Kutta Methods for Numerical Solution of ODE	305
4.8.1 <i>Runge-Kutta Methods for First Order ODE</i>	305
4.8.1.1 <i>Explicit Runge-Kutta Methods for First Order ODE</i>	307
4.8.1.2 <i>Implicit Runge-Kutta Methods for First Order ODE</i>	309
4.8.2 <i>Runge-Kutta Methods for Second Order ODE</i>	311
4.8.2.1 <i>Explicit Runge-Kutta Methods for Second Order ODE</i>	314
4.8.2.2 <i>Implicit Runge-Kutta Methods for Second Order ODE</i>	317
4.8.3 <i>Generalized Form of Second Order ODE</i>	318
4.8.4 <i>Second Order Trapezoidal Method</i>	319
4.8.5 <i>Third Derivatives for Variable Step Size Error Control</i>	320
4.9 Numerical Solution for Systems Governed by ODE	322
4.9.1 <i>Code 4.8 for Second Order ODE Solution</i>	322
4.9.2 <i>Numerical Examples of Systems Governed by ODE</i>	322
4.9.2.1 <i>Planar Double Pendulum</i>	322
4.9.2.2 <i>Planar Slider-Crank</i>	326
4.9.2.3 <i>Spatial Robotic Manipulator</i>	334
4.10 Constraint Forces and Differential-Algebraic Equations of Motion	334
4.10.1 <i>Generalized Constraint Reaction Forces</i>	334
4.10.2 <i>Differential-Algebraic Equations of Motion</i>	335
4.10.3 <i>Spatial Joint Constraint Reaction Force and Torque</i>	336
4.10.4 <i>Planar Joint Constraint Reaction Force and Torque</i>	338
4.10.5 <i>Evaluation of Constraint Reaction Forces</i>	339
Appendix 4.A ODE Solution Code	340
4.A.1 <i>Code 4.8 Second Order ODE Solvers</i>	340
4.A.1.1 <i>User Components of Code 4.8</i>	340
4.A.1.2 <i>Computational Components of Code</i>	341
4.A.2 <i>Code 4.A.1 Planar Double Pendulum</i>	343
4.A.3 <i>Code 4.A.2 Planar Slider-Crank</i>	343
4.A.4 <i>Code 4.A.3 Three Degree of Freedom Robotic Manipulator</i>	343
Appendix 4.B Key Formulas, Chapter 4	344
 5 TANGENT SPACE ODE FOR HOLONOMIC SYSTEMS	346
5.0 Introduction	346
5.1 A Spatial Model Problem That Requires Local Coordinates	348
5.1.1 <i>Tangent Space Generalized Coordinates</i>	349
5.1.2 <i>ODE of Motion</i>	351
5.1.3 <i>Numerical Solution</i>	353

5.2 Tangent Space ODE	356
5.2.1 <i>Tangent Space Kinematics with Time Dependent Constraints</i>	356
5.2.2 <i>Kinematic Relations</i>	359
5.2.3 <i>Tangent Space Kinematics with Time Independent Constraints</i>	361
5.2.4 <i>ODE of Motion</i>	362
5.2.5 <i>Calculation of Constraint Reaction Forces</i>	364
5.3 Numerical Solution of Tangent Space ODE	366
5.3.1 <i>Explicit Numerical Integration Algorithm</i>	366
5.3.2 <i>Derivatives for Implicit Integration</i>	367
5.3.3 <i>Organization of Derivative Calculation</i>	370
5.3.4 <i>Trapezoidal Implicit Numerical Integration</i>	372
5.3.5 <i>Runge-Kutta Implicit Numerical Integration</i>	373
5.3.6 <i>Implicit Numerical Integration Algorithm</i>	375
5.4 Numerical Examples with Tangent Space ODE	377
5.4.1 <i>Planar Double Pendulum</i>	377
5.4.2 <i>Top with Tip Fixed</i>	381
5.4.2.1 <i>Spin Stabilized Top</i>	381
5.4.2.2 <i>Transient Top</i>	384
5.4.3 <i>Top with Tip Sliding on x-y Plane</i>	384
5.4.4 <i>Spatial Double Pendulum</i>	390
5.4.5 <i>Need for Automated Equation Formulation and Solution</i>	394
5.5 Tangent Space Index 0 DAE	396
5.5.1 <i>Index 0 DAE Formulation</i>	396
5.5.2 <i>Explicit Integration of Index 0 DAE</i>	398
5.5.3 <i>Derivatives for Implicit Integration of Index 0 DAE</i>	399
5.5.4 <i>Implicit Trapezoidal Integration of Index 0 DAE</i>	400
5.5.5 <i>Implicit Runge-Kutta Integration of Index 0 DAE</i>	401
5.5.6 <i>Implicit Index 0 DAE Integration Algorithm</i>	403
5.5.7 <i>Third Derivatives for Error Control</i>	403
5.6 Numerical Examples with Tangent Space Index 0 DAE	406
5.6.1 <i>Top with Tip Fixed</i>	406
5.6.1.1 <i>Spin Stabilized Top</i>	406
5.6.1.2 <i>Transient Top</i>	407
5.6.2 <i>Spatial Double Pendulum</i>	407
5.7 Code 5.7 for Tangent Space Simulation of Planar Systems	410
5.7.1 <i>User Components of Code</i>	410
5.7.2 <i>Computational Components of Code</i>	414
5.7.3 <i>Code Output</i>	416
5.8 Planar System Simulation Using Code 5.7	418
5.8.1 <i>Double Pendulum</i>	418
5.8.1.1 <i>Model and Data Set</i>	418
5.8.1.2 <i>Nominal Results</i>	419
5.8.1.3 <i>Results for Stiff Systems</i>	419
5.8.1.4 <i>Results for Highly Oscillatory Systems</i>	422
5.8.1.5 <i>Level of Effort Required</i>	423
5.8.2 <i>Quick Return Mechanism</i>	424
5.8.2.1 <i>Model and Data Set</i>	424
5.8.2.2 <i>Simulation Results and Analysis</i>	425
5.8.2.3 <i>Level of Effort Required</i>	427
5.8.3 <i>Surge Waves in a Coil Spring</i>	428
5.8.3.1 <i>Model and Data Set</i>	428
5.8.3.2 <i>Simulation Results and Analysis</i>	429
5.8.4 <i>Three-Body Mechanism with Translational Joints in Ground</i>	431
5.8.4.1 <i>Model and Data Set</i>	431
5.8.4.2 <i>Simulation Results and Analysis</i>	432
5.8.4.3 <i>Impact of Body 3 With a Unilateral Spring</i>	433

<i>5.8.5 Slider-Crank</i>	434
<i>5.8.5.1 Model and Data Set</i>	434
<i>5.8.5.2 Simulation Results and Analysis</i>	435
<i>5.8.6 Rotating Disk with Translating Body</i>	436
<i>5.8.6.1 Model and Data Set</i>	436
<i>5.8.6.2 Simulation Results and Analysis</i>	437
5.9 Code 5.9 for Tangent Space Simulation of Spatial Systems	438
<i>5.9.1 User Components of Code</i>	438
<i>5.9.2 Computational Components of Code</i>	443
<i>5.9.3 Code Output</i>	445
5.10 Spatial System Simulation Using Code 5.9	447
<i>5.10.1 Spinning Top with Tip Fixed</i>	447
<i>5.10.2 Spinning Top with Tip Constrained to be a Unit Distance from a Fixed Point</i>	449
<i>5.10.3 Spatial Double Pendulum</i>	452
<i>5.10.4 Body in Cylindrical Joint with Ground</i>	456
<i>5.10.5 Spatial Slider-Crank</i>	459
<i>5.10.6 Four Body Mechanism with Translational Joints in Ground</i>	462
<i>5.10.7 Fly-Ball Governor</i>	465
5.11 Well Posed Formulations of Holonomic Mechanical System Dynamics	470
<i>5.11.1 Introduction</i>	470
<i>5.11.2 Design Dependence in Variational Equations of Motion</i>	470
<i>5.11.3 Initial Conditions as Functions of Problem Data</i>	471
<i>5.11.4 ODE of Motion</i>	472
<i>5.11.5 Well Posed ODE and Index 0 DAE of Motion</i>	476
<i>5.11.6 Well Posed Variational Equations of Motion</i>	477
<i>5.11.7 Well Posed Full DAE of Motion</i>	477
Appendix 5.A Holonomic Tangent Space Code	482
Appendix 5.B Kinematic and Kinetic Derivatives	483
<i>5.B.1 Derivatives of Kinematic Constraints</i>	483
<i>5.B.1.1 Derivatives of Planar Constraints</i>	483
<i>5.B.1.2 Derivatives of Spatial Constraints</i>	486
<i>5.B.2 Derivatives of Kinetic Quantities</i>	490
<i>5.B.2.1 Derivatives of Planar Kinetics</i>	490
<i>5.B.2.2 Derivatives of Spatial Kinetics</i>	491
<i>5.B.3 Derivatives of Generalized Forces</i>	492
<i>5.B.3.1 Derivatives of Planar Generalized Forces</i>	492
<i>5.B.3.2 Derivatives of Spatial Generalized Forces</i>	495
Appendix 5.C Key Formulas, Chapter 5	499
Appendix 5.D The Trouble with Maggi and Kane Equations	501
<i>5.D.1 Kinematics Formulation</i>	501
<i>5.D.2 Maggi/Kane Equations with Only Velocity and Acceleration Constraints</i>	502
<i>5.D.3 Inaccuracy of Maggi and Kane Equations for Holonomic Constraints</i>	504
Appendix 5.E Generalized Coordinate Partitioning	505
<i>5.E.1 Gaussian Elimination with the Constraint Jacobian Matrix</i>	505
<i>5.E.2 The Differential-Algebraic Equations of Motion</i>	506
<i>5.E.3 Partitioning the DAE of Dynamics</i>	507
<i>5.E.4 Second Order Partitioned ODE of Dynamics</i>	508
<i>5.E.5 Explicit Integration of Partitioned Equations of Motion</i>	509
<i>5.E.6 Implicit Integration of Second Order Partitioned ODE of Dynamics</i>	510
6 TANGENT SPACE ODE FOR NONHOLONOMIC SYSTEMS	512
6.0 Introduction	512
6.1 Tangent Space Kinematics for Nonholonomic Systems	513
<i>6.1.1 Differential Constraints</i>	513
<i>6.1.2 Examples of Nonholonomic Constraints</i>	514
<i>6.1.3 Nonholonomic System Kinematics</i>	519
<i>6.1.4 Configuration Tangent Space Parameterization</i>	520

<i>6.1.5 Velocity Tangent Space Parameterization</i>	522
<i>6.1.6 Kinematic Differential Equation</i>	523
<i>6.1.7 Kinematically Admissible Virtual Displacements</i>	524
6.2 Tangent Space ODE for Dynamics of Nonholonomic Systems	526
<i>6.2.1 ODE of Motion</i>	526
<i>6.2.2 Explicit Numerical Integration</i>	527
<i>6.2.3 Derivatives for Implicit Numerical Integration</i>	528
<i>6.2.4 Implicit Numerical Integration Algorithms</i>	530
<i>6.2.4.1 Trapezoidal Integration</i>	530
<i>6.2.4.2 SDIRK Integration</i>	532
<i>6.2.4.3 Implicit Integration Algorithm</i>	533
6.3 Numerical Examples with Nonholonomic ODE	534
<i>6.3.1 Planar Three Wheel Transporter</i>	534
<i>6.3.2 Planar Articulated Vehicle</i>	538
<i>6.3.3 Spatial Disk Rolling Without Slip on x-y Plane</i>	541
<i>6.3.4 Spatial Ellipsoid Rolling Without Slip on a Moving Surface</i>	544
6.4 Index 0 DAE Formulation for Nonholonomic Systems	553
<i>6.4.1 Index 0 DAE for Nonholonomic Systems</i>	553
<i>6.4.2 Explicit Integration of Nonholonomic Index 0 DAE</i>	555
<i>6.4.3 Derivatives for Implicit Numerical Integration</i>	556
<i>6.4.4 Implicit Numerical Integration Algorithms</i>	557
<i>6.4.4.1 Trapezoidal Integration</i>	557
<i>6.4.4.2 SDIRK Integration</i>	558
<i>6.4.4.3 Implicit Integration Algorithm</i>	559
6.5 Numerical Examples with Index 0 DAE	561
<i>6.5.1 Planar Articulated Vehicle</i>	561
<i>6.5.2 Spatial Disk Rolling Without Slip on x-y Plane</i>	563
6.6 Well Posed Formulations of Nonholonomic Mechanical System Dynamics	567
Appendix 6.A Tangent Space Nonholonomic Code	571
Appendix 6.B Key Formulas, Chapter 6	572
 7 DAE METHODS FOR HOLONOMIC SYSTEMS	573
7.0 Introduction	573
7.1 The DAE of Holonomic Systems	574
<i>7.1.1 Lagrange Multipliers</i>	574
<i>7.1.2 Index of DAE</i>	575
<i>7.1.3 An Alternative DAE Formulation</i>	577
<i>7.1.4 The Petzold Imperative; DAE are not ODE</i>	577
7.2 Formulations for Numerical Solution of DAE	579
<i>7.2.1 Index 1 DAE Formulation</i>	579
<i>7.2.2 Index 2 DAE Formulation</i>	580
<i>7.2.3 Index 3 DAE Formulation</i>	580
<i>7.2.4 Correction of Constraint Errors</i>	580
<i>7.2.5 Numerical Approaches to Solution of DAE</i>	582
<i>7.2.6 Third Derivative for Error Control</i>	584
7.3 Numerical Integration of Index 1, 2, and 3 DAE of Holonomic Systems	586
<i>7.3.1 Integration of Index 1 DAE</i>	586
<i>7.3.1.1 Explicit Integration of Index 1 DAE</i>	586
<i>7.3.1.2 Implicit Integration of Index 1 DAE</i>	586
<i>7.3.2 Integration of Index 2 DAE</i>	588
<i>7.3.3 Integration of Index 3 DAE</i>	589
7.4 Code 7.4 for DAE Simulation of Planar Systems	591
<i>7.4.1 User Components of Code</i>	591
<i>7.4.2 Computational Components of Code</i>	593
7.5 Planar System Simulation Using DAE Code 7.4	597
<i>7.5.1 Double Pendulum</i>	597
<i>7.5.2 Quick Return Mechanism</i>	600

7.6 Code 7.6 for DAE Simulation of Spatial Systems	602
7.6.1 <i>User Components of Code</i>	602
7.6.2 <i>Computational Components of Code</i>	604
7.7 Spatial System Simulation Using DAE Code 7.2	607
7.7.1 <i>Spin Stabilized Top</i>	607
7.7.2 <i>Double Pendulum</i>	609
7.7.3 <i>Slider-Crank</i>	609
Appendix 7.A DAE Solution Code	611
Appendix 7.B Key Formulas, Chapter 7	612
 8 SIMULATION OF FRICTION IN MULTIBODY DYNAMICS	 613
8.0 Introduction	613
8.1 Modeling Friction in Multibody Dynamics	614
8.1.1 <i>Static Coulomb Friction in a Constrained System</i>	614
8.1.2 <i>Three Mass Planar Model with Friction and Stiction</i>	615
8.1.2.1 <i>Mechanism Model</i>	615
8.1.2.2 <i>Continuous Friction-Slip Velocity Function</i>	617
8.1.2.3 <i>Tangent Space Index 0 Model with Friction</i>	618
8.1.2.4 <i>Explicit Integration of the Index 0 DAE</i>	619
8.1.2.5 <i>Implicit integration of the Index 0 DAE</i>	620
8.1.3. <i>Simulation of Three Mass planar Mechanism</i>	621
8.2 Analytical Stiction Criteria and Ranges of Stiction Forces	626
8.2.1 <i>Analytical Stiction Criteria for the Three Mass Model</i>	626
8.2.2 <i>Analytical Solution of Stiction Inequalities</i>	628
8.2.3 <i>Admissible Ranges of Stiction Forces</i>	630
8.2.4 <i>Numerical Solution of Coulomb Stiction Criteria</i>	632
8.3. Cartesian Coordinate Simulation of Model Problems	635
8.3.1 <i>Cartesian Coordinate Simulation of a Three Mass System</i>	635
8.3.2 <i>Cartesian Coordinate Simulation of a Four Mass System</i>	637
8.4 Modeling Planar Multibody Systems with Friction	642
8.4.1 <i>Constraint Reaction Forces and Friction Generalized Forces</i>	642
8.4.1.1 <i>Revolute Joint Reaction and Friction Forces</i>	642
8.4.1.2 <i>Translational Joint Reaction and Friction Forces</i>	643
8.4.2 <i>Derivatives of Friction Generalized Forces</i>	645
8.5 Modeling Spatial Multibody Systems with Friction	647
8.5.1 <i>Constraint Contact Reaction Forces</i>	647
8.5.2 <i>Friction Forces in Joints</i>	651
8.5.3 <i>Derivatives of Friction Forces</i>	654
8.6 Index 0 DAE Formulation for Systems with Friction	660
8.6.1 <i>Index 0 Equations of Motion</i>	660
8.6.2 <i>Explicit Numerical Integration Algorithm</i>	660
8.6.3 <i>Implicit Numerical Integration Algorithm</i>	662
8.7 Code 8.7 for Simulation of Planar Systems with Friction	665
8.7.1 <i>User Components of Code</i>	665
8.7.2 <i>Computational Components of Code</i>	667
8.8 Planar System Simulation with Friction Using Code 8.7	669
8.8.1 <i>Three slider Mechanism</i>	669
8.8.2 <i>Slider-Crank</i>	671
8.8.3 <i>Quick Return Mechanism</i>	674
8.8.4 <i>Surge Waves in a Coil Spring</i>	676
8.8.5 <i>Rotating Disk with Translating Body</i>	678
8.9 Code 8.9 for Simulation of Spatial Multibody Systems with Friction	682
8.9.1 <i>User Components of Code</i>	682
8.9.2 <i>Computational Components of Code</i>	684
8.10 Spatial System Simulation with Friction Using Code 8.9	686
8.10.1 <i>Body in Cylindrical Joint with Ground</i>	686
8.10.2 <i>Slider-Crank</i>	688

8.10.3 Four Body Slider	691
8.10.4 Rotating Disk with Translating Body	695
Appendix 8.A Tangent Space Index 0 DAE Friction Code	698
Appendix 8.B Key Formulas, Chapter 8	699
 9 MANIPULATOR KINEMATICS AND DYNAMICS	700
9.0 Introduction to Part I, Nonredundant Manipulators	700
9.1 Characteristics of Nonredundant Manipulators	702
9.1.1 Serial Manipulators	703
9.1.2 Explicit Parallel Manipulators	704
9.1.3 Implicit Manipulators with no Global Forward or Inverse Kinematic Mapping	706
9.1.4 Compound Manipulators	706
9.1.5 Relation Between the Present Formulation and the Manipulator Literature	708
9.2 Manipulator Kinematics	710
9.2.1 Manipulator Coordinates and Kinematic Equations	710
9.2.2 Forward and Inverse Kinematics	711
9.2.2.1 Forward Kinematics	711
9.2.2.2 Inverse Kinematics	712
9.2.3 Singular Configurations and Explicit Input-Output Equations	712
9.3 Serial Manipulator Kinematics	714
9.3.1 A Model Serial Manipulator	714
9.3.2 Serial Manipulator Regular Configuration Manifold	717
9.4 Explicit Parallel Manipulator Kinematics	719
9.4.1 A Model Explicit Parallel Manipulator	719
9.4.2 Explicit Parallel Manipulator Regular Configuration Manifold	723
9.5 Implicit Manipulator Kinematics	725
9.5.1 Model Implicit Manipulators	725
9.5.2 Implicit Manipulator Regular Configuration Manifold	727
9.6 Compound Manipulator Kinematics	728
9.6.1 A Model Compound Manipulator with Closed Kinematic Loop	728
9.6.2 Compound Manipulator Regular Configuration Manifold	730
9.6.3 Singularity Free Manifold Components for the Front-End Loader	731
9.7 Singularity Free Manifold Components and Manipulator Singularities	735
9.8 Foundations of Manipulator Dynamics	739
9.9 Forward and Inverse Velocity Mappings	741
9.9.1 Serial Manipulator Velocity Mappings	741
9.9.1.1 Serial Manipulator Forward Velocity Mapping	742
9.9.1.2 Serial Manipulator Inverse Velocity Mapping	742
9.9.2 Explicit Parallel Manipulator Velocity Mappings	742
9.9.2.1 Explicit Parallel Manipulator Inverse Velocity Mapping	743
9.9.2.2 Explicit Parallel Manipulator Forward Velocity Mapping	743
9.9.3 Implicit Manipulator Velocity Mappings	743
9.9.3.1 Implicit Manipulator Forward Velocity Mapping	743
9.9.3.2 Implicit Manipulator inverse Velocity Mapping	743
9.9.4 Compound Manipulator Velocity Mappings	743
9.9.4.1 Compound Manipulator Froward Velocity Mapping	743
9.9.4.2 Compound Manipulator Inverse Velocity Mapping	744
9.10 Computation of Forward and Inverse Kinematics on a Time Grid	745
9.10.1 Serial Manipulator Forward Kinematics	745
9.10.2 Explicit Parallel Manipulator Inverse Kinematics	745
9.10.3 Implicit Manipulator Kinematics	745
9.10.3.1 Implicit Manipulator Forward Kinematics	745
9.10.3.2 Implicit Manipulator Inverse Kinematics	748
9.10.4 Serial Manipulator Inverse Kinematics	749
9.10.5 Explicit Parallel Manipulator Forward Kinematics	750

9.10.6 Compound Manipulator Kinematics	750
9.10.6.1 Compound Manipulator Forward Kinematics	751
9.10.6.2 Compound Manipulator Inverse Kinematics	753
9.10.7 Real-Time Computation of Forward and Inverse Kinematics	754
9.11 ODE of Manipulator Dynamics	756
9.11.1 Compound Manipulator ODE	757
9.11.1.1 Compound Manipulator ODE of Dynamics in Input Coordinates	757
9.11.1.2 Compound Manipulator ODE of Dynamics in Output Coordinates	758
9.11.2 Serial Manipulator ODE	760
9.11.2.1 Serial Manipulator ODE of Dynamics in Input Coordinates	760
9.11.2.2 Serial Manipulator ODE of Dynamics in Output Coordinates	760
9.11.3 Explicit Parallel Manipulator ODE	760
9.11.3.1 Explicit Parallel Manipulator ODE of Dynamics in Input Coordinates	760
9.11.3.2 Explicit Parallel Manipulator ODE of Dynamics in Output Coordinates	760
9.11.4 Implicit Manipulator ODE	760
9.11.4.1 Implicit Manipulator ODE of Dynamics in Input Coordinates	760
9.11.4.2 Implicit Manipulator ODE of Dynamics in Output Coordinates	760
9.12 Equations of Dynamics for Model Manipulators	761
9.12.1 Variational Equation of Motion for Planar Systems	761
9.12.2 Model Serial Manipulator	762
9.12.3 Model Explicit Parallel Manipulator	763
9.12.4 Model Implicit Manipulator	764
9.12.5 Model Compound Manipulator	765
9.12.6 Real-Time Computation of the ODE of Dynamics	766
9.13 Introduction to Part II, Redundant Manipulators	768
9.14 Characteristics of Redundant Manipulators	769
9.14.1 A Redundant Serial Manipulator	770
9.14.2 A Redundant Compound Manipulator	772
9.15 Kinematically Redundant Serial Manipulators	775
9.15.1 Serial Manipulator Configuration Space	775
9.15.2 Regular Serial Manipulator Configuration Space	775
9.15.3 Serial Manipulator Inverse Kinematic Configuration Mapping	776
9.15.4 The Regular Serial Manipulator Configuration Differentiable Manifold	776
9.15.5 The Serial Manipulator Self-Motion Manifold	778
9.15.6 Mapping One-Dimensional Serial Manipulator Self-Motion Manifold Components	778
9.15.7 A Large Scale Redundant Serial Manipulator	781
9.15.8 Serial Manipulator Inverse Dynamics	785
9.16 Kinematically Redundant Compound Manipulators	787
9.16.1 Compound Manipulator Configuration Space	787
9.16.2 Regular Compound Manipulator Configuration Space	788
9.16.3 Compound Manipulator Inverse Kinematic Configuration Mapping	789
9.16.4 Parameterization of the Regular Compound Manipulator Configuration Space	790
9.16.5 The Compound Manipulator Self-Motion Manifold	791
9.16.6 Mapping One-Dimensional Compound Manipulator Self-Motion Manifolds	791
9.16.7 Model Kinematically Redundant Compound Manipulator Kinematics	792
9.16.8 Compound Manipulator Obstacle Avoidance Using Self-Motion Manifolds	793
9.16.9 Compound Manipulator Inverse Dynamics	796
Appendix 9.A: Computation of $h(v, z)$ and $B(v, z)$	798
Bibliography	799
Index	809

# CHAPTER 1

## Elements of Mechanical System Kinematics and Dynamics

### 1.0 Introduction

Knowledge related to dynamics of particles and celestial bodies was dramatically advanced by Newton in the 17th century (Newton, 1687), with the formulation of fundamental laws of motion of discrete masses and a foundation for calculus that enables quantification of the laws. A century later, d'Alembert (1743), Euler (1765), and Lagrange (1788) created a sound analytical foundation for the field of mechanical system dynamics that survives to the present time. In the subsequent quarter of a millennium, the brightest mathematical and scientific minds added to the foundation of knowledge and methods for obtaining governing equations of motion and characterizing properties of their solutions. Extreme nonlinearity and high dimensionality that characterizes the equations of mechanical system kinematics and dynamics, however, provided an insurmountable barrier to analytic solution of problems of immense scientific and engineering importance.

In the last third of the 20th century and first two decades of the 21st century, technology has emerged that makes simulation and analysis of the kinematics and dynamics of mechanical systems possible for applications of contemporary importance. Not surprisingly, this development parallels almost precisely advancements in digital computer technology and computational science, whose impact on kinematics and dynamics of mechanical systems has been nothing short of spectacular. Kinematics and dynamics of mechanical systems, sometimes called *multibody systems*, has advanced to the point that it is accepted as a discipline unto itself, with several journals devoted to research contributions. Thousands of papers and dozens of books on the subject have been published during the past four decades. Texts on computational aspects of mechanical system kinematics and dynamics, however, are accessible primarily to advanced graduate students and research engineers. In parallel, a rich mathematical and numerical analysis literature and applicable software base has evolved that provides the foundation for computational kinematics and dynamics. Unfortunately, some of the most important contributions are presented at a mathematical level that is beyond the comfort level of we mere mortal engineers.

The text Computer Aided Kinematics and Dynamics of Mechanical Systems, Volume 1: Basic Methods (Haug, 1989) was based on the state of computational kinematics and dynamics technology in 1989. The objective of this text is to present modern foundations that have evolved for computational formulation and solution of the equations of kinematics and dynamics of mechanical systems. These results enable students to master basics of the subject, carry out practical applications, and access the extensive literature and application software base that has evolved over time.

The purpose of this chapter is to (1) define the scope of the subject treated, (2) outline typical applications that illustrate capabilities desired, and (3) identify objectives, foundations, and challenges that face computer formulation and solution of the equations of mechanical system kinematics and dynamics.

## 1.1 Scope and Typical Applications

A *mechanical system*, also called a *multibody system*, is defined as a collection of interconnected bodies that move relative to one another, consistent with joints that limit relative motion of pairs of bodies. Typical mechanical systems are *vehicles*, *machine tools*, *agricultural and construction equipment*, *robots*, and *manipulators*. Initial conditions, internal forces due to friction and control inputs, and externally applied forces may be specified, in which case dynamics of the system is determined by equations of motion that are based on the geometry of kinematics and laws of dynamics. Kinematics and dynamics of mechanical systems are characterized by large amplitude motion, which leads to *geometric nonlinearity* that is reflected in algebraic equations of constraint and differential equations of motion. Nonlinearity of mechanical system kinematics and dynamics is often viewed as an annoyance. It must be realized, however, that some of the most valuable contributions made by mechanisms and machines to modern technology are due to their *nonlinearity*. The *slider-crank mechanism* of Section 1.1.1 is the foundation for numerous machines. It is highly nonlinear, transforming reciprocating motion of the slider to rotational motion of the crank in the internal combustion engine. Without it, the internal combustion engine that has powered transportation and the broader industrial revolution for centuries would not be possible.

An important consideration that serves to classify mechanical systems concerns the source of forces that act on such systems. This is particularly important for mechanical systems on which some form of *control* is exerted. Force effects due to electronic and hydraulic *feedback control* play a crucial role in the dynamics of modern mechanical systems. The scope of mechanical system dynamics is, therefore, heavily dependent on the classes of force systems that act. The most elementary form of force that acts on a mechanical system is *gravity*, which is normally taken as constant and acting perpendicular to the surface of the earth. Other forces that act on bodies in a system, due to interaction with their environment, include aerodynamic, friction, and damping forces that are due to relative motion of components of the system. Another important class of forces that act in a mechanical system is associated with *compliant components* such as *coil springs*, *leaf springs*, *tires*, *shock absorbers*, and a multitude of other *deformable components* that create forces and torques that act on the system.

It must be noted that most mechanical systems involve some form of *energy dissipation*. This is characteristic of systems with *friction* or *hydraulic dampers*, such as *shock absorbers* in vehicles that are designed to control motion of the system. Feedback control systems that act in mechanical systems involve *energy input* to the system. All this is in conflict with much of the classical literature on dynamics that treats dissipation or addition of energy as an anomaly that is outside the scope of the subject. Much of this literature is thus not applicable to systems treated in this text.

While components of mechanical systems deform under the action of forces imposed on them, in many applications the effects of *deformation* are negligible and bodies can be considered to be rigid. Most applications addressed in this text involve bodies that are considered to be rigid, called *rigid multibody systems*. Important applications in which deformation must be accounted for are beyond the scope of this introductory text, but rely on the foundations developed herein.

Finally, perhaps the greatest challenge facing effective modeling and simulation of mechanical system kinematics and dynamics is the dimensionality of equations involved. For k

bodies that move in space, subject to  $m$  equations of constraint, the number of variables that define the configuration of the system is  $7k$ . With  $\text{dof} = 7k - m$  *degrees of freedom*, on the order of 10 and  $k$  on the order of 10, approximately  $m = 7k - \text{dof} = 70 - 10 = 60$  *nonlinear equations of constraint* must be dealt with. This is modest dimensionality by modern standards. This means that even if 10 *independent coordinates* can be found in a change of variable to define 10 *ordinary differential equations of motion*, 60 variables must be eliminated from 60 highly nonlinear equations of constraint. The result is systems of equations that cannot be practically written in analytical form, much less solved analytically. The logical conclusion is that, since equations of kinematics for realistic systems cannot be analytically written and solved, methods must be created and implemented for digital computer assembly and solution of the equations of mechanical system kinematics and dynamics. While progress has been made in creating this capability, much remains to be done.

To be more concrete regarding classes of mechanical systems to be addressed, it is helpful to review a few typical engineering applications. Since the audience for the text is familiar with and relies upon self-propelled vehicles, most applications illustrated in this section are from *vehicle mechanics*.

### 1.1.1 Slider-Crank Mechanism

The engine shown in Fig. 1.1.1 contains many moving parts and illustrates a number of the most common mechanisms employed in machine design. The crankshaft of the engine rotates in *lubricated bearings* in the body of the engine and contains *rotational bearings* with *connecting rods*, which are coupled through rotational bearings to pistons that reciprocate in combustion cylinders. The crankshaft-connecting rod-piston assembly comprises what is commonly called a *slider-crank mechanism*, which is used to convert *reciprocating motion* of the piston to *rotation* of the crankshaft, a highly nonlinear relationship.

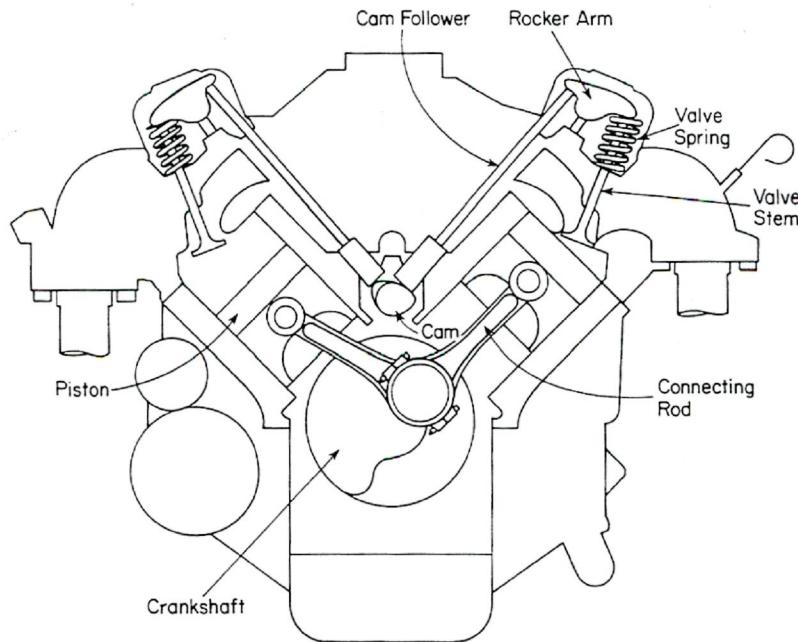


Figure 1.1.1 Cross Section of a V-2, 4, 6, or 8 Engine

Using models of combustion forces in the cylinder, equations of motion may be formulated and solved for rotational speed vs. torque delivered, as the engine interacts with the transmission. *Reaction forces on bearings* are determined by the equations of dynamics to support *design of bearings* and other moving components. Virtually all these relationships are nonlinear.

### 1.1.2 Fly-Ball Governor

A second example is the traditional *fly-ball governor* shown in Fig. 1.1.2. Relatively massive balls are attached to arms that are pivoted on a rotating shaft, so that they rotate with the shaft. Couplers are attached between the ball arms and a collar that is constrained to translate along the shaft. The entire mechanism rotates with *angular velocity*  $\omega$  of the shaft. As  $\omega$  increases, *centrifugal forces* act on the balls to throw them out, causing the collar to move upward, hence increasing the collar height  $s$  shown in Fig. 1.1.2. The purpose of the fly-ball governor is to control the operating speed of an engine. A mechanism couples the position  $s$  of the collar to the fuel feed of an internal combustion engine that drives the shaft. The mechanism is designed so that, at the desired speed of the engine, centrifugal forces on the balls and gravitational and spring forces that act on the mechanism reach a *steady state*, with the collar at a constant height. If an increased load is encountered that reduces engine speed, hence the angular velocity  $\omega$  of the shaft; e.g., a vehicle encountering a hill or a lawn mower encountering tall grass, the balls will drop and the collar will move downward. The mechanism that couples the position  $s$  of the collar with the fuel feed provides additional fuel, which in turn speeds the engine and causes centrifugal forces on the balls to increase, raising the balls toward their nominal height and returning the angular velocity of the shaft to its nominal value. This governor mechanism gives rise to the expression that the associated machine is going “*balls out*” when the engine runs at top speed. Motion of the mechanism and centrifugal force that enables the system to function as desired depend on numerous *nonlinear effects*. If machine dynamics were linear, this system could not perform its function.

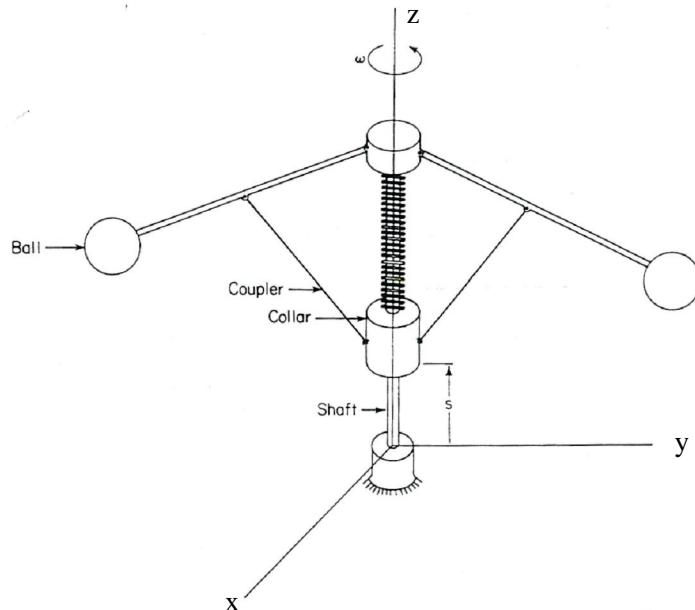


Figure 1.1.2 Fly-Ball Governor

### 1.1.3 Windshield Wiper

The *windshield wiper* mechanism shown in Fig. 1.1.3 is an application of *four-bar linkages* that transmit motor-driven rotation of the crank to reciprocating motion of the windshield wipers, a nonlinear relation. The *crank* and left *rocker arm* are pivoted in the vehicle frame at points A and B. A *coupler* is attached to the crank at point C and to the left rocker arm at point D. The crank, crank coupler, left rocker, and frame of the vehicle constitute a *four-bar linkage*. Since the distance from B to D is greater than the distance from A to C, a full rotation of the crank causes only a partial rotation of the left rocker arm, leading to the desired *reciprocating motion* of the left windshield wiper. The dimensions of the various links are selected to generate the desired range of motion. A second four-bar linkage is formed by the right rocker arm that is pivoted in the frame of the vehicle at point G and the rocker coupler that is connected to the left and right rocker arms at points E and F. This second linkage transmits reciprocating motion from the left rocker arm to the right rocker arm, hence controlling motion of the right windshield wiper. As with the slider-crank mechanism and the fly-ball governor, this mechanism depends on nonlinear geometric effects to perform its function. Its dynamic performance is effected by inertia of the moving components, motor torque delivered, and hydrodynamic and friction forces between the wiper blades and windshield.

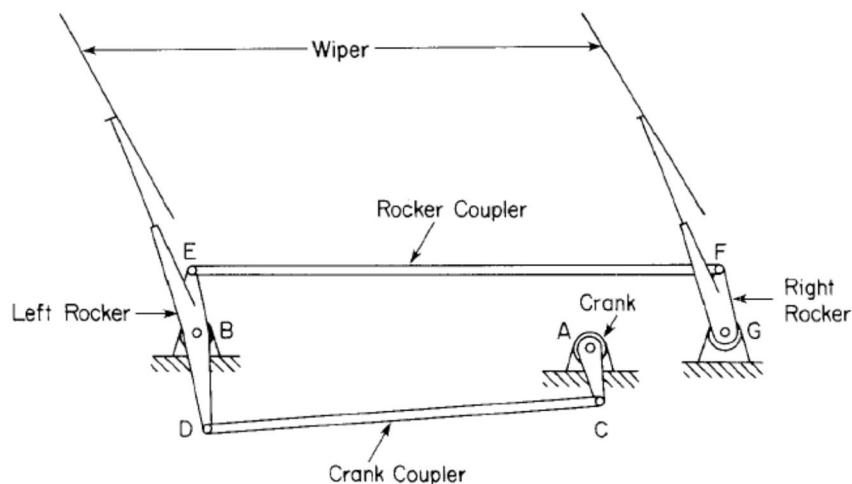


Figure 1.1.3 Windshield Wiper Mechanism

### 1.1.4 Automobile Suspension

The final vehicle related example, which illustrates the scope of applications that are addressed in this text, is the vehicle of Fig. 1.1.4, whose suspension system is shown schematically in Figs. 1.1.5 and 1.1.6. This commonly employed suspension consists of a *McPherson strut* front suspension and a *trailing arm* rear suspension. Each front wheel assembly is attached to the chassis of the vehicle through a lower control arm and a telescoping strut-spindle assembly, as shown in Fig. 1.1.6. Concentric with the strut are a *suspension spring* and shock absorber. *Ball joints* at the top and bottom of the strut permit steering rotation of the wheel assembly about the strut, controlled by the *tie rod* and the *steering rack*. The more elementary rear suspension shown in Fig. 1.1.5 is simply a *trailing arm* that is pivoted in the chassis to permit the rear wheel assembly to move relative to the chassis. Spring and *shock absorber* components that act between the trailing arm and chassis provide for support of the chassis and cushioning of nonlinear tire-road interaction force effects.

Dynamic analysis is carried out, using models of forces due to tire-road interaction, spring and shock absorber forces, forces at the driven wheel spindles due to engine torque, and steering rack displacement due to driver actions, to determine motion of the vehicle on the roadway. Many such analyses are carried out in designing subsystems of a vehicle. The nonlinear behavior of this intricate mechanical system enables vehicle performance characteristics that would not be possible if it were linear. As will be shown in the text, with wheel rotation, system kinematics are governed by 84 variables, called generalized coordinates, and 70 constraint equations. The vehicle system thus has 14 degrees of freedom. Only the most wild-eyed optimist could hope for analytical solutions of the nonlinear equations of kinematics and dynamics of such a system. This and other examples discussed here require the power of the modern digital computer and supporting software to both assemble and solve the equations of motion.

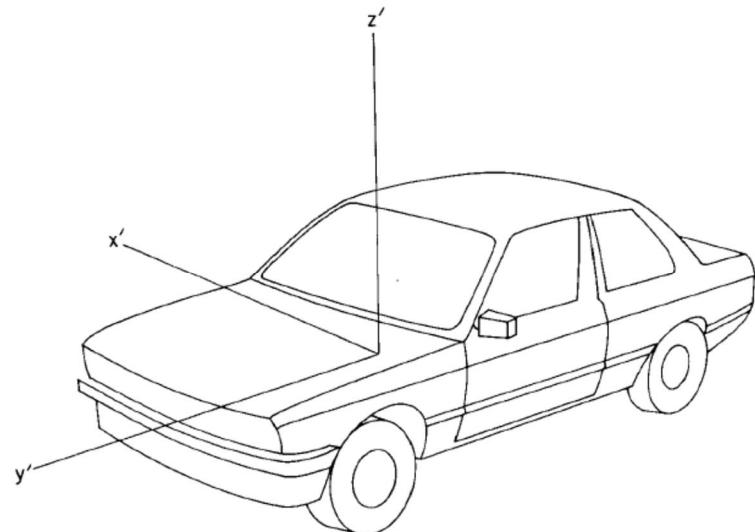


Figure 1.1.4 Automobile

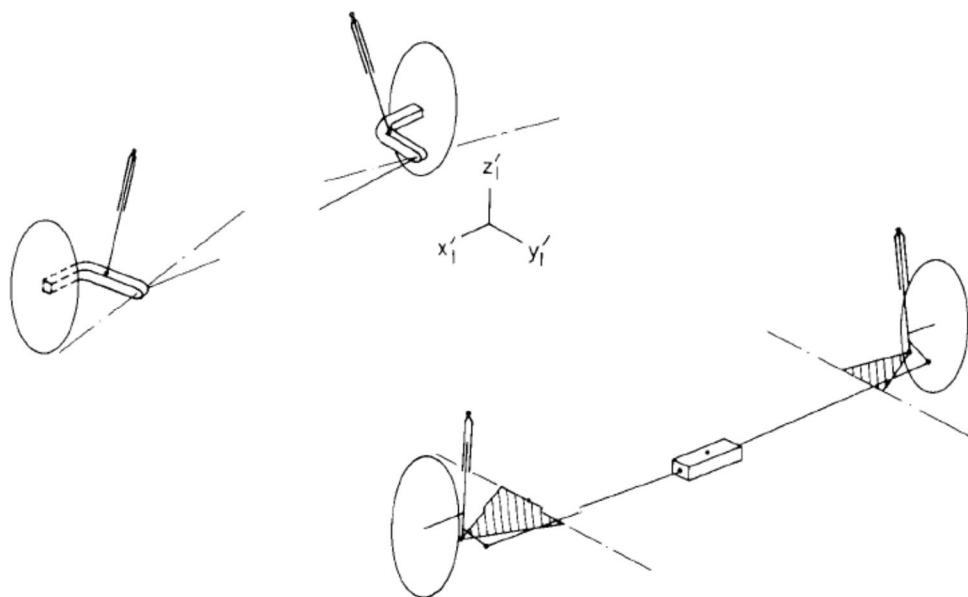


Figure 1.1.5 Automobile Suspension Schematic

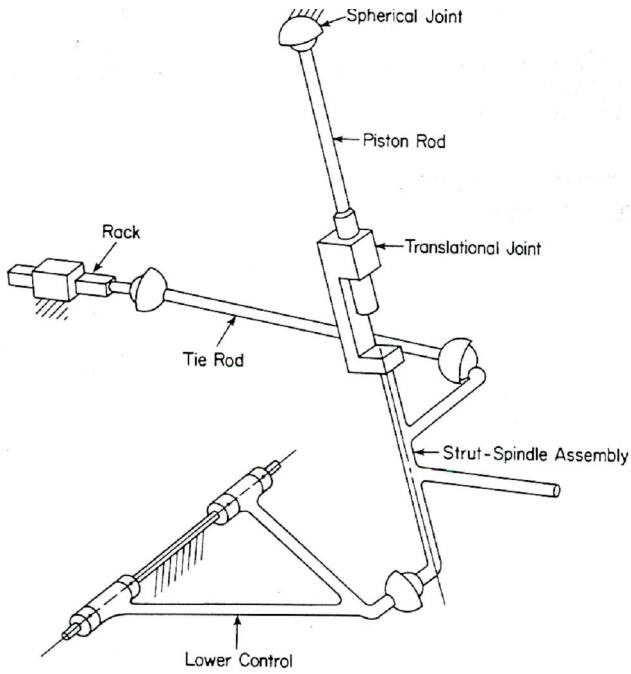


Figure 1.1.6 McPherson Strut Front Suspension

### 1.1.5 Serial Robotic Manipulator

The *robot*, or *manipulator*, shown in Fig. 1.1.7 is made up of nine bodies, including ground (body 1). The first degree of freedom is rotation  $q_1$  of the base (body 2) about a vertical axis fixed in ground. The second degree of freedom is rotation  $q_2$  of the pivot arm (body 3) about a horizontal axis fixed in body 2. The third degree of freedom is translation  $q_3$  of the boom (body 4) in a guide that is fixed in body 3. The fourth degree of freedom is rotation  $q_4$  of the first wrist pivot (body 5) relative to body 4. The fifth degree of freedom is rotation  $q_5$  of the second wrist pivot (body 6) relative to body 5. The sixth degree of freedom is rotation  $q_6$  of the hand mechanism (body 7) relative to body 6. The seventh degree of freedom is relative rotation  $q_7$  of the robot fingers (bodies 8 and 9). Such a mechanism permits the *end-effector* to grasp and *manipulate* a workpiece. As in previous examples, equations of kinematics and dynamics are nonlinear, enabling the robot to do things a linear system could not.

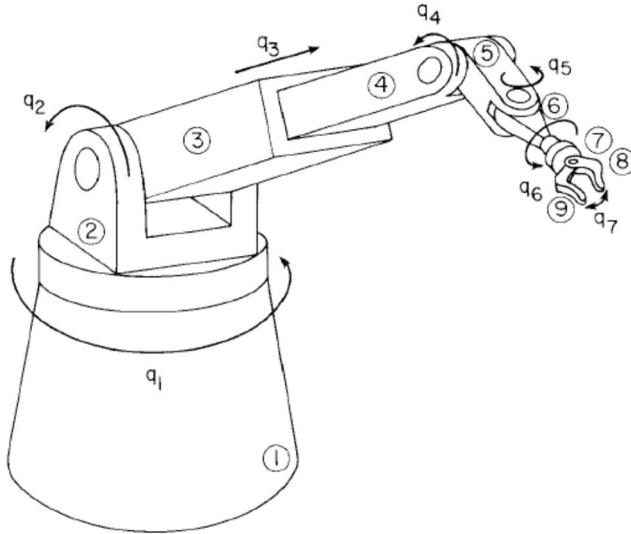


Figure 1.1.7 Serial Robotic Manipulator

This is a *serial manipulator*, in the sense that each of the inputs  $q_1, \dots, q_7$  determines the position and orientation of an outboard body, relative to its inboard body. This greatly simplifies the process of modeling and control of the robot. Considerations specific to a robot concern definition of admissible configurations; e.g., extreme position and orientation of the end-effector. Limits on configuration are determined by singularities in kinematic equations that define position and orientation of the end-effector, a special form of nonlinearity to must be dealt with.

### 1.1.6 Front-End Loader

The *front-end loader* shown in Fig. 1.1.8 is typical of high load capacity manipulators used in *construction*. The mechanism has three bodies that move relative to body 0 (the chassis), whose positions and orientations are defined in the chassis-fixed x-y frame by three *generalized coordinates* shown in Fig. 1.1.8,  $\mathbf{q} = [q_1 \quad q_2 \quad q_3]^T$ . *Hydraulic actuator inputs*  $\mathbf{y} = [y_1 \quad y_2]^T$  shown control motion of the system, and outputs are the orientation and elevation of the bucket,  $\mathbf{z} = [q_1 + q_3 \quad y_{D_2}]^T$ . Body 2 slides along the axis of body 1 (boom) and is connected to point D<sub>1</sub> on body 3 (bucket) by a bar of length 2 m, to control orientation of the bucket.

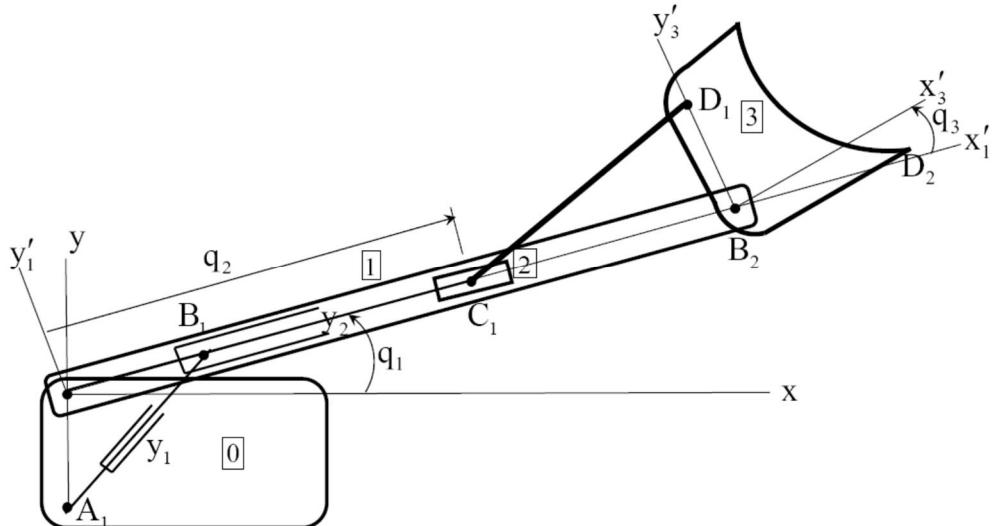


Figure 1.1.8 Front-End Loader

The mechanism of this manipulator contains a *closed kinematic loop* involving the slider, boom, bucket, and bar between the slider and bucket. This is called a *parallel manipulator*. It is more efficient in controlling motion under the influence of large loads than a serial manipulator, but the associated kinematics are much more complex. More specifically, if the lines C<sub>1</sub>D<sub>1</sub> and D<sub>1</sub>B<sub>2</sub> are colinear, the kinematic equations are *singular*. In case input q<sub>2</sub> is at its minimum value, the *singularity* precludes a decrease in q<sub>2</sub> and an increase in q<sub>2</sub> leads to one of two indeterminate rotations of the bucket. This form of kinematic singularity is highly nonlinear and must be dealt with in defining limits of functionality of the manipulator.

To assist the reader in highlighting key results from among often extensive derivations and analyses, a summary of each section is provided in a text box, such as the following:

Examples presented represent typical machines and machine elements that are encountered in mechanical system kinematics and dynamics. The breadth of such applications is extensive. A key feature that is pervasive in mechanical system kinematics and dynamics is nonlinearity in geometric and dynamic effects that must be accounted for. While applications differ greatly, many technical similarities permit the development of a uniform approach to computational kinematics and dynamics. Such an approach is presented in subsequent chapters of the text.

While the form of nonlinearity in kinematics and dynamics of systems discussed in this section will not be explicitly seen until Chapters 3 and 4, it is clear that output of these machines is not a linear function of input. Nonlinearity in machine kinematics and dynamics is both a curse and a blessing. Complexity of nonlinear systems is challenging, but is unavoidable in machines that exploit nonlinearity for the benefit of mankind.

## **1.2 Objectives and Foundations of Computational Kinematics and Dynamics**

The central objective of computational kinematics and dynamics is to create a formulation and associated digital computer implementation that (1) construct governing equations of kinematics and dynamics; (2) implement *numerical algorithms* that construct and solve the equations with specified precision, and (3) provide *digital computer software* that allows the engineer to input data that define a mechanical system, specify simulation inputs, specify forms of output desired, and carry out simulations. The essence of this objective is to make maximum use of the digital computer for rapid and accurate *data manipulation, model creation, and numerical computation*, hence relieving the engineer of tedious and error-prone manual calculations.

### **1.2.1 Emergence of the Discipline of Computational Dynamics**

The formulation of these objectives emerged from communications among an emerging community of engineers and scientists who contributed to their realization in the 1970s and 1980s. The first major gathering of this community in 1977 was a one-week symposium entitled “Dynamics of Multibody Systems” (Magnus, 1978). A second gathering in 1983 was a two-week NATO Advanced Study Institute on “Computer Aided Analysis and Optimization of Mechanical System Dynamics” (Haug, 1984). A third was a symposium in 1985 entitled “Dynamics of Multibody Systems” (Bianchi and Schiehlen, 1986). The fourth early symposium was in 1992, entitled “Advanced Multibody System Dynamics: Simulation and Software Tools” (Schiehlen, 1993). A noteworthy series of undocumented workshops on multibody simulation methods and software, involving leaders in mechanical and aerospace system dynamics, was hosted by the German Aerospace Establishment at Oberpfaffenhofen in the late 1980s and early 1990s. While many professional research symposia have since occurred, these five encounters of leaders in the emerging field of *multibody dynamics* were transformative. They brought together specialists that were addressing challenging applications in mechanical engineering, aerospace engineering, numerical analysis, and computational science, who were often unaware of each other’s work. The result was a cross fertilization that rarely occurs across diverse disciplines and, in this case, set off an eruption in the emerging field of *multibody dynamics*.

As became clear in the foregoing symposia and workshops, care must be taken to consider numerous alternatives that are available in selecting formulations and numerical methods to achieve significant advances in multibody kinematics and dynamics. Relative and absolute coordinate formulations were developed in the last quarter of the 20th century, using *topological analysis, symbolic computation, sparse matrix computation, and a variety of numerical integration methods* to automate the process of assembling and solving *equations of motion*. Computer codes that implement the methods, some of which are now available as commercial products, are outlined in a handbook and a monograph (Schiehlen, 1990; 1993).

### **1.2.2 Basic Elements of Mechanical System Dynamics**

To be specific regarding some of the issues that must be addressed in creating rigorous and practical methods that enable computer implementation of computational dynamics, an elementary example is studied. The goal of this exercise is to illustrate some of the challenges that must be overcome to achieve the objectives outlined in the foregoing.

---

**Example 1.2.1: Particle on Unit Sphere.** A particle of mass  $m$  is constrained to move on a *unit sphere*, as shown in Fig. 1.2.1. Denoting the coordinates of the particle in the  $x$ - $y$ - $z$  *inertial reference frame* as the  $3 \times 1$  matrix  $\mathbf{q} = [q_1 \ q_2 \ q_3]^T = [x \ y \ z]^T$ , where superscript T denotes matrix transpose, the *kinematic constraint* that the particle moves on the surface of the unit sphere is

$$\Phi(\mathbf{q}) = (\mathbf{q}^T \mathbf{q} - 1)/2 = (q_1^2 + q_2^2 + q_3^2 - 1)/2 = 0 \quad (1.2.1)$$

Defining the *constraint Jacobian* as the  $3 \times 1$  matrix

$$\Phi_q(\mathbf{q}) = [\partial\Phi / \partial q_1 \ \partial\Phi / \partial q_2 \ \partial\Phi / \partial q_3] = [q_1 \ q_2 \ q_3] = \mathbf{q}^T \quad (1.2.2)$$

and using the chain rule of differentiation, the associated *velocity constraint* and *acceleration constraint* are

$$\begin{aligned} \Phi_q \dot{\mathbf{q}} &= 0 \\ \Phi_q \ddot{\mathbf{q}} &= -\dot{\Phi}_q \dot{\mathbf{q}} = -\dot{\mathbf{q}}^T \dot{\mathbf{q}} = (\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2) \end{aligned} \quad (1.2.3)$$

where an over dot denotes derivative with respect to time; i.e.,  $\dot{\mathbf{q}} = d\mathbf{q}/dt$ . A *kinematically admissible virtual displacement* (see Section 4.12)  $\delta\mathbf{q} = [\delta q_1 \ \delta q_2 \ \delta q_3]^T$  is defined to be independent of time and to satisfy the linearized form of Eq. (1.2.1),

$$\Phi_q \delta\mathbf{q} = 0 \quad (1.2.4)$$

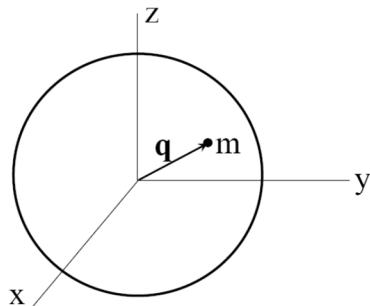


Figure 1.2.1 Particle on Unit Sphere

The d'Alembert form of the *variational equations of motion* (see Section 4.1.3), called the *fundamental equation* by Pars (1965), is

$$\delta\mathbf{q}^T (m\ddot{\mathbf{q}} + mg\mathbf{u}_z) = 0 \quad (1.2.5)$$

which must hold for all  $\delta\mathbf{q}$  that satisfy Eq. (1.2.4), where  $\mathbf{u}_z = [0 \ 0 \ 1]^T$  is the *unit vector* in the positive  $z$  direction and no constraint reaction forces are included. Note that Eq. (1.2.5) is much more than a statement  $\delta\mathbf{q}^T (m\ddot{\mathbf{q}} - \mathbf{F}) = 0$ . It is based on *d'Alembert's principle* (Lanczos, 1962; Pars, 1965), which states that *forces of constraint* do no work under the action of kinematically admissible virtual displacements.

There are two fundamentally different approaches to using Eqs. (1.2.4) and (1.2.5) to obtain equations of motion. The first is to work with *dependent generalized coordinates*  $\mathbf{q}$  that do not identically satisfy the constraints of Eqs. (1.2.1) and (1.2.3) and obtain the *Lagrange multiplier* form of *differential-algebraic equations (DAE)* of motion. The second approach is to define *independent generalized coordinates* (called *Lagrangian coordinates* (Pars, 1965) of dimension less than 3 that identically satisfy all three forms of the constraints of Eqs. (1.2.1) and (1.2.3) and obtain *ordinary differential equations (ODE)* of motion.

### **The DAE Approach**

In the DAE approach, since expressions in Eqs. (1.2.4) and Eq. (1.2.5) are linear in  $\delta\mathbf{q}$ , the *Lagrange multiplier theorem* (see Section 2.2.2) states that there exists a unique multiplier  $\lambda$  such that  $\delta\mathbf{q}^T(m\ddot{\mathbf{q}} + mg\mathbf{u}_z) + \lambda\Phi_q\delta\mathbf{q} = 0$  for arbitrary  $\delta\mathbf{q}$ , or  $\delta\mathbf{q}^T(m\ddot{\mathbf{q}} + \Phi_q^T\lambda + mg\mathbf{u}_z) = 0$ . Since  $\delta\mathbf{q}$  is arbitrary,

$$m\ddot{\mathbf{q}} + \Phi_q^T\lambda + mg\mathbf{u}_z = \mathbf{0} \quad (1.2.6)$$

Equation (1.2.6) is three scalar equations in the four variables  $\ddot{\mathbf{q}}$  and  $\lambda$ , the latter appearing algebraically. This equation, together with the algebraic constraint equation of Eq. (1.2.1) comprises a system of four differential and algebraic equations in the four variables  $\mathbf{q}$  and  $\lambda$ . This system of equations is not an ODE. It has been named a *DAE*, since  $\lambda$  appears algebraically and Eq. (1.2.1) is algebraic. A landmark paper (Petzold, 1982) shows that DAE are not ODE, and in fact they are much more difficult to solve numerically than ODE.

In an attempt to overcome the deficiency of Eq. (1.2.6) as three equations in four variables, some in the computational dynamics community append the second of Eqs. (1.2.3) to Eq. (1.2.6) to obtain the matrix equation

$$\begin{bmatrix} m\mathbf{I} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} -mg\mathbf{u}_z \\ -(\dot{\mathbf{q}}_1^2 + \dot{\mathbf{q}}_2^2 + \dot{\mathbf{q}}_3^2) \end{bmatrix} \quad (1.2.7)$$

This is four equations in four variables and the determinant of the coefficient matrix of Eq. (1.2.7), with  $\mathbf{q} = \mathbf{q}^T$  and  $\mathbf{q}^T\mathbf{q} = 1$ , is  $-m^2 \neq 0$ , so the equation has a unique solution for  $\ddot{\mathbf{q}}$  and  $\lambda$ . An ODE numerical integration formula (see Section 4.8) is then applied, with specified initial conditions, to determine  $\mathbf{q}$  and  $\dot{\mathbf{q}}$ . This computational approach suffers from two inconsistencies. First, applying an ODE integrator to a DAE is of questionable validity (Petzold, 1982). Second, the constraints of Eq. (1.2.1) and the first of Eqs. (1.2.3) are ignored, leading to unavoidable error in the approximate solution. Other manipulations outlined in Chapter 7 are often attempted in solution of DAE, but still lead to difficulties.

### **The ODE Approach**

As a second approach that avoids difficulties with DAE formulations, one may rely on the analytical dynamics mantra (Pars, 1965) of defining independent *Lagrangian generalized coordinates* that satisfy all three forms of the constraints of Eqs. (1.2.1) and (1.2.3) and apply the d'Alembert variational equation of motion of Eq. (1.2.5). Using independent *spherical coordinates*  $\theta$  and  $\phi$  shown in Fig. 1.2.2, the vector  $\mathbf{q}$  is written as a function of these independent coordinates,

$$\mathbf{q} = \begin{bmatrix} \cos \theta \cos \phi \\ \cos \theta \sin \phi \\ \sin \theta \end{bmatrix} \quad (1.2.8)$$

and their time derivatives and virtual displacements are

$$\dot{\mathbf{q}} = \begin{bmatrix} -\sin \theta \cos \phi & -\cos \theta \sin \phi \\ -\sin \theta \sin \phi & \cos \theta \cos \phi \\ \cos \theta & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix}$$

$$\delta \mathbf{q} = \begin{bmatrix} -\sin \theta \cos \phi & -\cos \theta \sin \phi \\ -\sin \theta \sin \phi & \cos \theta \cos \phi \\ \cos \theta & 0 \end{bmatrix} \begin{bmatrix} \delta \theta \\ \delta \phi \end{bmatrix} \quad (1.2.9)$$

$$\ddot{\mathbf{q}} = \begin{bmatrix} -\sin \theta \cos \phi & -\cos \theta \sin \phi \\ -\sin \theta \sin \phi & \cos \theta \cos \phi \\ \cos \theta & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} -\dot{\theta}^2 \cos \theta \cos \phi + 2\dot{\theta}\dot{\phi} \sin \theta \sin \phi - \dot{\phi}^2 \cos \theta \cos \phi \\ -\dot{\theta}^2 \cos \theta \sin \phi - 2\dot{\theta}\dot{\phi} \sin \theta \cos \phi - \dot{\phi}^2 \cos \theta \sin \phi \\ -\dot{\theta}^2 \sin \theta \end{bmatrix}$$

Substitution of  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  of Eqs. (1.2.8) and (1.2.9) into the constraints of Eqs. (1.2.1) and (1.2.3) it is verified that all three are satisfied identically; i.e.,  $\theta$  and  $\phi$  are Lagrangian generalized coordinates. Similarly, substituting  $\mathbf{q}$  of Eq. (1.2.9) into Eq. (1.2.5), that equation is identically satisfied.

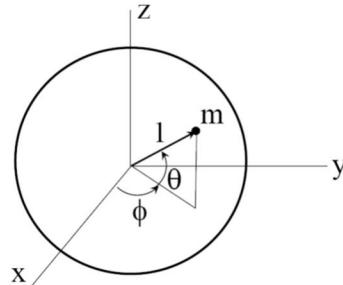


Figure 1.2.2 Spherical Coordinates for Particle on Unit Sphere

Substituting Eqs. (1.2.9) into Eq. (1.2.5) and simplifying results using trigonometric identities yields

$$[\delta\theta \quad \delta\phi] \left\{ m \begin{bmatrix} 1 & 0 \\ 0 & \cos^2 \theta \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} + m \begin{bmatrix} \dot{\phi}^2 \sin \theta \cos \theta \\ -2\dot{\theta}\dot{\phi} \cos \theta \sin \theta \end{bmatrix} + mg \begin{bmatrix} \cos \theta \\ 0 \end{bmatrix} \right\} = 0$$

Since  $\delta\theta$  and  $\delta\phi$  are arbitrary,

$$m \begin{bmatrix} 1 & 0 \\ 0 & \cos^2 \theta \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = -m \begin{bmatrix} \dot{\phi}^2 \sin \theta \cos \theta \\ -2\dot{\theta}\dot{\phi} \cos \theta \sin \theta \end{bmatrix} - mg \begin{bmatrix} \cos \theta \\ 0 \end{bmatrix} \quad (1.2.10)$$

This is a pair of *second order ODE* in  $\theta$  and  $\phi$ . However, the coefficient matrix of the acceleration vector is singular at  $\theta = \pm\pi/2$ ; i.e., at the poles on the z axis. If one attempts to

numerically solve Eq. (1.2.10), a division by zero occurs during solution for  $\ddot{\theta}$  and  $\ddot{\phi}$  when  $\dot{\alpha} = \pm \pi/2$ . This difficulty arises for a reason quite different from those encountered when using DAE.

In an attempt to resolve this difficulty, define a different set of *Lagrangian generalized coordinates*, such as  $\alpha$  and  $\beta$  in Fig. 1.2.3 where the vector  $\mathbf{q}$  is projected onto the x-z plane and the *Cartesian coordinates* of the particle are

$$\mathbf{q} = \begin{bmatrix} \cos \alpha \sin \beta \\ \sin \alpha \\ \cos \alpha \cos \beta \end{bmatrix} \quad (1.2.11)$$

Repeating the foregoing calculations, d'Alembert's equations of motion in the  $\alpha - \beta$  generalized coordinates are

$$m \begin{bmatrix} 1 & 0 \\ 0 & \cos^2 \alpha \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} = -m \begin{bmatrix} \dot{\beta}^2 \cos \alpha \sin \alpha \\ -2m\dot{\beta}\dot{\alpha} \cos \alpha \sin \alpha \end{bmatrix} - mg \begin{bmatrix} \sin \alpha \cos \beta \\ \cos \alpha \sin \beta \end{bmatrix} \quad (1.2.12)$$

In this formulation, a singularity occurs for  $\alpha = \pm \pi/2$ ; i.e., at the poles on the y axis. There is no singularity at the poles on the z axis, as was the case with the  $\theta - \phi$  generalized coordinates.

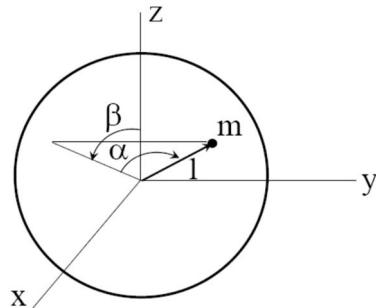


Figure 1.2.3 Alternative Coordinates for Particle on Unit Sphere

The problem here is with the coordinates used to parameterize the unit sphere, not with the smooth surface itself, which has no singularities. Topological arguments presented in Sections 3.5 and 3.6 show that there exists no pair of generalized coordinates that can *parameterize* the entire unit sphere without encountering a singularity; i.e., there exists no globally valid pair of Lagrangian generalized coordinates. This is true for even this elementary application, in which the surface is smooth. The moral is that if such fundamental difficulties are encountered in this trivial application, imagine the gremlins that lurk in real-world applications.

The point of Example 1.2.1 is that, if a system as simple as a particle constrained to move on a unit sphere is plagued with difficulties, real world applications are sure to be susceptible to even more complex problems. A rigorous and systematic formulation that defines *locally independent generalized coordinates* and formulates and solves the associated ODE of motion until analytical criteria dictate reformulation of the equations of motion and *continuation of integration* is presented in Chapter 5. This approach is the basis for *differential geometry* and *manifolds* that are introduced in advanced mathematics and physics literature (Arnold, V. I.,

1978; Abraham and Marsden, 1978). This broadly applicable approach is presented, with computer software implementation and numerical examples, in Chapter 5. Finally, it is emphasized that a closed form analytical solution is essentially never possible in mechanical system dynamics.

### 1.2.3 Foundations for Computational Mechanical System Kinematics and Dynamics

As illustrated by Example 1.2.1, the nonlinear nature of kinematics and dynamics of constrained mechanical systems is fraught with subtle challenges that can only be dealt with through a rigorous analytical foundation that can be systematically implemented in digital computation. The trivial example of a particle on a unit sphere shows that *singularities* are encountered due to selection of generalized coordinates if rational criteria are not established to avoid their occurrence; e.g., in the form of division by zero that must be avoided at all cost in digital computation. The classical theory of *analytical dynamics*, as best illustrated by the landmark text (Pars, 1965), assumes that generalized coordinates can be defined that are globally consistent with constraints on the system under consideration, called *Lagrangian generalized coordinates*. As shown in Example 1.2.1, this is not always possible. In fact, in representing orientation of bodies in space, even though there are three orientation degrees of freedom (see Section 2.4.3), any set of three orientation generalized coordinates must have *singular configurations* (Stuepnagel, 1964). A mathematically sound foundation for computational dynamics must therefore be established to avoid such singular configurations, through algorithms that select and modify independent generalized coordinates, as necessary, during simulation. In this sense, creation of rigorous computational formulations for mechanical system dynamics is even more challenging than in the ad-hoc approaches used in classical *analytical dynamics*.

In addition to singularities that may be associated with inappropriate selection of generalized coordinates, simulation of trial designs that exhibit physically *singular behavior*, such as *lock-up* or *bifurcation*, must adapt to sense onset of such singular behavior and report it. *Condition numbers* of near singular matrices may be monitored (see Section 2.2.7) and increasing numbers of iterations required for solution of nonlinear equations (see Section 2.2.6) may be used as computable tests of onset of singular behavior. Similarly, *stiff behavior* (see Section 4.7) of the equations of dynamics for well-designed systems that use high levels of damping to suppress unwanted high frequency oscillation lead to severe difficulties if numerical integration methods are used that are unable to adapt to such behavior.

Extensive research in numerical analysis has been carried out since the 1980's, focused on effective methods of solving ODE and DAE that arise in numerous fields of application. Landmark texts that document the state of the art of numerical solution methods have been published by Hairer and coauthors (1993; 1996). The DAE of mechanical system dynamics, unfortunately, fall into a class called *index three DAE* that is difficult to solve numerically (Ascher and Petzold, 1998). An extensive treatment of theoretical and numerical analysis of DAE has been presented by Rabier and Rheinboldt (2002), at a level of mathematical sophistication beyond the reach of we mere mortal engineers. Because the solution of ODE and DAE is critical for mechanical system dynamics, and because the literature on the subject is relatively new and tends to be mathematically abstract, attention is given in the text to basic concepts, at a mathematical level that is accessible to the intended engineering audience.

An extensive mathematical literature on *differential geometry* has emerged (Spivak, 1999; Dundas, 2018) that holds enormous potential for contribution to kinematics and dynamics

of mechanical systems. Unfortunately, both the mathematical literature and its application to date to kinematics and dynamics (Arnold, 1978; Abraham and Marsden, 1978) are at a level of mathematical abstraction that preclude its digestion by most engineers. Basic concepts from this literature have been adapted in this text that hopefully makes these important tools accessible to the intended audience (see Sections 3.5 and 3.6).

In addition to the requirement that rigorous mathematical and numerical analysis foundations be established for computational dynamics, there must be a basic software underpinning that permits experimentation and practical learning. The author obtained a MATLAB license in 2015 and began relearning programming, after a 49 year period without having written a line of code. This humbling experience has led to MATLAB programs that are documented in the text and contained in appendices to chapters two through nine that implement methods presented. The reader is encouraged to use these codes to try out ideas and designs and to report what are surely bugs and inadequacies in the programs to the author, including suggested improvements, if possible.

Computational kinematics and dynamics emerged in the last third of the 20<sup>th</sup> century, in parallel with evolution of the digital computer, and has become a vibrant discipline that supports numerous areas of engineering and applied science. Realization of its full potential relies on rigorous methods of analytical dynamics, mathematics, numerical analysis, and computational science. Excellent texts and monographs in each of these disciplines exist, written primarily for specialists in their discipline, but are not readily accessible to the engineer who is attempting to advance or use the technology of mechanical system kinematics and dynamics. A goal of this text is to bridge the gap and bring this modern technology to bear on computational methods of mechanical system kinematics and dynamics.

### ***1.3 Organization of the Text***

Building on the foundations outline in Section 1.2.3, basic methods of vector spaces and multivariable calculus are employed in Chapter 3 to define the position and orientation  $n$  of bodies in the plane and in three-dimensional space. A summary of vector analysis and linear algebra is presented in Sections 2.1 and 2.2, including the Lagrange multiplier theorem of linear algebra and the matrix condition number. The implicit function theorem of multivariable calculus is presented as a key tool in mechanical system kinematics. Euler parameter orientation coordinates are defined and numerous identities that form the foundation for orientation of bodies in space are derived. MATLAB computer codes for implementation of results presented are included in Appendix 2.B.

A broadly applicable formulation for kinematics of planar and spatial multibody systems is presented in Chapter 3. Libraries of kinematic constraints and their derivatives that are required for kinematic configuration, velocity, and acceleration analysis are derived and implemented in MATLAB computer codes of Appendix 3.A, for kinematic analysis of planar and spatial systems. Concepts of configuration spaces and kinematic constraint manifolds are introduced, in the context of differential geometry, in Sections 3.4 through 3.6, providing a foundation for analysis of singular configurations of mechanisms in Section 3.7. Formulations and computer codes for general purpose kinematic analysis of planar and spatial systems are presented in Sections 3.8 through 3.12, including numerical applications.

Equations of motion for systems of particles, planar bodies, and spatial bodies are derived in Sections 4.1 through 4.4, using only Newton's three laws of motion for a particle and d'Alembert's principle of virtual work. Internal forces and kinematic constraints between bodies that comprise a multibody system are used with d'Alembert's principle to obtain variational equations of motion in Sections 4.5 and 4.6. ODE of dynamics are derived in Section 4.7 for systems in which independent generalized coordinates can be defined. Runge-Kutta numerical integration methods for ODE of dynamics are presented in Section 4.8 and implemented in MATLAB computer code in Section 4.8 and Appendix 4.A. Finally, constraint reaction forces and DAE of multibody system dynamics are derived in Section 4.9.

Methods of differential geometry are used with the kinematics formulation of Chapter 3 and the variational equations of dynamics of Chapter 4 to create a broadly applicable ODE formulation for dynamics of holonomically constrained multibody systems in Chapter 5. Numerical solution methods and results of Section 4.8 are derived and implemented in MATLAB computer codes for dynamic simulation of planar and spatial multibody systems. Numerical examples are presented using general purpose MATLAB codes for planar and spatial multibody dynamics in Sections 5.7 through 5.10. Finally, it is proved in Section 5.11 that four formulations of multibody dynamics that have been derived are equivalent and well-posed; i.e., they have unique solutions that are differentiable functions of design variables and model parameters.

Parts of an earlier version of the text were used as the basis for a graduate course on multibody dynamics with Professor Bahram Ravani at the University of California-Davis in the fall of 2020. Chapters 1 through 5 were used as the basis for a one semester graduate course on multibody dynamics with Professor Corina Sandu at Virginia Tech in the fall of 2021. The remainder of the text was used as a second semester graduate course at Virginia Tech on

advanced dynamics in the spring of 2022. The author thanks Professors Ravani and Sandu for the opportunity to use and refine the text based on interaction with their graduate students.

The differential geometric approach to ODE of Holonomic multibody systems of Chapter 5 is extended to treat equality constrained nonholonomic systems in Chapter 6. A formulation that includes holonomic and velocity equality constraints is based on parameterizations at the configuration and velocity levels that lead to a system of first order ODE that are shown to be well-posed. Numerical examples are presented that verify theoretical properties of the ODE formulation and its computational implementation.

Three approximate DAE formulations of holonomic multibody dynamics that have been used in recent literature are presented in Chapter 7; the Index I, II, and III formulations. Numerical implementation of the methods in MATLAB computer code for nonstiff and stiff multibody systems provide insight into positive and negative aspects of the three formulations, consistent with theoretical results presented by Petzold (1982).

Methods for simulation of multibody systems with friction are presented in Chapter 8. An initial study of model problems with the discontinuous Coulomb friction model suggests that alternative formulations are required for multibody dynamic simulation. A recently developed continuous friction model (Brown and McPhee, 2016) is implemented in the general purpose planar and spatial formulations and MATLAB codes of Chapter 5. Simulation of model problems shows that results obtained are good approximations to both dynamic and stiction effects. It is shown that explicit numerical integration methods perform well on model problems presented in the literature that are reported as stiff when using DAE solution methods. This shows that these model problems are not stiff and suggests that numerical difficulties reported are associated with use of DAE integration methods for simulation of multibody systems with friction.

The final chapter of the text, Chapter 9, treats manipulator kinematics and dynamics. Part I of the chapter deals with nonredundant manipulators for which there are equal numbers of input and output coordinates. A manipulator differentiable manifold formulation is developed that enables classification of manipulators into four distinct categories, each with special analytical and computational attributes. It is shown that the regular manipulator configuration space is naturally partitioned into maximal, singularity free, connected components on which differentiable forward and inverse kinematic configuration, velocity, and acceleration mappings exist and are practically computable. The mapping results are used with the variational equations of motion of Chapter 4 to obtain ODE of manipulator dynamics in both input and output coordinates. Part II of the chapter deals with kinematically redundant manipulators, for which the dimension of input coordinates is greater than that of the output coordinates. As a result, while forward kinematics is the same as with nonredundant manipulators, inverse kinematics is set valued, leading to self-motion manifolds that require the full range of capabilities offered by differential geometry.

The first five chapters of the text provide a relatively complete treatment of planar and spatial rigid body kinematics and dynamics. The development employs classical methods of vector analysis and multivariable calculus, together with modest extensions that are based on modern methods of differential geometry in Euclidean space. A broadly applicable variational formulation of multibody system dynamics is developed, using only Newton's laws for particles and d'Alembert's principle. The ODE formulation of Chapter 5 is combined with Runge-Kutta numerical integration methods to obtain general-purpose MATLAB simulation software. The last four chapters of the text provide independent treatments of nonholonomic system dynamics, DAE numerical solution methods, dynamics of systems with friction, and manipulator kinematics and dynamics.

## CHAPTER 2

### Position and Orientation of Bodies

#### 2.0 Introduction

Kinematics of a multibody system defines admissible configurations and motions of bodies in the system, consistent with *joints* that constrain relative motion of pairs of bodies. Admissible configurations (positions and orientations) of bodies that make up the system are described by vectors, matrices, and *generalized coordinates* that define the position and orientation of each body in the system. A summary of basic concepts of vector analysis is presented in Section 2.1, beginning with conventional *geometric vectors* and proceeding to *algebraic vectors* that are comprised of components of geometric vectors relative to an orthogonal reference frame. Algebraic vectors are shown to be ideally suited for computer implementation, in contrast to their geometric counterparts. Time derivatives of vectors are introduced, as the basis for velocity and acceleration analysis that is required for multibody kinematics and dynamics.

Section 2.2 continues with a summary of basic results from *matrix algebra* and *multipariable calculus* that form the foundation for computational kinematics and dynamics. A Lagrange multiplier theorem of linear algebra is proved, as the basis for representing constraint reaction forces in system dynamics. Key elements of multivariable calculus that form the foundation for system kinematic analysis are summarized, including the *implicit function theorem* that assures existence and differentiability of solutions of the nonlinear equations of kinematics. Concepts of scalar product of n-dimensional vectors and distance between vectors are introduced to define the *Euclidean space*  $R^n$ . Continuity of functions in this setting is defined, to form the foundation for *differential geometry on  $R^n$* , the bedrock on which mechanical system kinematics and dynamics is built. Finally, the *Newton-Raphson* iterative numerical method is presented for computer solution of nonlinear equations in Euclidean space, with definition of matrix *condition number* that is used in *numerical error control*.

The remainder of the chapter is focused on methods for defining the position and orientation of bodies that make up multibody systems. A characterization of *orientation of bodies* that move in a plane is presented in Section 2.3, as the basis for *planar system kinematics*. The foundation for defining orientation of bodies in *three-dimensional space* is presented in Section 2.4, in terms of *orientation transformation matrices*. It is shown that orientation of a body in space has three degrees of freedom, independent of generalized coordinates that may be used to define orientation. *Velocity* and *acceleration* relations are derived, as the basis for spatial system analysis. *Virtual displacements* needed for system dynamics are defined, using analogous relations.

*Euler parameter generalized coordinates* that define orientation of bodies in space and serve as the foundation for spatial kinematics and dynamics throughout the remainder of the text are presented in Section 2.5. It is shown that there is locally a one-to-one relation between Euler parameters and elements of the orientation transformation matrix, so Euler parameters are singularity free. *Identities and derivative operators* involving Euler parameters that serve as the basis for kinematic and dynamic modeling and computer simulation are derived in Section 2.6. These identities and operators form the basis for a library of computer subroutines in Appendix 2.C for implementation of kinematic equations in digital computation.

Classical *Euler angles* are defined in Appendix 2.A, as three generalized coordinates that characterize orientation of bodies in space. The orientation transformation matrix is written as a function of Euler angles and is shown to have a singularity that must be avoided in practice. Literature is cited that shows singularities arise for any set of three orientation generalized coordinates. Appendix 2.B focuses on issues and progress in defining and using generalized coordinates that effectively represent orientation of bodies in space and are well suited for digital computation.

MATLAB computer code is provided in Appendix 2.C to implement vector and matrix operators that are derived in the chapter. Code 2.4 evaluates the angle of relative rotation and its derivatives between bodies in space that are constrained to admit relative rotation about a common axis. Code 2.5 evaluates the orientation transformation matrix for a body, using vector data or Euler's theorem, and evaluates Euler parameters that characterize orientation for a given orthogonal orientation transformation matrix. Code 2.6 evaluates Euler parameter derivative operators that are derived in Section 2.6 and form the basis for numerical methods throughout the text. Finally, a compilation of key formulas derived in Chapter 2 is presented in Appendix 2.D, summarizing important and often used results that need not be memorized.

## 2.1 Vector Analysis

Vectors play a crucial role in locating points in the plane and in space and characterizing geometric relationships such as *orthogonality* and *parallelism*. The conventional geometric definition of vectors serves these purposes, but is not well suited for computation. Following a brief summary of *geometric vector analysis*, an equivalent matrix form of *algebraic vector analysis* that is ideally suited for computation is introduced. A reader who is not conversant in the basics of matrices should read Section 2.2 before studying the contents of this section.

### 2.1.1 Geometric Vectors

The concept of a vector in space may be introduced in a geometric setting, with no requirement for identification of a reference frame. In this setting, a *geometric vector*, or simply a *vector*, is defined as the *directed line segment* from one point in space to another point in space. Vector  $\vec{a}$  in Fig. 2.1.1 is a directed line segment that begins at point A and ends at point B. The *magnitude of a vector*  $\vec{a}$  is its length (the distance between A and B) and is denoted by  $a$ , or  $|\vec{a}|$ . Note that the length of a vector is positive if points A and B do not coincide and is zero if and only if they coincide. A vector with zero length is denoted as  $\vec{0}$  and is called the *zero vector*.

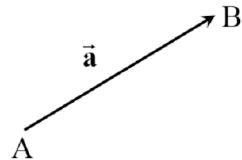


Figure 2.1.1 Vector  $\vec{a}$  From Point A to Point B

Multiplication of a vector  $\vec{a}$  by a nonnegative *scalar*  $\geq 0$  is defined as the vector  $\vec{a}$  in the same direction as  $\vec{a}$ , but having magnitude  $|\vec{a}|$ . A *unit vector*, having length 1 unit, in the direction  $\vec{a} \neq \vec{0}$  is  $(1/a)\vec{a}$ . Multiplication of a vector  $\vec{a}$  by a negative scalar  $< 0$  is defined as the vector with magnitude  $| |\vec{a}| |$  and direction opposite to that of  $\vec{a}$ . The *negative of a vector* is obtained by multiplying the vector by  $-1$ . It is the vector with the same magnitude but opposite direction.

**Example 2.1.1:** Let points A and B in Fig. 2.1.1 be located in an x-y-z *orthogonal reference frame* shown in Fig. 2.1.2; i.e., a reference frame with orthogonal axes.

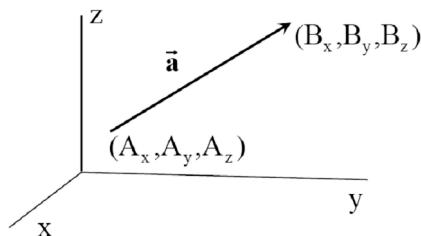


Figure 2.1.2 Vector  $\vec{a}$  Located in Orthogonal Reference Frame

The distance between points A and B, with coordinates  $(A_x, A_y, A_z)$  and  $(B_x, B_y, B_z)$ , respectively, is the length of  $\vec{a}$ ; i.e.,

$$|\vec{a}| = \left[ (B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2 \right]^{1/2}$$


---

Nonzero vectors  $\vec{a}$  and  $\vec{b}$  are added according to the *parallelogram rule*, as shown in Fig. 2.1.3. The parallelogram used in this construction is formed in the plane that contains the intersecting vectors  $\vec{a}$  and  $\vec{b}$ . The *vector sum* is written as

$$\vec{c} = \vec{a} + \vec{b} \quad (2.1.1)$$

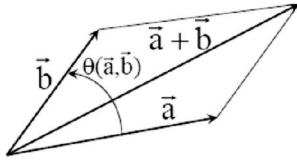


Figure 2.1.3 Addition of Vectors

Addition of vectors and multiplication of vectors by scalars obey the following rules (Kreyszig, 2011)

$$\begin{aligned} \vec{a} + \vec{b} &= \vec{b} + \vec{a} \\ (\lambda + \mu)\vec{a} &= \lambda\vec{a} + \mu\vec{a} \end{aligned} \quad (2.1.2)$$

where  $\lambda$  and  $\mu$  are scalars.

Using the definition of vector addition, the plane formed by intersecting vectors  $\vec{a}$  and  $\vec{b}$  in Fig. 2.1.3 may be called the x'-y' plane, with the x' axis along vector  $\vec{a}$  and the y' axis perpendicular to the x' axis, as shown in Fig. 2.1.4. The x'-y' frame is called right handed if rotating the x' axis to the y' axis with the right hand leads to the thumb pointing out of the plane toward the viewer. Plane trigonometry can now be used to calculate the length of vector  $\vec{a} + \vec{b}$  and the angle  $\theta(\vec{a}, \vec{a} + \vec{b})$  between  $\vec{a}$  and  $\vec{a} + \vec{b}$ ,

$$\begin{aligned} |\vec{a} + \vec{b}| &= \left[ (a + b \cos \theta(\vec{a}, \vec{b}))^2 + (b \sin \theta(\vec{a}, \vec{b}))^2 \right]^{1/2} \\ \theta(\vec{a}, \vec{a} + \vec{b}) &= \text{Arctan} \left[ \frac{b \sin \theta(\vec{a}, \vec{b})}{a + b \cos \theta(\vec{a}, \vec{b})} \right] \end{aligned} \quad (2.1.3)$$

These calculations define the vector  $\vec{a} + \vec{b}$ , but are not convenient for computation.

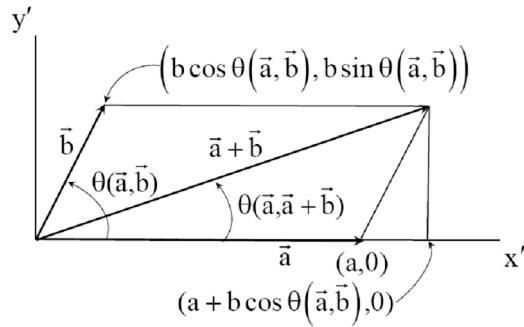


Figure 2.1.4 Sum of Vectors in  $x'$ - $y'$  Plane of Intersection

*Orthogonal reference frames* are used extensively in representing vectors. Use in this text is limited to *right-hand orthogonal reference frames*; i.e., with mutually orthogonal  $x$ ,  $y$ , and  $z$  axes that are ordered by the finger structure of the right hand, as shown in Fig. 2.1.5. Such a frame is called a *Cartesian reference frame*.

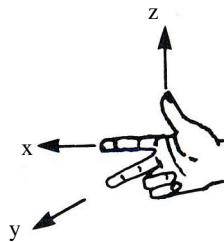


Figure 2.1.5 Right-Hand Orthogonal Reference Frame

A vector  $\vec{a}$  can be resolved into components  $a_x$ ,  $a_y$ , and  $a_z$  along the  $x$ ,  $y$ , and  $z$  axes of a Cartesian reference frame, via orthogonal projection onto the axes, as shown in Fig. 2.1.6. These components are called the *Cartesian components of a vector*. The *unit coordinate vectors*  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$  are unit vectors that are directed along the positive  $x$ ,  $y$ , and  $z$  axes, respectively, as shown in Fig. 2.1.6. In vector notation,

$$\vec{a} = a_x \vec{i} + a_y \vec{j} + a_z \vec{k} \quad (2.1.4)$$

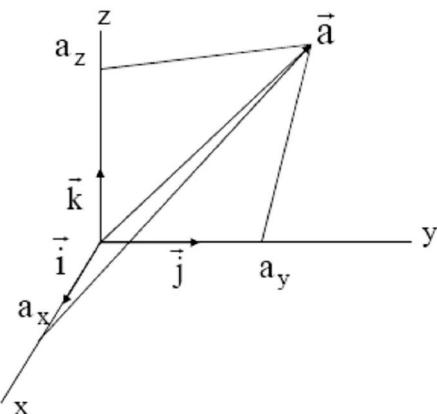


Figure 2.1.6 Components of a Vector

Denote the angle from vector  $\vec{a}$  to vector  $\vec{b}$  in the plane that contains them by  $(\vec{a}, \vec{b})$ , with *counterclockwise as positive* about the normal to the plane of the vectors that points toward the viewer, as shown in Fig. 2.1.3. In terms of angles between the vector  $\vec{a}$  and unit vectors along the positive x, y, and z axes, the *Cartesian components of a vector*, or simply *components of a vector*  $\vec{a}$  relative to the x-y-z frame are

$$\begin{aligned} a_x &= \cos(\vec{i}, \vec{a}) \\ a_y &= \cos(\vec{j}, \vec{a}) \\ a_z &= \cos(\vec{k}, \vec{a}) \end{aligned} \quad (2.1.5)$$

The quantities  $\cos(\vec{i}, \vec{a})$ ,  $\cos(\vec{j}, \vec{a})$ , and  $\cos(\vec{k}, \vec{a})$  are called the *direction cosines* of vector  $\vec{a}$ . Note that if the viewer had been on the back side of the plane of Fig. 2.1.3, the counterclockwise angle from  $\vec{a}$  to  $\vec{b}$  would be  $2 - (\vec{a}, \vec{b})$ . However,  $\cos(2 - ) = \cos$ , so the viewpoint does not influence the value of direction cosines.

Addition of vectors  $\vec{a}$  and  $\vec{b}$  may be expressed in terms of their components, using Eq. (2.1.4), as

$$\begin{aligned} \vec{c} &= \vec{a} + \vec{b} = (a_x + b_x)\vec{i} + (a_y + b_y)\vec{j} + (a_z + b_z)\vec{k} \\ &\equiv c_x\vec{i} + c_y\vec{j} + c_z\vec{k} \end{aligned} \quad (2.1.6)$$

where  $c_x = a_x + b_x$ ,  $c_y = a_y + b_y$ , and  $c_z = a_z + b_z$  are the components of vector  $\vec{c}$ . Thus, addition of vectors occurs component-by-component. Using this idea, three vectors  $\vec{a}$ ,  $\vec{b}$ , and  $\vec{c}$  may be added to show that

$$(\vec{a} + \vec{b}) + \vec{c} = \vec{a} + (\vec{b} + \vec{c}) \quad (2.1.7)$$

The *scalar product* (sometimes called the *dot product*) of vectors  $\vec{a}$  and  $\vec{b}$  is defined as the product of their magnitudes and the cosine of the angle between them; i.e.,

$$\vec{a} \cdot \vec{b} = ab \cos(\vec{a}, \vec{b}) \quad (2.1.8)$$

This definition is purely geometric, so it is independent of any reference frame in which the vectors may be represented.

Note that if two vectors  $\vec{a}$  and  $\vec{b}$  are nonzero; i.e.,  $a \neq 0 \neq b$ , then their scalar product is zero if and only if  $\cos(\vec{a}, \vec{b}) = 0$ . Two nonzero vectors are said to be *orthogonal vectors* if their scalar product is zero.

Since  $(\vec{b}, \vec{a}) = 2 - (\vec{a}, \vec{b})$  and  $\cos(2 - ) = \cos$ , the order of terms appearing on the right side of Eq. (2.1.8) is immaterial. Thus,

$$\vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a} \quad (2.1.9)$$

The scalar product of two vectors may be interpreted as the product of the magnitude of one of the vectors times the orthogonal projection of the other vector onto that vector. To see this, refer to Fig. 2.1.4, where the projection of vector  $\vec{b}$  onto  $\vec{a}$  has length  $b \cos(\vec{a}, \vec{b})$ .

Based on the definition of scalar product, the following identities hold for the unit coordinate vectors  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$ :

$$\begin{aligned}\vec{i} \cdot \vec{j} &= \vec{j} \cdot \vec{k} = \vec{k} \cdot \vec{i} = 0 \\ \vec{i} \cdot \vec{i} &= \vec{j} \cdot \vec{j} = \vec{k} \cdot \vec{k} = 1\end{aligned}\quad (2.1.10)$$

For any vector  $\vec{a}$ ,

$$\vec{a} \cdot \vec{a} = a \cos 0 = a^2$$

While not obvious on geometrical grounds, the scalar product satisfies the relation (Kreyszig, 2011)

$$(\vec{a} + \vec{b}) \cdot \vec{c} = \vec{a} \cdot \vec{c} + \vec{b} \cdot \vec{c} \quad (2.1.11)$$

Using Eq. (2.1.11) and the identities of Eq. (2.1.10), a direct calculation yields

$$\vec{a} \cdot \vec{b} = a_x b_x + a_y b_y + a_z b_z \quad (2.1.12)$$

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

$$\vec{a} \times \vec{b} \equiv a \sin(\vec{a}, \vec{b}) \vec{u} \quad (2.1.13)$$

where  $\vec{u}$  is the unit vector that is orthogonal (perpendicular) to the plane of intersection of vectors  $\vec{a}$  and  $\vec{b}$ , taken in the positive right-hand coordinate direction, as shown in Fig. 2.1.7. If the viewer were behind the plane of Fig. 2.1.7, the unit normal to the plane would be  $-\vec{u}$  and the counterclockwise angle from  $\vec{a}$  to  $\vec{b}$  would be  $2 - (\vec{a}, \vec{b})$ . Then, the vector product would be

$$\vec{a} \times \vec{b} \equiv a \sin(2 - (\vec{a}, \vec{b}))(-\vec{u}) = a \sin(\vec{a}, \vec{b}) \vec{u}$$

since  $\sin(2 - \theta) = \sin \theta$ . This is the same result as in Eq. (2.1.13), so the viewpoint does not influence the value of the vector product. Since the definition of vector product is purely geometrical, the result is independent of the reference frame.

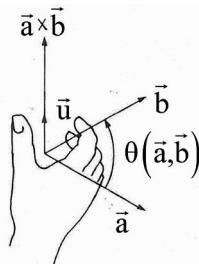


Figure 2.1.7 Vector Product

It is important to note that if  $|\vec{a}| \neq 0 \neq |\vec{b}|$  and  $\vec{a} \times \vec{b} = \vec{0}$ , then  $\sin(\vec{a}, \vec{b}) = 0$  and the vectors are parallel. Equivalently, if  $\vec{a} \times \vec{b} = \vec{0}$ , then  $\vec{a} = \alpha \vec{b}$  for some scalar  $\alpha$ .

Since reversal of the order of vectors  $\vec{a}$  and  $\vec{b}$  in Eq. (2.1.13) yields the opposite direction for the unit vector  $\vec{u}$ ,

$$\vec{a} \times \vec{b} = -\vec{b} \times \vec{a} \quad (2.1.14)$$

Analogous to Eq. (2.1.11), the vector product satisfies (Kreyszig, 2011)

$$(\vec{a} + \vec{b}) \times \vec{c} = \vec{a} \times \vec{c} + \vec{b} \times \vec{c} \quad (2.1.15)$$

Since  $(\vec{a}, \vec{a}) = 0$ , for any vector  $\vec{a}$ ,

$$\vec{a} \times \vec{a} = a^2 \sin 0 \vec{u} = \vec{0}$$

From the definition of unit coordinate vectors and vector product, the following identities are valid:

$$\begin{aligned} \vec{i} \times \vec{i} &= \vec{j} \times \vec{j} = \vec{k} \times \vec{k} = \vec{0} \\ \vec{i} \times \vec{j} &= -\vec{j} \times \vec{i} = \vec{k} \\ \vec{j} \times \vec{k} &= -\vec{k} \times \vec{j} = \vec{i} \\ \vec{k} \times \vec{i} &= -\vec{i} \times \vec{k} = \vec{j} \end{aligned} \quad (2.1.16)$$

Using the identities of Eqs. (2.1.15) and (2.1.16), the vector product of two vectors may be expanded and written in terms of their components as

$$\begin{aligned} \vec{c} = \vec{a} \times \vec{b} &= (a_y b_z - a_z b_y) \vec{i} + (a_z b_x - a_x b_z) \vec{j} + (a_x b_y - a_y b_x) \vec{k} \\ &\equiv c_x \vec{i} + c_y \vec{j} + c_z \vec{k} \end{aligned} \quad (2.1.17)$$

where  $c_x = a_y b_z - a_z b_y$ ,  $c_y = a_z b_x - a_x b_z$ , and  $c_z = a_x b_y - a_y b_x$ .

## 2.1.2 Algebraic Vectors

Recall from Eq. (2.1.4) that a vector  $\vec{a}$  can be written in component form in a Cartesian x-y-z frame as

$$\vec{a} = a_x \vec{i} + a_y \vec{j} + a_z \vec{k}$$

The geometric vector  $\vec{a}$  is thus uniquely defined by its Cartesian components, which may be written in matrix notation as

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T \quad (2.1.18)$$

This is the *algebraic representation of a geometric vector*. Note that the algebraic representation of vectors is dependent on the Cartesian reference frame selected; i.e., vectors  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$ . Some

of the purely geometric properties of vectors are thus lost, and properties of the reference frame that is used in defining components of vectors come into play. This apparent problem will in fact become a valuable tool in spatial kinematics and dynamics.

An *algebraic vector* is defined as a *column matrix*, or *column vector*. When an algebraic vector represents a geometric vector in three-dimensional space, it has three components. Algebraic vectors with more than three components will also be employed in kinematic and dynamic analysis of multibody systems. If  $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_n]^T$ , the algebraic vector  $\mathbf{a}$  is called an *n-vector* that belongs to *n-dimensional Euclidean space*, denoted  $\mathbb{R}^n$ .

If two geometric vectors  $\vec{a}$  and  $\vec{b}$  are represented in algebraic form as

$$\begin{aligned}\mathbf{a} &= \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T \\ \mathbf{b} &= \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T\end{aligned}\tag{2.1.19}$$

then Eq. (2.1.6) shows that their vector sum is represented in algebraic form as

$$\mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{bmatrix} a_x + b_x & a_y + b_y & a_z + b_z \end{bmatrix}^T\tag{2.1.20}$$

This simple algebraic relation, which is ideal for computation, is in contrast with the trigonometric complexity of the parallelogram rule of vector addition in Fig. 2.1.3.

**Example 2.1.2:** The algebraic representation of vectors

$$\begin{aligned}\vec{a} &= \vec{i} + 2\vec{j} + 3\vec{k} \\ \vec{b} &= -\vec{i} + \vec{j} - \vec{k}\end{aligned}$$

is

$$\begin{aligned}\mathbf{a} &= \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T \\ \mathbf{b} &= \begin{bmatrix} -1 & 1 & -1 \end{bmatrix}^T\end{aligned}$$

The algebraic representation of the sum  $\vec{c} = \vec{a} + \vec{b} = \vec{i} + 2\vec{j} + 3\vec{k} - \vec{i} + \vec{j} - \vec{k} = 0\vec{i} + 3\vec{j} + 2\vec{k}$  is

$$\mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{bmatrix} 0 & 3 & 2 \end{bmatrix}^T$$

Geometric vectors  $\vec{a}$  and  $\vec{b}$  are *equal vectors* if and only if their Cartesian components are equal; i.e.,  $\mathbf{a} = \mathbf{b}$ . Multiplication of a vector  $\vec{a}$  by a scalar  $\alpha$  occurs component-by-component, so the geometric vector

$$\vec{a} = (a_x \vec{i} + a_y \vec{j} + a_z \vec{k}) = a_x \vec{i} + a_y \vec{j} + a_z \vec{k}$$

is represented by the algebraic vector  $\mathbf{a}$ .

Since there is a one-to-one correspondence between geometric vectors and  $3 \times 1$  algebraic vectors that are formed from their components in a specified Cartesian reference frame, no distinction other than notation will be made between them in the remainder of the text.

The *scalar product* of geometric vectors  $\vec{a}$  and  $\vec{b}$  may be expressed in algebraic form, using the result of Eq.(2.1.12) , as

$$\vec{a} \cdot \vec{b} = a_x b_x + a_y b_y + a_z b_z = \mathbf{a}^T \mathbf{b} \quad (2.1.21)$$

This simple algebraic relation, which is ideally suited for digital computation, contrasts with the calculation of Eq. (2.1.3).

---

**Example 2.1.3:** The scalar product of vectors  $\vec{a}$  and  $\vec{b}$ , or  $\mathbf{a}$  and  $\mathbf{b}$  , in Example 2.1.2 is

$$\vec{a} \cdot \vec{b} = \mathbf{a}^T \mathbf{b} = [1 \quad 2 \quad 3] \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = -2$$

From the definition of scalar product,

$$\mathbf{a}^T \mathbf{b} = \vec{a} \cdot \vec{b} = abc \cos (\vec{a}, \vec{b})$$

and, with  $a = (\mathbf{a}^T \mathbf{a})^{1/2} = \sqrt{14}$  and  $b = (\mathbf{b}^T \mathbf{b})^{1/2} = \sqrt{3}$  ,  $\cos (\vec{a}, \vec{b}) = -2 / \sqrt{42}$  . Thus,

$(\vec{a}, \vec{b}) = 1.88$  or  $4.40$  rad. Additional information is required to uniquely determine  $(\vec{a}, \vec{b})$  .

---

A  $3 \times 3$  *skew-symmetric matrix*  $\tilde{\mathbf{a}}$  ; i.e.,  $\tilde{\mathbf{a}} = -\tilde{\mathbf{a}}^T$  , associated with a  $3 \times 1$  algebraic vector  $\mathbf{a} = [a_x, a_y, a_z]^T$  is defined as

$$\tilde{\mathbf{a}} \equiv \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (2.1.22)$$

Note that the overhead symbol  $\tilde{\cdot}$  (pronounced *tilde*) indicates that the components of the associated vector are used to generate a skew-symmetric  $3 \times 3$  matrix. Conversely, any  $3 \times 3$  skew-symmetric matrix  $\mathbf{B}$  of the form

$$\mathbf{B} \equiv \begin{bmatrix} 0 & b_{12} & b_{13} \\ -b_{12} & 0 & b_{23} \\ -b_{13} & -b_{23} & 0 \end{bmatrix}$$

can be written as  $\mathbf{B} = \tilde{\mathbf{b}}$  , where  $\mathbf{b} = [-b_{23} \quad b_{13} \quad -b_{12}]^T$  ; i.e.,

$$\mathbf{B} = \begin{bmatrix} 0 & b_{12} & b_{13} \\ -b_{12} & 0 & b_{23} \\ -b_{13} & -b_{23} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \equiv \tilde{\mathbf{b}} = \begin{pmatrix} \overbrace{-b_{23}} \\ \overbrace{b_{13}} \\ \overbrace{-b_{12}} \end{pmatrix} \quad (2.1.23)$$

The *vector product*  $\vec{c} = \vec{a} \times \vec{b}$ , which is expanded in component form in Eq. (2.1.17), can be written in algebraic vector form as

$$\mathbf{c} = \tilde{\mathbf{a}}\mathbf{b} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix} \quad (2.1.24)$$

This result is the reason the  $\tilde{\cdot}$  operator is introduced. It gives a computationally practical means of evaluating the vector product of two vectors that are represented in algebraic form. The practicality is clear by comparing Eqs. (2.1.13) and (2.1.24).

**Example 2.1.4:** The algebraic representation of the vector product  $\vec{c} = \vec{a} \times \vec{b}$  for vectors in Example 2.1.2 is

$$\mathbf{c} = \tilde{\mathbf{a}}\mathbf{b} = \begin{bmatrix} 0 & -3 & 2 \\ 3 & 0 & -1 \\ -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -5 \\ -2 \\ 3 \end{bmatrix}$$

### 2.1.3 Vector Identities

It is helpful to develop identities involving the  $\tilde{\cdot}$  operation. First, note that

$$\tilde{\mathbf{a}}^T = \begin{bmatrix} 0 & a_z & -a_y \\ -a_z & 0 & a_x \\ a_y & -a_x & 0 \end{bmatrix} = -\tilde{\mathbf{a}} \quad (2.1.25)$$

Also, for a scalar  $c$ ,

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} = (\widetilde{\mathbf{a}}) \quad (2.1.26)$$

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

$$\tilde{\mathbf{a}}\mathbf{b} = -\tilde{\mathbf{b}}\mathbf{a} \quad (2.1.27)$$

which agrees with the vector product relation  $\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}$  of Eq. (2.1.14). Direct calculation shows that

$$\tilde{\mathbf{a}}\mathbf{a} = \mathbf{0} \quad (2.1.28)$$

which agrees with  $\vec{a} \times \vec{a} = 0$ . By Eq. (2.1.25),

$$(\tilde{\mathbf{a}}\mathbf{a})^T = \mathbf{a}^T \tilde{\mathbf{a}}^T = -\mathbf{a}^T \tilde{\mathbf{a}} = \mathbf{0} \quad (2.1.29)$$

Direct calculation shows that

$$\tilde{\mathbf{a}}\tilde{\mathbf{b}} = \mathbf{b}\mathbf{a}^T - \mathbf{a}^T\mathbf{b}\mathbf{I} \quad (2.1.30)$$

where  $\mathbf{I}$  is the  $3 \times 3$  identity matrix. Direct calculation also shows that

$$\widetilde{\mathbf{a}}\mathbf{b} = \mathbf{b}\mathbf{a}^T - \mathbf{a}\mathbf{b}^T \quad (2.1.31)$$

$$\widetilde{\mathbf{a}}\tilde{\mathbf{b}} = \tilde{\mathbf{a}}\tilde{\mathbf{b}} - \tilde{\mathbf{b}}\tilde{\mathbf{a}} \quad (2.1.32)$$

From Eqs. (2.1.31) and (2.1.32),

$$\tilde{\mathbf{a}}\tilde{\mathbf{b}} + \mathbf{a}\mathbf{b}^T = \tilde{\mathbf{b}}\tilde{\mathbf{a}} + \mathbf{b}\mathbf{a}^T \quad (2.1.33)$$

Finally, direct calculation shows that

$$\widetilde{(\mathbf{a} + \mathbf{b})} = \tilde{\mathbf{a}} + \tilde{\mathbf{b}} \quad (2.1.34)$$

Matrix implementation of vector operations permits the systematic organization of calculations, which is essential for computer application.

**Example 2.1.5:** To define the position and orientation of a Cartesian  $x'$  -  $y'$  -  $z'$  reference frame with its origin at point P in an  $x$ - $y$ - $z$  Cartesian reference frame, it is first essential to locate point P with vector  $\mathbf{r}^P$ , as shown in Fig. 2.1.8.

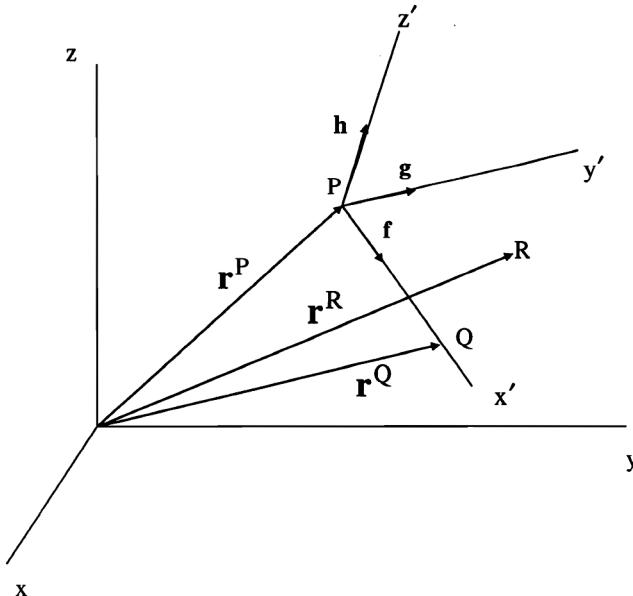


Figure 2.1.8 Points that Define a Cartesian  $x'$ - $y'$ - $z'$  Reference Frame

If a second point Q that is not coincident with P is located on the  $x'$  axis by vector  $\mathbf{r}^Q$ , the unit vector  $\mathbf{f}$  along the  $x'$  axis is

$$\mathbf{f} = \frac{1}{|\mathbf{r}^Q - \mathbf{r}^P|} (\mathbf{r}^Q - \mathbf{r}^P)$$

If a third point R that is not coincident with P and not on the  $x'$  axis is located in the  $x'-y'$  plane by vector  $\mathbf{r}^R$ , then the vector product of  $\mathbf{f}$  and  $\mathbf{r}^R - \mathbf{r}^P \neq \mathbf{0}$  is a vector in the direction of the  $z'$  axis; i.e.,

$$\mathbf{h} = \frac{1}{|\tilde{\mathbf{f}}(\mathbf{r}^R - \mathbf{r}^P)|} \tilde{\mathbf{f}}(\mathbf{r}^R - \mathbf{r}^P)$$

Point R could, but need not, be on the  $y'$  axis. Finally, the unit vector  $\mathbf{g}$  along the  $y'$  axis is

$$\mathbf{g} = \tilde{\mathbf{h}}\mathbf{f}$$

Thus, three distinct noncollinear points P, Q, and R in the  $x'-y'$  plane define unit vectors  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{h}$  along the  $x'$ ,  $y'$ , and  $z'$  axes, hence defining the position and orientation of the  $x'-y'-z'$  Cartesian reference frame

#### 2.1.4 Time Derivatives of Vectors

In analyzing velocities and accelerations, the time derivatives of a vector that locates a point must be calculated. Consider a *time-dependent vector*  $\vec{a}(t)$  with components

$\mathbf{a}(t) = [a_x(t) \ a_y(t) \ a_z(t)]^T$  in a *stationary Cartesian reference frame*; i.e., it does not depend on time and  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$  are constant vectors. The *time derivative of a vector*  $\vec{a}(t)$  is the *velocity* of the point that is located by the vector in a stationary x-y-z frame,

$$\dot{\mathbf{a}} = \frac{d}{dt} (a_x(t) \vec{i} + a_y(t) \vec{j} + a_z(t) \vec{k}) = \frac{d}{dt} a_x(t) \vec{i} + \frac{d}{dt} a_y(t) \vec{j} + \frac{d}{dt} a_z(t) \vec{k} = \dot{a}_x \vec{i} + \dot{a}_y \vec{j} + \dot{a}_z \vec{k}$$

Note that this is only valid if  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$  are not time dependent. In algebraic vector notation, this is

$$\dot{\mathbf{a}} = \frac{d}{dt} \mathbf{a}(t) = [\dot{a}_x \quad \dot{a}_y \quad \dot{a}_z]^T \quad (2.1.35)$$

Thus, for a vector that is represented by its components relative to a stationary Cartesian reference frame, its derivative is obtained by differentiating its components.

**Example 2.1.6:** A particle P moves with constant speed along a circle of radius R in the x-y plane of a stationary Cartesian reference frame, as shown in Fig. 2.1.9. Its position vector is thus

$$\mathbf{r}(t) = \begin{bmatrix} R \cos t \\ R \sin t \\ 0 \end{bmatrix}$$

where  $R$  and  $\omega$  are constant. Its *velocity* is thus

$$\dot{\mathbf{r}}(t) = \begin{bmatrix} -R \sin t \\ R \cos t \\ 0 \end{bmatrix}$$

Its *speed* is the magnitude of its velocity; i.e.,  $|\dot{\mathbf{r}}(t)| = \sqrt{(R\omega)^2(\sin^2 \omega t + \cos^2 \omega t)} = R\omega$ . Since

$$\mathbf{r}^T \dot{\mathbf{r}} = -R^2 \omega \cos t \sin t + R^2 \omega \cos t \sin t = 0$$

the vectors  $\mathbf{r}$  and  $\dot{\mathbf{r}}$  are orthogonal and  $\dot{\mathbf{r}}$  is tangent to the circle

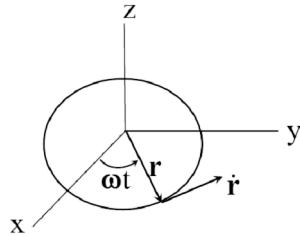


Figure 2.1.9 Particle Moving on Circular Path with Constant Speed

The derivative of the sum of two algebraic vectors is

$$\frac{d}{dt}(\mathbf{a}(t) + \mathbf{b}(t)) = \dot{\mathbf{a}} + \dot{\mathbf{b}} \quad (2.1.36)$$

which is the differentiation rule of calculus that the derivative of a sum is the sum of the derivatives.

The following vector forms of the *product rule of differentiation* are also valid:

$$\frac{d}{dt}(\mathbf{a} \cdot \mathbf{b}) = \dot{\mathbf{a}} \cdot \mathbf{b} + \mathbf{a} \cdot \dot{\mathbf{b}} \quad (2.1.37)$$

$$\frac{d}{dt}(\mathbf{a}^T \mathbf{b}) = \dot{\mathbf{a}}^T \mathbf{b} + \mathbf{a}^T \dot{\mathbf{b}} = \mathbf{b}^T \dot{\mathbf{a}} + \mathbf{a}^T \dot{\mathbf{b}} \quad (2.1.38)$$

$$\frac{d}{dt}(\tilde{\mathbf{a}} \mathbf{b}) = \dot{\tilde{\mathbf{a}}} \mathbf{b} + \tilde{\mathbf{a}} \dot{\mathbf{b}} = \tilde{\mathbf{a}} \dot{\mathbf{b}} - \tilde{\mathbf{b}} \dot{\mathbf{a}} \quad (2.1.39)$$

where  $\tilde{\mathbf{a}}(t)$  is a scalar function of time and  $\mathbf{a}$  and  $\mathbf{b}$  are time dependent vectors. Note also that

$$\tilde{\tilde{\mathbf{a}}} = \ddot{\mathbf{a}} \quad (2.1.40)$$

If the length of a vector  $\mathbf{a}(t)$  is constant; i.e.  $\mathbf{a}(t)^T \mathbf{a}(t) = c$ , where  $c$  is a constant, then Eq. (2.1.38) yields  $\dot{\mathbf{a}}^T \mathbf{a} + \mathbf{a}^T \ddot{\mathbf{a}} = 2\dot{\mathbf{a}}^T \mathbf{a} = 0$ , so

$$\dot{\mathbf{a}}^T \mathbf{a} = 0 \quad (2.1.41)$$

If  $\mathbf{a}$  is a position vector that locates a point, then  $\dot{\mathbf{a}}$  is the velocity of that point. Hence, Eq. (2.1.41) shows that the velocity of a point whose distance from the origin is constant is orthogonal to the position vector of the point. Note that Eq. (2.1.41) is satisfied by the velocity vector of Example 2.1.6, which is to be expected since  $\mathbf{r}^T \mathbf{r} = R^2$ .

The second time derivative of  $\mathbf{a}(t)$  is the *acceleration* of the point that is located in the stationary Cartesian x-y-z frame by the vector  $\mathbf{a}(t)$ ; i.e.,

$$\ddot{\mathbf{a}} = \frac{d}{dt}(\dot{\mathbf{a}}(t)) = \left[ \frac{d^2}{dt^2} a_x(t) \quad \frac{d^2}{dt^2} a_y(t) \quad \frac{d^2}{dt^2} a_z(t) \right]^T \quad (2.1.42)$$

Thus, for vectors that are written in terms of their components in a stationary Cartesian reference frame, acceleration may be calculated in terms of the second time derivatives of the components of the vector.

**Example 2.1.7:** The acceleration of the particle located by the vector  $\mathbf{r}(t)$  in Example 2.1.6 is

$$\ddot{\mathbf{r}}(t) = \begin{bmatrix} -R^2 \cos t \\ -R^2 \sin t \\ 0 \end{bmatrix} = -R^2 \begin{bmatrix} \cos t \\ \sin t \\ 0 \end{bmatrix} = -R^2 \mathbf{r}(t)$$

This is the classical *centripetal acceleration* of a point that moves with constant velocity on a circular path, since the direction of  $\ddot{\mathbf{r}}$  is opposite to the direction of  $\mathbf{r}(t)$ .

Since vector relations derived in this section are used often in kinematic modeling of mechanical systems, the most common relations are summarized as “Key Formulas” at the end of the section. Such summaries are presented at the end of sections in the text in which such relations are derived.

Geometric vectors, with the operations of addition, scalar product, and vector product, have the virtue of being independent of reference frames. Their downside is the difficulty in representing vector operations and differentiation with respect to time in a form that is suitable for digital computation.

Algebraic vectors, in contrast, suffer from being dependent on the Cartesian reference frame in which they are represented. Their saving grace is that they are ideally suited for matrix representation and digital computer implementation. Time derivatives of algebraic vectors are also easily defined and operated on by the rules of calculus and implemented in digital computation.

### **Key Formulas**

$$\tilde{\mathbf{a}} \equiv \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (2.1.22)$$

$$\mathbf{B} = \begin{bmatrix} 0 & b_{12} & b_{13} \\ -b_{12} & 0 & b_{23} \\ -b_{13} & -b_{23} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \equiv \tilde{\mathbf{b}} = \begin{pmatrix} \overbrace{-b_{23}} \\ b_{13} \\ -b_{12} \end{pmatrix} \quad (2.1.23)$$

$$\tilde{\mathbf{a}}^T = -\tilde{\mathbf{a}} \quad \tilde{\mathbf{a}}\tilde{\mathbf{b}} = -\tilde{\mathbf{b}}\tilde{\mathbf{a}} \quad \tilde{\mathbf{a}}\mathbf{0} = \mathbf{0} \quad (2.1.25) \quad (2.1.27) \quad (2.1.28)$$

$$\tilde{\mathbf{a}}\tilde{\mathbf{b}} = \mathbf{b}\mathbf{a}^T - \mathbf{a}^T\mathbf{b}\mathbf{I} \quad \widetilde{\tilde{\mathbf{a}}\tilde{\mathbf{b}}} = \tilde{\mathbf{a}}\tilde{\mathbf{b}} - \tilde{\mathbf{b}}\tilde{\mathbf{a}} \quad (2.1.30) \quad (2.1.32)$$

$$\tilde{\mathbf{a}}\tilde{\mathbf{b}} + \mathbf{a}\mathbf{b}^T = \tilde{\mathbf{b}}\tilde{\mathbf{a}} + \mathbf{b}\mathbf{a}^T \quad (2.1.33)$$

## 2.2 Matrix Algebra and Multivariable Calculus

Matrix algebra and multivariable calculus are the underpinning of computational kinematics and dynamics. Definitions and basic results in matrix and linear algebra are reviewed, in light of their pervasive use in the text, including the *Lagrange multiplier theorem of linear algebra* that is critically important in mechanical system dynamics. Similarly, basic concepts, notation, and results of multivariable calculus and topology that are used throughout the text are presented. Continuity of functions is defined using metric properties of the Euclidean vector space and extended to representation in terms of open sets that provide the foundation for Euclidean space as a topological vector space. Important properties of the space, in which kinematics and dynamics take place are summarized for future use. The *implicit function theorem* that is indispensable in determining when there are solutions of nonlinear equations is presented and applied in constructing such solutions. Finally, the iterative Newton-Raphson method that is the cornerstone for numerical solution of nonlinear equations of kinematics and dynamics is presented.

### 2.2.1 Matrix Algebra

Matrix algebra permits systematic representation of linear equations. Matrix manipulation also allows for organized development, simplification, and solution of systems of linear equations. A *matrix* is defined as a rectangular array of real numbers. If a matrix has  $m$  *rows* and  $n$  *columns*, its *matrix dimension* is  $m \times n$ . A matrix is denoted by a boldface capital letter if  $m$  and  $n$  are greater than one, and is written in the form

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad (2.2.1)$$

where the scalar element  $a_{ij}$  is located in 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  $\cdot^T$ . If  $a_{ij}$  is the  $i$ - $j$  element of matrix  $\mathbf{A}$ ,  $a_{ji}$  is the  $i$ - $j$  element of its transpose, denoted  $\mathbf{A}^T$ .

A matrix with only one column is called a *column matrix*, or *column vector*, as in Section 2.1, and is denoted by a boldface lowercase letter; i.e.,  $\mathbf{a}$ . A matrix with only one row is called a *row matrix*, or *row vector*, and is denoted by a boldface lowercase letter. An  $m \times n$  matrix can be considered as being constructed of  $n$  column matrices  $\mathbf{a}_j = [a_{1j} \ \cdots \ a_{mj}]^T$ ,  $j = 1, \dots, n$ , or  $m$  row matrices  $\mathbf{b}_i = [a_{i1} \ \cdots \ a_{in}]$ ,  $i = 1, \dots, m$ ; i.e.,

$$\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_n] = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_m \end{bmatrix} \quad (2.2.2)$$

---

**Example 2.2.1:** The  $2 \times 3$  matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 3 & 1 \end{bmatrix}$$

may be thought of as made up of three  $2 \times 1$  column matrices,

$$\mathbf{a}_1 = [1 \ 2]^T, \quad \mathbf{a}_2 = [2 \ 3]^T, \quad \mathbf{a}_3 = [0 \ 1]^T$$

or two  $1 \times 3$  row matrices,

$$\mathbf{b}_1 = [1 \ 2 \ 0], \quad \mathbf{b}_2 = [2 \ 3 \ 1]$$


---

A *square matrix* has an equal number of rows and columns. A *diagonal matrix* is a square matrix with  $a_{ij} = 0$  if  $i \neq j$ , and at least one nonzero diagonal element. An  $n \times n$  diagonal matrix  $\mathbf{A}$  can be represented as

$$\mathbf{A} = \text{diag}(a_{11} \ a_{22} \ \cdots \ a_{nn}) \quad (2.2.3)$$

The  $n \times n$  *identity matrix*, denoted  $\mathbf{I}$ , or  $\mathbf{I}_n$ , is the diagonal matrix with unit values on the diagonal. A *zero matrix* of any dimension, designated by a bold  $\mathbf{0}$ , is a matrix with all elements equal to zero.

If matrices  $\mathbf{A}$  and  $\mathbf{B}$  have the same dimension, they are defined to be *equal matrices* if  $a_{ij} = b_{ij}$ , for all  $i$  and  $j$ ; i.e., all elements in the same positions are equal. The *sum of matrices*  $\mathbf{A}$  and  $\mathbf{B}$  that have the same dimension is a matrix with the same dimension, defined as

$$\mathbf{C} = \mathbf{A} + \mathbf{B} \quad (2.2.4)$$

where  $c_{ij} = a_{ij} + b_{ij}$ , for all  $i$  and  $j$ ; i.e., matrices with the same dimension add element by element. The difference between matrices  $\mathbf{A}$  and  $\mathbf{B}$  of the same dimension is defined as

$$\mathbf{D} = \mathbf{A} - \mathbf{B}$$

where  $d_{ij} = a_{ij} - b_{ij}$ , for all  $i$  and  $j$ . If matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  have the same dimension,

$$(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C}) = \mathbf{A} + \mathbf{B} + \mathbf{C} \quad (2.2.5)$$

Similarly, for matrices  $\mathbf{A}$  and  $\mathbf{B}$  with the same dimension,

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A} \quad (2.2.6)$$


---

**Example 2.2.2:** The sum of the  $2 \times 2$  matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix}$$

is

$$\mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix}$$

and the difference is

$$\mathbf{D} = \mathbf{A} - \mathbf{B} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}$$


---

Let  $\mathbf{A}$  be an  $m \times p$  matrix and  $\mathbf{B}$  be a  $p \times n$  matrix, written in the form

$$\mathbf{A} \equiv \begin{bmatrix} a_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_m \end{bmatrix}, \quad \mathbf{B} \equiv \begin{bmatrix} b_{ij} \end{bmatrix} = [\mathbf{b}_1 \quad \cdots \quad \mathbf{b}_n] \quad (2.2.7)$$

The *matrix product* of  $\mathbf{A}$  and  $\mathbf{B}$  is defined as the  $m \times n$  matrix

$$\mathbf{C} = \mathbf{AB} \quad (2.2.8)$$

where

$$c_{ij} = \sum_{k=1}^p a_{ik} b_{kj} \quad (2.2.9)$$

or, in terms of the rows of  $\mathbf{A}$  and columns of  $\mathbf{B}$ ,

$$c_{ij} = \mathbf{d}_i \mathbf{b}_j \quad (2.2.10)$$


---

**Example 2.2.3:** The product of the  $2 \times 2$  matrix  $\mathbf{B}$  of Example 2.2.2 and the  $2 \times 3$  matrix  $\mathbf{A}$  of Example 2.2.1 is the  $2 \times 3$  matrix

$$\mathbf{C} = \mathbf{BA} = \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 6 & 2 \\ 3 & 5 & 1 \end{bmatrix}$$


---

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. Even if matrix multiplications  $\mathbf{AB}$  and  $\mathbf{BA}$  are defined; i.e., the matrices are square and of the same dimension, in general,

$$\mathbf{AB} \neq \mathbf{BA} \quad (2.2.11)$$

To see that this is the case, the matrices of Example 2.2.2 yield

$$\mathbf{AB} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 5 \end{bmatrix} \neq \begin{bmatrix} 4 & 2 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} = \mathbf{BA}$$

If  $\mathbf{A}$  and  $\mathbf{B}$  are  $m \times p$  matrices and  $\mathbf{C}$  is a  $p \times n$  matrix, direct manipulation verifies that

$$(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC} \quad (2.2.12)$$

Similarly, if  $\mathbf{A}$  is an  $m \times p$  matrix,  $\mathbf{B}$  is a  $p \times q$  matrix, and  $\mathbf{C}$  is a  $q \times n$  matrix,

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC}) = \mathbf{ABC} \quad (2.2.13)$$

*Multiplication of a matrix  $\mathbf{A}$  by a scalar* is defined as

$$\mathbf{C} = k\mathbf{A} \quad (2.2.14)$$

where  $c_{ij} = k a_{ij}$ , for all  $i$  and  $j$ ; i.e., all elements in the matrix are multiplied by the scalar.

If  $a_{ij} = a_{ji}$ , for all  $i$  and  $j$ , the square matrix  $\mathbf{A}$  is called a *symmetric matrix*; i.e.,  $\mathbf{A} = \mathbf{A}^T$ .

If  $a_{ij} = -a_{ji}$ , for all  $i$  and  $j$ , the square matrix  $\mathbf{A}$  is called a *skew-symmetric matrix*; i.e.,  $\mathbf{A} = -\mathbf{A}^T$ .

Note that for skew-symmetric matrices,  $a_{ii} = 0$ , for all  $i$ .

The transpose of the sum of two matrices is the sum of their transposes; i.e.,

$$(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T \quad (2.2.15)$$

Also, if  $\mathbf{A}$  is an  $m \times p$  matrix and  $\mathbf{B}$  is a  $p \times n$  matrix, then

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T \quad (2.2.16)$$

A set of  $n \times 1$  column matrices  $\mathbf{a}_j$ ,  $j = 1, \dots, m$ , is called *linearly dependent* if there are constants  $c_j$ ,  $j = 1, \dots, m$ , that are not all zero such that

$$\sum_{j=1}^m c_j \mathbf{a}_j = \mathbf{0} \quad (2.2.17)$$

If a set of column matrices is not linearly dependent, it is called *linearly independent*.

Equivalently, column matrices  $\mathbf{a}_j$ ,  $j = 1, \dots, m$ , are linearly independent if and only if Eq. (2.2.17) implies that  $c_j = 0$ ,  $j = 1, \dots, m$ .

**Example 2.2.4:** To see if the column matrices

$$\mathbf{b}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$$

are linearly independent, form  $\sum_{j=1}^m c_j \mathbf{b}_j = \mathbf{0}$ . This may be written as a matrix equation in the  $c_i$ ,

$$[\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3] \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \mathbf{0}$$

For any  $c_3$ ,  $c_1 = -2c_3$  and  $c_2 = c_3$  satisfy these equations and the  $\mathbf{b}_j$  are linearly dependent.

---

Consider the  $n \times m$  matrix  $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_m]$  with columns  $\mathbf{a}_1, \dots, \mathbf{a}_m$ . If a linear combination of the columns of  $\mathbf{A}$  is zero; i.e.,

$$\mathbf{A}\alpha = \sum_{j=1}^m \alpha_j \mathbf{a}_j = \mathbf{0} \quad (2.2.18)$$

for some  $\alpha = [ \alpha_1 \ \cdots \ \alpha_m ]^T \neq \mathbf{0}$ , then the columns of  $\mathbf{A}$  are linearly dependent. Otherwise, they

are linearly independent. Rows  $\mathbf{d}_1, \dots, \mathbf{d}_n$  of  $\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_n \end{bmatrix}$  are linearly dependent if

$$\beta^T \mathbf{D} = \sum_{i=1}^n \beta_i \mathbf{d}_i = \mathbf{0} \quad (2.2.19)$$

for some  $\beta = [ \beta_1 \ \cdots \ \beta_n ]^T \neq \mathbf{0}$ . Otherwise they are linearly independent.

The *row rank (column rank)* of a matrix is defined as the largest number of linearly independent rows (columns) in the matrix. The row and column ranks of any matrix are equal (Kreyszig, 2011), hence defining the *rank* of the matrix. The rank of a matrix is also equal to the dimension of the largest square submatrix that is obtained by deleting rows and columns with nonzero determinant (Kreyszig, 2011). A square matrix with linearly independent rows or columns is said to have *full rank*. A square matrix has full rank if and only as its determinant is not zero.

When a square matrix does not have full rank; i.e.,  $|\mathbf{A}| = 0$ , it is called a *singular matrix*. For a singular matrix with columns  $\mathbf{a}_i, i = 1 \dots n$ , there are scalars  $u_i, i = 1, \dots, n$ , that are not all 0 such that  $\sum_{i=1}^n u_i \mathbf{a}_i = \mathbf{0}$ . With  $\mathbf{u} = [ u_1 \dots u_n ]^T$ , this is the matrix equation

$$\mathbf{A}\mathbf{u} = \mathbf{0} \quad (2.2.20)$$

This shows that a square matrix  $\mathbf{A}$  is singular if and only if there is a nonzero column vector  $\mathbf{u}$  such that Eq. (2.2.20) holds.

A square matrix  $\mathbf{A}$  of full rank is called a *nonsingular matrix*. For such a matrix, there is an *inverse matrix* (Kreyszig, 2011), denoted  $\mathbf{A}^{-1}$ , such that

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \quad (2.2.21)$$

Using the definition of Eq. (2.2.21) and Eq. (2.2.16), if square matrix  $\mathbf{A}$  is nonsingular,

$$(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} \quad (2.2.22)$$

and if square matrices  $\mathbf{A}$  and  $\mathbf{B}$  of the same dimension are nonsingular,

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \quad (2.2.23)$$

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

$$\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = \mathbf{I} \quad (2.2.24)$$

That is, from the definition of Eq. (2.2.21),

$$\mathbf{A}^{-1} = \mathbf{A}^T \quad (2.2.25)$$

Since constructing the inverse of a nonsingular matrix is time consuming, it is important to know when a matrix is orthogonal. In case a matrix is orthogonal, its inverse is easily constructed using Eq. (2.2.25), which requires no arithmetic operations.

---

**Example 2.2.5:** The  $2 \times 2$  matrix

$$\mathbf{A} = \begin{bmatrix} \cos & -\sin \\ \sin & \cos \end{bmatrix}$$

is orthogonal, for any value of  $\theta$ , since

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} \cos^2 + \sin^2 & 0 \\ 0 & \cos^2 + \sin^2 \end{bmatrix} = \mathbf{I}$$


---

## 2.2.2 Lagrange Multiplier Theorem of Linear Algebra

A result that plays an important role in constrained multibody dynamics is the *Lagrange multiplier theorem* of linear algebra.

**Lagrange Multiplier Theorem:** Let  $\mathbf{c}$  be an  $n$ -vector of constants,  $\mathbf{x}$  be an  $n$ -vector of variables, and  $\mathbf{B}$  be an  $m \times n$  constant matrix. If

$$\mathbf{x}^T \mathbf{c} = 0 \quad (2.2.26)$$

for all  $\mathbf{x}$  that satisfy

$$\mathbf{Bx} = \mathbf{0} \quad (2.2.27)$$

then there exists an  $m$ -vector  $\lambda$  of *Lagrange multipliers* such that

$$\mathbf{c}^T \mathbf{x} + \lambda^T \mathbf{Bx} = 0 \quad (2.2.28)$$

for arbitrary  $\mathbf{x}$ . Thus,  $\mathbf{c}^T + \lambda^T \mathbf{B} = \mathbf{0}$ , or

$$\mathbf{c} + \mathbf{B}^T \lambda = \mathbf{0} \quad (2.2.29)$$

Further, if  $\mathbf{B}$  has full row rank,  $\lambda$  is unique.

Since this theorem is not found in the literature on mechanical system kinematics and dynamics, a short proof is presented here. To prove the theorem, define the *null space* of  $\mathbf{B}$  as  $N(\mathbf{B}) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Bx} = \mathbf{0}\}$  and the *range* of  $\mathbf{B}^T$  as  $R(\mathbf{B}^T) = \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y} = \mathbf{B}^T \alpha, \text{ for some } \alpha \in \mathbb{R}^n\}$ .

The *fundamental theorem of algebra* (Strang, 1980) implies

$R(\mathbf{B}^T) = (N(\mathbf{B}))^\perp \equiv \{\mathbf{y} \in \mathbb{R}^m : \mathbf{x}^T \mathbf{y} = 0 \text{ for all } \mathbf{x} \in N(\mathbf{B})\}$ . The statement that Eq. (2.2.26) holds for

all  $\mathbf{x}$  that satisfy Eq.(2.2.27) is equivalent to  $\mathbf{c} \in N(\mathbf{B})^\perp = R(\mathbf{B}^T)$ . Thus, there exists an  $\alpha \in R^m$  such that  $\mathbf{c} = \mathbf{B}^T \alpha$ . Setting  $\lambda = -\alpha$  yields the result of Eq. (2.2.29). The result of Eq. (2.2.28) follows. To see that  $\lambda$  is unique if  $\mathbf{B}$  has full row rank, assume there is a  $\bar{\lambda} \neq \lambda$  that satisfies Eq. (2.2.29). Then,  $\mathbf{B}^T (\lambda - \bar{\lambda}) = \mathbf{0}$  and, since  $\mathbf{B}^T$  has full column rank,  $\lambda - \bar{\lambda} = \mathbf{0}$  and  $\lambda$  is unique. This completes the proof.

### 2.2.3 Multivariable Calculus

Just as in differentiation of a vector whose components are functions of  $t$ , the *derivative of a matrix* whose elements depend on  $t$  may be defined. Consider a matrix  $\mathbf{B}(t) = [b_{ij}(t)]$ . The derivative of  $\mathbf{B}(t)$  with respect to  $t$  is defined as

$$\dot{\mathbf{B}} = \frac{d}{dt} \mathbf{B} = \left[ \frac{db_{ij}}{dt} \right] = \left[ \dot{b}_{ij} \right] \quad (2.2.30)$$

With this definition, elementary rules of differentiation for matrices of compatible dimension are

$$\frac{d}{dt} (\mathbf{B} + \mathbf{C}) = \dot{\mathbf{B}} + \dot{\mathbf{C}} \quad (2.2.31)$$

$$\frac{d}{dt} (\mathbf{BC}) = \dot{\mathbf{B}}\mathbf{C} + \mathbf{B}\dot{\mathbf{C}} \quad (2.2.32)$$

$$\frac{d}{dt} (\mathbf{B} \cdot \mathbf{B}) = \mathbf{B} \cdot \dot{\mathbf{B}} + \dot{\mathbf{B}} \cdot \mathbf{B} \quad (2.2.33)$$

where  $\cdot = \cdot(t)$  is a scalar function of  $t$ .

In dealing with systems of nonlinear equations in several variables, which govern the kinematics and dynamics of mechanical systems, it is essential that *multivariable calculus* be employed. A basic construct in multivariable calculus is definition of *real Euclidean space*  $R^n \equiv \{ \mathbf{q} = [q_1 \ \cdots \ q_n]^T : q_i \text{ is real, } i = 1, \dots, n \}$ , and the scalar product of  $\mathbf{u}$  and  $\mathbf{v}$  in  $R^n$  is defined as  $\langle \mathbf{u}, \mathbf{v} \rangle \equiv \mathbf{u}^T \mathbf{v} = \sum_{i=1}^n u_i v_i$ . To introduce notation used here, let  $\mathbf{q} \in R^n$ ,  $a(\mathbf{q})$  be a scalar differentiable function of  $\mathbf{q}$ , and  $\mathbf{f}(\mathbf{q}) = [f_1(\mathbf{q}) \ \cdots \ f_m(\mathbf{q})]^T$  be an  $m$ -vector of scalar differentiable functions of  $\mathbf{q}$ . Using  $i$  as row index and  $j$  as column index, the following *matrix calculus notation* is defined:

$$a_q(\mathbf{q}) \equiv \frac{\partial a(\mathbf{q})}{\partial \mathbf{q}} = \left[ \frac{\partial a(\mathbf{q})}{\partial q_j} \right]_{1 \times n} \quad (2.2.34)$$

$$\mathbf{f}_q(\mathbf{q}) \equiv \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} = \left[ \frac{\partial f_i(\mathbf{q})}{\partial q_j} \right]_{m \times n} \quad (2.2.35)$$

Note that the derivative of a scalar function with respect to a vector variable in Eq. (2.2.34) is a row matrix. This is one of the few matrix symbols in the text that is a row matrix,

rather than the more common column matrix. Note also that, as defined by Eq. (2.2.35), the derivative of a column vector function, whose elements are scalar functions of a vector variable, is a matrix. The *subscript notation* used here to denote differentiation is helpful in deriving needed relations, without becoming entangled in cumbersome index and partial differentiation notation. To take advantage of this notation, however, it is critically important that correct matrix definitions of derivatives be used.

The partial derivative of the scalar product of m-vector functions

$\mathbf{g}(\mathbf{q}) = [g_1(\mathbf{q}) \ \cdots \ g_m(\mathbf{q})]^T$  and  $\mathbf{h}(\mathbf{q}) = [h_1(\mathbf{q}) \ \cdots \ h_m(\mathbf{q})]^T$  of an n-vector variable  $\mathbf{q}$ , by careful manipulation, yields the *product rule of differentiation*

$$\begin{aligned} (\mathbf{g}^T \mathbf{h})_{\mathbf{q}} &= \frac{\partial}{\partial \mathbf{q}} (\mathbf{g}^T \mathbf{h}) = \frac{\partial}{\partial \mathbf{q}} \left( \sum_{k=1}^m g_k h_k \right) = \left[ \frac{\partial}{\partial q_j} \left( \sum_{k=1}^m g_k h_k \right) \right] \\ &= \left[ \sum_{k=1}^m \left( \frac{\partial g_k}{\partial q_j} h_k + g_k \frac{\partial h_k}{\partial q_j} \right) \right] = \left[ \left( \sum_{k=1}^m h_k \frac{\partial g_k}{\partial q_j} \right) \right] + \left[ \left( \sum_{k=1}^m g_k \frac{\partial h_k}{\partial q_j} \right) \right] \quad (2.2.36) \\ &= \mathbf{h}^T \mathbf{g}_{\mathbf{q}} + \mathbf{g}^T \mathbf{h}_{\mathbf{q}} \end{aligned}$$

Note that what might have intuitively appeared to be the appropriate product rule of differentiation is not even defined, much less valid; i.e.,  $(\mathbf{g}^T \mathbf{h})_{\mathbf{q}} \neq (\mathbf{g}_{\mathbf{q}})^T \mathbf{h} + \mathbf{g}^T \mathbf{h}_{\mathbf{q}}$ .

If  $\mathbf{f}(\mathbf{g}) = [f_1(\mathbf{g}) \ \cdots \ f_m(\mathbf{g})]^T$  and  $\mathbf{g}(\mathbf{q}) = [g_1(\mathbf{q}) \ \cdots \ g_k(\mathbf{q})]^T$ , where  $\mathbf{q}$  is an n-vector, the *chain rule of differentiation* is

$$\mathbf{f}(\mathbf{g}(\mathbf{q}))_{\mathbf{q}} = \left[ \frac{\partial f_i(\mathbf{g}(\mathbf{q}))}{\partial q_j} \right]_{m \times n} = \left[ \sum_{\ell=1}^k \left( \frac{\partial f_i}{\partial g_{\ell}} \frac{\partial g_{\ell}}{\partial q_j} \right) \right]_{m \times n} = \left[ \frac{\partial f_i}{\partial g_j} \right]_{m \times k} \left[ \frac{\partial g_i}{\partial q_j} \right]_{k \times n} = \mathbf{f}_g \mathbf{g}_{\mathbf{q}} \quad (2.2.37)$$

If  $\mathbf{B}$  is a constant  $m \times n$  matrix and  $\mathbf{p}$  and  $\mathbf{q}$  are m-and n-vectors of variables, respectively, the following useful relations may be verified:

$$\frac{\partial}{\partial \mathbf{q}} (\mathbf{B} \mathbf{q}) = (\mathbf{B} \mathbf{q})_{\mathbf{q}} = \mathbf{B} \quad (2.2.38)$$

$$\frac{\partial}{\partial \mathbf{p}} (\mathbf{p}^T \mathbf{B} \mathbf{q}) = (\mathbf{p}^T \mathbf{B} \mathbf{q})_{\mathbf{p}} = (\mathbf{q}^T \mathbf{B}^T \mathbf{p})_{\mathbf{p}} = \mathbf{q}^T \mathbf{B}^T \quad (2.2.39)$$

$$\frac{d}{dt} (\mathbf{p}^T \mathbf{B} \mathbf{q}) = \mathbf{p}^T \mathbf{B} \dot{\mathbf{q}} + \mathbf{q}^T \mathbf{B}^T \dot{\mathbf{p}} \quad (2.2.40)$$

## 2.2.4 Euclidean Space, Continuity of Functions, and Topologies

To deal with continuity of functions of several variables and their derivatives, the concept of distance between vector variables is required. First, the *n-dimensional Euclidean space* is defined as

$$\mathbb{R}^n = \left\{ \mathbf{q} = [q_1 \ \cdots \ q_n]^T, q_i \text{ real} \right\} \quad (2.2.41)$$

in which kinematics and dynamics of multibody systems take place. With addition of vectors and multiplication of a vector by a real scalar defined by  $\mathbf{q}^1 + \mathbf{q}^2 = [q_1^1 + q_1^2 \quad \dots \quad q_n^1 + q_n^2]^T$  and

$\alpha\mathbf{q} = [\alpha q_1 \quad \dots \quad \alpha q_n]^T$ ,  $R^n$  is a *vector space* of dimension n. Thus, there exist n linearly independent vectors  $\xi^i \in R^n$  that *span*  $R^n$ ; i.e., for every  $\mathbf{q} \in R^n$ , there are unique scalars  $\alpha_i$ ,  $i=1,\dots,n$ , such that  $\mathbf{q} = \sum_{i=1}^n \alpha_i \xi^i$ . With *scalar product*, *norm*, and *distance function* defined as

$$\langle \mathbf{q}^1, \mathbf{q}^2 \rangle \equiv \mathbf{q}^{1T} \mathbf{q}^2 = \sum_1^n q_i^1 q_i^2 \quad (2.2.42)$$

$$\|\mathbf{q}\| \equiv \sqrt{\langle \mathbf{q}, \mathbf{q} \rangle} \quad (2.2.43)$$

$$d(\mathbf{r}, \mathbf{s}) \equiv \|\mathbf{r} - \mathbf{s}\|_n \quad (2.2.44)$$

$R^n$  is a *metric space* (Mendleson, 1962). The norm has the important properties

$$\begin{aligned} \|\mathbf{q}\|_n &\geq 0 \\ \|\mathbf{r} + \mathbf{s}\|_n &\leq \|\mathbf{r}\|_n + \|\mathbf{s}\|_n \\ \|\mathbf{q}\|_n &= 0 \text{ if and only if } \mathbf{q} = \mathbf{0} \end{aligned} \quad (2.2.45)$$

for all  $\mathbf{q}$ ,  $\mathbf{r}$ , and  $\mathbf{s}$  in  $R^n$ . When the dimension n of  $R^n$  is understood, the subscript n of  $\|\mathbf{q}\|_n$  will be suppressed.

A function  $\mathbf{f}(\mathbf{q}) = [f_1(\mathbf{q}) \quad \dots \quad f_m(\mathbf{q})]^T \in R^m$  of  $\mathbf{q} \in R^n$ , denoted  $\mathbf{f}: R^n \rightarrow R^m$  is defined to be *continuous* at  $\mathbf{q}^0 \in R^n$  if, for every  $\varepsilon > 0$ , there exists a  $\delta > 0$  such that  $\|\mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{q}^0)\|_m < \varepsilon$ , for all  $\mathbf{q} \in R^n$  such that  $\|\mathbf{q} - \mathbf{q}^0\|_n < \delta$ . The set of functions  $\mathbf{f}(\mathbf{q})$  that are continuous at all  $\mathbf{q}$  in a subset  $\Omega$  of  $R^n$ , denoted  $\Omega \subset R^n$ , is called  $C^0(\Omega)$ . The set of all functions  $\mathbf{f}(\mathbf{q})$  for which all partial derivatives of order up to and including k of each component  $f_i(\mathbf{q})$  are continuous at all  $\mathbf{q}$  in  $\Omega$  is called  $C^k(\Omega)$ .

A set  $N \subset R^n$  is defined to be an *open set* if for every vector  $\mathbf{q}^0 \in N$  there exists a  $\delta > 0$  such that the *open ball* of radius  $\delta$  centered at  $\mathbf{q}^0$ ,

$$B(\mathbf{q}^0, \delta) \equiv \left\{ \mathbf{q} \in R^n : \|\mathbf{q} - \mathbf{q}^0\| < \delta \right\} \quad (2.2.46)$$

is a subset of N. The *Euclidean space* with *topology*  $\tau$ , denoted  $(R^n, \tau)$ , is a *topological space*, where  $\tau$  is a collection of open sets in  $R^n$  with the following properties (Mendleson, 1962):

(1)  $R^n$  and  $\emptyset$  (the *empty set*) are open

(2) for any finite collection  $N^i$ ,  $i=1,\dots,k$ , of open sets, the intersection  $\cap_{i=1}^k N^i$  is open

(3) for a possibly infinite collection  $N^i$ ,  $i \in I$ , of open sets, the union  $\cup_{i \in I} N^i$  is open

(4) there exists a collection of open sets  $N^j$ ,  $j \in J$ , such that  $R^n = \cup_{j \in J} N^j$

With its vector space and topological space structure,  $R^{n_{\text{gc}}}$  is a *topological vector space*. The combination of these two properties forms the foundation for a configuration manifold that establishes theoretical results and practical methods for formulation and solution of Lagrange multiplier-free ODE of multibody dynamics.

Topological properties of  $R^n$  that are important for kinematics and dynamics of multibody systems are as follows (Mendelson, 1962):

(1)  $R^n$  is *Hausdorff*; i.e., for any vectors  $\mathbf{q}^1 \neq \mathbf{q}^2$  in  $R^n$ , there exist open sets

$N^1$  and  $N^2$  such that  $\mathbf{q}^i \in N^i$ ,  $i = 1, 2$  and  $N^1 \cap N^2 = \emptyset$

(2) sets  $V \subset R^n$  that are *connected*; i.e., there exist no nonempty open subsets

$U^1$  and  $U^2$  such that  $V = U^1 \cup U^2$  and  $U^1 \cap U^2 = \emptyset$ , are *path connected*; i.e., for any  $\mathbf{q}^1$  and  $\mathbf{q}^2$  in  $V$ , there is a continuous curve  $\mathbf{q}(t) = \omega(t)$ ,  $0 \leq t \leq 1$ , in  $V$ , defined by a continuous vector function  $\omega(t)$ , such that  $\mathbf{q}^1 = \omega(0)$  and  $\mathbf{q}^2 = \omega(1)$

(3) a subset  $W$  of  $R^n$  with open sets defined as  $\{\mathbf{q} \in N \cap W : N \text{ open in } R^n\} \in \tau_W$  is a topological space. The topology  $\tau_W$  is called the *induced topology* on  $W$

(4) every topological subspace  $W$  of  $R^n$  can be *partitioned* into maximal, path connected, open subsets  $W_\alpha \subset W$ ,  $\alpha \in I$ , called *components*, such that

(a)  $W_\alpha \cap W_\beta = \emptyset$  if  $\alpha \neq \beta$

(b)  $\cup_{\alpha \in I} W_\alpha = W$

Since  $R^n$  is a vector space, it is in fact a *topological vector space* (Horvath, 1966). For any subset  $X$  of  $R^n$  with open sets in  $X$  defined as  $\bar{\tau} = \{\Omega \cap X : \Omega \subset R^n \text{ open}\}$ ,  $(X, \bar{\tau})$  is a *topological subspace* of  $R^n$  with the *induced topology* (Mendelson, 1962). Since a subset  $X$  of  $R^n$  may not be closed under the operations of addition and multiplication by a scalar, it may not be a vector space. For example, the set  $X = \{\mathbf{q} \in R^2 : q_1^2 + q_2^2 = 1\}$ , the unit circle in  $R^2$ , is not a vector space since for  $\mathbf{q} \in X$ ,  $2\mathbf{q}$  is not in  $X$ .

An equivalent *characterization* of the foregoing  $\varepsilon$ - $\delta$  definition of continuity is that  $\mathbf{f} : R^n \rightarrow R^m$  is continuous at  $\mathbf{q}^0 \in R^n$  if, for every open set  $\Omega \subset R^m$  such that  $\mathbf{f}(\mathbf{q}^0) \in \Omega$ , the *inverse image* of  $\Omega$ ; i.e.,  $\mathbf{f}^{-1}(\Omega) = \{\mathbf{q} \in R^n : \mathbf{f}(\mathbf{q}) \in \Omega\}$ , is open (Mendelson, 1962). This result is needed in defining sets of configurations that share important kinematic and kinetic properties.

## 2.2.5 Implicit Function Theorem

A system of  $n$  equations in an  $n$ -vector  $\mathbf{u}$  and an  $m$ -vector  $\mathbf{v}$  of variables,

$$\mathbf{f}(\mathbf{u}, \mathbf{v}) = \begin{bmatrix} f_1(\mathbf{u}, \mathbf{v}) & \cdots & f_n(\mathbf{u}, \mathbf{v}) \end{bmatrix}^T = \mathbf{0} \quad (2.2.47)$$

can, under certain circumstances, be solved for  $\mathbf{u}$  as a function of  $\mathbf{v}$ . Conditions that assure existence, uniqueness, and smoothness of a solution are given by the *implicit function theorem* (Corwin and Szczerba, 1982).

**Implicit Function Theorem:** Let the functions  $f_i(\mathbf{u}, \mathbf{v})$  in Eq. (2.2.47) have  $k \geq 1$  continuous derivatives with respect to vector variables  $\mathbf{u}$  and  $\mathbf{v}$  in a neighborhood of  $\mathbf{u}^0$  and  $\mathbf{v}^0$  and let Eq. (2.2.47) be satisfied by vectors  $\mathbf{u}^0$  and  $\mathbf{v}^0$ ; i.e.,  $\mathbf{f}(\mathbf{u}^0, \mathbf{v}^0) = \mathbf{0}$ . If the

$n \times n$  sub-Jacobian  $\mathbf{f}_u(\mathbf{u}^0, \mathbf{v}^0) = \left[ \frac{\partial f_i(\mathbf{u}^0, \mathbf{v}^0)}{\partial u_j} \right]$  is nonsingular, then there exists a unique solution  $\mathbf{u}$  of Eq. (2.2.47) that is a  $k$ -times continuously differentiable function of  $\mathbf{v}$ ; i.e.,

$$\mathbf{u} = \mathbf{h}(\mathbf{v}) \quad (2.2.48)$$

such that  $\mathbf{u}^0 = \mathbf{h}(\mathbf{v}^0)$  and

$$\mathbf{f}(\mathbf{h}(\mathbf{v}), \mathbf{v}) = \mathbf{0} \quad (2.2.49)$$

for all  $\mathbf{v}$  in a neighborhood  $V^0$  of  $\mathbf{v}^0$ .

It is important to be precise about what this theorem says and what it does not say. First, while it assures existence of a smooth solution, given by Eq. (2.2.48), it does not prescribe a specific formula for the solution. Second, it is critical that the sub-Jacobian  $\mathbf{f}_u(\mathbf{u}, \mathbf{v})$  be evaluated with vectors  $\mathbf{u}^0$  and  $\mathbf{v}^0$  for which  $\mathbf{f}(\mathbf{u}^0, \mathbf{v}^0) = \mathbf{0}$ . If the sub-Jacobian is nonsingular at  $(\mathbf{u}^0, \mathbf{v}^0)$  but  $\mathbf{f}(\mathbf{u}^0, \mathbf{v}^0) \neq \mathbf{0}$ , then the theorem says nothing; i.e., it does not apply. Failure to verify that vectors  $\mathbf{u}^0$  and  $\mathbf{v}^0$  satisfy Eq. (2.2.47) is a common error in attempting to use the implicit function theorem. Third, the theorem assures that the solution exists and is unique only in a neighborhood  $V^0$  of  $\mathbf{v}^0$ , which may be a tiny set or it may be a set of large extent, and there may or may not be solutions outside  $V^0$ . Finally, the existence of  $k$  continuous derivatives of the solution is important in both the theory and numerical methods that are used in kinematics and dynamics. In particular, iterative numerical solution methods for the equations of kinematics and dynamics require existence of derivatives for use in computation. The implicit function theorem, in many cases, assures that the needed derivatives exist.

**Example 2.2.6** With  $\mathbf{u} \in \mathbb{R}^2$  and  $\mathbf{v} \in \mathbb{R}^1$ , consider the equations

$$\mathbf{f}(\mathbf{u}, \mathbf{v}) = \begin{bmatrix} u_1 - u_2^2 \\ u_2 - u_1 - v \end{bmatrix} = \mathbf{0} \quad (2.2.50)$$

The subJacobian with respect to  $\mathbf{u}$  is  $\mathbf{f}_u(\mathbf{u}, v) = \begin{bmatrix} 1 & -2u_2 \\ -1 & 1 \end{bmatrix}$  which is nonsingular provided  $|\mathbf{f}_u(\mathbf{u}, v)| = 1 - 2u_2 \neq 0$ . With  $\mathbf{u}^0 = \mathbf{0}$  and  $v^0 = 1$ ,  $|\mathbf{f}_u(\mathbf{u}^0, v^0)| = 1 \neq 0$ . Does this assure existence of continuous  $\mathbf{h}(v)$  such that  $\mathbf{u} = \mathbf{h}(v)$  satisfies Eq. (2.2.50)? To check, substitute  $v = 1$  into Eq. (2.2.50), solve for  $u_2 = u_1 + 1$ , and obtain the quadratic equation  $u_1 - (u_1 + 1)^2 = -u_1^2 - u_1 - 1 = 0$ , which has solutions  $u_1 = (-1 \pm \sqrt{1-4})/2$ , which are not real. Thus, Eq. (2.2.50) has no real valued solution for  $\mathbf{u}$  as a function of  $v$  in a neighborhood of  $v^0 = 1$ , appearing to contradict the implicit function theorem. What is wrong?

The resolution of the foregoing dilemma is provided by Fig. 2.2.1. Solutions of Eq. (2.2.50) are intersections of the straight line and parabola. The dashed straight line with  $v = 1$ , does not intersect the parabola, so there is no solution. The error in the foregoing analysis is misapplication of the implicit function theorem. Since  $\mathbf{f}(\mathbf{0}, 1) = [0 \ -1]^T \neq \mathbf{0}$ , the theorem says nothing about solutions near  $(\mathbf{u}^0, v^0) = (\mathbf{0}, 1)$ .

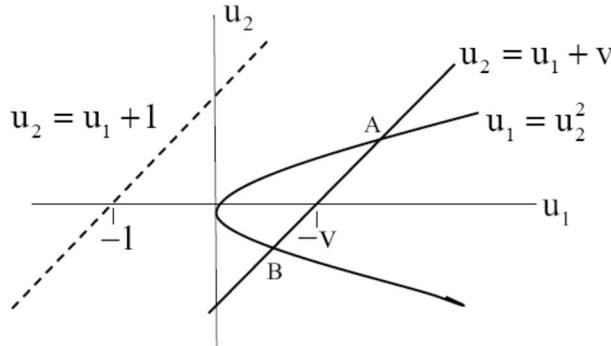


Figure 2.2.1 Intersections of Straight Line and Parabola

If  $v^0 = -1$ ,  $u_2 = u_1 - 1$  is the solid line in Fig. 2.2.1, so  $u_1 - (u_1 - 1)^2 = -u_1^2 + 3u_1 - 1 = 0$ , with solutions  $u_1 = 3/2 \pm \sqrt{5/4}$ . Selecting  $\mathbf{u}^0 = [3/2 + \sqrt{5/4} \ 1/2 + \sqrt{5/4}]^T$ , which is point A in Fig. 2.2.1,  $\mathbf{f}(\mathbf{u}^0, v^0) = \mathbf{0}$  and the implicit function theorem applies, yielding the solution  $\mathbf{u} = \mathbf{h}(v) = (1/2)[(1-2v) + \sqrt{1-4v} \ (-1-2v) + \sqrt{1-4v}]^T$  in a neighborhood of  $v^0 = -1$ . Note that this is not a global solution of Eq. (2.2.50) in a neighborhood of  $v^0 = -1$ . If  $\mathbf{u}^0 = [3/2 - \sqrt{5/4} \ 1/2 - \sqrt{5/4}]^T$ , which is point B in Fig. 2.2.1, implicit function theorem criteria are satisfied and  $\mathbf{u} = \mathbf{h}(v) = (1/2)[(1-2v) - \sqrt{1-4v} \ (-1-2v) - \sqrt{1-4v}]^T$  is a local solution. The moral is; be very careful when applying the implicit function theorem.

## 2.2.6 Newton-Raphson Method for Solution of Nonlinear Equations

At the intersection of multivariable calculus and matrix theory are iterative methods for solving nonlinear algebraic equations. Let  $\mathbf{x}$  be an n-vector of unknowns and

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \in \mathbb{R}^n \quad (2.2.51)$$

be a system of  $n$  nonlinear equations to be solved for  $\mathbf{x}$ , where the vector function has continuous derivatives and the *Jacobian matrix*  $\mathbf{F}_x(\mathbf{x})$  is nonsingular in an open set  $D$  of  $\mathbb{R}^n$ . With an estimate  $\mathbf{x}^0 \in D$  of the solution, a first order Taylor expansion that seeks a perturbation  $\Delta\mathbf{x}^0$  of  $\mathbf{x}^0$  that better approximates the solution as  $\mathbf{x}^1 = \mathbf{x}^0 + \Delta\mathbf{x}^0$  is

$$\mathbf{F}(\mathbf{x}^0) + \mathbf{F}_x(\mathbf{x}^0)\Delta\mathbf{x}^0 = \mathbf{0}$$

After  $j$  iterations of this perturbation computation, the incremental equation is

$$\begin{aligned} \mathbf{F}_x(\mathbf{x}^j)\Delta\mathbf{x}^j &= -\mathbf{F}(\mathbf{x}^j) \\ \mathbf{x}^{j+1} &= \mathbf{x}^j + \Delta\mathbf{x}^j \end{aligned} \quad (2.2.52)$$

which is continued until  $\|\mathbf{F}(\mathbf{x}^{j+1})\| \leq \text{Tol}$ , where  $\text{Tol}$  is a *solution error tolerance*.

This is the *Newton-Raphson method* (Atkinson, 1989) for solution of Eq. (2.2.51). The method has the attractive property that if  $\mathbf{x}^0$  is sufficiently close to the solution, the method *converges quadratically* to the solution; i.e.,

$$\|\bar{\mathbf{x}} - \mathbf{x}^{j+1}\| \leq k \|\bar{\mathbf{x}} - \mathbf{x}^j\|^2 \quad (2.2.53)$$

where  $\bar{\mathbf{x}}$  is the solution and  $k$  is a constant. However, a good *initial estimate* is often difficult to obtain, and the method may diverge for a poor initial estimate. Fortunately, in kinematic and dynamic simulation applications on a time grid  $t_{i+1} = t_i + h$ ,  $i=1,2, \dots$ , with  $h$  small, the solution at  $t_i$  can be used as the initial estimate at  $t_{i+1}$  and convergence is likely.

MATLAB Code 2.2 that implements the Newton-Raphson method is presented in Section 2.B.1 of Appendix 2.B.

## 2.2.7 Matrix Condition Number

The Jacobian matrix  $\mathbf{F}_x(\mathbf{x})$  being nonsingular is not adequate to assure good performance of the Newton-Raphson method. The *condition number* of the Jacobian plays a crucial role (Strang, 1980). With the norm of a vector defined by Eq. (2.2.43), the *matrix norm* of the Jacobian is

$$\|\mathbf{F}_x(\mathbf{x})\|_M = \max_{\mathbf{y} \neq 0} \left( \|\mathbf{F}_x(\mathbf{x})\mathbf{y}\|_M / \|\mathbf{y}\| \right) \quad (2.2.54)$$

and the *condition number* of the Jacobian is defined as (Strang, 1980)

$$c = \|\mathbf{F}_x(\mathbf{x})\|_M \|\mathbf{F}_x(\mathbf{x})^{-1}\|_M \quad (2.2.55)$$

Fortunately, there are methods to obtain good approximations of the condition number in MATLAB, without calculation of the inverse (Atkinson, 1989).

To see the significance of the condition number, consider the case in which the error in evaluating the right side of the matrix equation  $\mathbf{A}\mathbf{x} = \mathbf{b}$  is  $\delta\mathbf{b}$ . A bound on the magnitude of the perturbation  $\mathbf{x}$  in the resulting solution is (Strang, 1980)

$$\| \mathbf{x} \| / \| \mathbf{x} \| \leq c (\| \mathbf{b} \| / \| \mathbf{A} \|_M) \quad (2.2.56)$$

If the condition number  $c$  of matrix  $\mathbf{A}$  is small, the solution is not significantly affected by a perturbation in the right side. If  $c$  is large, however, large error and possibly divergence can occur. Similarly, if the matrix is perturbed by  $\mathbf{A}$ , a bound on solution error is (Strang, 1980)

$$\| \mathbf{x} \| / \| \mathbf{x} + \mathbf{e} \| \leq c (\| \mathbf{A} \|_M / \| \mathbf{A} \|_M) \quad (2.2.57)$$

As above, if the condition number  $c$  is small, the solution is not significantly affected by a perturbation in the Jacobian. If  $c$  is large, however, large error and possibly divergence can occur.

The condition number of a matrix  $\mathbf{A}$  in an equation  $\mathbf{Ax} = \mathbf{b}$  thus serves as a check on accuracy of solution. If  $c = \| \mathbf{A} \|_M \| \mathbf{A}^{-1} \|_M$  is large, problems occur. Unfortunately, if  $\mathbf{A}$  is a scalar,  $\| \mathbf{A} \|_M = |A|$  and  $\| \mathbf{A}^{-1} \|_M = 1/|A|$ . Thus,  $c = |A|/|A| = 1$  and no information is obtained. In this case,  $c \equiv 1/|A|$  serves as a check on accuracy of the solution. If  $1/|A|$  is large, problems arise. This serves as an extension of condition number when  $\mathbf{A}$  is a scalar.

As may be noted in MATLAB code presented in the text for iterative solution of nonlinear equations, the condition number is used as a warning of impending divergence of the Newton-Raphson algorithm.

Matrix algebra and multivariable calculus provide indispensable tools for mechanical system kinematics and dynamics. Both are ideally suited for digital computation. While the Lagrange multiplier theorem of linear algebra has a somewhat abstract proof, it is easily applied as a basic tool of multibody dynamics.

The Euclidean space  $\mathbb{R}^n$ , with the definition of scalar product, norm, and distance between vectors, is indispensable in definition of continuity of functions and is the basis for differential geometry, the foundation of kinematics and dynamics of mechanical systems.

The implicit function theorem is key in assuring existence of solutions of nonlinear equations. Finally, the Newton-Raphson iterative solution algorithm and the matrix condition number play pivotal roles in numerical solution of equations of kinematics and dynamics.

### **Key Formulas**

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T \quad (\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} \quad (\mathbf{AB})^{-1} = \mathbf{B}^{-1} \mathbf{A}^{-1} \quad (2.2.16) \quad (2.2.22) \quad (2.2.23)$$

$$\mathbf{a}_q \equiv \frac{\partial \mathbf{a}}{\partial \mathbf{q}} = \left[ \frac{\partial \mathbf{a}}{\partial q_j} \right]_{1 \times k} \quad \mathbf{f}_q \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \left[ \frac{\partial f_i}{\partial q_j} \right]_{n \times k} \quad (2.2.34) \quad (2.2.35)$$

$$\left(\mathbf{g}^T \mathbf{h}\right)_q = \mathbf{h}^T \mathbf{g}_q + \mathbf{g}^T \mathbf{h}_q \quad \left(\Phi(\mathbf{g}(\mathbf{q}))\right)_q = \Phi_g \mathbf{g}_q \quad (2.2.36) \quad (2.2.37)$$

$$R^n = \left\{ \mathbf{q} = [q_1 \quad \cdots \quad q_n]^T : q_i \text{ real}, i = 1, \dots, n \right\} \quad (2.2.41)$$

$$\langle \mathbf{r}, \mathbf{s} \rangle \equiv \mathbf{r}^T \mathbf{s} = \sum_{i=1}^n r_i s_i \quad \|\mathbf{q}\|_n \equiv \sqrt{\langle \mathbf{q}, \mathbf{q} \rangle} \quad d(\mathbf{r}, \mathbf{s}) \equiv \|\mathbf{r} - \mathbf{s}\|_n \quad (2.2.42) \quad (2.2.43) \quad (2.2.44)$$

$$B(\mathbf{q}^0, r) \equiv \left\{ \mathbf{q} \in R^n : \|\mathbf{q} - \mathbf{q}^0\| < r \right\} \quad (2.2.46)$$

## 2.3 Position and Orientation of a Body in a Plane

A *planar rigid body* has constant thickness normal to the plane in which it lies and mass that is distributed in the plane. It is located and oriented in the plane by an orthogonal  $x'$ - $y'$  reference frame that is attached to the body. Care is taken to show that the body has just one orientation degree of freedom and that it is the angle of rotation from the  $x$ -axis to the  $x'$ -axis.

### 2.3.1 Reference Frames for Location and Orientation in a Plane

Consider a point P that is fixed in a right hand orthogonal  $x'$ - $y'$  *body fixed reference frame* that translates and rotates relative to a right hand orthogonal  $x$ - $y$  *global reference frame*, as shown in Fig. 2.3.1. Using geometric vector notation, with the vector  $\vec{s}^P$  fixed in the  $x'$ - $y'$  frame locating point P in that frame, point P may be located relative to the  $x$ - $y$  frame by the vector equation

$$\vec{r}^P = \vec{r} + \vec{s}^P \quad (2.3.1)$$

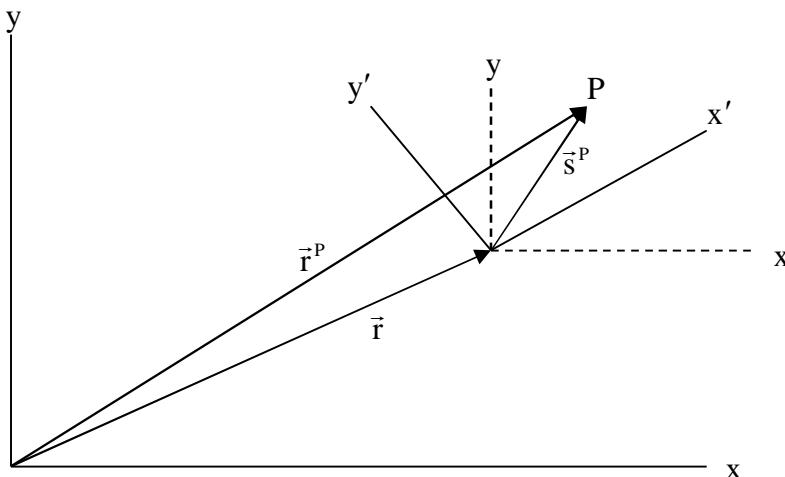


Figure 2.3.1 Vector Location of Point in Moving Reference Frame

In order to quantify the effect of rotation of a reference frame in the plane, consider the  $x'$ - $y'$  and  $x$ - $y$  orthogonal frames with common origin shown in Fig. 2.3.2. The vector  $\vec{s}$  shown can be represented using its components relative to unit vectors in each of the reference frames,

$$\vec{s} = s_x \vec{i} + s_y \vec{j} \quad (2.3.2)$$

where

$$s_x = \vec{s} \cdot \vec{i}, \quad s_y = \vec{s} \cdot \vec{j} \quad (2.3.3)$$

and

$$\vec{s} = s_{x'} \vec{f} + s_{y'} \vec{g} \quad (2.3.4)$$

where

$$s_{x'} = \vec{s} \cdot \vec{f}, \quad s_{y'} = \vec{s} \cdot \vec{g} \quad (2.3.5)$$

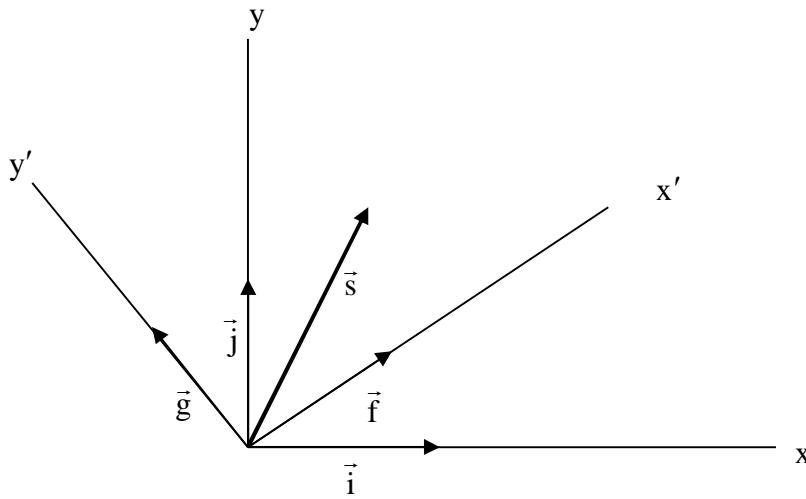


Figure 2.3.2 Orthogonal Reference Frames with Common Origin

Algebraic vectors that define  $\vec{s}$  in the two frames are

$$\mathbf{s} = \begin{bmatrix} s_x & s_y \end{bmatrix}^T \quad (2.3.6)$$

in the  $x$ - $y$  frame and

$$\mathbf{s}' = \begin{bmatrix} s_{x'} & s_{y'} \end{bmatrix}^T \quad (2.3.7)$$

in the  $x'$ - $y'$  frame.

It is clear that there is a relation between  $\mathbf{s}$  and  $\mathbf{s}'$ , since they are defined by the same geometric vector  $\vec{s}$ . To establish this relation, the  $\vec{f}$  and  $\vec{g}$  unit vectors in the  $x'$ - $y'$  frame may be written in terms of the  $\vec{i}$  and  $\vec{j}$  unit vectors in the  $x$ - $y$  frame as

$$\begin{aligned} \vec{f} &= a_{11}\vec{i} + a_{21}\vec{j} \\ \vec{g} &= a_{12}\vec{i} + a_{22}\vec{j} \end{aligned} \quad (2.3.8)$$

where  $a_{ij}$  are the *direction cosines*

$$\begin{aligned} a_{11} &= \vec{i} \cdot \vec{f} = \cos(\vec{i}, \vec{f}), & a_{12} &= \vec{i} \cdot \vec{g} = \cos(\vec{i}, \vec{g}) \\ a_{21} &= \vec{j} \cdot \vec{f} = \cos(\vec{j}, \vec{f}), & a_{22} &= \vec{j} \cdot \vec{g} = \cos(\vec{j}, \vec{g}) \end{aligned} \quad (2.3.9)$$

Substituting from Eq. (2.3.8) into Eq. (2.3.4) yields

$$\vec{s} = (a_{11}s_{x'} + a_{12}s_{y'})\vec{i} + (a_{21}s_{x'} + a_{22}s_{y'})\vec{j} \quad (2.3.10)$$

Equating coefficients of  $\vec{i}$  and  $\vec{j}$  in this representation of  $\vec{s}$  with those in Eq. (2.3.2),

$$\begin{aligned} s_x &= a_{11}s_{x'} + a_{12}s_{y'} \\ s_y &= a_{21}s_{x'} + a_{22}s_{y'} \end{aligned} \quad (2.3.11)$$

In matrix form, this is

$$\mathbf{s} = \mathbf{A}\mathbf{s}' \quad (2.3.12)$$

where  $\mathbf{A}$  is called the *direction cosine matrix* or *orientation transformation matrix*,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (2.3.13)$$

The orientation transformation matrix  $\mathbf{A}$  has very special properties. If the x-y component forms of unit vectors  $\vec{f}$  and  $\vec{g}$  are denoted by  $\mathbf{f}$  and  $\mathbf{g}$  and the x-y components of unit vectors  $\vec{i}$  and  $\vec{j}$  are denoted by  $\mathbf{i}$  and  $\mathbf{j}$ , then

$$\mathbf{i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{j} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.3.14)$$

and Eq. (2.3.8) shows that

$$\mathbf{f} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} \quad (2.3.15)$$

Therefore, the matrix  $\mathbf{A}$  of Eq. (2.3.13) can be written as

$$\mathbf{A} = [\mathbf{f} \quad \mathbf{g}] \quad (2.3.16)$$

Expanding the matrix product,

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} \mathbf{f}^T \\ \mathbf{g}^T \end{bmatrix} [\mathbf{f} \quad \mathbf{g}] = \begin{bmatrix} \mathbf{f}^T \mathbf{f} & \mathbf{f}^T \mathbf{g} \\ \mathbf{g}^T \mathbf{f} & \mathbf{g}^T \mathbf{g} \end{bmatrix} \quad (2.3.17)$$

Since the unit vectors  $\mathbf{f}$  and  $\mathbf{g}$  are orthogonal unit vectors, this is

$$\mathbf{A}^T \mathbf{A} = \mathbf{I} \quad (2.3.18)$$

Thus,  $\mathbf{A}^T = \mathbf{A}^{-1}$ , and the orientation transformation matrix  $\mathbf{A}$  is an *orthogonal matrix*. This special property permits an easy solution of Eq. (2.3.12),

$$\mathbf{s}' = \mathbf{A}^T \mathbf{s} \quad (2.3.19)$$

Transforming between algebraic vectors that represent the same physical vector in the x-y and x'-y' Cartesian reference frames is thus a trivial matter, given by Eqs. (2.3.12) and (2.3.19).

The vector that locates a fixed point P in the x'-y' frame in Fig. 2.3.1, written in terms of geometric vectors in Eq. (2.3.1), can now be written in algebraic form as

$$\mathbf{r}^P = \mathbf{r} + \mathbf{s}^P = \mathbf{r} + \mathbf{A}\mathbf{s}'^P \quad (2.3.20)$$

### 2.3.2 Orientation Degrees of Freedom

The four direction cosines in matrix  $\mathbf{A}$  define the orientation of the x'-y' frame relative to the x-y reference frame, but they are not independent. In component form, Eq. (2.3.18) is

$$\sum_{k=1}^2 a_{ki} a_{kj} = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases}, \quad i, j = 1, 2 \quad (2.3.21)$$

Since interchanging  $i$  and  $j$  yields the same equation, Eq. (2.3.18) provides just three equations among the four direction cosines.

To show that there is one degree of freedom of rotation in the plane, it must be shown that Eq. (2.3.18), equivalently Eq. (2.3.21), imposes three independent constraints on the four components of  $\mathbf{A}$ . First note that, by Eq. (2.3.16), the four components of  $\mathbf{A}$  are the components of vectors  $\mathbf{f}$  and  $\mathbf{g}$ . Define the four variables comprising components of  $\mathbf{A}$  as

$$\mathbf{q} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$$

The constraints imposed on these variables by the diagonal and upper triangular parts of Eq. (2.3.18), using the expansion of Eq. (2.3.17), may be written in the form

$$\Phi(\mathbf{q}) = \begin{bmatrix} \frac{1}{2}(\mathbf{f}^T \mathbf{f} - 1) \\ \frac{1}{2}(\mathbf{g}^T \mathbf{g} - 1) \\ \mathbf{f}^T \mathbf{g} \end{bmatrix} = \mathbf{0} \quad (2.3.22)$$

Let  $\mathbf{q}^0 = [\mathbf{f}^{0T} \quad \mathbf{g}^{0T}]^T$  satisfy Eq. (2.3.22), so  $\mathbf{A}^0 = [\mathbf{f}^0 \quad \mathbf{g}^0]$  is orthogonal. Then, the Jacobian of Eq. (2.3.22), evaluated at  $\mathbf{q}^0$ , is

$$\Phi_q(\mathbf{q}^0) = \begin{bmatrix} \mathbf{f}^{0T} & \mathbf{0} \\ \mathbf{0} & \mathbf{g}^{0T} \\ \mathbf{g}^{0T} & \mathbf{f}^{0T} \end{bmatrix} \quad (2.3.23)$$

The three rows of this Jacobian are linearly independent; i.e., it has full row rank, if the three columns of its transpose are linearly independent; i.e., if the equation

$$\Phi_q^T(\mathbf{q}^0) \mathbf{u} = \begin{bmatrix} \mathbf{f}^0 & \mathbf{0} & \mathbf{g}^0 \\ \mathbf{0} & \mathbf{g}^0 & \mathbf{f}^0 \end{bmatrix} \mathbf{u} = \mathbf{0} \quad (2.3.24)$$

implies  $\mathbf{u} = [u_1 \quad u_2 \quad u_3]^T = \mathbf{0}$ . Expanding the product of Eq. (2.3.24),

$$\begin{aligned} \mathbf{f}^0 u_1 + \mathbf{g}^0 u_3 &= [\mathbf{f}^0 \quad \mathbf{g}^0] \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} = \mathbf{A}^0 \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} = \mathbf{0} \\ \mathbf{g}^0 u_2 + \mathbf{f}^0 u_3 &= [\mathbf{f}^0 \quad \mathbf{g}^0] \begin{bmatrix} u_3 \\ u_2 \end{bmatrix} = \mathbf{A}^0 \begin{bmatrix} u_3 \\ u_2 \end{bmatrix} = \mathbf{0} \end{aligned} \quad (2.3.25)$$

Since  $\mathbf{A}^0$  is orthogonal, hence nonsingular, all components of  $\mathbf{u}$  are zero and the Jacobian of Eq. (2.3.23) has full row rank. The implicit function theorem thus shows that Eq. (2.3.22) can be solved for three of its four variables as a function of the remaining independent variable. Thus, there is *one rotation degree of freedom in the plane*.

Define the rotation of the  $x'$ - $y'$  frame relative to the  $x$ - $y$  frame by angle  $\phi$  shown in Fig. 2.3.3, with counterclockwise taken as positive. Recalling the relations of Eq. (2.3.9) and using the geometry of Fig. 2.3.3,

$$\begin{aligned} a_{11} &= \vec{i} \cdot \vec{f} = \cos(\vec{i}, \vec{f}) = \cos\phi \\ a_{12} &= \vec{i} \cdot \vec{g} = \cos(\vec{i}, \vec{g}) = \cos(\phi + \pi/2) = \cos\pi/2 \cos\phi - \sin\pi/2 \sin\phi = -\sin\phi \\ a_{21} &= \vec{j} \cdot \vec{f} = \cos(\vec{j}, \vec{f}) = \cos(\pi/2 - \phi) = \cos\pi/2 \cos\phi + \sin\pi/2 \sin\phi = \sin\phi \\ a_{22} &= \vec{j} \cdot \vec{g} = \cos(\vec{j}, \vec{g}) = \cos\phi \end{aligned} \quad (2.3.26)$$

This shows that the orientation matrix is

$$\mathbf{A}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \quad (2.3.27)$$

and the angle  $\phi$  shown in Fig. 2.3.3 is an independent generalized coordinate that defines orientation of the  $x'$ - $y'$  reference frame in the plane. This is consistent with the foregoing analysis that showed there is one rotation degree of freedom.

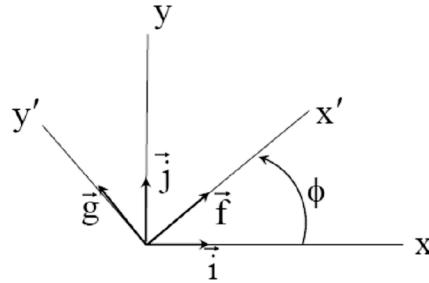


Figure 2.3.3 Relative Rotation of Planar Reference Frames

A trivial calculation shows that  $\mathbf{A}(\phi)$  of Eq. (2.3.27) is orthogonal, for all  $\phi$ ; i.e.,

$$\begin{aligned} \mathbf{A}^T(\phi)\mathbf{A}(\phi) &= \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \\ &= \begin{bmatrix} \cos^2\phi + \sin^2\phi & -\cos\phi\sin\phi + \cos\phi\sin\phi \\ -\cos\phi\sin\phi + \cos\phi\sin\phi & \cos^2\phi + \sin^2\phi \end{bmatrix} = \mathbf{I}_3 \end{aligned}$$

### 2.3.3 Orientation Identities

Since  $\mathbf{A}(\phi)$  transforms a vector that is fixed in the  $x'$ - $y'$  frame to a vector in the  $x$ - $y$  frame, as a result of rotation of the  $x'$ - $y'$  frame, it is called an *orientation transformation matrix*. It has a number of special properties that are helpful in kinematic modeling of planar systems. First, a vector  $\mathbf{v}$  may be rotated counterclockwise by  $\pi/2$  to create a vector  $\mathbf{v}^\perp$  that is perpendicular to  $\mathbf{v}$ ,

$$\mathbf{v}^\perp = \mathbf{A}(\pi/2)\mathbf{v} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{v} \equiv \mathbf{P}\mathbf{v} \quad (2.3.28)$$

That is,

$$\mathbf{P} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (2.3.29)$$

Since  $\mathbf{P}$  is an orientation transformation with  $\phi = \pi/2$ , it is orthogonal; i.e.,

$$\mathbf{P}^T \mathbf{P} = \mathbf{I} \quad (2.3.30)$$

Direct manipulation shows that  $\mathbf{A}$  and  $\mathbf{P}$  have the following properties:

$$\begin{aligned} \frac{d}{d\phi} \mathbf{A}(\phi) &= \mathbf{P} \mathbf{A}(\phi) \\ \frac{d}{d\phi} \mathbf{A}^T(\phi) &= -\mathbf{A}^T(\phi) \mathbf{P} \end{aligned} \quad (2.3.31)$$

$$\mathbf{P} \mathbf{P} = -\mathbf{I} \quad (2.3.32)$$

$$\begin{aligned} \mathbf{P} \mathbf{A} &= \mathbf{A} \mathbf{P} \\ \mathbf{A}^T \mathbf{P} \mathbf{A} &= \mathbf{P} \end{aligned} \quad (2.3.33)$$

$$\mathbf{P} \mathbf{A} \mathbf{P} = -\mathbf{A} \quad (2.3.34)$$

$$\mathbf{P}^T \mathbf{A} \mathbf{P} = \mathbf{A} \quad (2.3.35)$$

$$\mathbf{P}^T \mathbf{A}^T \mathbf{P}^T = -\mathbf{A}^T \quad (2.3.36)$$

It is important to note that once an orientation transformation matrix  $\mathbf{A}$  has been determined to define the orientation of a planar body, the angle in Fig. 2.3.3 can be determined. In particular, let

$$\begin{aligned} \cos \phi &= a_{11} \\ \sin \phi &= a_{21} \end{aligned} \quad (2.3.37)$$

and  $\mathbf{A}(\phi)$  satisfy Eq. (2.3.18). Expanding  $\mathbf{A}^T \mathbf{A} = \mathbf{I}$  yields  $a_{11}^2 + a_{21}^2 = 1$  in the (1,1) component, which is required if Eq. (2.3.37) indeed defines trigonometric functions. Once  $\sin \phi$  and  $\cos \phi$  are known, the value of  $\phi$ ,  $0 \leq \phi \leq 2\pi$ , may be uniquely determined by taking the Arcsin of both sides of the second of Eqs. (2.3.37) and using the algebraic sign of  $\cos \phi$  from the first of Eqs. (2.3.37) to uniquely evaluate  $\phi$ . With

$$-\frac{\pi}{2} \leq \text{Arcsin } a_{21} \leq \frac{\pi}{2} \quad (2.3.38)$$

a trigonometric analysis that is detailed in Section 2.4.4 shows that  $\phi$  is given by

$$\phi = \begin{cases} -\pi - \text{Arcsin } a_{21}, & \text{if } a_{21} \leq 0 \text{ and } a_{11} \leq 0 \\ \text{Arcsin } a_{21}, & \text{if } a_{11} \geq 0 \\ \pi - \text{Arcsin } a_{21}, & \text{if } a_{21} \geq 0 \text{ and } a_{11} \leq 0 \end{cases} \quad (2.3.39)$$

When the origins of the x-y and x'-y' frames do not coincide, as shown in Fig. 2.3.1, point P is located by

$$\mathbf{r}^P = \mathbf{r} + \mathbf{A}(\phi) \mathbf{s}'^P \quad (2.3.40)$$

where  $\mathbf{s}'^P$  is fixed in the x'-y' frame. Thus,  $\mathbf{r}$  and  $\phi$  define the location of all points in a planar rigid body, so they may be regarded as *generalized coordinates* of the body. The generalized coordinates of a planar rigid body are the three components of the vector of *planar Cartesian generalized coordinates*,

$$\mathbf{q} = \begin{bmatrix} \mathbf{r} \\ \phi \end{bmatrix} \quad (2.3.41)$$

### 2.3.4 Time Derivatives and Variations in Rotation and Displacement

In applications, an x'-y' Cartesian reference frame is fixed in a moving body to define its position and orientation over time, relative to a stationary global x-y reference frame. Consider a point P that is fixed in an x'-y' frame, as shown in Fig. 2.3.1. The vector that locates P in the x-y frame is given by Eq. (2.3.40), where  $\mathbf{s}'^P$  is the constant vector of coordinates of P in the x'-y' frame. Since the x'-y' frame is moving and changing its orientation with time, the vector  $\mathbf{r} = \mathbf{r}(t)$  and the angle  $\phi = \phi(t)$ , so the transformation matrix  $\mathbf{A} = \mathbf{A}(\phi(t))$ , are functions of time.

Differentiating both sides of Eq. (2.3.40) with respect to time and using Eq. (2.3.31) yields the velocity of point P as

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\mathbf{A}}(\phi) \mathbf{s}'^P = \dot{\mathbf{r}} + \dot{\phi} \mathbf{PA}(\phi) \mathbf{s}'^P \quad (2.3.42)$$

Taking the time derivative of Eq. (2.3.42) and using Eqs. (2.3.31) and (2.3.32) yields the acceleration of point P as

$$\begin{aligned} \ddot{\mathbf{r}}^P &= \ddot{\mathbf{r}} + \ddot{\phi} \mathbf{PA}(\phi) \mathbf{s}'^P + \dot{\phi}^2 \mathbf{PA}(\phi) \mathbf{s}'^P \\ &= \ddot{\mathbf{r}} + \ddot{\phi} \mathbf{PA}(\phi) \mathbf{s}'^P - \dot{\phi}^2 \mathbf{A}(\phi) \mathbf{s}'^P \end{aligned} \quad (2.3.43)$$

The form of Eqs. (2.3.42) and (2.3.43) is characteristic of that to be encountered in spatial kinematics. Velocity equations are linear in first derivatives of generalized coordinates, with coefficients that depend on generalized coordinates. Acceleration equations are linear in second derivatives, with coefficients that depend on generalized coordinates, but quadratic in first derivatives of generalized coordinates. The quadratic velocity terms in the acceleration equations are reminiscent of *Coriolis terms* that arise in classical equations of motion.

Taking the *differential* of Eq. (2.3.27), as with the derivative in Eq. (2.3.31),

$$\delta \mathbf{A}(\phi) = \frac{d}{d\phi} \mathbf{A}(\phi) \delta \phi = \delta \phi \mathbf{PA}(\phi) \quad (2.3.44)$$

Similarly, taking the differential of both sides of Eq. (2.3.40) and using Eq. (2.3.44) yields

$$\delta\mathbf{r}^P = \delta\mathbf{r} + \delta\mathbf{A}(\phi)\mathbf{s}'^P = \delta\mathbf{r} + \delta\phi\mathbf{PA}(\phi)\mathbf{s}'^P \quad (2.3.45)$$

where  $\delta\mathbf{r}$  and  $\delta\phi$  are *differentials*, or *variations*, of  $\mathbf{r}$  and  $\phi$ .

The position and orientation of a body-fixed reference frame, relative to an inertial reference frame, define the position and orientation of a planar body, using two position coordinates and a single orientation coordinate. Algebraic vectors and a  $2 \times 2$  orthogonal orientation transformation matrix provide the basis for characterizing the position, orientation, velocity, and acceleration of all points in the body. Identities involving orientation provide expressions for derivatives and variations of coordinates of points that are to be used in digital computation.

### **Key Formulas**

$$\mathbf{A} = \mathbf{A}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \quad \mathbf{P} = \mathbf{A}(\pi/2) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (2.3.27) \quad (2.3.29)$$

$$\mathbf{s} = \mathbf{As}' \quad \mathbf{PP} = -\mathbf{I} \quad \mathbf{PA} = \mathbf{AP} \quad (2.3.12) \quad (2.3.32) \quad (2.3.33)$$

$$\frac{d}{d\phi}\mathbf{A}(\phi) = \mathbf{PA}(\phi) \quad \frac{d}{d\phi}\mathbf{A}^T(\phi) = -\mathbf{A}^T(\phi)\mathbf{P} \quad (2.3.31)$$

$$\mathbf{r}^P = \mathbf{r} + \mathbf{A}(\phi)\mathbf{s}'^P \quad (2.3.40)$$

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\phi}\mathbf{PA}(\phi)\mathbf{s}'^P \quad \ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} + \ddot{\phi}\mathbf{PA}(\phi)\mathbf{s}'^P - \dot{\phi}^2\mathbf{A}(\phi)\mathbf{s}'^P \quad (2.3.42) \quad (2.3.43)$$

$$\delta\mathbf{A}(\phi) = \delta\phi\mathbf{PA}(\phi) \quad \delta\mathbf{r}^P = \delta\mathbf{r} + \delta\mathbf{A}(\phi)\mathbf{s}'^P = \delta\mathbf{r} + \delta\phi\mathbf{PA}(\phi)\mathbf{s}'^P \quad (2.3.44) \quad (2.3.45)$$

## 2.4 Position and Orientation of a Body in Space

A rigid body in space is defined as being made up of a continuum of particles that are constrained not to move relative to one another. While actual bodies are never perfectly rigid, deformation effects are often negligible when considering the motion of a machine that is comprised of multiple bodies. For this reason, a significant component of the study of kinematics and dynamics of machines is devoted to modeling individual bodies in a system as being rigid. Relations that define the position and orientation of a rigid body in space for multibody kinematics and dynamics are derived in this section, using the approach introduced in Section 2.3 for planar bodies. The matter of representing *orientation in space*, however is much more intricate.

### 2.4.1 Reference Frames for Location and Orientation in Space

The geometry of the body shown in Fig. 2.4.1 may be defined in an  $x'$ - $y'$ - $z'$  Cartesian reference frame that is fixed to the body, called a *body-fixed reference frame*. A point P may be defined in the body by a vector with algebraic representation  $\mathbf{s}^P$  relative to the  $x'$ - $y'$ - $z'$  frame. Representing the same vector as  $\mathbf{s}^P$  in the global  $x$ - $y$ - $z$  reference frame, point P may be located in the global frame by the vector relation

$$\mathbf{r}^P = \mathbf{r} + \mathbf{s}^P \quad (2.4.1)$$

Thus, data that define the geometry of the body are specified in the  $x'$ - $y'$ - $z'$  frame and transformed by vector relations into the  $x$ - $y$ - $z$  frame.

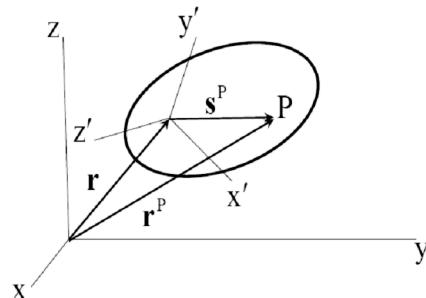


Figure 2.4.1 Body in Space with General Orientation

*Translation* of a body is defined by the vector  $\mathbf{r}$  from the origin of the  $x$ - $y$ - $z$  reference frame to the origin of the  $x'$ - $y'$ - $z'$  frame. Specification of *orientation* is, however, more complex. Nevertheless, once the  $x'$ - $y'$ - $z'$  frame is oriented relative to the  $x$ - $y$ - $z$  reference frame, all points of interest in the body can be defined in the  $x$ - $y$ - $z$  frame. Most of the attention in this section focuses on the transformation from vectors that are represented in a body-fixed  $x'$ - $y'$ - $z'$  frame to their representation in the  $x$ - $y$ - $z$  reference frame.

To see that the orientation of a body in space raises delicate technical questions, consider two rotation sequences for a body from the reference orientation shown in Fig. 2.4.2.

*Counterclockwise rotations* of the body by angles of  $\pi/2$  are successively made about its body-fixed  $x'$  and  $y'$  axes. Consider first the rotation sequence shown in Fig. 2.4.3. The first rotation is about the body-fixed  $x'$  axis, which initially coincides with the global reference

frame  $x$  axis. The second rotation is about the  $y'$  axis, as it coincides with the global reference frame  $z$  axis, resulting in the orientation shown on the right of Fig. 2.4.3.

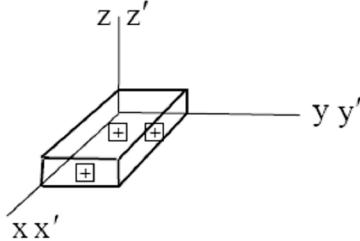


Figure 2.4.2 Rectangular Body in Reference Orientation

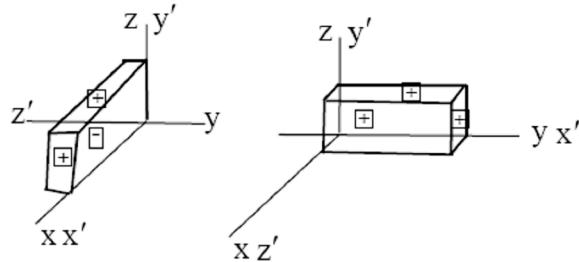


Figure 2.4.3 Rotation of Body by  $\pi/2$  About  $x'$  Axis and Then by  $\pi/2$  About  $y'$  Axis

Consider next the same rotations about body-fixed  $x'$  and  $y'$  axes, but in reverse order, as shown in Fig. 2.4.4. First, the body is rotated  $\pi/2$  about its body-fixed  $y'$  axis, which initially coincides with the global reference frame  $y$  axis. The body is then rotated  $\pi/2$  about its body-fixed  $x'$  axis, as it coincides with the negative global reference frame  $z$  axis, resulting in the orientation shown on the right of Fig. 2.4.4.

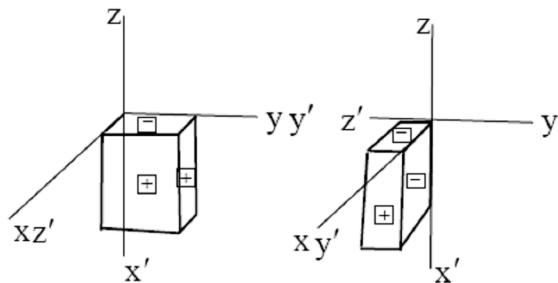


Figure 2.4.4 Rotation of Body by  $\pi/2$  About  $y'$  Axis and Then by  $\pi/2$  About  $x'$  Axis

Since the orientations shown on the right of Figs. 2.4.3 and 2.4.4 are distinctly different, it is clear that the order of rotation is important in defining the orientation of a body in space. Much as in the case of matrix multiplication, in which the order of terms in the product of two matrices is critically important, the order of rotations in space is likewise important. This shows that *rotation is not a vector quantity*, since the order of rotations is not commutative; i.e., *orders of rotation cannot be interchanged*.

## 2.4.2 Transformation of Coordinates

It is shown in Section 2.1 that a geometric vector is uniquely represented by an algebraic vector that contains components of the geometric vector in a Cartesian reference frame. The *components of a vector*, however, are defined in a specific Cartesian reference frame. Consider a Cartesian  $x'$ - $y'$ - $z'$  frame with the same origin as the  $x$ - $y$ - $z$  frame, as shown in Fig. 2.4.5. Unit  $x'$ ,  $y'$ , and  $z'$  coordinate vectors are denoted by  $\vec{f}$ ,  $\vec{g}$ , and  $\vec{h}$ , respectively, and unit  $x$ ,  $y$ , and  $z$  coordinate vectors are denoted by  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$ . A vector  $\vec{s}$  in space can be represented in either of the frames as

$$\vec{s} = s_x \vec{i} + s_y \vec{j} + s_z \vec{k} \quad (2.4.2)$$

or

$$\vec{s} = s_{x'} \vec{f} + s_{y'} \vec{g} + s_{z'} \vec{h} \quad (2.4.3)$$

where

$$s_x = \vec{s} \cdot \vec{i}, \quad s_y = \vec{s} \cdot \vec{j}, \quad s_z = \vec{s} \cdot \vec{k} \quad (2.4.4)$$

and

$$s_{x'} = \vec{s} \cdot \vec{f}, \quad s_{y'} = \vec{s} \cdot \vec{g}, \quad s_{z'} = \vec{s} \cdot \vec{h} \quad (2.4.5)$$

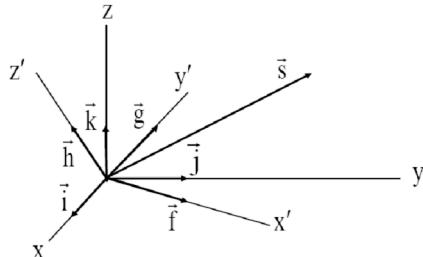


Figure 2.4.5 Cartesian Reference Frames with Common Origin

Algebraic vectors that define  $\vec{s}$  in the  $x$ - $y$ - $z$  and  $x'$ - $y'$ - $z'$  frames are

$$\mathbf{s} = \begin{bmatrix} s_x & s_y & s_z \end{bmatrix}^T \quad (2.4.6)$$

$$\mathbf{s}' = \begin{bmatrix} s_{x'} & s_{y'} & s_{z'} \end{bmatrix}^T \quad (2.4.7)$$

It is clear that there is a relation between  $\mathbf{s}$  and  $\mathbf{s}'$ , since they are defined by the same geometric vector  $\vec{s}$ . To establish this relation, the  $\vec{f}$ ,  $\vec{g}$ , and  $\vec{h}$  unit vectors may be written in terms of the  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$  unit vectors as

$$\begin{aligned} \vec{f} &= a_{11} \vec{i} + a_{21} \vec{j} + a_{31} \vec{k} \\ \vec{g} &= a_{12} \vec{i} + a_{22} \vec{j} + a_{32} \vec{k} \\ \vec{h} &= a_{13} \vec{i} + a_{23} \vec{j} + a_{33} \vec{k} \end{aligned} \quad (2.4.8)$$

where  $a_{ij}$  are the following *direction cosines*:

$$\begin{aligned} a_{11} &= \vec{i} \cdot \vec{f} = \cos(\vec{i}, \vec{f}), \quad a_{12} = \vec{i} \cdot \vec{g} = \cos(\vec{i}, \vec{g}), \quad a_{13} = \vec{i} \cdot \vec{h} = \cos(\vec{i}, \vec{h}) \\ a_{21} &= \vec{j} \cdot \vec{f} = \cos(\vec{j}, \vec{f}), \quad a_{22} = \vec{j} \cdot \vec{g} = \cos(\vec{j}, \vec{g}), \quad a_{23} = \vec{j} \cdot \vec{h} = \cos(\vec{j}, \vec{h}) \\ a_{31} &= \vec{k} \cdot \vec{f} = \cos(\vec{k}, \vec{f}), \quad a_{32} = \vec{k} \cdot \vec{g} = \cos(\vec{k}, \vec{g}), \quad a_{33} = \vec{k} \cdot \vec{h} = \cos(\vec{k}, \vec{h}) \end{aligned} \quad (2.4.9)$$

Substituting from Eq. (2.4.8) into Eq. (2.4.3) yields

$$\begin{aligned} \vec{s} = & \left( a_{11}s_{x'} + a_{12}s_{y'} + a_{13}s_{z'} \right) \vec{i} \\ & + \left( a_{21}s_{x'} + a_{22}s_{y'} + a_{23}s_{z'} \right) \vec{j} \\ & + \left( a_{31}s_{x'} + a_{32}s_{y'} + a_{33}s_{z'} \right) \vec{k} \end{aligned} \quad (2.4.10)$$

Equating coefficients of  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$  in this representation of  $\vec{s}$  with those in Eq. (2.4.2),

$$\begin{aligned} s_x &= a_{11}s_{x'} + a_{12}s_{y'} + a_{13}s_{z'} \\ s_y &= a_{21}s_{x'} + a_{22}s_{y'} + a_{23}s_{z'} \\ s_z &= a_{31}s_{x'} + a_{32}s_{y'} + a_{33}s_{z'} \end{aligned} \quad (2.4.11)$$

In matrix form, this is

$$\mathbf{s} = \mathbf{As}' \quad (2.4.12)$$

where  $\mathbf{A}$  is the spatial *direction cosine matrix*, or *orientation transformation matrix*,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (2.4.13)$$

The orientation transformation matrix  $\mathbf{A}$  has very special properties. If the x-y-z component forms of unit vectors  $\vec{f}$ ,  $\vec{g}$ , and  $\vec{h}$  are denoted by  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{h}$  and the x-y-z components of unit vectors  $\vec{i}$ ,  $\vec{j}$ , and  $\vec{k}$  are denoted by  $\mathbf{i}$ ,  $\mathbf{j}$ , and  $\mathbf{k}$ , then

$$\mathbf{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.4.14)$$

and Eq. (2.4.8) shows that

$$\mathbf{f} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} \quad (2.4.15)$$

Therefore, the matrix  $\mathbf{A}$  of Eq. (2.4.13) can be written as

$$\mathbf{A} = [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}] \quad (2.4.16)$$

As can be verified by direct calculation,

$$\begin{aligned} |\mathbf{A}| &= \begin{vmatrix} f_x & g_x & h_x \\ f_y & g_y & h_y \\ f_z & g_z & h_z \end{vmatrix} = f_x(g_y h_z - g_z h_y) + f_y(g_z h_x - g_x h_z) + f_z(g_x h_y - g_y h_x) \\ &= \mathbf{f}^T \begin{bmatrix} 0 & -g_z & g_y \\ g_z & 0 & -g_x \\ -g_y & g_x & 0 \end{bmatrix} \mathbf{h} = \mathbf{f}^T \tilde{\mathbf{g}} \mathbf{h} \end{aligned}$$

Since the vectors  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{h}$  form a right-hand Cartesian reference frame,  $\vec{f} = \vec{g} \times \vec{h}$ , or in component form,  $\mathbf{f} = \tilde{\mathbf{g}}\mathbf{h}$ . Thus,

$$|\mathbf{A}| = \mathbf{f}^T \tilde{\mathbf{g}} \mathbf{h} = \mathbf{f}^T \mathbf{f} = 1 \quad (2.4.17)$$

Expanding the matrix product  $\mathbf{A}\mathbf{A}^T$  yields

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} \mathbf{f}^T \\ \mathbf{g}^T \\ \mathbf{h}^T \end{bmatrix} \begin{bmatrix} \mathbf{f} & \mathbf{g} & \mathbf{h} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^T \mathbf{f} & \mathbf{f}^T \mathbf{g} & \mathbf{f}^T \mathbf{h} \\ \mathbf{g}^T \mathbf{f} & \mathbf{g}^T \mathbf{g} & \mathbf{g}^T \mathbf{h} \\ \mathbf{h}^T \mathbf{f} & \mathbf{h}^T \mathbf{g} & \mathbf{h}^T \mathbf{h} \end{bmatrix} \quad (2.4.18)$$

Since the unit vectors  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{h}$  form an *orthogonal triad*, this is

$$\mathbf{A}^T \mathbf{A} = \mathbf{I} \quad (2.4.19)$$

Thus,  $\mathbf{A}^T = \mathbf{A}^{-1}$  and the orientation transformation matrix  $\mathbf{A}$  is an *orthogonal matrix*. This special property permits an easy solution of Eq. (2.4.12),

$$\mathbf{s}' = \mathbf{A}^T \mathbf{s} \quad (2.4.20)$$

Transforming algebraic vectors that represent the same geometric vector in the x-y-z and x'-y'-z' Cartesian reference frames is thus a trivial matter, using Eqs. (2.4.12) and (2.4.20), once matrix  $\mathbf{A}$  is known.

When the origins of the x-y-z and x'-y'-z' frames do not coincide, the foregoing analysis is applied between the x'-y'-z' frame and a translated x-y-z frame, as shown in Fig. 2.4.1. If the algebraic vector  $\mathbf{s}'^P$  locates point P in the x'-y'-z' frame, then in the translated x-y-z frame this vector is

$$\mathbf{s}^P = \mathbf{A} \mathbf{s}'^P \quad (2.4.21)$$

Thus,

$$\mathbf{r}^P = \mathbf{r} + \mathbf{s}^P = \mathbf{r} + \mathbf{A} \mathbf{s}'^P \quad (2.4.22)$$

where  $\mathbf{r}$  is the vector from the origin of the x-y-z frame to the origin of the x'-y'-z' frame, as shown in Fig. 2.4.1.

### 2.4.3 Orientation Degrees of Freedom

The nine direction cosines in matrix  $\mathbf{A}$  define the orientation of the  $x'-y'-z'$  frame, relative to the  $x-y-z$  reference frame, but they are not independent. In component form, Eq. (2.4.19) is

$$\sum_{k=1}^3 a_{ki} a_{kj} = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases}, \quad i, j = 1, 2, 3 \quad (2.4.23)$$

Since interchanging  $i$  and  $j$  yields the same equation, Eq. (2.4.19) provides just six equations among the nine direction cosines.

To show that there are three orientation degrees of freedom in space, it must be shown that Eq. (2.4.19), equivalently Eq. (2.4.23), imposes six independent constraints on the nine components of  $\mathbf{A}$ . First note that by Eq. (2.4.16), the nine components of  $\mathbf{A}$  are the components of vectors  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{h}$ , so the nine variables comprising components of matrix  $\mathbf{A}$  are

$$\mathbf{q} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \\ \mathbf{h} \end{bmatrix}$$

The constraints imposed on these variables by the diagonal and upper triangular parts of Eq. (2.4.19), using the expansion of Eq. (2.4.18), may be written in the form

$$\Phi(\mathbf{q}) = \begin{bmatrix} \frac{1}{2}(\mathbf{f}^T \mathbf{f} - 1) \\ \frac{1}{2}(\mathbf{g}^T \mathbf{g} - 1) \\ \frac{1}{2}(\mathbf{h}^T \mathbf{h} - 1) \\ \mathbf{f}^T \mathbf{g} \\ \mathbf{f}^T \mathbf{h} \\ \mathbf{g}^T \mathbf{h} \end{bmatrix} = \mathbf{0} \quad (2.4.24)$$

Let the 9-vector  $\mathbf{q}^0 = [\mathbf{f}^{0T} \quad \mathbf{g}^{0T} \quad \mathbf{h}^{0T}]^T$  satisfy the six constraint equations of Eq. (2.4.24). The Jacobian of the constraint function  $\Phi(\mathbf{q})$  of Eq. (2.4.24), with the superscript 0 suppressed for notational convenience, is

$$\Phi_q = \begin{bmatrix} \mathbf{f}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{g}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}^T \\ \mathbf{g}^T & \mathbf{f}^T & \mathbf{0} \\ \mathbf{h}^T & \mathbf{0} & \mathbf{f}^T \\ \mathbf{0} & \mathbf{h}^T & \mathbf{g}^T \end{bmatrix}$$

The rows of the  $6 \times 9$  matrix  $\Phi_q$  are linearly independent if and only if the columns of the  $9 \times 6$  matrix  $\Phi_q^T$  are linearly independent; i.e., if and only if

$$\Phi_q^T \mathbf{u} = \mathbf{0}$$

implies that  $\mathbf{u} = [u_1 \cdots u_6]^T = \mathbf{0}$ . Expanding the product on the left, this is

$$\Phi_q^T \mathbf{u} = \begin{bmatrix} f & 0 & 0 & g & h & 0 \\ 0 & g & 0 & f & 0 & h \\ 0 & 0 & h & 0 & f & g \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \mathbf{0}$$

Expanding and reordering variables yields

$$\begin{aligned} [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}] \begin{bmatrix} u_1 \\ u_4 \\ u_5 \end{bmatrix} &= \mathbf{A} \begin{bmatrix} u_1 \\ u_4 \\ u_5 \end{bmatrix} = \mathbf{0} \\ [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}] \begin{bmatrix} u_4 \\ u_2 \\ u_6 \end{bmatrix} &= \mathbf{A} \begin{bmatrix} u_4 \\ u_2 \\ u_6 \end{bmatrix} = \mathbf{0} \\ [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}] \begin{bmatrix} u_5 \\ u_6 \\ u_3 \end{bmatrix} &= \mathbf{A} \begin{bmatrix} u_5 \\ u_6 \\ u_3 \end{bmatrix} = \mathbf{0} \end{aligned}$$

Since  $\mathbf{A} = [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}]$  is nonsingular, these three equations imply that all components of  $\mathbf{u}$  are zero, hence  $\mathbf{u} = \mathbf{0}$ . Thus,  $\Phi_q$  has full row rank and it has a  $6 \times 6$  submatrix with nonzero determinant. Defining the variables in  $\mathbf{q}$  that correspond to columns of the nonsingular submatrix as dependent, the implicit function theorem of Section 2.2.5 guarantees that these six variables can be written as unique functions of the remaining three variables, in some neighborhood of  $\mathbf{q}^0$ . Thus, there are indeed *three orientation degrees of freedom in space*.

While the nine direction cosines, subject to six constraints of Eq. (2.4.19), could be adopted as generalized coordinates that define orientation of the x'-y'-z' frame, this is neither practical nor convenient. Thus, other orientation generalized coordinates are sought.

It is fascinating and very important that a rigid body in the two-dimensional plane has one orientation degree of freedom, whereas a rigid body in three-dimensional space has three orientation degrees of freedom. As shown in Section 2.3, a single rotation angle generalized coordinate unambiguously defines orientation of a planar body and yields very simple orientation transformation relations. As will be seen throughout the remainder of this chapter, the search for generalized coordinates that define orientation of a body in space is more problematic.

For over two centuries, it has been known that a body in space has three orientation degrees of freedom. The search for a set of three globally valid generalized coordinates to characterize orientation, however, has continued to be about as productive as the eternal search

for the "fountain of youth". In fact, in 1964 a proof was published (Stuelpnagel, 1964) that there exists no set of three orientation generalized coordinates that is globally valid; i.e., without singularities. This shows that singularities are unavoidable if one uses any possible set of three orientation generalized coordinates. This remarkable paper further shows that orientation generalized coordinates of dimension four or more that introduce normalization constraints are valid, without singularities. The difficulty here is not the result of a pathological orientation transformation matrix, but with the mathematical impossibility of globally representing it with three orientation generalized coordinates. This is analogous to the difficulty highlighted in Example 1.2.1 of Section 1.2.2 in parameterization of the unit sphere; i.e., there is no pair of singularity free generalized coordinates that represents the entire surface of the smooth sphere.

---

**Example 2.4.1:** As a special case, consider the x-y-z and x'-y'-z' frames shown in Fig. 2.4.6, with z and z' axes that point out of the page coincident. The angle of rotation of the x' axis, relative to the x axis, with counterclockwise as positive, is denoted as  $\phi$ .

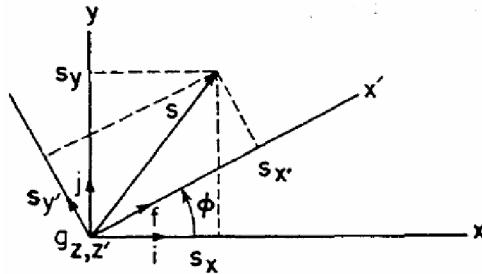


Figure 2.4.6 Reference Frames with Coincident z and z' Axes

From Eq. (2.4.9),

$$\begin{aligned} a_{11} &= a_{22} = \cos\phi \\ a_{21} &= \cos\left(\frac{\pi}{2} - \phi\right) = \sin\phi \\ a_{12} &= \cos\left(\frac{\pi}{2} + \phi\right) = -\sin\phi \\ a_{33} &= 1 \\ a_{31} &= a_{32} = a_{13} = a_{23} = 0 \end{aligned}$$

Thus, Eq. (2.4.12) yields the transformation

$$\mathbf{s} = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{x'} \\ s_{y'} \\ s_{z'} \end{bmatrix} \equiv \mathbf{As}' \quad (2.4.25)$$

which agrees with the planar relation in Eq. (2.3.27).

#### 2.4.4 Identities Involving the Orientation Transformation Matrix

Useful relations may be obtained by noting that, for any vectors  $\mathbf{s}'$  and  $\mathbf{v}'$  in the  $x'-y'-z'$  frame, the vector product of  $\mathbf{s}$  and  $\mathbf{v}$  in the  $x-y-z$  frame may be obtained by forming the vector product of  $\mathbf{s}'$  and  $\mathbf{v}'$  in the  $x'-y'-z'$  frame and transforming the result to the  $x-y-z$  frame; i.e.,

$$\tilde{\mathbf{s}}\mathbf{v} = \mathbf{A}(\tilde{\mathbf{s}}'\mathbf{v}')$$

Using  $\mathbf{v} = \mathbf{Av}'$ , this is

$$\tilde{\mathbf{s}}(\mathbf{Av}') = (\tilde{\mathbf{s}}\mathbf{A})\mathbf{v}' = (\mathbf{A}\tilde{\mathbf{s}}')\mathbf{v}'$$

Since this must hold for arbitrary vectors  $\mathbf{v}'$ ,

$$\tilde{\mathbf{s}}\mathbf{A} = \mathbf{A}\tilde{\mathbf{s}}' \quad (2.4.26)$$

Substituting  $\mathbf{s} = \mathbf{As}'$ , multiplying on the right by  $\mathbf{A}^T$ , and using Eq. (2.4.19) yields

$$\tilde{\mathbf{s}} = \widetilde{\mathbf{As}'} = \mathbf{A}\widetilde{\mathbf{s}'}\mathbf{A}^T \quad (2.4.27)$$

for arbitrary  $\mathbf{s}'$ . Similarly,

$$\tilde{\mathbf{s}}' = \widetilde{\mathbf{A}^T\mathbf{s}} = \mathbf{A}^T\widetilde{\mathbf{s}}\mathbf{A} \quad (2.4.28)$$

for arbitrary  $\mathbf{s}$ .

Consider the pair of Cartesian  $x'_i-y'_i-z'_i$  and  $x'_j-y'_j-z'_j$  frames shown in Fig. 2.4.7. An arbitrary vector  $\mathbf{s}$  in the  $x-y-z$  frame has representations  $\mathbf{s}'_i$  and  $\mathbf{s}'_j$  in the  $x'_i-y'_i-z'_i$  and  $x'_j-y'_j-z'_j$  frames, respectively; i.e.,

$$\mathbf{s} = \mathbf{A}_i\mathbf{s}'_i = \mathbf{A}_j\mathbf{s}'_j \quad (2.4.29)$$

where  $\mathbf{A}_i$  and  $\mathbf{A}_j$  are transformation matrices from the  $x'_i-y'_i-z'_i$  and  $x'_j-y'_j-z'_j$  frames to the  $x-y-z$  frame, respectively. Since  $\mathbf{A}_i$  and  $\mathbf{A}_j$  are orthogonal matrices, Eq. (2.4.29) yields

$$\mathbf{s}'_i = \mathbf{A}_i^T\mathbf{A}_j\mathbf{s}'_j \equiv \mathbf{A}_{ij}\mathbf{s}'_j \quad (2.4.30)$$

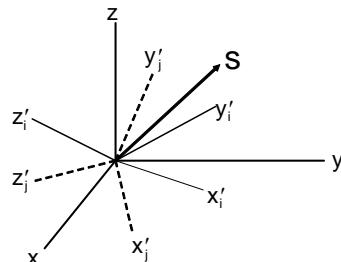


Figure 2.4.7 Reference Frames with Coincident Origins

Since  $\mathbf{s}'_j$  is an arbitrary vector,

$$\mathbf{A}_{ij} = \mathbf{A}_i^T \mathbf{A}_j \quad (2.4.31)$$

is the transformation matrix from the  $x'_j-y'_j-z'_j$  frame to the  $x'_i-y'_i-z'_i$  frame. A direct calculation shows that  $\mathbf{A}_{ij}$  is an *orthogonal matrix*.

In numerous applications, kinematic constraints dictate that certain axes of the  $x'_i-y'_i-z'_i$  and  $x'_j-y'_j-z'_j$  frames remain parallel; e.g., the unit vectors  $\mathbf{h}_i$  and  $\mathbf{h}_j$  shown in Fig. 2.4.8 are to be *collinear*, with  $\mathbf{h}_i = \mathbf{h}_j$ . The angle  $\theta_{ij}$  of rotation, measured positive as counterclockwise from  $\mathbf{f}_i$  to  $\mathbf{f}_j$ , is to be calculated.

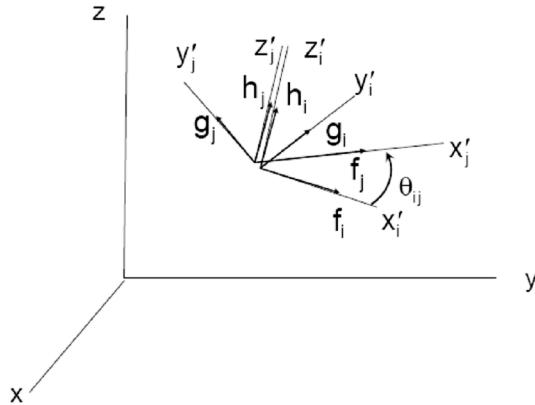


Figure 2.4.8 Triads with  $z'$  Axes Coincident

From the definition of scalar product and the fact that the coordinate vectors shown in Fig. 2.4.8 are unit vectors,

$$\mathbf{f}_i^T \mathbf{f}_j = \cos\theta_{ij} \quad (2.4.32)$$

Similarly, from the definition of vector product,

$$\tilde{\mathbf{f}}_i \mathbf{f}_j = \mathbf{h}_i \sin\theta_{ij} \quad (2.4.33)$$

Taking the scalar product of both sides of Eq. (2.4.33) with  $\mathbf{h}_i$  and using the fact that  $\tilde{\mathbf{f}}_i \mathbf{h}_i = -\mathbf{g}_i$ ,

$$\sin\theta_{ij} = \mathbf{h}_i^T \tilde{\mathbf{f}}_i \mathbf{f}_j = \mathbf{g}_i^T \mathbf{f}_j \quad (2.4.34)$$

Writing the unit vectors in terms of the respective reference frames in which they are fixed and using the transformation matrices from these frames to the  $x-y-z$  frame, Eq. (2.4.32) becomes

$$\cos\theta_{ij} = \mathbf{f}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j \quad (2.4.35)$$

Similarly, Eq. (2.4.34) becomes

$$\sin\theta_{ij} = \mathbf{g}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j \quad (2.4.36)$$

To simplify notation, define  $c \equiv \cos\theta_{ij} = \mathbf{f}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j$  and  $s \equiv \sin\theta_{ij} = \mathbf{g}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j$ .

As seen in Fig. 2.4.9, in the interval  $[-\pi/2 \leq \theta_{ij} \leq \pi/2]$ ,  $c \geq 0$  and  $\theta_{ij} = \text{Arcsin}(\mathbf{g}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j)$ .

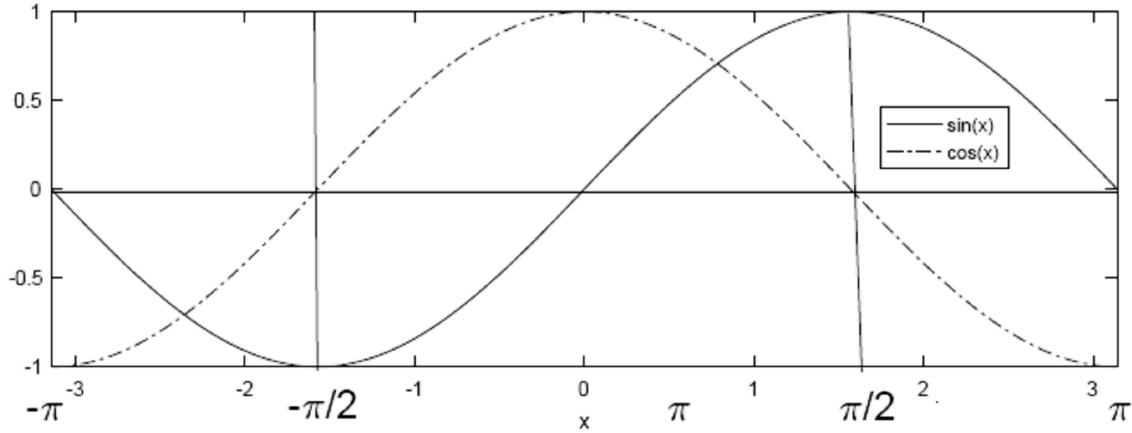


Figure 2.4.9  $\sin(x)$  and  $\cos(x)$  on Interval  $[-\pi \leq x \leq \pi]$

In the interval  $[-\pi \leq \theta_{ij} \leq -\pi/2]$ ,  $c \leq 0$  and  $s \leq 0$ . Symmetry of the sin function about  $-\pi/2$  yields  $\sin(-\pi/2 + \phi) = \sin(-\pi/2 - \phi) = -\sin(\pi/2 + \phi)$ . With  $\theta_{ij} = -\pi/2 - \phi$ ,  $\phi = -\pi/2 - \theta_{ij}$  and  $s = \sin(\theta_{ij}) = \sin(-\pi/2 - \pi/2 - \theta_{ij}) = -\sin(\theta_{ij} + \pi)$ . Since  $0 \leq \theta_{ij} + \pi \leq \pi/2$ ,  $\theta_{ij} + \pi = \text{Arcsin}(-s) = -\text{Arcsin}(s)$  and  $\theta_{ij} = -\pi - \text{Arcsin}(\mathbf{g}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j)$ .

In the interval  $[\pi/2 \leq \theta_{ij} \leq \pi]$ ,  $c \leq 0$  and  $s \geq 0$ . Symmetry of the sin function about  $\pi/2$  yields  $\sin(\pi/2 + \phi) = \sin(\pi/2 - \phi)$ . With  $\theta_{ij} = \pi/2 + \phi$ ,  $\phi = \theta_{ij} - \pi/2$  and  $s = \sin(\theta_{ij}) = \sin(\pi/2 + \pi/2 - \theta_{ij}) = \sin(\pi - \theta_{ij})$ . Since  $0 \leq \pi - \theta_{ij} \leq \pi/2$ ,  $\pi - \theta_{ij} = \text{Arcsin}(s)$  and  $\theta_{ij} = \pi - \text{Arcsin}(\mathbf{g}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j)$ .

In summary, with  $c \equiv \cos\theta_{ij} = \mathbf{f}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j$  and  $s \equiv \sin\theta_{ij} = \mathbf{g}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{f}'_j$  known, the angle  $\theta_{ij}$  is uniquely determined in the interval  $[-\pi \leq \theta_{ij} \leq \pi]$  as

$$\theta_{ij} = \begin{cases} -\pi - \text{Arcsin}(s), & \text{if } s \leq 0 \text{ and } c < 0 \\ \text{Arcsin}(s), & \text{if } c \geq 0 \\ \pi - \text{Arcsin}(s), & \text{if } s \geq 0 \text{ and } c < 0 \end{cases} \quad (2.4.37)$$

which is implemented in MATLAB Code 2.4 in Section 2.B.2 of Appendix 2.B.

#### 2.4.5 Velocity, Acceleration, Angular Velocity, and Angular Acceleration

An  $x'$ - $y'$ - $z'$  frame is fixed in a moving body to define the body's position and orientation, relative to an  $x$ - $y$ - $z$  global reference frame. Consider a point P that is fixed in the  $x'$ - $y'$ - $z'$  frame, as shown in Fig. 2.4.1. The vector that locates P in the  $x$ - $y$ - $z$  reference frame is given by Eq. (2.4.22),

$$\mathbf{r}^P = \mathbf{r} + \mathbf{A}\mathbf{s}'^P \quad (2.4.38)$$

where  $\mathbf{s}'^P$  is the constant vector of coordinates of P in the  $x'-y'-z'$  frame and  $\mathbf{A}$  is the orientation transformation matrix from the  $x'-y'-z'$  frame relative to the stationary  $x-y-z$  frame.

Since the  $x'-y'-z'$  frame is moving and changing its orientation with time, the vector  $\mathbf{r}$  and transformation matrix  $\mathbf{A}$  are functions of time. The *differentiation rules* of Section 2.1.4 can be used to obtain the time derivative of  $\mathbf{r}^P$  as

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\mathbf{A}}\mathbf{s}'^P \quad (2.4.39)$$

Using Eq. (2.4.20), this result may be rewritten as

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\mathbf{A}}\mathbf{A}^T\mathbf{s}^P \quad (2.4.40)$$

To interpret terms in Eq.(2.4.40), it is helpful to derive an identity that involves the orientation transformation matrix  $\mathbf{A}$ . Differentiating both sides of  $\mathbf{AA}^T = \mathbf{I}$  with respect to time,

$$\dot{\mathbf{A}}\mathbf{A}^T + \mathbf{A}\dot{\mathbf{A}}^T = \mathbf{0} \quad (2.4.41)$$

Thus,

$$\dot{\mathbf{A}}\mathbf{A}^T = -\mathbf{A}\dot{\mathbf{A}}^T = -(\dot{\mathbf{A}}\mathbf{A}^T)^T \quad (2.4.42)$$

so  $\dot{\mathbf{A}}\mathbf{A}^T$  is skew-symmetric. As noted following the definition of Eq. (2.1.22) in Section 2.1, there exists a vector  $\boldsymbol{\omega}$  such that  $\tilde{\boldsymbol{\omega}} = \dot{\mathbf{A}}\mathbf{A}^T$ . Using Eq. (2.4.26), the body fixed representation of this vector is  $\tilde{\boldsymbol{\omega}}' = \mathbf{A}^T(\dot{\mathbf{A}}\mathbf{A}^T)\mathbf{A} = \mathbf{A}^T\dot{\mathbf{A}}$ . Together, these relations are

$$\begin{aligned} \tilde{\boldsymbol{\omega}} &= \dot{\mathbf{A}}\mathbf{A}^T \\ \tilde{\boldsymbol{\omega}}' &= \mathbf{A}^T\dot{\mathbf{A}} \end{aligned} \quad (2.4.43)$$

This is the definition of  $\boldsymbol{\omega}$  in terms of the transformation matrix  $\mathbf{A}$  and its time derivative.

Substituting the first of Eqs. (2.4.43) into Eq. (2.4.40) yields the *velocity equation*

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \tilde{\boldsymbol{\omega}}\mathbf{s}^P \quad (2.4.44)$$

In geometric vector notation, this is the familiar relationship

$$\vec{\mathbf{r}}^P = \vec{\mathbf{r}} + \vec{\boldsymbol{\omega}} \times \vec{\mathbf{s}}^P \quad (2.4.45)$$

where  $\vec{\boldsymbol{\omega}}$  is the angular velocity of the  $x'-y'-z'$  frame. Thus, the vector  $\boldsymbol{\omega}$  defined by Eq. (2.4.43) is the *angular velocity* of the  $x'-y'-z'$  frame, relative to the  $x-y-z$  frame. Since the vector product in Eq. (2.4.44) is with the vector  $\mathbf{s}^P$  represented in the  $x-y-z$  frame, the vector  $\boldsymbol{\omega}$  is represented in the  $x-y-z$  frame.

It is important to recall that angular velocity is defined by Eq. (2.4.43), rather than a physical argument based on rotation speed about some axis. Such arguments introducing angular velocity are generally incorrect, other than in a special case of rotation about an axis that is fixed in space; i.e., not time dependent.

Since  $\boldsymbol{\omega}$  is a vector represented in the  $x-y-z$  frame, it may also be represented in the  $x'-y'-z'$  frame as

$$\omega' = \mathbf{A}^T \omega \quad (2.4.46)$$

Taking the vector product called for in Eq. (2.4.44) in the  $x'-y'-z'$  frame and transforming to the  $x-y-z$  frame yields

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \mathbf{A}\tilde{\omega}'\mathbf{s}'^P \quad (2.4.47)$$


---

**Example 2.4.2:** The transformation matrix  $\mathbf{A}$  of Example 2.4.1 is defined in Eq. (2.4.25) as a function of the angle  $\phi$ . For this matrix, Eq. (2.4.43) yields

$$\begin{aligned} \tilde{\omega} &= \dot{\mathbf{A}}\mathbf{A}^T = \dot{\phi} \begin{bmatrix} -\sin\phi & -\cos\phi & 0 \\ \cos\phi & -\sin\phi & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\dot{\phi} & 0 \\ \dot{\phi} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \equiv \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \end{aligned}$$

Thus,  $\omega = [0 \ 0 \ \dot{\phi}]^T$ , which agrees with the concept of angular velocity about a fixed axis.

---

Multiplying both sides of the first of Eqs. (2.4.43) on the right by  $\mathbf{A}$  yields the useful relationship

$$\dot{\mathbf{A}} = \tilde{\omega}\mathbf{A} \quad (2.4.48)$$

and on the left by  $\mathbf{A}$  in the second of Eqs. (2.4.43) yields

$$\dot{\mathbf{A}} = \mathbf{A}\tilde{\omega}' \quad (2.4.49)$$

Equations (2.4.48) and (2.4.49) provide frequently used relationships between the time derivative of the transformation matrix  $\mathbf{A}$  and the *angular velocity*  $\omega$  of the  $x'-y'-z'$  frame.

Equation (2.4.39) may be differentiated with respect to time to obtain

$$\ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} + \ddot{\mathbf{A}}\mathbf{s}'^P \quad (2.4.50)$$

Differentiating Eq. (2.4.48) and using Eq. (2.4.48), the following relationships for the second time derivative of the transformation matrix  $\mathbf{A}$  are obtained:

$$\begin{aligned} \ddot{\mathbf{A}} &= \tilde{\omega}\mathbf{A} + \tilde{\omega}\dot{\mathbf{A}} \\ &= \tilde{\omega}\mathbf{A} + \tilde{\omega}\tilde{\omega}\mathbf{A} \end{aligned} \quad (2.4.51)$$

where  $\dot{\omega}$  is *angular acceleration* of the  $x'-y'-z'$  frame. Similarly, using Eq. (2.4.49),

$$\ddot{\mathbf{A}} = \mathbf{A}\tilde{\omega}' + \mathbf{A}\tilde{\omega}'\tilde{\omega}' \quad (2.4.52)$$

Substituting from Eq. (2.4.51) into Eq. (2.4.50) yields

$$\begin{aligned}\ddot{\mathbf{r}}^P &= \ddot{\mathbf{r}} + \tilde{\boldsymbol{\omega}}\mathbf{s}^P + \tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{s}^P \\ &= \ddot{\mathbf{r}} + \mathbf{A}\tilde{\boldsymbol{\omega}}'\mathbf{s}'^P + \mathbf{A}\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{s}'^P\end{aligned}\quad (2.4.53)$$

The form of Eqs. (2.4.53) and (2.4.47) are identical to the form of Eqs. (2.3.23) and (2.3.24), and is characteristic of what is to be encountered in the equations of multibody kinematics and dynamics. Velocity equations are linear in first derivatives of generalized coordinates and angular velocities, with coefficients that depend on generalized coordinates. Acceleration equations are linear in second derivatives of generalized coordinates and angular acceleration, with coefficients that depend on generalized coordinates, but contain quadratic terms in first derivatives of generalized coordinates and angular velocities, with coefficients that depend on generalized coordinates. The quadratic velocity terms in the acceleration equations are reminiscent of *Coriolis terms* that arise in classical equations of motion. These nonlinear effects reinforce the arguments in Chapter 1 that mechanical systems are highly nonlinear.

**Example 2.4.3:** Using the results of Example 2.4.1 and Eq. (2.4.47), the velocity of a point P that is defined by  $\mathbf{s}'^P = [1 \ 1 \ 1]^T$  in the x'-y'-z' frame is

$$\dot{\mathbf{r}}^P = \mathbf{A}\tilde{\boldsymbol{\omega}}'\mathbf{s}'^P = \dot{\phi}[-\cos\phi - \sin\phi \ \cos\phi - \sin\phi \ 0]^T$$

where  $\dot{\mathbf{r}} = \mathbf{0}$ . The acceleration of point P, from Eqs. (2.4.50) and (2.4.51), is

$$\ddot{\mathbf{r}}^P = \ddot{\mathbf{A}}\mathbf{s}'^P = \ddot{\phi}\begin{bmatrix} -\cos\phi - \sin\phi \\ \cos\phi - \sin\phi \\ 0 \end{bmatrix} + \dot{\phi}^2\begin{bmatrix} -\cos\phi + \sin\phi \\ -\cos\phi - \sin\phi \\ 0 \end{bmatrix}$$

Derivative relations derived here serve as the foundation for kinematics and dynamics of rigid bodies in space. The reader is encouraged to become proficient in their use, in preparation for dynamic analysis. As an aid in the use of the relations derived, a summary of key formulas is provided at the end of the section.

## 2.4.6 Infinitesimal Rotation and Displacement

Much as in the case of angular velocity, the identity  $\mathbf{AA}^T = \mathbf{I}$  may be manipulated using the *differential*, rather than time derivative. In this setting, the differential may be interpreted as an *infinitesimal variation* at a given instant in time; i.e., with time held fixed. Taking the differential of both sides of the identity of Eq. (2.4.19) yields

$$\mathbf{AA}^T + \mathbf{A} \ \mathbf{A}^T = \mathbf{O}(\mathbf{A}^2) \quad (2.4.54)$$

where the term  $\mathbf{O}(\mathbf{A}^2)$  denotes *infinitesimal quantities* that approach zero faster than  $\mathbf{A}$ . In terms of the matrix norm  $\|\mathbf{B}\| = \max_{i,j} |b_{ij}|$ , any matrix  $\mathbf{D}$  in  $\mathbf{O}(\mathbf{A}^2)$  satisfies the property

$$\lim_{\|\delta\mathbf{A}\| \rightarrow 0} \left( \frac{\|\mathbf{D}\|}{\|\mathbf{A}\|} \right) = 0$$

In the sense of *infinitesimals*, the right side of Eq.(2.4.54) is said to be zero; i.e.,

$$\mathbf{A}\mathbf{A}^T + \mathbf{A}'\mathbf{A}^T = \mathbf{0} \quad (2.4.55)$$

Thus,

$$\mathbf{A}\mathbf{A}^T = -\mathbf{A}'\mathbf{A}^T = -(\mathbf{A}\mathbf{A}^T)^T \quad (2.4.56)$$

and the matrix  $\mathbf{A}\mathbf{A}^T$  is skew symmetric.

Since any *skew symmetric matrix* can be written as the tilde operator applied to some 3-vector, there exists a vector denoted  $\boldsymbol{\pi}$  such that

$$\mathbf{A}\mathbf{A}^T = \tilde{\boldsymbol{\pi}} \quad (2.4.57)$$

or,

$$\mathbf{A} = \tilde{\boldsymbol{\pi}}\mathbf{A} \quad (2.4.58)$$

Taking the differential of the vector relationship  $\mathbf{a} = \mathbf{A}\mathbf{a}'$ , with  $\mathbf{a}'$  constant,

$$\mathbf{a} = \mathbf{A}\mathbf{a}' = \tilde{\boldsymbol{\pi}}\mathbf{A}\mathbf{a}' = \tilde{\boldsymbol{\pi}}\mathbf{a} \quad (2.4.59)$$

Equation (2.4.59) shows that the vector product on the right is between  $\boldsymbol{\pi}$  and  $\mathbf{a}$ . Since  $\mathbf{a}$  is a vector represented in the global reference frame,  $\boldsymbol{\pi}$  is also represented in the global reference frame.

To see if the matrix  $\mathbf{A} + \mathbf{A}' = \mathbf{A} + \tilde{\boldsymbol{\pi}}\mathbf{A}$  is orthogonal, as must be the case if the variation defined by  $\boldsymbol{\pi}$  represents a rotation, expanding the product

$$\begin{aligned} (\mathbf{A} + \tilde{\boldsymbol{\pi}}\mathbf{A})^T(\mathbf{A} + \tilde{\boldsymbol{\pi}}\mathbf{A}) &= \mathbf{A}^T\mathbf{A} + \mathbf{A}^T\tilde{\boldsymbol{\pi}}\mathbf{A} - \mathbf{A}^T\tilde{\boldsymbol{\pi}}\mathbf{A} - \mathbf{A}^T\tilde{\boldsymbol{\pi}}\tilde{\boldsymbol{\pi}}\mathbf{A} \\ &= \mathbf{I} + (\tilde{\boldsymbol{\pi}}\mathbf{A})^T\tilde{\boldsymbol{\pi}}\mathbf{A} = \mathbf{I} + \mathbf{A}^T\mathbf{A} \\ &= \mathbf{I} + \mathbf{O}(\mathbf{A}^2) \end{aligned} \quad (2.4.60)$$

since the product  $\mathbf{A}^T\mathbf{A}$  is  $\mathbf{O}(\mathbf{A}^2)$ . Thus, in the sense of infinitesimals,  $\mathbf{A} + \mathbf{A}' = \mathbf{A} + \tilde{\boldsymbol{\pi}}\mathbf{A}$  is indeed orthogonal and  $\boldsymbol{\pi}$  represents an *infinitesimal rotation* of the x'-y'-z' frame.

In terms of the body fixed representation of infinitesimal rotation  $\boldsymbol{\pi}'$ ,  $\boldsymbol{\pi} = \mathbf{A}'\boldsymbol{\pi}'$ , and

$$\mathbf{A} = \tilde{\boldsymbol{\pi}}\mathbf{A} = \mathbf{A}'\tilde{\boldsymbol{\pi}}'\mathbf{A}^T\mathbf{A} = \mathbf{A}'\tilde{\boldsymbol{\pi}}' \quad (2.4.61)$$

where the identity  $\tilde{\mathbf{b}} = \mathbf{A}\tilde{\mathbf{b}}'\mathbf{A}^T$  of Eq. (2.4.27) has been used with  $\mathbf{b} = \boldsymbol{\pi}$ . Thus,  $\boldsymbol{\pi}$  and  $\boldsymbol{\pi}'$  play a role similar to that of angular velocity.

Taking the differential of the vector relation

$$\mathbf{r}^P = \mathbf{r} + \mathbf{s}^P = \mathbf{r} + \mathbf{A}\mathbf{s}'^P \quad (2.4.62)$$

and using Eq. (2.4.61) yields

$$\mathbf{r}^P = \mathbf{r} + \mathbf{A}\mathbf{s}'^P = \mathbf{r} + \tilde{\boldsymbol{\pi}}\mathbf{s}^P = \mathbf{r} + \mathbf{A}'\tilde{\boldsymbol{\pi}}'\mathbf{s}'^P \quad (2.4.63)$$

where  $\mathbf{r}$  is regarded as an *infinitesimal displacement*.

In contrast to characterizing the orientation of a body in a plane, representation of orientation of a body in space is at best intricate. The vector locating the origin of a body-fixed reference frame in space is an easy extension of the planar case. For orientation, however, it is shown that the transformation matrix from the body-fixed frame to the global frame is a  $3 \times 3$  orthogonal matrix. The implicit function theorem shows that there are three orientation degrees of freedom, but does not identify three globally applicable orientation variables. This is because there are none, as proven only in 1964 by Stuelpnagel.

Identities involving the orientation transformation matrix and its time derivative lead to the definition of angular velocity and angular acceleration, each a three-dimensional vector. Vector and differential calculus define the position, velocity, acceleration, and variation of all points in a body, which are required for establishing equations of motion of bodies in space.

### **Key Formulas**

$$\mathbf{s} = \mathbf{A}\mathbf{s}' \quad \mathbf{A} = [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}] \quad \mathbf{s}' = \mathbf{A}^T \mathbf{s} \quad \mathbf{A}^T \mathbf{A} = \mathbf{I} \quad (2.4.12) \quad (2.4.16) \quad (2.4.20) \quad (2.4.19)$$

$$\mathbf{r}^P = \mathbf{r} + \mathbf{s}^P = \mathbf{r} + \mathbf{A}\mathbf{s}'^P \quad (2.4.22)$$

$$\tilde{\mathbf{s}} = \mathbf{A}\tilde{\mathbf{s}}' \mathbf{A}^T \quad \tilde{\mathbf{s}}' = \mathbf{A}^T \tilde{\mathbf{s}} \mathbf{A} \quad (2.4.27) \quad (2.4.28)$$

$$\tilde{\boldsymbol{\omega}} = \dot{\mathbf{A}}\mathbf{A}^T \quad \tilde{\boldsymbol{\omega}}' = \mathbf{A}^T \dot{\mathbf{A}} \quad (2.4.43)$$

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \tilde{\boldsymbol{\omega}}\mathbf{s}^P \quad \dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \mathbf{A}\tilde{\boldsymbol{\omega}}'\mathbf{s}'^P \quad (2.4.44) \quad (2.4.47)$$

$$\dot{\mathbf{A}} = \tilde{\boldsymbol{\omega}}\mathbf{A} \quad \dot{\mathbf{A}} = \mathbf{A}\tilde{\boldsymbol{\omega}}' \quad (2.4.48) \quad (2.4.49)$$

$$\ddot{\mathbf{A}} = \tilde{\boldsymbol{\omega}}\mathbf{A} + \tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{A} \quad \ddot{\mathbf{A}} = \mathbf{A}\tilde{\boldsymbol{\omega}}' + \mathbf{A}\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}' \quad (2.4.51) \quad (2.4.52)$$

$$\ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} - \tilde{\mathbf{s}}^P \tilde{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{s}^P = \ddot{\mathbf{r}} + \mathbf{A}\tilde{\boldsymbol{\omega}}'\mathbf{s}'^P + \mathbf{A}\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{s}'^P \quad (2.4.53)$$

$$\mathbf{A} = \tilde{\boldsymbol{\pi}}\mathbf{A} \quad \mathbf{A} = \mathbf{A} \tilde{\boldsymbol{\pi}}' \quad (2.4.58) \quad (2.4.61)$$

$$\mathbf{r}^P = \mathbf{r} + \tilde{\boldsymbol{\pi}}\mathbf{s}^P = \mathbf{r} + \mathbf{A} \tilde{\boldsymbol{\pi}}'\mathbf{s}'^P \quad (2.4.63)$$

## 2.5 Euler Parameter Orientation Coordinates for a Body in Space

A set of four Euler parameters that satisfy a normalization condition is introduced to define orientation of a body in space. Euler parameters are shown to avoid singularities that are encountered with any set of three coordinates are used to define orientation and to have attractive analytical and computational properties.

### 2.5.1 Invariants of the Orientation Transformation

Using Eq. (2.4.19) and the notation  $\|\mathbf{a}\| = \sqrt{(\mathbf{a}^T \mathbf{a})}$  for length of a vector,

$$\|\mathbf{a}\|^2 = \mathbf{a}^T \mathbf{a} = \mathbf{a}'^T \mathbf{A}^T \mathbf{A} \mathbf{a}' = \mathbf{a}'^T \mathbf{a}' = \|\mathbf{a}'\|^2 \quad (2.5.1)$$

Thus, orientation transformation preserves length of a vector; i.e.,

$$\|\mathbf{a}\| = \|\mathbf{A} \mathbf{a}'\| = \|\mathbf{a}'\| \quad (2.5.2)$$

Using this result and the scalar product relation of Eq. (2.1.20),

$$\begin{aligned} \vec{\mathbf{a}} \cdot \vec{\mathbf{b}} &= \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b} = \mathbf{a}'^T \mathbf{A}^T \mathbf{A} \mathbf{b}' \\ &= \mathbf{a}'^T \mathbf{b}' = \|\mathbf{a}'\| \|\mathbf{b}'\| \cos \theta(\mathbf{a}', \mathbf{b}') = \vec{\mathbf{a}}' \cdot \vec{\mathbf{b}}' \end{aligned} \quad (2.5.3)$$

Thus, scalar product is preserved and

$$\cos \theta(\mathbf{a}, \mathbf{b}) = \cos \theta(\mathbf{a}', \mathbf{b}') \quad (2.5.4)$$

so orientation transformation preserves the angle between vectors.

An important property of the orientation transformation matrix  $\mathbf{A}$ , shown in Eq. (2.4.17), is that its determinant is unity; i.e.,

$$|\mathbf{A}| = 1 \quad (2.5.5)$$

Direct manipulation yields the following relationship:

$$(\mathbf{A} - \mathbf{I}) \mathbf{A}^T = \mathbf{A}^T \mathbf{A} - \mathbf{A}^T = \mathbf{I} - \mathbf{A}^T = (\mathbf{I} - \mathbf{A})^T = -(\mathbf{A} - \mathbf{I})^T \quad (2.5.6)$$

Key facts (Kreyszig, 2011) regarding determinants are that the determinant of a product of square matrices  $\mathbf{A}$  and  $\mathbf{B}$  is the product of their determinants; i.e.,  $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$ , the determinant of the transpose of a matrix is its determinant, and the determinant of the negative of a  $3 \times 3$  matrix is  $(-1)^3 = -1$  times its determinant. Using these facts and Eq. (2.5.6), for any orthogonal matrix  $\mathbf{A}$  with unit determinant,

$$|\mathbf{A} - \mathbf{I}| = |\mathbf{A} - \mathbf{I}| |\mathbf{A}^T| = |(\mathbf{A} - \mathbf{I}) \mathbf{A}^T| = |\mathbf{I} - \mathbf{A}^T| = |-(\mathbf{A} - \mathbf{I})^T| = |-(\mathbf{A} - \mathbf{I})| = -|\mathbf{A} - \mathbf{I}| \quad (2.5.7)$$

Thus,  $|\mathbf{A} - \mathbf{I}| = 0$ . By Eq. (2.2.20), there exists a unit vector  $\mathbf{u}$  such that  $(\mathbf{A} - \mathbf{I})\mathbf{u} = \mathbf{0}$ , or

$$\mathbf{A}\mathbf{u} = \mathbf{u} \quad (2.5.8)$$

By definition,  $\mathbf{A}\mathbf{u}' = \mathbf{u}$  and Eq. (2.5.8) shows that

$$\mathbf{u} = \mathbf{u}' \quad (2.5.9)$$

Thus, the transformation  $\mathbf{A}$  leaves the vector  $\mathbf{u}$  unchanged, so the x-y-z frame may be brought into coincidence with the  $x'-y'-z'$  frame by a rotation by an angle  $\theta$  about the vector  $\mathbf{u}$ , as shown in Fig. 2.5.1. This result may be interpreted geometrically in the form of *Euler's theorem* (Palais, Palais, and Rodi, 2009).

**Euler's Theorem:** If the origins of two right-hand Cartesian reference frames coincide, they may be brought into coincidence by a single rotation about some axis.

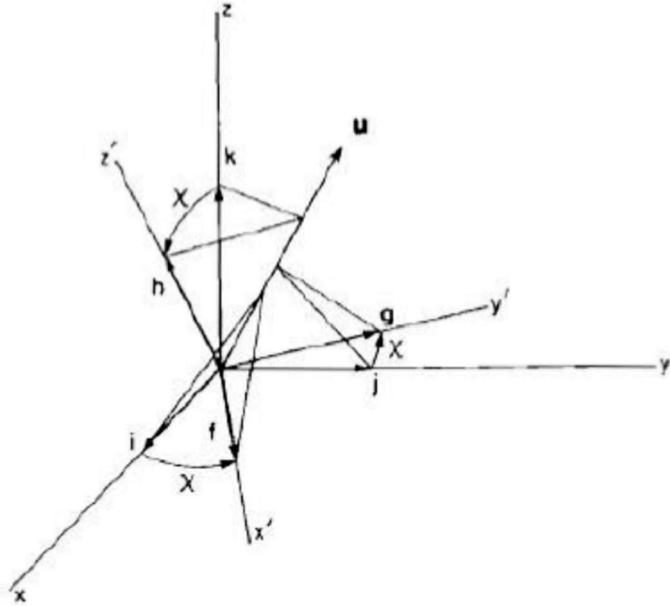


Figure 2.5.1 Euler Rotation

## 2.5.2 Orientation of Reference Frames

As shown in Fig. 2.5.2, the Euler rotation transforms the unit vector  $\mathbf{i}$  on the surface of a cone until it comes into coincidence with unit vector  $\mathbf{f}$  and  $\cos\theta(\mathbf{u},\mathbf{i}) = \cos\theta(\mathbf{u},\mathbf{f})$ . Since the ends of  $\mathbf{f}$  and  $\mathbf{i}$  lie in a plane that is orthogonal to  $\mathbf{u}$ , as shown in Fig. 2.5.2, both  $\mathbf{f}$  and  $\mathbf{i}$  have the same projection on  $\mathbf{u}$ , given by

$$\mathbf{v} = (\mathbf{i}^T \mathbf{u}) \mathbf{u} = (\mathbf{f}^T \mathbf{u}) \mathbf{u} \quad (2.5.10)$$

Since  $\mathbf{i} - \mathbf{v}$  is rotated by the transformation to  $\mathbf{f} - \mathbf{v}$  and vector length is preserved by the transformation,

$$\|\mathbf{i} - \mathbf{v}\| = \|\mathbf{f} - \mathbf{v}\| = \sin\theta(\mathbf{i}, \mathbf{u}) = \|\tilde{\mathbf{u}}\mathbf{i}\| \quad (2.5.11)$$

so  $\mathbf{i} - \mathbf{v}$  and  $\mathbf{f} - \mathbf{v}$  form radii of a circle in the plane orthogonal to  $\mathbf{u}$ , as shown in Fig. 2.5.2.

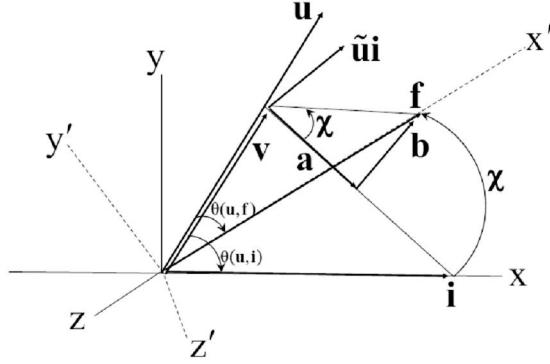


Figure 2.5.2. Relation Between Transformed Vectors  $\mathbf{f}$  and  $\mathbf{i}$

The objective is to relate unit vector  $\mathbf{i}$  and its counterpart  $\mathbf{f}$  in the  $x$ - $y$ - $z$  reference frame. The vector  $\mathbf{v} = (\mathbf{u}^T \mathbf{i}) \mathbf{u}$  is the intersection of  $\mathbf{u}$  with the plane containing the endpoints of  $\mathbf{f}$  and  $\mathbf{i}$ . The vector  $\mathbf{b}$  perpendicular to  $\mathbf{i} - \mathbf{v}$  to the tip of vector  $\mathbf{f}$  in Fig 2.5.2 is perpendicular to the plane of intersection of  $\mathbf{u}$  and  $\mathbf{i}$ , so it is parallel to  $\tilde{\mathbf{u}}\mathbf{i}$ . Further,  $\mathbf{b}$  has length  $\|\mathbf{b}\| = \|\mathbf{f} - \mathbf{v}\| \sin \chi = \|\mathbf{i} - \mathbf{v}\| \sin \chi$ . Equation (2.5.11) thus shows that  $\mathbf{b} = (\tilde{\mathbf{u}}\mathbf{i}) \sin \chi$ . Using Eq. (2.5.11), vector  $\mathbf{a}$  has length  $\|\mathbf{a}\| = \|\mathbf{f} - \mathbf{v}\| \cos \chi = \|\mathbf{i} - \mathbf{v}\| \cos \chi$  and since it is along vector  $\mathbf{i} - \mathbf{v}$ ,  $\mathbf{a} = (\mathbf{i} - \mathbf{v}) \cos \chi = (\mathbf{i} - (\mathbf{u}^T \mathbf{i}) \mathbf{u}) \cos \chi$ . With these results, and using the geometry of Fig. 2.5.2 to represent the vector sum  $\mathbf{v} + \mathbf{a} + \mathbf{b}$ ,

$$\mathbf{f} = \mathbf{v} + \mathbf{a} + \mathbf{b} = (\mathbf{u}^T \mathbf{i}) \mathbf{u} + (\mathbf{i} - (\mathbf{u}^T \mathbf{i}) \mathbf{u}) \cos \chi + (\tilde{\mathbf{u}}\mathbf{i}) \sin \chi \quad (2.5.12)$$

The same analysis can be used to relate  $\mathbf{g}$  and  $\mathbf{j}$  and  $\mathbf{h}$  and  $\mathbf{k}$ , obtaining

$$\begin{aligned} \mathbf{g} &= (\mathbf{u}^T \mathbf{j}) \mathbf{u} + (\mathbf{j} - (\mathbf{u}^T \mathbf{j}) \mathbf{u}) \cos \chi + (\tilde{\mathbf{u}}\mathbf{j}) \sin \chi \\ \mathbf{h} &= (\mathbf{u}^T \mathbf{k}) \mathbf{u} + (\mathbf{k} - (\mathbf{u}^T \mathbf{k}) \mathbf{u}) \cos \chi + (\tilde{\mathbf{u}}\mathbf{k}) \sin \chi \end{aligned} \quad (2.5.13)$$

Substituting from Eqs. (2.5.12) and (2.5.13) into Eq. (2.4.16) yields

$$\begin{aligned} \mathbf{A} &= [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}] \\ &= (1 - \cos \chi) [\mathbf{u}(\mathbf{u}^T \mathbf{i}) \quad \mathbf{u}(\mathbf{u}^T \mathbf{j}) \quad \mathbf{u}(\mathbf{u}^T \mathbf{k})] \\ &\quad + \cos \chi [\mathbf{i} \quad \mathbf{j} \quad \mathbf{k}] + \sin \chi \tilde{\mathbf{u}} [\mathbf{i} \quad \mathbf{j} \quad \mathbf{k}] \\ &= (1 - \cos \chi) \mathbf{u} \mathbf{u}^T [\mathbf{i} \quad \mathbf{j} \quad \mathbf{k}] + \cos \chi \mathbf{I} + \sin \chi \tilde{\mathbf{u}} \\ &= (1 - \cos \chi) \mathbf{u} \mathbf{u}^T + \cos \chi \mathbf{I} + \sin \chi \tilde{\mathbf{u}} \end{aligned} \quad (2.5.14)$$

**Example 2.5.1:** Consider the special case in which the vector about which rotation takes place is  $\mathbf{u} = [0 \quad 0 \quad 1]^T$ ; i.e., rotation occurs as the counterclockwise angle  $\phi$  about the  $z$  axis in Fig. 2.4.8. As shown in Example 2.4.2, the orientation transformation matrix is

$$\mathbf{A}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5.15)$$

Substituting  $\mathbf{u} = [0 \ 0 \ 1]^T$  for this case into Eq. (2.5.14) yields

$$\mathbf{A}(\chi) = \begin{bmatrix} \cos\chi & -\sin\chi & 0 \\ \sin\chi & \cos\chi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5.16)$$

which is consistent with Euler's theorem and the angle of rotation  $\chi$  in Fig. 2.5.2.

### 2.5.3 Euler Parameters

Using the trigonometric identities

$$\begin{aligned} 1 - \cos \theta &= 2\sin^2 \frac{\theta}{2} \\ \sin \theta &= 2\sin \frac{\theta}{2} \cos \frac{\theta}{2} \\ \cos \theta &= 2\cos^2 \frac{\theta}{2} - 1 \\ \cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} &= 1 \end{aligned} \quad (2.5.17)$$

the orientation transformation matrix of Eq. (2.5.14) may be written as

$$\begin{aligned} \mathbf{A} &= 2\sin^2 \frac{\theta}{2} \mathbf{u}\mathbf{u}^T + \left( 2\cos^2 \frac{\theta}{2} - 1 \right) \mathbf{I} + 2\sin \frac{\theta}{2} \cos \frac{\theta}{2} \tilde{\mathbf{u}} \\ &= 2\sin^2 \frac{\theta}{2} \mathbf{u}\mathbf{u}^T + \left( \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} \right) \mathbf{I} + 2\sin \frac{\theta}{2} \cos \frac{\theta}{2} \tilde{\mathbf{u}} \end{aligned} \quad (2.5.18)$$

Defining four *Euler parameters* as

$$\begin{aligned} e_0 &\equiv \cos \frac{\theta}{2} \\ \mathbf{e} &\equiv \sin \frac{\theta}{2} \mathbf{u} \end{aligned} \quad (2.5.19)$$

the orientation transformation matrix on the second line of Eq. (2.5.18) is

$$\mathbf{A} = (e_0^2 - \mathbf{e}^T \mathbf{e}) \mathbf{I} + 2\mathbf{e}\mathbf{e}^T + 2e_0 \tilde{\mathbf{e}} \quad (2.5.20)$$

where the first and second terms in Eq. (2.5.20) are symmetric matrices and the third term is skew-symmetric. In component form, Eq. (2.5.20) is

$$\mathbf{A} = \begin{bmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_1e_2 - e_0e_3) & 2(e_1e_3 + e_0e_2) \\ 2(e_1e_2 + e_0e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2e_3 - e_0e_1) \\ 2(e_1e_3 - e_0e_2) & 2(e_2e_3 + e_0e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{bmatrix} \quad (2.5.21)$$

The quadratic form of terms in  $\mathbf{A}$ , as functions of Euler parameters, will play a crucial role in obtaining important identities and in efficient computation.

If the form of the orientation transformation matrix on the first line of Eq. (2.5.18) had been used, the orientation transformation matrix could be written in terms of Euler parameters in the equivalent form

$$\bar{\mathbf{A}} = (2e_0^2 - 1)\mathbf{I} + 2(\mathbf{ee}^T + e_0\tilde{\mathbf{e}}) \quad (2.5.22)$$

This form of the orientation transformation matrix has been used in some textbooks (Wittenburg, 1977; Haug, 1989). It suffers, however, from requiring satisfaction of the Euler parameter normalization condition, in order that a key factorization of  $\bar{\mathbf{A}}$  as a product of two matrices that are linear in Euler parameters holds. This condition is not satisfied during iterative numerical solution procedures, hence precluding use of the factorization in important applications. It is interesting to note that the form of Eq. (2.5.20) is in fact used in some classical developments; e.g., (Goldstein, 1980).

The 4-vector of *Euler parameters* is defined as

$$\mathbf{p} = [e_0 \quad e_1 \quad e_2 \quad e_3]^T = [e_0 \quad \mathbf{e}^T]^T \quad (2.5.23)$$

Equation (2.5.19) and  $\mathbf{u}^T\mathbf{u} = 1$  yield the *Euler parameter normalization condition*,

$$\mathbf{p}^T\mathbf{p} = e_0^2 + \mathbf{e}^T\mathbf{e} = \cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} = \mathbf{u}^T\mathbf{u} = \cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} = 1$$

This relation and its time derivative yield the identities

$$\begin{aligned} \mathbf{p}^T\mathbf{p} &= 1 \\ \mathbf{p}^T\dot{\mathbf{p}} &= 0 \end{aligned} \quad (2.5.24)$$

MATLAB Code 2.5 outlined in Section 2.5.6 and contained in Appendix 2.B implements the Euler rotation definition of the orientation transformation matrix.

#### 2.5.4 Mapping from an Orientation Transformation Matrix to Euler Parameters

Given an orthogonal transformation matrix  $\mathbf{A}$ , it is often required to determine an associated set of Euler parameters; i.e., a vector  $\mathbf{p} = [e_0 \quad \mathbf{e}^T]^T$  such that  $\mathbf{A} = \mathbf{A}(\mathbf{p})$ . In particular, in defining an initial orientation for a body, unit vectors  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{h}$  can often be defined along the body fixed  $x'$ - $y'$ - $z'$  axes and the orientation transformation matrix for the body is given by Eq. (2.4.16),  $\mathbf{A} = [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}]$ . Euler parameters associated with this orthogonal matrix are then required to initiate a simulation.

The *trace* of an orthogonal transformation matrix  $\mathbf{A}$ , denoted  $\text{tr}\mathbf{A}$ , is defined as the sum of the diagonal elements of the matrix; i.e.,

$$\text{tr}\mathbf{A} \equiv a_{11} + a_{22} + a_{33} \quad (2.5.25)$$

Equating the sum of diagonal terms in the orientation transformation matrix of Eq. (2.5.21) to the trace of the given orthogonal transformation matrix  $\mathbf{A}$  and using the first identity of Eq. (2.5.24) ,

$$\text{tr}\mathbf{A} = 3e_0^2 - e_1^2 - e_2^2 - e_3^2 = 4e_0^2 - 1 \quad (2.5.26)$$

Thus,

$$e_0^2 = \frac{\text{tr}\mathbf{A} + 1}{4} \quad (2.5.27)$$

For any  $3 \times 3$  orthogonal matrix  $\mathbf{A}$ ,  $-1 \leq \text{tr}\mathbf{A} \leq 3$  (Bernstein, 2009), so  $0 \leq e_0^2 \leq 1$ .

To be precise, the method and theory of determining the transformation from  $\mathbf{A}$  to Euler parameters is stated as a theorem.

**Inverse Euler Parameter Mapping Theorem:** Let  $\mathbf{A}$  be an orthogonal matrix with  $|\mathbf{A}|=1$ . There exist locally unique Euler parameters  $\mathbf{p} = [e_0 \ e^T]^T$ ,  $e_0 = \cos(\chi/2)$  and  $\mathbf{e} = \sin(\chi/2)\mathbf{u}$ , that are determined by Eq. (2.5.20), as follows:

(a) if  $\mathbf{A}^T \neq \mathbf{A}$ ; i.e.,  $\mathbf{A}$  is not symmetric (by far the most common case),

$$\begin{aligned} e_0 &= \pm(1/2)\sqrt{\text{tr}\mathbf{A} + 1} \neq 0 \\ \mathbf{e} &= \pm\left(\frac{1}{2\sqrt{\text{tr}\mathbf{A} + 1}}\right) \begin{bmatrix} a_{32} - a_{23} \\ a_{13} - a_{31} \\ a_{21} - a_{12} \end{bmatrix} \end{aligned} \quad (2.5.28)$$

where the same choice of sign is made in both expressions.

(b) if  $\mathbf{A}^T = \mathbf{A}$ , either  $\mathbf{e} = \mathbf{0}$  or  $e_0 = 0$ . If  $e_0 = \pm(1/2)\sqrt{\text{tr}\mathbf{A} + 1} = \pm 1$ ,  $\mathbf{e} = \mathbf{0}$  and  $\mathbf{A} = \mathbf{I}$ .

Otherwise,  $e_0 = \pm(1/2)\sqrt{\text{tr}\mathbf{A} + 1} = 0$  and

$$\begin{aligned} |e_1| &= \sqrt{(a_{11} + 1)/2}, \quad e_1 = s_1 |e_1| \\ |e_2| &= \sqrt{(a_{22} + 1)/2}, \quad e_2 = s_2 |e_2| \\ |e_3| &= \sqrt{(a_{33} + 1)/2}, \quad e_3 = s_3 |e_3| \end{aligned} \quad (2.5.29)$$

where the signs  $s_i = \pm 1$  or  $\mp 1$  are selected to satisfy the conditions

$$\begin{aligned} a_{12} &= 2e_1 e_2 = 2s_1 s_2 |e_1| |e_2| \\ a_{13} &= 2e_1 e_3 = 2s_1 s_3 |e_1| |e_3| \\ a_{23} &= 2e_2 e_3 = 2s_2 s_3 |e_2| |e_3| \end{aligned} \quad (2.5.30)$$

as follows: (1) if  $|e_i| = 0$ , set  $s_i = 0$ ; (2) if  $|e_j| = \max_i(|e_i|)$ , set  $s_j = \pm 1$ ; and (3) for  $|e_k| \neq 0$ ,  $k \neq j$ , choose  $s_k$  to satisfy Eq. (2.5.30).

**Proof:** In case (a),  $\mathbf{A}$  is not symmetric, so the third term in Eq. (2.5.20) cannot be zero and  $e_0 \neq 0$ . The vector  $\mathbf{u}$  in Eq. (2.5.19) is the normalized solution of

$$\mathbf{A}\mathbf{u} = \mathbf{u} \quad (2.5.31)$$

Since  $\mathbf{A}^T = \mathbf{A}^{-1}$ , Eq. (2.5.31) may be multiplied on the left by  $\mathbf{A}^T$ , yielding

$$\mathbf{A}^T \mathbf{A} \mathbf{u} = \mathbf{u} = \mathbf{A}^T \mathbf{u} \quad (2.5.32)$$

Subtracting Eqs. (2.5.31) and (2.5.32),  $\mathbf{u}$  satisfies

$$(\mathbf{A} - \mathbf{A}^T) \mathbf{u} = \mathbf{0} \quad (2.5.33)$$

Since  $(\mathbf{A} - \mathbf{A}^T)^T = \mathbf{A}^T - \mathbf{A} = -(\mathbf{A} - \mathbf{A}^T)$ ,  $\mathbf{A} - \mathbf{A}^T$  is skew symmetric and can be written as

$$\mathbf{A} - \mathbf{A}^T = \begin{bmatrix} 0 & a_{12} - a_{21} & a_{13} - a_{31} \\ a_{21} - a_{12} & 0 & a_{23} - a_{32} \\ a_{31} - a_{13} & a_{32} - a_{23} & 0 \end{bmatrix} = \tilde{\mathbf{b}} \quad (2.5.34)$$

where

$$\mathbf{b} = \begin{bmatrix} a_{32} - a_{23} \\ a_{13} - a_{31} \\ a_{21} - a_{12} \end{bmatrix} \neq \mathbf{0} \quad (2.5.35)$$

Equation (2.5.33) may thus be written as

$$\tilde{\mathbf{b}}\mathbf{u} = \mathbf{0} \quad (2.5.36)$$

Since  $\mathbf{b} \neq \mathbf{0} \neq \mathbf{u}$ ,  $|\tilde{\mathbf{b}}\mathbf{u}| = |\mathbf{b}||\mathbf{u}|\sin(\mathbf{b}, \mathbf{u}) = 0$ , so  $\mathbf{u}$  and  $\mathbf{b}$  are parallel and  $\mathbf{u} = \alpha\mathbf{b}$  for some real  $\alpha$ .

Thus, a normalized solution of Eq. (2.5.31) is

$$\mathbf{u} = (1/\|\mathbf{b}\|)\mathbf{b} \quad (2.5.37)$$

Substituting  $\mathbf{e}_0$  and  $\mathbf{u}$  from Eqs. (2.5.27) and (2.5.37), Eq. (2.5.19) is written as

$$\begin{aligned} \mathbf{e}_0 &= \cos(\chi/2) = \pm\sqrt{\text{tr}\mathbf{A} + 1}/2 \\ \mathbf{e} &= \sin(\chi/2)(1/\|\mathbf{b}\|)\mathbf{b} \end{aligned} \quad (2.5.38)$$

Subtracting symmetrically placed off-diagonal terms in the given matrix  $\mathbf{A}$  and the matrix of Eq. (2.5.21), equating results, and substituting from Eqs. (2.5.38) and (2.5.35) yields

$$\begin{aligned} a_{12} - a_{21} &= -4\mathbf{e}_0\mathbf{e}_3 = 4\cos(\chi/2)\sin(\chi/2)(a_{12} - a_{21})/\|\mathbf{b}\| \\ a_{13} - a_{31} &= 4\mathbf{e}_0\mathbf{e}_2 = 4\cos(\chi/2)\sin(\chi/2)(a_{13} - a_{31})/\|\mathbf{b}\| \\ a_{23} - a_{32} &= -4\mathbf{e}_0\mathbf{e}_1 = 4\cos(\chi/2)\sin(\chi/2)(a_{23} - a_{32})/\|\mathbf{b}\| \end{aligned} \quad (2.5.39)$$

Since  $\mathbf{b} \neq \mathbf{0}$ , Eq. (2.5.39) implies  $4\cos(\chi/2)\sin(\chi/2)/\|\mathbf{b}\| = 1$  and, using Eq. (2.5.27),

$$\sin(\chi/2) = \|\mathbf{b}\|/(4\cos(\chi/2)) = \pm\|\mathbf{b}\|/(2\sqrt{\text{tr}\mathbf{A} + 1}) \quad (2.5.40)$$

Substituting this result into Eq. (2.5.38),

$$\begin{aligned} e_0 &= \pm\sqrt{\text{tr}\mathbf{A}+1}/2 \\ \mathbf{e} &= \pm(1/(2\sqrt{\text{tr}\mathbf{A}+1}))\mathbf{b} \end{aligned} \quad (2.5.41)$$

which is Eq. (2.5.28).

In case (b),  $\mathbf{A}^T = \mathbf{A}$ , so the third term in Eq. (2.5.20) must be zero and either  $e_0 = 0$  or  $\mathbf{e} = \mathbf{0}$ . If  $\mathbf{e} = \mathbf{0}$ ,  $e_0 = \pm 1$  and, since all off diagonal terms are zero, Eq. (2.5.21) shows that  $\mathbf{A} = \mathbf{I}$  and  $\mathbf{p} = \pm[1 \ 0 \ 0]^T$ .

In case (c), if  $\mathbf{A}^T = \mathbf{A}$  and  $e_0 = 0$ , equating diagonal terms of the given matrix and the matrix of Eq. (2.5.21) yields  $a_{11} = 2e_1^2 - 1$ ,  $a_{22} = 2e_2^2 - 1$ , and  $a_{33} = 2e_3^2 - 1$ . This yields the magnitudes of Euler parameters in Eq. (2.5.29). Equating terms above the diagonal in the given matrix and the matrix of Eq. (2.5.21) yields the conditions of Eq. (2.5.30) that determine the signs of the Euler parameters.

In all three cases, choosing the plus sign yields Euler parameter vector  $\mathbf{p}^+$  and the minus sign yields  $\mathbf{p}^-$ ; i.e.,  $\mathbf{p}^- = -\mathbf{p}^+$ . Thus,  $\|\mathbf{p}^+ - \mathbf{p}^-\| = \|2\mathbf{p}^+\| = 2\sqrt{\mathbf{p}^{+T}\mathbf{p}^+} = 2$  and the alternative values of Euler parameters determined by the given orientation transformation matrix  $\mathbf{A}$  are far apart, so the solution is locally unique. This completes the proof of the theorem.

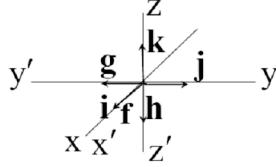
MATLAB Code 2.5 that is outlined in Section 2.5.6 and contained in Appendix 2.B, implements the inverse Euler parameter mapping theorem.

The large distance between the alternate Euler parameter solutions  $\mathbf{p}^-$  and  $\mathbf{p}^+$  is important in time-based analysis of the motion of mechanical systems, since Euler parameters are determined on a finely spaced discrete time grid. Thus, it is not possible to jump from one of the alternatives to the other when progressing from one grid point to the next. For practical purposes, either selection of sign is acceptable to begin analysis on the time grid, and no singularity is encountered (Stuelpnagel, 1964). Hopf proved in 1940 that a five parameter representation of  $\mathbf{A}$  is required to obtain a globally valid one-to-one representation of orientation. While avoiding the foregoing alternative representation by  $\mathbf{p}^-$  or  $\mathbf{p}^+$ , the extra cost of introducing five coordinates that must satisfy two constraints would be oppressive.

### Example 2.5.2: The symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

has determinant  $|\mathbf{A}| = 1$  and is easily verified to be orthogonal. Since  $\text{Tr}\mathbf{A} = -1$ ,  $e_0 = 0 = \cos(\theta/2)$ , so  $\theta = \pm 90^\circ$ . From Eq. (2.5.29),  $|e_1| = 1$ ,  $|e_2| = 0$ , and  $|e_3| = 0$ . Thus  $\mathbf{e} = \pm[1 \ 0 \ 0]^T = \pm\mathbf{i}$  and  $\mathbf{u} = \mathbf{i}$ . As shown, this represents a rotation of the primed frame by angle  $90^\circ$  about the x-axis.



Consider the perturbation of Euler parameters to  $e_0 = \epsilon$ ,  $e_1 = \sqrt{1 - \epsilon^2}$ ,  $e_2 = 0$ , and  $e_3 = 0$ . The normalization constraint is satisfied and from Eq. (2.5.21),

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1+2^{-2} & -2\sqrt{1-\epsilon^2} \\ 0 & 2\sqrt{1-\epsilon^2} & -1+2^{-2} \end{bmatrix}$$

A direct manipulation shows that this matrix is orthogonal, has determinant 1, and it is clearly not symmetric. Its trace is  $\text{tr}\mathbf{A} = -1 + 4^{-2}$ , so from Eq. (2.5.28),  $\epsilon_0 = \pm\sqrt{4^{-2}}/2 = \pm\epsilon$  and

$$\mathbf{e} = \pm\left(\frac{1}{2\sqrt{4^{-2}}}\right) \begin{bmatrix} 4\sqrt{1-\epsilon^2} \\ 0 \\ 0 \end{bmatrix} = \pm \begin{bmatrix} \sqrt{1-\epsilon^2} \\ 0 \\ 0 \end{bmatrix} \quad \text{which is consistent with the perturbed Euler parameters}$$

selected. This demonstrates consistency of the alternatives that exist in the theorem and continuity of the orientation transformation matrix as a function of Euler parameters, even in the neighborhood of transition from a symmetric matrix to a nonsymmetric matrix.

### 2.5.5 Point Definition of an Orthogonal Reference Frame

One method of defining an orthogonal  $x'-y'-z'$  reference frame, relative to the global  $x-y-z$  frame, is to define point O as the origin of the  $x'-y'-z'$  frame with vector  $\mathbf{r}^O$ , a point P on the  $x'$  axis different from point O with vector  $\mathbf{r}^P$ , and a point Q in the  $x'-y'$  plane that is not on the  $x'$  axis with vector  $\mathbf{r}^Q$ , as shown in Fig. 2.5.3. The unit vector along the positive  $x'$  axis, represented in the global frame, is  $\mathbf{f} = (1/\|\mathbf{r}^P - \mathbf{r}^O\|)(\mathbf{r}^P - \mathbf{r}^O)$ . Since  $\mathbf{r}^Q$  is in the  $x'-y'$  plane and  $\mathbf{r}^Q - \mathbf{r}^O \neq \mathbf{0}$ ,  $\tilde{\mathbf{f}}(\mathbf{r}^Q - \mathbf{r}^O) \neq \mathbf{0}$  is orthogonal to the  $x'-y'$  plane, hence it is along the positive  $z'$  axis. The unit vector along that axis is thus  $\mathbf{h} = (1/\|\tilde{\mathbf{f}}(\mathbf{r}^Q - \mathbf{r}^O)\|)\tilde{\mathbf{f}}(\mathbf{r}^Q - \mathbf{r}^O)$ . Finally, the unit vector along the positive  $y'$  axis is  $\mathbf{g} = -\tilde{\mathbf{f}}\mathbf{h}$ . The transformation matrix from the  $x'-y'-z'$  frame to the  $x-y-z$  frame is, from Eq. (2.4.16),  $\mathbf{A} = [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}]$ .

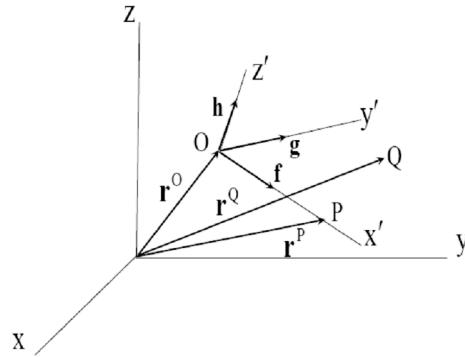


Figure 2.5.3 Points that Define an Orthogonal  $x'$ - $y'$ - $z'$  Frame

MATLAB Code 2.5 in Appendix 2.B implements the point definition approach.

**Example 2.5.3:** As an example of using the foregoing definition of an orthogonal  $x'$ - $y'$ - $z'$  reference frame and the inverse Euler parameter mapping theorem to determine Euler parameters that characterize its orientation, let  $\mathbf{r}^O = \mathbf{0}$ ,  $\mathbf{r}^P = [1 \ 1 \ 1]^T$  and  $\mathbf{r}^Q = [-1 \ -1 \ 0]^T$ . Then,  $\mathbf{f} = (1/\sqrt{3})[1 \ 1 \ 1]^T$ ,  $\tilde{\mathbf{f}}\mathbf{r}^Q = (1/\sqrt{3})[1 \ -1 \ 0]^T$ ,  $\mathbf{h} = (1/\sqrt{2})[1 \ -1 \ 0]^T$ , and  $\mathbf{g} = -\tilde{\mathbf{f}}\mathbf{h} = (1/\sqrt{6})[-1 \ -1 \ 2]^T$ . The orthogonal transformation matrix from the  $x'$ - $y'$ - $z'$  frame to the  $x$ - $y$ - $z$  frame is

$$\mathbf{A} = \begin{bmatrix} 1/\sqrt{3} & -1/\sqrt{6} & 1/\sqrt{2} \\ 1/\sqrt{3} & -1/\sqrt{6} & -1/\sqrt{2} \\ 1/\sqrt{3} & 2/\sqrt{6} & 0 \end{bmatrix}$$

Since  $\mathbf{A}$  is not symmetric and  $\text{tr}\mathbf{A} = 1/\sqrt{3} - 1/\sqrt{6}$ , Eq. (2.5.28) yields

$$\mathbf{e}_0 = \pm(1/2)\sqrt{1/\sqrt{3} - 1/\sqrt{6} + 1}$$

$$\mathbf{e} = \pm \left( \frac{1}{2\sqrt{1/\sqrt{3} - 1/\sqrt{6} + 1}} \right) \begin{bmatrix} 2/\sqrt{6} + 1/\sqrt{2} \\ 1/\sqrt{2} - 1/\sqrt{3} \\ 1/\sqrt{3} + 1/\sqrt{6} \end{bmatrix}$$

While it would be virtually impossible to use purely geometric analysis to specify the unit vector  $\mathbf{u}$  and rotational angle  $\chi$  in Euler's theorem that define the orientation of the foregoing  $x'$ - $y'$ - $z'$  frame, analytical results using the inverse mapping theorem are easily obtained.

## 2.5.6 Code 2.5, Point Definition of A and Solution for p with Given A

MATLAB Code 2.5 for point definition of an orientation transformation matrix and for computation of Euler parameters for a given orientation transformation matrix is presented in Fig. 2.5.4. Functions AT, atil, and partA are defined in the code.

```

1 %AA_Spatial_Orientation_Transformation_Matrix_and_Inverse for p
3 Part=[]; %Part=1, Define orthogonal transformation matrix
4 %Part=2, Compute p for given orthogonal transformation matrix A
7 if Part==1 %Define orthogonal transformation matrix
9 mode=[]; %mode=1, enter A; mode=2, point definition of A;
10 %mode=3, Euler's Theorem
12 if mode==1
13 A=[,,];
14 end
16 if mode==2
17 %Define orthogonal transformation matrix A using point definition of
18 %Section 2.5.5.
20 rO=[,]; %Enter vector rO to origin
21 rP=[,]; %Enter vector rP to point P
22 rQ=[,]; %Enter vector rQ to point Q
24 f=(1/norm(rP-rO))*(rP-rO);
25 h=(1/norm(ati(f)*(rQ-rO)))*ati(f)*(rQ-rO);
26 g=-ati(f)*h;
28 A=[f,g,h];
30 end
32 if mode==3 %Define orthogonal transformation matrix A using Euler's
33 %Theorem of Section 2.5.3.
35 %Unit vector u about which rotation angle chi brings x-y-z frame
36 %into coincidence with x'-y'-z' frame
38 ubar=[,]; %Enter vector ubar about which rotation occurs,
39 %not necessarily normalized
41 chi=[]; %Enter angle chi of rotation
43 u=(1/norm(ubar))*ubar;
44 e0=cos(chi/2);
45 e=sin(chi/2)*u;
46 A=(e0^2-e'*e)*eye(3)+2*e'*e+e0*ati(e);
47 end
49 end
51 %%%%%%%%%%%%%%
52 if Part==2 %Given orthogonal A, compute Euler parameter vector p in
53 %Section 2.5.4 so A=A(p)
55 [a11,a12,a13,a21,a22,a23,a31,a32,a33] = partA(A);
C:\Users\echau\Docume...\\AA_Spatial_Orientation_Transformation_Matrix_and_Inverse.m Page 2
57 if norm(A'-A)>0
58 trA=a11+a22+a33;
59 e0=0.5*sqrt(trA+1);
60 e=(1/(2*sqrt(trA+1)))*[a32-a23;a13-a31;a21-a12];
61 end
63 if norm(A'-A)==0
64 e0=1;
65 e=zeros(3,1);
66 end
68 p=[e0;e];
70 end

```

Figure 2.5.4 Code 2.5, Point Definition of **A** and Solution for **p** with Given **A**

Properties of the spatial orthogonal orientation transformation matrix lead to Euler's theorem that states that every orientation of a body-fixed reference frame, hence of a rigid body in space, can be realized by a rotation about a unit vector in space that brings the global reference frame into coincidence with the body-fixed frame. This leads to definition of a four-dimensional vector of Euler parameters that characterize orientation of the body, but must satisfy a normalization condition, thus three orientation degrees of freedom. The orientation transformation matrix is written as a quadratic function of Euler parameters, a property that enables practical and efficient computation in mechanical system kinematics and dynamics.

A classical set of three Euler angles that define orientation, but suffer an unavoidable singularity, is defined in Appendix 2.A and contrasted with Euler parameters.

Relations are derived that define an orthogonal reference frame and represent its orientation transformation matrix in terms of vectors that define three points in space. Conversely, relations are derived that determine locally unique Euler parameters for a specified orthogonal transformation matrix. Both relations are implemented in MATLAB computer code 2.5 in Appendix 2.B that is summarized in Section 2.5.6.

### **Key Formulas**

$$e_0 \equiv \cos \frac{\theta}{2} \quad \mathbf{e} \equiv \sin \frac{\theta}{2} \mathbf{u} \quad (2.5.19)$$

$$\mathbf{A} = \left( e_0^2 - \mathbf{e}^T \mathbf{e} \right) \mathbf{I} + 2\mathbf{e}\mathbf{e}^T + 2e_0 \tilde{\mathbf{e}} \quad (2.5.20)$$

$$\mathbf{A} = \begin{bmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_1 e_2 - e_0 e_3) & 2(e_1 e_3 + e_0 e_2) \\ 2(e_1 e_2 + e_0 e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2 e_3 - e_0 e_1) \\ 2(e_1 e_3 - e_0 e_2) & 2(e_2 e_3 + e_0 e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{bmatrix} \quad (2.5.21)$$

$$\mathbf{p} = [e_0 \quad e_1 \quad e_2 \quad e_3]^T = [e_0 \quad \mathbf{e}^T]^T \quad \mathbf{p}^T \mathbf{p} = 1 \quad \mathbf{p}^T \dot{\mathbf{p}} = 0 \quad (2.5.23) \quad (2.5.24)$$

$$e_0 = \pm \sqrt{\text{tr} \mathbf{A} + 1} / 2 \quad \mathbf{b} = \begin{bmatrix} a_{32} - a_{23} \\ a_{13} - a_{31} \\ a_{21} - a_{12} \end{bmatrix} \quad (2.5.28)$$

$$\mathbf{e} = \pm \left( 1 / (2\sqrt{\text{tr} \mathbf{A} + 1}) \right) \mathbf{b}$$

## 2.6 Euler Parameter Identities and Derivatives

A number of identities and derivative relationships involving Euler parameters follow from the quadratic form of terms in the orientation transformation matrix of Eq. (2.5.21), as functions of Euler parameters. Relations based on these properties that are presented in this section are critical in deriving and effectively implementing often complicated equations of multibody kinematics and dynamics. Derivatives of kinematic constraint equations up to third order arise in representing the geometry of nonlinear kinematic and kinetic equations that must be incorporated into the equations of multibody dynamics and used in their numerical solution.

The identities and derivative relations presented in this section are used many times in deriving equations of motion and, more important, in deriving and implementing numerical methods for their computer solution. These results make practical what would otherwise be hopelessly complex expressions required in computer programming. The reader is encouraged to study the approach presented to obtain identities, which is based on differential calculus at the sophomore level. To avoid the need to memorize results and for ease of reference, key formulas derived are summarized at the end of the section. This is similar to the common practice in differential calculus of tabulating derivatives of large numbers of transcendental functions in book appendices, to avoid the need to memorize them.

MATLAB implementation of identities and derivatives is presented in Section 2.B.4 of Appendix 2.B and in Code 2.6 of software that accompanies the text.

### 2.6.1 Euler Parameter Identities

Define  $3 \times 4$  matrices that are linear functions of Euler parameters,

$$\mathbf{E}(\mathbf{p}) \equiv [-\mathbf{e} \quad \tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}] \quad (2.6.1)$$

$$\mathbf{G}(\mathbf{p}) \equiv [-\mathbf{e} \quad -\tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}] \quad (2.6.2)$$

Multiplication and manipulation with vector identities of Section 2.1 verifies that

$$\begin{aligned} \mathbf{E}(\mathbf{p})\mathbf{G}(\mathbf{p})^T &= [-\mathbf{e} \quad \tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}] \begin{bmatrix} -\mathbf{e}^T \\ \tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I} \end{bmatrix} = \mathbf{e}\mathbf{e}^T + \tilde{\mathbf{e}}\tilde{\mathbf{e}} + 2\mathbf{e}_0\tilde{\mathbf{e}} + \mathbf{e}_0^2\mathbf{I} \\ &= \mathbf{e}\mathbf{e}^T + \mathbf{e}\mathbf{e}^T - \mathbf{e}^T\mathbf{e}\mathbf{I} + 2\mathbf{e}_0\tilde{\mathbf{e}} + \mathbf{e}_0^2\mathbf{I} = (\mathbf{e}_0^2 - \mathbf{e}^T\mathbf{e})\mathbf{I} + 2\mathbf{e}\mathbf{e}^T + 2\mathbf{e}_0\tilde{\mathbf{e}} \end{aligned} \quad (2.6.3)$$

Note that the right sides of Eqs. (2.5.20) and (2.6.3) are identical; i.e.,

$$\mathbf{A}(\mathbf{p}) = \mathbf{E}(\mathbf{p})\mathbf{G}^T(\mathbf{p}) \quad (2.6.4)$$

Two observations are important at this point. First, Eq. (2.6.4) was obtained without the requirement that the vector  $\mathbf{p}$  is normalized. Second, from Eqs. (2.6.3) and (2.6.4), it is observed that all the entries in the matrix  $\mathbf{A}(\mathbf{p})$  are quadratic in components of  $\mathbf{p}$ , whereas each of the factors on the right of Eq. (2.6.4) is a matrix whose entries are linear in  $\mathbf{p}$ .

Some of the relations presented in Section 2.5 require that the vector of Euler parameters satisfy the normalization condition  $\mathbf{p}^T\mathbf{p} = 1$ . They should therefore not be used in iterative numerical computations when  $\mathbf{p}^T\mathbf{p} \neq 1$ . A number of useful identities that involve Euler parameters can be obtained without requiring satisfaction of the normalization condition; e.g.,

Eq. (2.6.4). These identities are valuable in developing relationships for derivatives of kinematic expressions with respect to Euler parameters that are used in iterative computation, during which the Euler parameter normalization condition may not be satisfied. To signify whether Euler parameter normalization is required for validity of identities derived in this section, only equations with bold equation numbers in the Key Formulas section at the end of the section require Euler parameter normalization.

Carrying out the multiplications indicated shows that

$$\mathbf{E}(\mathbf{p})\mathbf{p} = [-\mathbf{e} \quad \tilde{\mathbf{e}} + e_0 \mathbf{I}] \begin{bmatrix} e_0 \\ \mathbf{e} \end{bmatrix} = -e_0 \mathbf{e} + \tilde{\mathbf{e}} \mathbf{e} + e_0 \mathbf{e} = \mathbf{0} \quad (2.6.5)$$

where no Euler parameter normalization condition was required. Similarly,

$$\mathbf{G}(\mathbf{p})\mathbf{p} = \mathbf{0} \quad (2.6.6)$$

Carrying out the multiplications indicated, for any  $\mathbf{p} \in \mathbb{R}^4$ , yields

$$\begin{aligned} \mathbf{E}(\mathbf{p})\mathbf{E}^T(\mathbf{p}) &= [-\mathbf{e} \quad \tilde{\mathbf{e}} + e_0 \mathbf{I}] \begin{bmatrix} -\mathbf{e}^T \\ -\tilde{\mathbf{e}} + e_0 \mathbf{I} \end{bmatrix} \\ &= \mathbf{e}\mathbf{e}^T - \tilde{\mathbf{e}}\tilde{\mathbf{e}} + e_0^2 \mathbf{I} = \mathbf{e}\mathbf{e}^T - \mathbf{e}\mathbf{e}^T + \mathbf{e}^T \mathbf{e} \mathbf{I} + e_0^2 \mathbf{I} = \mathbf{p}^T \mathbf{p} \mathbf{I} \end{aligned} \quad (2.6.7)$$

Only if the Euler parameter normalization condition is satisfied is  $\mathbf{E}(\mathbf{p})\mathbf{E}^T(\mathbf{p}) = \mathbf{I}$ . Similarly, for any vector  $\mathbf{p} \in \mathbb{R}^4$ ,

$$\mathbf{G}(\mathbf{p})\mathbf{G}^T(\mathbf{p}) = \mathbf{p}^T \mathbf{p} \mathbf{I} \quad (2.6.8)$$

and only if  $\mathbf{p}$  is normalized is  $\mathbf{G}(\mathbf{p})\mathbf{G}^T(\mathbf{p}) = \mathbf{I}$ .

Carrying out the multiplications indicated yields

$$\begin{aligned} \mathbf{E}^T(\mathbf{p})\mathbf{E}(\mathbf{p}) &= [-\mathbf{e} \quad \tilde{\mathbf{e}} + e_0 \mathbf{I}]^T [-\mathbf{e} \quad \tilde{\mathbf{e}} + e_0 \mathbf{I}] = \begin{bmatrix} \mathbf{e}^T \mathbf{e} & -e_0 \mathbf{e}^T \\ -e_0 \mathbf{e} & -\tilde{\mathbf{e}}\tilde{\mathbf{e}} + e_0^2 \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{e}^T \mathbf{e} + e_0^2 - e_0^2 & -e_0 \mathbf{e}^T \\ -e_0 \mathbf{e} & -\mathbf{e}\mathbf{e}^T + \mathbf{e}^T \mathbf{e} \mathbf{I} + e_0^2 \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^T \mathbf{p} & \mathbf{0} \\ \mathbf{0} & \mathbf{p}^T \mathbf{p} \mathbf{I} \end{bmatrix} - \begin{bmatrix} e_0^2 & e_0 \mathbf{e}^T \\ e_0 \mathbf{e} & \mathbf{e}\mathbf{e}^T \end{bmatrix} \\ &= \mathbf{p}^T \mathbf{p} \mathbf{I}_4 - \mathbf{p} \mathbf{p}^T \end{aligned} \quad (2.6.9)$$

Only if  $\mathbf{p}$  is normalized is  $\mathbf{E}^T(\mathbf{p})\mathbf{E}(\mathbf{p}) = \mathbf{I}_4 - \mathbf{p} \mathbf{p}^T$ . Using the same form of calculation,

$$\mathbf{G}^T(\mathbf{p})\mathbf{G}(\mathbf{p}) = \mathbf{p}^T \mathbf{p} \mathbf{I}_4 - \mathbf{p} \mathbf{p}^T \quad (2.6.10)$$

and only if  $\mathbf{p}$  is normalized is  $\mathbf{G}^T(\mathbf{p})\mathbf{G}(\mathbf{p}) = \mathbf{I}_4 - \mathbf{p} \mathbf{p}^T$ .

A direct manipulation, using Eqs. (2.6.4), (2.6.7), (2.6.8), (2.6.9), and (2.6.10), suppressing arguments, shows that

$$\mathbf{A}\mathbf{A}^T = \mathbf{E}\mathbf{G}^T\mathbf{G}\mathbf{E}^T = \mathbf{E}(\mathbf{p}^T \mathbf{p} \mathbf{I} - \mathbf{p} \mathbf{p}^T) \mathbf{E}^T = \mathbf{p}^T \mathbf{p} \mathbf{E} \mathbf{E}^T = (\mathbf{p}^T \mathbf{p})^2 \mathbf{I} \quad (2.6.11)$$

and

$$\mathbf{A}^T \mathbf{A} = \mathbf{G} \mathbf{E}^T \mathbf{E} \mathbf{G}^T = \mathbf{G} (\mathbf{p}^T \mathbf{p} \mathbf{I} - \mathbf{p} \mathbf{p}^T) \mathbf{G}^T = \mathbf{p}^T \mathbf{p} \mathbf{G} \mathbf{G}^T = (\mathbf{p}^T \mathbf{p})^2 \mathbf{I} \quad (2.6.12)$$

Thus, only if  $\mathbf{p}^T \mathbf{p} = 1$ ,  $\mathbf{A}$  is *orthogonal*; i.e.,  $\mathbf{A} \mathbf{A}^T = \mathbf{I}$  and  $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ , in which case,  $\mathbf{A}^T = \mathbf{A}^{-1}$ . Nevertheless,  $\mathbf{A} \mathbf{A}^T = \mathbf{A}^T \mathbf{A}$ , even if  $\mathbf{p}$  is not normalized.

Carrying out the multiplications indicated, manipulating, and using Eq. (2.1.24),

$$\begin{aligned} \mathbf{E}(\mathbf{p}_i) \mathbf{p}_j &= \begin{bmatrix} -\mathbf{e}_i & \tilde{\mathbf{e}}_i + \mathbf{e}_{0_i} \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{e}_{0_j} \\ \mathbf{e}_j \end{bmatrix} = -\mathbf{e}_{0_j} \mathbf{e}_i + \tilde{\mathbf{e}}_i \mathbf{e}_j + \mathbf{e}_{0_i} \mathbf{e}_j \\ &= -(-\mathbf{e}_{0_i} \mathbf{e}_j + \tilde{\mathbf{e}}_j \mathbf{e}_i + \mathbf{e}_{0_j} \mathbf{e}_i) = -\begin{bmatrix} -\mathbf{e}_j & \tilde{\mathbf{e}}_j + \mathbf{e}_{0_j} \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{e}_{0_i} \\ \mathbf{e}_i \end{bmatrix} \end{aligned} \quad (2.6.13)$$

which, for any  $\mathbf{p}_i$  and  $\mathbf{p}_j$  in  $\mathbb{R}^4$  is simply

$$\mathbf{E}(\mathbf{p}_i) \mathbf{p}_j = -\mathbf{E}(\mathbf{p}_j) \mathbf{p}_i \quad (2.6.14)$$

A similar manipulation yields the identity

$$\mathbf{G}(\mathbf{p}_i) \mathbf{p}_j = -\mathbf{G}(\mathbf{p}_j) \mathbf{p}_i \quad (2.6.15)$$

Direct manipulation, using Eqs. (2.1.24) and (2.1.32), yields

$$\begin{aligned} \mathbf{E}(\mathbf{p}_i) \mathbf{G}^T(\mathbf{p}_j) &= \begin{bmatrix} -\mathbf{e}_i & \tilde{\mathbf{e}}_i + \mathbf{e}_{0_i} \mathbf{I} \end{bmatrix} \begin{bmatrix} -\mathbf{e}_j & -\tilde{\mathbf{e}}_j + \mathbf{e}_{0_j} \mathbf{I} \end{bmatrix}^T \\ &= \mathbf{e}_i \mathbf{e}_j^T + \tilde{\mathbf{e}}_i \tilde{\mathbf{e}}_j + \mathbf{e}_{0_j} \tilde{\mathbf{e}}_i + \mathbf{e}_{0_i} \tilde{\mathbf{e}}_j + \mathbf{e}_{0_i} \mathbf{e}_{0_j} \mathbf{I} \\ &= \mathbf{e}_j \mathbf{e}_i^T + \tilde{\mathbf{e}}_j \tilde{\mathbf{e}}_i + \mathbf{e}_{0_i} \tilde{\mathbf{e}}_j + \mathbf{e}_{0_j} \tilde{\mathbf{e}}_i + \mathbf{e}_{0_j} \mathbf{e}_{0_i} \mathbf{I} \end{aligned} \quad (2.6.16)$$

Since the last two lines of Eq. (2.6.16) are identical, with the exception that indices  $i$  and  $j$  are interchanged, for any  $\mathbf{p}_i$  and  $\mathbf{p}_j$  in  $\mathbb{R}^4$ ,

$$\mathbf{E}(\mathbf{p}_i) \mathbf{G}^T(\mathbf{p}_j) = \mathbf{E}(\mathbf{p}_j) \mathbf{G}^T(\mathbf{p}_i) \quad (2.6.17)$$

Similarly, for any  $\mathbf{p}_i$  and  $\mathbf{p}_j$  in  $\mathbb{R}^4$ ,

$$\mathbf{G}(\mathbf{p}_i) \mathbf{E}^T(\mathbf{p}_j) = \mathbf{G}(\mathbf{p}_j) \mathbf{E}^T(\mathbf{p}_i) \quad (2.6.18)$$

Define  $4 \times 4$  matrices

$$\mathbf{R}(\mathbf{a}) \equiv \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & \tilde{\mathbf{a}} \end{bmatrix} \quad (2.6.19)$$

$$\mathbf{T}(\mathbf{a}) \equiv \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & -\tilde{\mathbf{a}} \end{bmatrix} \quad (2.6.20)$$

where  $\mathbf{a}$  is a  $3 \times 1$  column vector. Direct manipulation yields

$$\begin{aligned}\mathbf{R}(\mathbf{a})\mathbf{p} &= \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & \tilde{\mathbf{a}} \end{bmatrix} \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} -\mathbf{a}^T \mathbf{e} \\ \mathbf{e}_0 \mathbf{a} + \tilde{\mathbf{a}} \mathbf{e} \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{e}^T \mathbf{a} \\ \mathbf{e}_0 \mathbf{a} - \tilde{\mathbf{a}} \mathbf{e} \end{bmatrix} = \begin{bmatrix} -\mathbf{e}^T \\ \mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{a}} \end{bmatrix} \mathbf{a} = \begin{bmatrix} -\mathbf{e} & \tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I} \end{bmatrix}^T \mathbf{a}\end{aligned}\quad (2.6.21)$$

where no use was made of Euler parameter normalization. From the definition of Eq. (2.6.1), this is the identity

$$\mathbf{R}(\mathbf{a})\mathbf{p} = \mathbf{E}^T(\mathbf{p})\mathbf{a} \quad (2.6.22)$$

A similar manipulation yields the identity

$$\mathbf{T}(\mathbf{a})\mathbf{p} = \mathbf{G}^T(\mathbf{p})\mathbf{a} \quad (2.6.23)$$

## 2.6.2 Euler Parameter Derivatives

Throughout this section,  $\mathbf{a}' \in \mathbb{R}^3$  and  $\mathbf{p}, \gamma \in \mathbb{R}^4$ , where  $\mathbf{p}$  and  $\gamma$  are not necessarily normalized and  $\mathbf{a}'$  and  $\gamma$  do not depend on  $\mathbf{p}$ . Expanding the product indicated and taking the derivative with respect to  $\mathbf{p}$ ,

$$\begin{aligned}(\mathbf{A}(\mathbf{p})\mathbf{a}')_p &= \left[ (\mathbf{e}_0^2 - \mathbf{e}^T \mathbf{e}) \mathbf{a}' + 2(\mathbf{e} \mathbf{e}^T + \mathbf{e}_0 \tilde{\mathbf{e}}) \mathbf{a}' \right]_p \\ &= \left[ 2\mathbf{e}_0 \mathbf{a}' + 2\tilde{\mathbf{e}} \mathbf{a}' \quad -2\tilde{\mathbf{e}} \mathbf{a}' + 2\mathbf{e} \mathbf{a}'^T - 2\mathbf{e}_0 \tilde{\mathbf{a}}' \right] \\ &= 2 \left[ (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \mathbf{a}' \quad \mathbf{e} \mathbf{a}'^T - (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \tilde{\mathbf{a}}' \right]\end{aligned}\quad (2.6.24)$$

where, using Eqs. (2.1.24) and (2.1.27),  $(\mathbf{e}^T \mathbf{e} \mathbf{a}')_e = (\mathbf{a}' \mathbf{e}^T \mathbf{e})_e = \mathbf{a}' (\mathbf{e}^T \mathbf{e})_e = 2\mathbf{a}' \mathbf{e}^T$  and  $(\mathbf{e} \mathbf{e}^T \mathbf{a}')_e = \{\mathbf{e} (\mathbf{a}'^T \mathbf{e})\}_e = \mathbf{e} \mathbf{a}'^T + (\mathbf{a}'^T \mathbf{e}) \mathbf{I}$ . Using Eq. (2.1.29),  $-2\mathbf{a}' \mathbf{e}^T + 2\mathbf{e}^T \mathbf{a}' \mathbf{I} = -2\tilde{\mathbf{e}} \mathbf{a}'$ . Defining

$$\mathbf{B}(\mathbf{p}, \mathbf{a}') \equiv 2 \left[ (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \mathbf{a}' \quad \mathbf{e} \mathbf{a}'^T - (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \tilde{\mathbf{a}}' \right] = 2\mathbf{E}(\mathbf{p})\mathbf{T}(\mathbf{a}') \quad (2.6.25)$$

where the second equality on the right is verified by direct manipulation, the following derivative identity that does not require normalization of  $\mathbf{p}$  is obtained from Eq. (2.6.24):

$$(\mathbf{A}(\mathbf{p})\mathbf{a}')_p = \mathbf{B}(\mathbf{p}, \mathbf{a}') \quad (2.6.26)$$

Expanding the product indicated, using Eqs. (2.1.24) and (2.1.32),

$$\begin{aligned}\mathbf{B}(\mathbf{p}_i, \mathbf{a}')_{\mathbf{p}_j} &= 2 \left[ (\mathbf{e}_{0_i} \mathbf{I} + \tilde{\mathbf{e}}_i) \mathbf{a}' \quad \mathbf{e}_i \mathbf{a}'^T - (\mathbf{e}_{0_i} \mathbf{I} + \tilde{\mathbf{e}}_i) \tilde{\mathbf{a}}' \right]_{\mathbf{p}_j} \\ &= 2 \left[ (\mathbf{e}_{0_i} \mathbf{e}_{0_j} \mathbf{I} + \mathbf{e}_{0_j} \tilde{\mathbf{e}}_i) \mathbf{a}' + \mathbf{e}_i \mathbf{a}'^T \mathbf{e}_j + \mathbf{e}_{0_i} \tilde{\mathbf{e}}_j \mathbf{a}' + \tilde{\mathbf{e}}_i \tilde{\mathbf{e}}_j \mathbf{a}' \right] \\ &= 2 \left[ (\mathbf{e}_{0_j} \mathbf{e}_{0_i} \mathbf{I} + \mathbf{e}_{0_i} \tilde{\mathbf{e}}_j) \mathbf{a}' + \mathbf{e}_i \mathbf{e}_j^T \mathbf{a}' + \mathbf{e}_{0_i} \tilde{\mathbf{e}}_j \mathbf{a}' + \tilde{\mathbf{e}}_i \tilde{\mathbf{e}}_j \mathbf{a}' \right] \\ &= 2 \left[ (\mathbf{e}_{0_j} \mathbf{e}_{0_i} \mathbf{I} + \mathbf{e}_{0_i} \tilde{\mathbf{e}}_j) \mathbf{a}' + \mathbf{e}_j \mathbf{e}_i^T \mathbf{a}' + \mathbf{e}_{0_j} \tilde{\mathbf{e}}_i \mathbf{a}' + \tilde{\mathbf{e}}_j \tilde{\mathbf{e}}_i \mathbf{a}' \right] \\ &= 2 \left[ (\mathbf{e}_{0_j} \mathbf{e}_{0_i} \mathbf{I} + \mathbf{e}_{0_i} \tilde{\mathbf{e}}_j) \mathbf{a}' + \mathbf{e}_j \mathbf{a}'^T \mathbf{e}_i + \mathbf{e}_{0_j} \tilde{\mathbf{e}}_i \mathbf{a}' + \tilde{\mathbf{e}}_j \tilde{\mathbf{e}}_i \mathbf{a}' \right]\end{aligned}\quad (2.6.27)$$

Since the second and last expressions on the right are identical, with the exception that the roles of  $i$  and  $j$  have been reversed, this yields the identity

$$\mathbf{B}(\mathbf{p}_i, \mathbf{a}')\mathbf{p}_j = \mathbf{B}(\mathbf{p}_j, \mathbf{a}')\mathbf{p}_i \quad (2.6.28)$$

which does not require  $\mathbf{p}_i$  or  $\mathbf{p}_j$  in  $\mathbb{R}^4$  to be normalized. Using the approach employed to obtain Eq. (2.6.26),

$$\begin{aligned} (\mathbf{A}^T(\mathbf{p})\mathbf{a})_p &= \left( (\mathbf{e}_0^2 - \mathbf{e}^T\mathbf{e})\mathbf{a} + 2\mathbf{e}\mathbf{e}^T\mathbf{a} - 2\mathbf{e}_0\tilde{\mathbf{e}}\mathbf{a} \right)_p \\ &= \left[ 2\mathbf{e}_0\mathbf{a} - 2\tilde{\mathbf{e}}\mathbf{a} \quad -2\mathbf{a}\mathbf{e}^T + 2\mathbf{a}^T\mathbf{e}\mathbf{I} + 2\mathbf{e}\mathbf{a}^T + 2\mathbf{e}_0\tilde{\mathbf{a}} \right] \\ &= 2\left[ (\mathbf{e}_0\mathbf{I} - \tilde{\mathbf{e}})\mathbf{a} \quad +\mathbf{e}\mathbf{a}^T - \tilde{\mathbf{e}}\tilde{\mathbf{a}} + \mathbf{e}_0\tilde{\mathbf{a}} \right] \\ &= 2\left[ (\mathbf{e}_0\mathbf{I} - \tilde{\mathbf{e}})\mathbf{a} \quad +\mathbf{e}\mathbf{a}^T + (\mathbf{e}_0\mathbf{I} - \tilde{\mathbf{e}})\tilde{\mathbf{a}} \right] \end{aligned} \quad (2.6.29)$$

and a derivative relation is obtained, with no requirement for normalization of  $\mathbf{p}$ ,

$$(\mathbf{A}^T(\mathbf{p})\mathbf{a})_p = \mathbf{C}(\mathbf{p}, \mathbf{a}) \quad (2.6.30)$$

where

$$\mathbf{C}(\mathbf{p}, \mathbf{a}) \equiv 2\left[ (\mathbf{e}_0\mathbf{I} - \tilde{\mathbf{e}})\mathbf{a} \quad \mathbf{e}\mathbf{a}^T + (\mathbf{e}_0\mathbf{I} - \tilde{\mathbf{e}})\tilde{\mathbf{a}} \right] = 2\mathbf{G}(\mathbf{p})\mathbf{R}(\mathbf{a}) \quad (2.6.31)$$

the second equality being verified by direct manipulation. Using the same approach employed to derive Eq. (2.6.28), the following identity, not requiring  $\mathbf{p}_i$  or  $\mathbf{p}_j$  in  $\mathbb{R}^4$  to be normalized, is obtained:

$$\mathbf{C}(\mathbf{p}_i, \mathbf{a})\mathbf{p}_j = \mathbf{C}(\mathbf{p}_j, \mathbf{a})\mathbf{p}_i \quad (2.6.32)$$

Equation (2.6.28) provides an easy result for the derivative

$$(\mathbf{B}(\mathbf{p}_i, \mathbf{a}')\mathbf{p}_j)_{\mathbf{p}_i} = (\mathbf{B}(\mathbf{p}_j, \mathbf{a}')\mathbf{p}_i)_{\mathbf{p}_i} = \mathbf{B}(\mathbf{p}_j, \mathbf{a}') \quad (2.6.33)$$

without Euler parameter normalization. Similarly, Eq. (2.6.32) yields

$$(\mathbf{C}(\mathbf{p}_i, \mathbf{a})\mathbf{p}_j)_{\mathbf{p}_i} = (\mathbf{C}(\mathbf{p}_j, \mathbf{a})\mathbf{p}_i)_{\mathbf{p}_i} = \mathbf{C}(\mathbf{p}_j, \mathbf{a}) \quad (2.6.34)$$

without Euler parameter normalization.

Expanding the products indicated, using Eqs. (2.1.22), (2.1.24), (2.1.32), and (2.1.27),

$$\begin{aligned} \mathbf{B}^T(\mathbf{p}, \mathbf{a}')\mathbf{b} &= 2\left[ (\mathbf{e}_0\mathbf{I} + \tilde{\mathbf{e}})\mathbf{a}' \quad \mathbf{e}\mathbf{a}'^T - (\mathbf{e}_0\mathbf{I} + \tilde{\mathbf{e}})\tilde{\mathbf{a}}' \right]^T \mathbf{b} \\ &= 2\left[ \begin{array}{c|c} \mathbf{e}_0\mathbf{a}'^T\mathbf{b} - \mathbf{a}'^T\tilde{\mathbf{e}}\mathbf{b} & \\ \hline \mathbf{a}'\mathbf{e}^T\mathbf{b} + \mathbf{e}_0\tilde{\mathbf{a}}'\mathbf{b} - \tilde{\mathbf{a}}'\tilde{\mathbf{e}}\mathbf{b} & \end{array} \right] = 2\left[ \begin{array}{c|c} \mathbf{e}_0\mathbf{a}'^T\mathbf{b} + \mathbf{a}'^T\tilde{\mathbf{e}}\mathbf{b} & \\ \hline \mathbf{e}_0\tilde{\mathbf{a}}'\mathbf{b} + \mathbf{a}'\mathbf{b}^T\mathbf{e} + \tilde{\mathbf{a}}'\tilde{\mathbf{e}}\mathbf{b} & \end{array} \right] \\ &= 2\left[ \begin{array}{c|c} \mathbf{a}'^T\mathbf{b} & \mathbf{a}'^T\tilde{\mathbf{e}}\mathbf{b} \\ \hline \tilde{\mathbf{a}}'\mathbf{b} & \mathbf{a}'\mathbf{b}^T + \tilde{\mathbf{a}}'\tilde{\mathbf{e}}\mathbf{b} \end{array} \right] \mathbf{p} = 2\left[ \begin{array}{c|c} \mathbf{a}'^T\mathbf{b} & \mathbf{a}'^T\tilde{\mathbf{e}}\mathbf{b} \\ \hline \tilde{\mathbf{a}}'\mathbf{b} & \mathbf{a}'\mathbf{b}^T + \mathbf{b}\mathbf{a}'^T - \mathbf{a}'^T\mathbf{b}\mathbf{I} \end{array} \right] \mathbf{p} \end{aligned} \quad (2.6.35)$$

with no normalization of  $\mathbf{p}$ . This yields the identity

$$\mathbf{B}^T(\mathbf{p}, \mathbf{a}')\mathbf{b} = \mathbf{K}(\mathbf{a}', \mathbf{b})\mathbf{p} \quad (2.6.36)$$

where

$$\mathbf{K}(\mathbf{a}', \mathbf{b}) \equiv 2 \begin{bmatrix} \mathbf{a}'^T \mathbf{b} & \mathbf{a}'^T \tilde{\mathbf{b}} \\ \tilde{\mathbf{a}}' \mathbf{b} & \mathbf{a}' \mathbf{b}^T + \mathbf{b} \mathbf{a}'^T - \mathbf{a}'^T \mathbf{b} \mathbf{I} \end{bmatrix} \quad (2.6.37)$$

A direct manipulation shows that  $\mathbf{K}(\mathbf{a}', \mathbf{b})$  is symmetric; i.e., for any values of  $\mathbf{a}'$  and  $\mathbf{b}$ ,  $\mathbf{K}^T(\mathbf{a}', \mathbf{b}) \equiv (\mathbf{K}(\mathbf{a}', \mathbf{b}))^T = \mathbf{K}(\mathbf{a}', \mathbf{b})$ . Use of Eq. (2.6.36) yields the derivative relation

$$(\mathbf{B}^T(\mathbf{p}, \mathbf{a}')\gamma)_p = (\mathbf{K}(\mathbf{a}', \gamma)\mathbf{p})_p = \mathbf{K}(\mathbf{a}', \gamma) \quad (2.6.38)$$

Manipulations identical to those used in obtaining Eq. (2.6.35) yield

$$\begin{aligned} \mathbf{C}^T(\mathbf{p}, \mathbf{a})\mathbf{b} &= 2 \left[ (\mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{e}}) \mathbf{a} - \mathbf{e} \mathbf{a}^T + (\mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{e}}) \tilde{\mathbf{a}} \right]^T \mathbf{b} \\ &= 2 \begin{bmatrix} \mathbf{e}_0 \mathbf{a}^T \mathbf{b} + \mathbf{a}^T \tilde{\mathbf{e}} \mathbf{b} \\ \mathbf{a} \mathbf{e}^T \mathbf{b} - \mathbf{e}_0 \tilde{\mathbf{a}} \mathbf{b} - \tilde{\mathbf{a}} \tilde{\mathbf{e}} \mathbf{b} \end{bmatrix} = 2 \begin{bmatrix} \mathbf{e}_0 \mathbf{a}^T \mathbf{b} - \mathbf{a}^T \tilde{\mathbf{b}} \mathbf{e} \\ \mathbf{a} \mathbf{b}^T \mathbf{e} - \mathbf{e}_0 \tilde{\mathbf{a}} \mathbf{b} + \tilde{\mathbf{a}} \tilde{\mathbf{b}} \mathbf{e} \end{bmatrix} \\ &= 2 \begin{bmatrix} \mathbf{a}^T \mathbf{b} & -\mathbf{a}^T \tilde{\mathbf{b}} \\ -\tilde{\mathbf{a}} \mathbf{b} & \mathbf{a} \mathbf{b}^T + \tilde{\mathbf{a}} \tilde{\mathbf{b}} \end{bmatrix} \mathbf{p} = 2 \begin{bmatrix} \mathbf{a}^T \mathbf{b} & -\mathbf{a}^T \tilde{\mathbf{b}} \\ -\tilde{\mathbf{a}} \mathbf{b} & \mathbf{a} \mathbf{b}^T + \mathbf{b} \mathbf{a}^T - \mathbf{a}^T \mathbf{b} \mathbf{I} \end{bmatrix} \mathbf{p} \end{aligned} \quad (2.6.39)$$

where no normalization of  $\mathbf{p}$  is required. This yields the identity

$$\mathbf{C}^T(\mathbf{p}, \mathbf{a})\mathbf{b} = \mathbf{L}(\mathbf{a}, \mathbf{b})\mathbf{p} \quad (2.6.40)$$

where

$$\mathbf{L}(\mathbf{a}, \mathbf{b}) \equiv 2 \begin{bmatrix} \mathbf{a}^T \mathbf{b} & -\mathbf{a}^T \tilde{\mathbf{b}} \\ -\tilde{\mathbf{a}} \mathbf{b} & \mathbf{a} \mathbf{b}^T + \mathbf{b} \mathbf{a}^T - \mathbf{a}^T \mathbf{b} \mathbf{I} \end{bmatrix} \quad (2.6.41)$$

and  $\mathbf{L}(\mathbf{a}', \mathbf{b})$  is symmetric. Equation (2.6.40) yields the derivative relation

$$(\mathbf{C}^T(\mathbf{p}, \mathbf{a})\mathbf{b})_p = (\mathbf{L}(\mathbf{a}, \mathbf{b})\mathbf{p})_p = \mathbf{L}(\mathbf{a}, \mathbf{b}) \quad (2.6.42)$$

Using the identities of Eqs. (2.6.14) and (2.6.15), the following derivatives are obtained, without requiring that  $\mathbf{p}$  or  $\gamma$  in  $\mathbb{R}^4$  be normalized:

$$(\mathbf{E}(\mathbf{p})\gamma)_p = -\mathbf{E}(\gamma) \quad (2.6.43)$$

$$(\mathbf{G}(\mathbf{p})\gamma)_p = -\mathbf{G}(\gamma) \quad (2.6.44)$$

Expanding the product indicated, using Eqs. (2.1.23) and (2.1.25),

$$\mathbf{E}^T(\mathbf{p})\mathbf{a} = [-\mathbf{e} \quad \tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}]^T \mathbf{a} = \begin{bmatrix} -\mathbf{e}^T \mathbf{a} \\ -\tilde{\mathbf{e}} \mathbf{a} + \mathbf{e}_0 \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & \tilde{\mathbf{a}} \end{bmatrix} \mathbf{p} \quad (2.6.45)$$

Using the definition of Eq. (2.6.19), this is the identity

$$\mathbf{E}^T(\mathbf{p})\mathbf{a} = \mathbf{R}(\mathbf{a})\mathbf{p} \quad (2.6.46)$$

which, with no normalization of  $\mathbf{p}$ , yields the derivative relationship

$$(\mathbf{E}^T(\mathbf{p})\mathbf{a})_p = \mathbf{R}(\mathbf{a}) \quad (2.6.47)$$

Using the same manipulations yields analogous identities and derivative relationships,

$$\mathbf{G}^T(\mathbf{p})\mathbf{a} = [-\mathbf{e} \quad -\tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}]^T \mathbf{a} = \begin{bmatrix} -\mathbf{e}^T \mathbf{a} \\ \tilde{\mathbf{e}} \mathbf{a} + \mathbf{e}_0 \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & -\tilde{\mathbf{a}} \end{bmatrix} \mathbf{p} \quad (2.6.48)$$

$$\mathbf{G}^T(\mathbf{p})\mathbf{a} = \mathbf{T}(\mathbf{a})\mathbf{p} \quad (2.6.49)$$

$$(\mathbf{G}^T(\mathbf{p})\mathbf{a})_p = \mathbf{T}(\mathbf{a}) \quad (2.6.50)$$

where no normalization of  $\mathbf{p}$  is required.

It is of value for manipulations to be carried out in kinematic and dynamic analysis to note that the operators  $\mathbf{B}(\mathbf{p}, \mathbf{a}')$ ,  $\mathbf{C}(\mathbf{p}, \mathbf{a})$ ,  $\mathbf{K}(\mathbf{a}', \mathbf{b})$ , and  $\mathbf{L}(\mathbf{a}, \mathbf{b})$  of Eqs. (2.6.25), (2.6.31), (2.6.37), and (2.6.41) are *bilinear* in their arguments. That is, with either argument fixed, they are linear in the other argument.

### 2.6.3 Time Derivatives and Variations of Euler Parameters

Since orientation of a body in space varies with time, the orientation transformation matrix is a function of time; i.e.,  $\mathbf{A} = \mathbf{A}(t)$ . Thus, the Euler parameter vector  $\mathbf{p}$  that defines  $\mathbf{A}(t)$  must be a function of time; i.e.,  $\mathbf{A}(t) = \mathbf{A}(\mathbf{p}(t))$ . Since the Euler parameter normalization constraint must hold at all orientations; i.e. for all  $t$ ,

$$\mathbf{p}^T(t)\mathbf{p}(t) = 1 \quad (2.6.51)$$

Equation (2.6.51) may be differentiated with respect to time to obtain  $\mathbf{p}^T \dot{\mathbf{p}} + \dot{\mathbf{p}}^T \mathbf{p} = 2\mathbf{p}^T \dot{\mathbf{p}} = 0$ , equivalently,

$$\mathbf{p}^T \dot{\mathbf{p}} = 0 \quad (2.6.52)$$

It is important to recall that this condition is only satisfied if the Euler parameter normalization condition is imposed for all time.

Using the identity  $\mathbf{E}(\mathbf{p}_i)\mathbf{G}^T(\mathbf{p}_j) = \mathbf{E}(\mathbf{p}_j)\mathbf{G}^T(\mathbf{p}_i)$  of (2.6.17) with  $\mathbf{p}_i = \dot{\mathbf{p}}$  and  $\mathbf{p}_j = \mathbf{p}$ ,

$$\mathbf{E}(\dot{\mathbf{p}})\mathbf{G}^T(\mathbf{p}) = \mathbf{E}(\mathbf{p})\mathbf{G}^T(\dot{\mathbf{p}}) \quad (2.6.53)$$

Differentiating Eq. (2.6.4) in the form  $\mathbf{A}(t) = \mathbf{E}(\mathbf{p}(t))\mathbf{G}^T(\mathbf{p}(t))$  with respect to time, using the fact that  $\mathbf{E}(\mathbf{p})$  and  $\mathbf{G}(\mathbf{p})$  are linear in  $\mathbf{p}$ , and using (2.6.53),

$$\begin{aligned} \dot{\mathbf{A}} &= \dot{\mathbf{E}}\mathbf{G}^T(\mathbf{p}) + \mathbf{E}(\mathbf{p})\dot{\mathbf{G}}^T = \mathbf{E}(\dot{\mathbf{p}})\mathbf{G}^T(\mathbf{p}) + \mathbf{E}(\mathbf{p})\mathbf{G}^T(\dot{\mathbf{p}}) \\ &= 2\mathbf{E}(\mathbf{p})\mathbf{G}^T(\dot{\mathbf{p}}) = 2\mathbf{E}(\mathbf{p})\dot{\mathbf{G}}^T = 2\dot{\mathbf{E}}\mathbf{G}^T(\mathbf{p}) \end{aligned} \quad (2.6.54)$$

where  $\dot{\mathbf{E}} = \mathbf{E}(\dot{\mathbf{p}})$  and  $\dot{\mathbf{G}} = \mathbf{G}(\dot{\mathbf{p}})$ . This result holds even if  $\mathbf{p}$  is not normalized.

The product  $\mathbf{E}(\mathbf{p})\dot{\mathbf{p}}$  may be expanded as

$$\mathbf{E}(\mathbf{p})\dot{\mathbf{p}} = [-\mathbf{e} \quad \tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}] \begin{bmatrix} \dot{\mathbf{e}}_0 \\ \dot{\mathbf{e}} \end{bmatrix} = -\dot{\mathbf{e}}_0 \mathbf{e} + \tilde{\mathbf{e}} \dot{\mathbf{e}} + \mathbf{e}_0 \dot{\mathbf{e}} \quad (2.6.55)$$

Applying the tilde operator to both sides of this equation and manipulating, using Eqs. (2.1.32), (2.1.30), and the second of Eqs. (2.5.24),

$$\begin{aligned} \widetilde{\mathbf{E}(\mathbf{p})\dot{\mathbf{p}}} &= \widetilde{-\dot{\mathbf{e}}_0 \mathbf{e} + \tilde{\mathbf{e}} \dot{\mathbf{e}} + \mathbf{e}_0 \dot{\mathbf{e}}} = -\dot{\mathbf{e}}_0 \tilde{\mathbf{e}} + \widetilde{\tilde{\mathbf{e}} \dot{\mathbf{e}}} + \mathbf{e}_0 \widetilde{\dot{\mathbf{e}}} = -\dot{\mathbf{e}}_0 \tilde{\mathbf{e}} + \tilde{\mathbf{e}} \tilde{\dot{\mathbf{e}}} - \tilde{\mathbf{e}} \tilde{\mathbf{e}} + \mathbf{e}_0 \tilde{\mathbf{e}} \\ &= -\dot{\mathbf{e}}_0 \tilde{\mathbf{e}} + \tilde{\mathbf{e}} \tilde{\dot{\mathbf{e}}} - \mathbf{e}^T \mathbf{e} \mathbf{I} + \mathbf{e}_0 \tilde{\mathbf{e}} = -\dot{\mathbf{e}}_0 \tilde{\mathbf{e}} + \tilde{\mathbf{e}} \tilde{\dot{\mathbf{e}}} - \mathbf{e}^T \mathbf{e} \mathbf{I} + \mathbf{e}_0 \tilde{\mathbf{e}} \\ &= -\mathbf{E}(\mathbf{p}) \mathbf{E}^T(\dot{\mathbf{p}}) \end{aligned}$$

where  $\mathbf{e}^T \dot{\mathbf{e}} = -\mathbf{e}_0 \dot{\mathbf{e}}_0$  has been used from  $\mathbf{p}^T \dot{\mathbf{p}} = 0$  of Eq. (2.5.24). Thus, this equation holds only if  $\mathbf{p}^T \dot{\mathbf{p}} = 0$ . Taking the transpose of both sides of this equation yields

$$\widetilde{\mathbf{E}(\mathbf{p})\dot{\mathbf{p}}} = \mathbf{E}(\dot{\mathbf{p}}) \mathbf{E}^T(\mathbf{p}) = \dot{\mathbf{E}} \mathbf{E}^T \quad (2.6.56)$$

which only holds only if  $\mathbf{p}^T \dot{\mathbf{p}} = 0$ .

From Eqs. (2.4.48) and (2.4.49), using Eqs. (2.6.4),

(2.6.54), (2.6.9), (2.6.10), (2.6.5), and (2.6.6) and  $\mathbf{p}^T \mathbf{p} = 1$ ,

$$\begin{aligned} \tilde{\omega} &= \dot{\mathbf{A}} \mathbf{A}^T(\mathbf{p}) = 2\dot{\mathbf{E}} \mathbf{G}^T(\mathbf{p}) \mathbf{G}(\mathbf{p}) \mathbf{E}^T(\mathbf{p}) = 2\dot{\mathbf{E}} (\mathbf{I} - \mathbf{p} \mathbf{p}^T) \mathbf{E}^T(\mathbf{p}) = 2\dot{\mathbf{E}} \mathbf{E}^T(\mathbf{p}) \\ \tilde{\omega}' &= \mathbf{A}^T(\mathbf{p}) \dot{\mathbf{A}} = 2\mathbf{G}(\mathbf{p}) \mathbf{E}^T(\mathbf{p}) \mathbf{E}(\mathbf{p}) \dot{\mathbf{G}}^T = 2\mathbf{G}(\mathbf{p}) (\mathbf{I} - \mathbf{p} \mathbf{p}^T) \dot{\mathbf{G}}^T = 2\mathbf{G}(\mathbf{p}) \dot{\mathbf{G}}^T \end{aligned} \quad (2.6.57)$$

Equation (2.6.56) yields

$$\tilde{\omega} = \widetilde{2\mathbf{E}(\mathbf{p})\dot{\mathbf{p}}} \quad (2.6.58)$$

equivalently, the desired relation

$$\omega = 2\mathbf{E}(\mathbf{p})\dot{\mathbf{p}} \quad (2.6.59)$$

Multiplying both sides of Eq. (2.6.59) on the left by  $\mathbf{E}^T(\mathbf{p})$  yields

$$\mathbf{E}^T(\mathbf{p})\omega = 2\mathbf{E}^T(\mathbf{p})\mathbf{E}(\mathbf{p})\dot{\mathbf{p}} = 2(\mathbf{p}^T \mathbf{p} \mathbf{I} - \mathbf{p} \mathbf{p}^T)\dot{\mathbf{p}} \quad (2.6.60)$$

Providing  $\mathbf{p}^T \mathbf{p} = 1$  and  $\mathbf{p}^T \dot{\mathbf{p}} = 0$ , this yields the inverse relationship

$$\dot{\mathbf{p}} = (1/2)\mathbf{E}^T(\mathbf{p})\omega \quad (2.6.61)$$

Thus, even though the matrix  $\mathbf{E}(\mathbf{p})$  is not square, it behaves much like an orthogonal matrix in Eqs. (2.6.59) and (2.6.61).

From Eq. (2.6.59),

$$\omega' = \mathbf{A}^T(\mathbf{p})\omega = 2\mathbf{G}(\mathbf{p})\mathbf{E}^T(\mathbf{p})\mathbf{E}(\mathbf{p})\dot{\mathbf{p}} = 2\mathbf{G}(\mathbf{p})(\mathbf{I} - \mathbf{p} \mathbf{p}^T)\dot{\mathbf{p}} \quad (2.6.62)$$

Since  $\mathbf{G}(\mathbf{p})\mathbf{p} = \mathbf{0}$ ,

$$\omega' = 2\mathbf{G}(\mathbf{p})\dot{\mathbf{p}} \quad (2.6.63)$$

Multiplying both sides by  $\mathbf{G}^T(\mathbf{p})$ ,  $\mathbf{G}^T\omega' = 2\mathbf{G}^T(\mathbf{p})\mathbf{G}(\mathbf{p})\dot{\mathbf{p}} = 2(\mathbf{I} - \mathbf{p}\mathbf{p}^T)\dot{\mathbf{p}}$  and, provided  $\mathbf{p}^T\mathbf{p} = 1$ , the inverse relation follows, just as in Eq. (2.6.61); i.e.,

$$\dot{\mathbf{p}} = (1/2)\mathbf{G}^T(\mathbf{p})\omega' \quad (2.6.64)$$

It is important to note that Eqs. (2.6.57) through (2.6.64) hold only if  $\mathbf{p}^T\mathbf{p} = 1$  for all time.

#### 2.6.4 Euler Parameter Differentials and Infinitesimal Rotation

Repeating the manipulations of Eqs. (2.6.52) through (2.6.64), replacing derivative  $d/dt$  with the differential  $\dot{\phantom{x}}$  of Section 2.4.6 and suppressing  $\mathbf{p}$  arguments,

$$\mathbf{p}^T \mathbf{p} = 0 \quad (2.6.65)$$

$$\mathbf{A} = 2\mathbf{E}(\mathbf{p}) \quad \mathbf{G}^T = 2 \quad \mathbf{E}\mathbf{G}^T(\mathbf{p}) \quad (2.6.66)$$

$$\widetilde{\mathbf{E}(\mathbf{p})} \quad \mathbf{p} = -\mathbf{E}(\mathbf{p}) \quad \mathbf{E}^T \quad (2.6.67)$$

$$\begin{aligned} \pi &= 2\mathbf{E}(\mathbf{p}) \quad \mathbf{p} \\ \pi' &= 2\mathbf{G}(\mathbf{p}) \quad \mathbf{p} \end{aligned} \quad (2.6.68)$$

$$\mathbf{p} = (1/2)\mathbf{E}^T(\mathbf{p}) \quad \pi = (1/2)\mathbf{G}^T(\mathbf{p}) \quad \pi' \quad (2.6.69)$$

Each of Eqs. (2.6.65) through (2.6.69) is valid only if  $\mathbf{p}$  is normalized and  $\mathbf{p}^T \mathbf{p} = 0$ .

#### 2.6.5 Nonintegrability of Angular Velocity

To determine whether angular velocity can be obtained as the time derivative of some function of Euler parameters, expanding Eq. (2.6.59) and using Eq. (2.6.2) yields

$$\boldsymbol{\omega} = 2\mathbf{E}(\mathbf{p})\dot{\mathbf{p}} = 2[-\mathbf{e} \quad \tilde{\mathbf{e}} + \mathbf{e}_0\mathbf{I}] \begin{bmatrix} \dot{\mathbf{e}}_0 \\ \dot{\mathbf{e}} \end{bmatrix} = 2(-\mathbf{e}\dot{\mathbf{e}}_0 + \tilde{\mathbf{e}}\dot{\mathbf{e}} + \mathbf{e}_0\dot{\mathbf{e}}) \quad (2.6.70)$$

The first component of Eq. (2.6.70) is

$$a_x = 2(-\mathbf{e}_1\dot{\mathbf{e}}_0 + \mathbf{e}_0\dot{\mathbf{e}}_1 - \mathbf{e}_3\dot{\mathbf{e}}_2 + \mathbf{e}_2\dot{\mathbf{e}}_3) \equiv 2 \sum_{i=0}^3 a_i \dot{e}_i \quad (2.6.71)$$

The Exact Differential Theorem (Lovelock and Rund, 1975) (see Section 6.1) can be used to determine whether the differential form on the right of Eq. (2.6.71) is exact. It is the time

derivative of a function of  $\mathbf{p}$  if and only if  $\frac{\partial a_i}{\partial e_j} = \frac{\partial a_j}{\partial e_i}$  for all  $i$  and  $j$  in the range 0 to 3. With  $i = 0$  and  $j = 1$ ,

$$\frac{\partial a_0}{\partial e_1} = \frac{\partial(-2e_1)}{\partial e_1} = -2 \neq 2 = \frac{\partial(2e_0)}{\partial e_0} = \frac{\partial a_1}{\partial e_0} \quad (2.6.72)$$

Thus, the differential form of Eq. (2.6.71); i.e., the  $x$  component of angular velocity, is not exact, so it is not the time derivative of any function of Euler parameters. This shows that angular velocity cannot be integrated to obtain a function of Euler parameters. Similarly, while  $\delta\pi$  is an infinitesimal rotation, it is not the differential of any function of Euler parameters.

### 2.6.6 Angular Acceleration as a Function of Euler Parameter Derivatives

Differentiating both sides of

$$\mathbf{a} = \mathbf{A}(\mathbf{p})\mathbf{a}' \quad (2.6.73)$$

where  $\mathbf{a}'$  is constant and using the identity of Eq. (2.6.26),

$$\dot{\mathbf{a}} = (\mathbf{A}(\mathbf{p})\mathbf{a}')_p \dot{\mathbf{p}} = \mathbf{B}(\mathbf{p}, \mathbf{a}') \dot{\mathbf{p}} \quad (2.6.74)$$

Taking the time derivative of both sides of Eq. (2.6.74) and using the identity of Eq. (2.6.28),

$$\begin{aligned} \ddot{\mathbf{a}} &= \mathbf{B}(\mathbf{p}, \mathbf{a}') \ddot{\mathbf{p}} + (\mathbf{B}(\mathbf{p}, \mathbf{a}') \dot{\mathbf{p}})_p \dot{\mathbf{p}} \\ &= \mathbf{B}(\mathbf{p}, \mathbf{a}') \ddot{\mathbf{p}} + \mathbf{B}(\dot{\mathbf{p}}, \mathbf{a}') \dot{\mathbf{p}} \end{aligned} \quad (2.6.75)$$

As expected, acceleration is a linear function of Euler parameter second derivatives, with coefficients that depend on Euler parameters, and it has quadratic terms in Euler parameter first derivatives.

Relations for angular acceleration as a function of Euler parameter second derivatives may be obtained from Eqs. (2.6.59) and (2.6.63),

$$\dot{\boldsymbol{\omega}} = 2\mathbf{E}(\mathbf{p})\ddot{\mathbf{p}} + 2\mathbf{E}(\dot{\mathbf{p}})\dot{\mathbf{p}} = 2\mathbf{E}(\mathbf{p})\ddot{\mathbf{p}} \quad (2.6.76)$$

where Eq. (2.6.5) has been used with  $\mathbf{p}$  replaced by  $\dot{\mathbf{p}}$ , and

$$\dot{\boldsymbol{\omega}}' = 2\mathbf{G}(\mathbf{p})\ddot{\mathbf{p}} + 2\mathbf{G}(\dot{\mathbf{p}})\dot{\mathbf{p}} = 2\mathbf{G}(\mathbf{p})\ddot{\mathbf{p}} \quad (2.6.77)$$

where Eq. (2.6.6) has been used with  $\mathbf{p}$  replaced by  $\dot{\mathbf{p}}$ . These relations hold only if  $\mathbf{p}$  is normalized, for all time.

In order to determine  $\ddot{\mathbf{p}}$  as a function of  $\dot{\boldsymbol{\omega}}$  and  $\dot{\boldsymbol{\omega}}'$ , Eqs. (2.6.76) and (2.6.77) may be multiplied on the left by  $\mathbf{E}^T$  and  $\mathbf{G}^T$ , suppressing arguments, to obtain

$$\mathbf{E}^T \dot{\boldsymbol{\omega}} = 2\mathbf{E}^T \mathbf{E} \ddot{\mathbf{p}} = 2(\mathbf{I} - \mathbf{p} \mathbf{p}^T) \ddot{\mathbf{p}} = 2\ddot{\mathbf{p}} - 2\mathbf{p} \mathbf{p}^T \ddot{\mathbf{p}} \quad (2.6.78)$$

$$\mathbf{G}^T \dot{\boldsymbol{\omega}}' = 2\mathbf{G}^T \mathbf{G} \ddot{\mathbf{p}} = 2(\mathbf{I} - \mathbf{p} \mathbf{p}^T) \ddot{\mathbf{p}} = 2\ddot{\mathbf{p}} - 2\mathbf{p} \mathbf{p}^T \ddot{\mathbf{p}} \quad (2.6.79)$$

Differentiation of the identity  $\mathbf{p}^T \dot{\mathbf{p}} = 0$  of Eq. (2.6.52) and using  $\dot{\mathbf{p}} = (1/2)\mathbf{G}^T \boldsymbol{\omega}'$  of Eq. (2.6.64) and  $\mathbf{G}\mathbf{G}^T = \mathbf{p}\mathbf{p}^T \mathbf{I}_3$  of Eq. (2.6.8),

$$\mathbf{p}^T \ddot{\mathbf{p}} = -\dot{\mathbf{p}}^T \dot{\mathbf{p}} = -(1/4)\boldsymbol{\omega}'^T \boldsymbol{\omega}' (\mathbf{p}^T \mathbf{p}) = -(1/4)(\mathbf{A}^T \boldsymbol{\omega})^T \mathbf{A}^T \boldsymbol{\omega} (\mathbf{p}^T \mathbf{p}) = -(1/4)\boldsymbol{\omega}^T \boldsymbol{\omega} (\mathbf{p}^T \mathbf{p})$$

Substituting this result in Eqs. (2.6.78) and (2.6.79) and enforcing  $\mathbf{p}^T \mathbf{p} = 1$  yields

$$\ddot{\mathbf{p}} = \frac{1}{2} \mathbf{E}^T \dot{\boldsymbol{\omega}} - \frac{1}{4} \boldsymbol{\omega}^T \boldsymbol{\omega} \mathbf{p} \quad (2.6.80)$$

$$\ddot{\mathbf{p}} = \frac{1}{2} \mathbf{G}^T \dot{\boldsymbol{\omega}}' - \frac{1}{4} \boldsymbol{\omega}'^T \boldsymbol{\omega}' \mathbf{p} \quad (2.6.81)$$

Thus, there are explicit transformations from angular acceleration to Euler parameter second derivatives. However, they involve terms that are quadratic in Euler parameter first derivatives and they require  $\mathbf{p}$  to be normalized.

### 2.6.7 Special Identities

Identities that are helpful in the equations of dynamics for specific applications are derived. Forming products from Eqs. (2.6.1) and (2.6.2) and from Eqs. (2.6.19) and (2.6.20) and manipulating using vector identities from Section 2.1 to move factors involving  $\mathbf{a}$  to the left, yields

$$\begin{aligned} \mathbf{G}(\mathbf{p})\mathbf{T}^T(\mathbf{a}) &= [-\mathbf{e} \quad -\tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}] \begin{bmatrix} 0 & \mathbf{a}^T \\ -\mathbf{a} & \tilde{\mathbf{a}} \end{bmatrix} \\ &= [-\mathbf{a}\mathbf{e}_0 - \tilde{\mathbf{a}}\mathbf{e} \quad -\mathbf{e}\mathbf{a}^T + \tilde{\mathbf{a}}\mathbf{e}_0 - \tilde{\mathbf{e}}\tilde{\mathbf{a}}] \\ &= [-\mathbf{a}\mathbf{e}_0 - \tilde{\mathbf{a}}\mathbf{e} \quad -\mathbf{e}\mathbf{a}^T - \tilde{\mathbf{a}}\tilde{\mathbf{e}} + \mathbf{e}\mathbf{a}^T - \mathbf{a}\mathbf{e}^T + \mathbf{e}_0\tilde{\mathbf{a}}] \quad (2.6.82) \\ &= \tilde{\mathbf{a}}[-\mathbf{e} \quad (\mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{e}})] - \mathbf{a}[\mathbf{e}_0 \quad \mathbf{e}^T] \\ &= \tilde{\mathbf{a}}\mathbf{G}(\mathbf{p}) - \mathbf{a}\mathbf{p}^T \\ \mathbf{E}(\mathbf{p})\mathbf{R}^T(\mathbf{a}) &= -\tilde{\mathbf{a}}\mathbf{E}(\mathbf{p}) - \mathbf{a}\mathbf{p}^T \end{aligned}$$

which do not require normalization of  $\mathbf{p}$ .

A helpful identity in applications is obtained by carrying out the following manipulations, using Eqs. (2.6.25), (2.6.82), (2.6.4), (2.6.5), (2.6.31), and (2.6.6), with no Euler parameter normalization:

$$\begin{aligned} \mathbf{G}(\mathbf{p})\mathbf{B}^T(\mathbf{p}, \mathbf{a}') &= 2\mathbf{G}(\mathbf{p})\mathbf{T}^T(\mathbf{a}')\mathbf{E}^T(\mathbf{p}) = 2(\tilde{\mathbf{a}}'\mathbf{G}(\mathbf{p}) - \mathbf{a}'\mathbf{p}^T)\mathbf{E}^T(\mathbf{p}) \\ &= 2\tilde{\mathbf{a}}'(\mathbf{E}(\mathbf{p})\mathbf{G}^T(\mathbf{p}))^T - 2\mathbf{a}'(\mathbf{E}(\mathbf{p})\mathbf{p})^T \\ &= 2\tilde{\mathbf{a}}'\mathbf{A}^T(\mathbf{p}) \quad (2.6.83) \\ \mathbf{E}(\mathbf{p})\mathbf{C}^T(\mathbf{p}, \mathbf{a}') &= 2\mathbf{E}(\mathbf{p})\mathbf{R}^T(\mathbf{a}')\mathbf{G}^T(\mathbf{p}) = 2(-\tilde{\mathbf{a}}'\mathbf{E}(\mathbf{p}) - \mathbf{a}'\mathbf{p}^T)\mathbf{G}^T(\mathbf{p}) \\ &= -2\tilde{\mathbf{a}}'\mathbf{E}(\mathbf{p})\mathbf{G}^T(\mathbf{p}) - 2\mathbf{a}'(\mathbf{G}(\mathbf{p})\mathbf{p})^T \\ &= -2\tilde{\mathbf{a}}'\mathbf{A}(\mathbf{p}) \end{aligned}$$

A final helpful relation involves the matrix  $[\mathbf{p} \quad \mathbf{G}^T(\mathbf{p})]$ ,

$$[\mathbf{p} \quad \mathbf{G}^T(\mathbf{p})] = \begin{bmatrix} \mathbf{p}^T \\ \mathbf{G}(\mathbf{p}) \end{bmatrix}^T = \begin{bmatrix} \mathbf{p}^T \\ \mathbf{G}(\mathbf{p}) \end{bmatrix}^{-1} \quad (2.6.84)$$

which holds, based on Eq. (2.6.10), as long as  $\mathbf{p}$  is normalized.

### 2.6.8 Derivatives with Respect to Model Parameters

In modeling systems for applications, parameter vectors that define geometry of the model are occasionally used as generalized coordinates. Derivatives with respect to these parameters are thus needed. For example, forming the products indicated and manipulating,

$$\begin{aligned}\mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i) \mathbf{p}_j &= 2 \left[ \left( \mathbf{e}_{0_i} \mathbf{I} + \tilde{\mathbf{e}}_i \right) \mathbf{a}'_i - \left( \mathbf{e}_{0_i} \mathbf{I} + \tilde{\mathbf{e}}_i \right) \tilde{\mathbf{a}}'_i \right] \mathbf{p}_j \\ &= 2 \left\{ \left( \mathbf{e}_{0_i} \mathbf{e}_{0_j} \mathbf{I} + \mathbf{e}_{0_j} \tilde{\mathbf{e}}_i \right) \mathbf{a}'_i + \mathbf{e}_i \mathbf{e}_j^T \mathbf{a}'_i + \left( \mathbf{e}_{0_i} \mathbf{I} + \tilde{\mathbf{e}}_i \right) \tilde{\mathbf{e}}_j \mathbf{a}'_i \right\}\end{aligned}$$

Differentiating with respect to  $\mathbf{a}'_i$  yields

$$(\mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i) \mathbf{p}_j)_{\mathbf{a}'_i} = 2 \left\{ \left( \mathbf{e}_{0_i} \mathbf{e}_{0_j} \mathbf{I} + \mathbf{e}_{0_j} \tilde{\mathbf{e}}_i \right) + \mathbf{e}_i \mathbf{e}_j^T + \left( \mathbf{e}_{0_i} \mathbf{I} + \tilde{\mathbf{e}}_i \right) \tilde{\mathbf{e}}_j \right\}$$

Carrying out expansion of products indicated yields the relation

$$(\mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i) \mathbf{p}_j)_{\mathbf{a}'_i} = 2 \left\{ \mathbf{E}(\mathbf{p}_i) \mathbf{G}^T(\mathbf{p}_j) \right\} \equiv \mathbf{M}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{M}(\mathbf{p}_j, \mathbf{p}_i) \quad (2.6.85)$$

where Eqs. (2.6.17) and (2.6.85) are used. Similarly, forming the product

$$\mathbf{C}(\mathbf{p}_i, \mathbf{a}_i) \mathbf{p}_j = 2 \left[ \left( \mathbf{e}_{0_i} \mathbf{I} - \tilde{\mathbf{e}}_i \right) \mathbf{a}_i - \mathbf{e}_i \mathbf{a}_i^T + \left( \mathbf{e}_{0_i} \mathbf{I} - \tilde{\mathbf{e}}_i \right) \tilde{\mathbf{a}}_i \right] \mathbf{p}_j$$

and differentiating with respect to  $\mathbf{a}'_i$ , an identical manipulation to that above yields

$$(\mathbf{C}(\mathbf{p}_i, \mathbf{a}_i) \mathbf{p}_j)_{\mathbf{a}_i} = 2 \left\{ \mathbf{G}(\mathbf{p}_i) \mathbf{E}^T(\mathbf{p}_j) \right\} \equiv \mathbf{N}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{N}(\mathbf{p}_j, \mathbf{p}_i) \quad (2.6.86)$$

Differentiation of Eqs. (2.6.85) and (2.6.86), using Eq. (2.6.43), with no normalization of  $\mathbf{p}$ , yields

$$\begin{aligned}(\mathbf{M}(\mathbf{p}, \gamma) \mathbf{b})_{\mathbf{p}} &= 2 \left( \mathbf{E}(\mathbf{p}) \mathbf{G}^T(\gamma) \mathbf{b} \right)_{\mathbf{p}} = -2 \mathbf{E}(\mathbf{G}^T(\gamma) \mathbf{b}) \equiv \mathbf{Z}(\gamma, \mathbf{b}) \\ (\mathbf{N}(\mathbf{p}, \gamma) \mathbf{b})_{\mathbf{p}} &= 2 \left( \mathbf{G}(\mathbf{p}) \mathbf{E}^T(\gamma) \mathbf{b} \right)_{\mathbf{p}} = -2 \mathbf{G}(\mathbf{E}^T(\gamma) \mathbf{b}) \equiv \mathbf{Y}(\gamma, \mathbf{b})\end{aligned} \quad (2.6.87)$$

Finally, forming the product indicated, differentiating, and manipulating,

$$\begin{aligned}(\mathbf{B}^T(\mathbf{p}, \mathbf{a}') \mathbf{b})_{\mathbf{a}'} &= 2 \left[ \begin{array}{c} \mathbf{a}'^T (\mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{e}}) \mathbf{b} \\ \mathbf{a}' \mathbf{e}^T \mathbf{b} + \tilde{\mathbf{a}}' (\mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{e}}) \mathbf{b} \end{array} \right]_{\mathbf{a}'} = 2 \left[ \begin{array}{c} \mathbf{b}^T (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \mathbf{a}' \\ \mathbf{e}^T \mathbf{b} \mathbf{a}' - \mathbf{e}_0 \tilde{\mathbf{b}} \mathbf{a}' + (\tilde{\mathbf{e}} \mathbf{b}) \mathbf{a}' \end{array} \right]_{\mathbf{a}'} \\ &= 2 \left[ \begin{array}{c} \mathbf{b}^T (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \\ -\mathbf{e}_0 \tilde{\mathbf{b}} + \mathbf{b} \mathbf{e}^T - \tilde{\mathbf{b}} \tilde{\mathbf{e}} \end{array} \right] = 2 \left[ \begin{array}{c} \mathbf{b}^T (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \\ \mathbf{b} \mathbf{e}^T - \tilde{\mathbf{b}} (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \end{array} \right] \\ &= \mathbf{C}^T(\mathbf{p}, \mathbf{b})\end{aligned} \quad (2.6.88)$$

A similar manipulation shows that

$$(\mathbf{C}^T(\mathbf{p}, \mathbf{b}) \mathbf{a}')_{\mathbf{b}} = \mathbf{B}^T(\mathbf{p}, \mathbf{a}') \quad (2.6.89)$$

None of the relations of this section require normalization of  $\mathbf{p}$ .

A number of algebraic identities follow from quadratic dependence of the orientation transformation matrix on Euler parameters. They are extended to time derivatives and differentials that define angular velocity, angular acceleration, and virtual rotations, serving as the basis for a practical theory and computational method for characterizing kinematics and dynamics of mechanical systems.

Derivative relations obtained using basic differential calculus lead to a family of differential operators that form the foundation for practical computer formulation and solution of equations of multibody kinematics and dynamics. Without the identities, detailed manipulations that were involved in their derivation would be repeated ad nauseum; i.e., to a sickening degree that would be intractable if an ad-hoc differential calculus approach were adopted. Identities relating to system design parameters that have been found helpful in applications are also provided. Key formulas from this and preceding sections are provided as a central reference in Appendix 2.C. MATLAB computer code that implements identities and differential operators is provided as Code 2.6 in Appendix 2.B.

### ***Key Formulas (Only formulas with bold italic equation numbers require $\mathbf{p}^T \mathbf{p} = 1$ )***

$$\mathbf{E}(\mathbf{p}) \equiv [-\mathbf{e} \quad \tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}] \quad \mathbf{G}(\mathbf{p}) \equiv [-\mathbf{e} \quad -\tilde{\mathbf{e}} + \mathbf{e}_0 \mathbf{I}] \quad (2.6.1) \quad (2.6.2)$$

$$\mathbf{A}(\mathbf{p}) = \mathbf{E}(\mathbf{p})\mathbf{G}(\mathbf{p})^T \quad \mathbf{E}(\mathbf{p})\mathbf{p} = \mathbf{0} \quad \mathbf{G}(\mathbf{p})\mathbf{p} = \mathbf{0} \quad (2.6.4) \quad (2.6.5) \quad (2.6.6)$$

$$\mathbf{E}(\mathbf{p})\mathbf{E}^T(\mathbf{p}) = \mathbf{p}^T \mathbf{p} \mathbf{I} \quad \mathbf{G}(\mathbf{p})\mathbf{G}^T(\mathbf{p}) = \mathbf{p}^T \mathbf{p} \mathbf{I} \quad (2.6.7) \quad (2.6.8)$$

$$\mathbf{E}^T(\mathbf{p})\mathbf{E}(\mathbf{p}) = \mathbf{p}^T \mathbf{p} \mathbf{I}_4 - \mathbf{p} \mathbf{p}^T \quad \mathbf{G}^T(\mathbf{p})\mathbf{G}(\mathbf{p}) = \mathbf{p}^T \mathbf{p} \mathbf{I}_4 - \mathbf{p} \mathbf{p}^T \quad (2.6.9) \quad (2.6.10)$$

$$\mathbf{E}(\mathbf{p}_i)\mathbf{p}_j = -\mathbf{E}(\mathbf{p}_j)\mathbf{p}_i \quad \mathbf{G}(\mathbf{p}_i)\mathbf{p}_j = -\mathbf{G}(\mathbf{p}_j)\mathbf{p}_i \quad (2.6.14) \quad (2.6.15)$$

$$\mathbf{R}(\mathbf{a}) \equiv \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & \tilde{\mathbf{a}} \end{bmatrix} \quad \mathbf{T}(\mathbf{a}) \equiv \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & -\tilde{\mathbf{a}} \end{bmatrix} \quad (2.6.19) \quad (2.6.20)$$

$$\mathbf{B}(\mathbf{p}, \mathbf{a}') \equiv 2 \left[ (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \mathbf{a}' - \mathbf{e} \mathbf{a}'^T - (\mathbf{e}_0 \mathbf{I} + \tilde{\mathbf{e}}) \tilde{\mathbf{a}}' \right] = 2 \mathbf{E}(\mathbf{p}) \mathbf{T}(\mathbf{a}') \quad (2.6.25)$$

$$\mathbf{C}(\mathbf{p}, \mathbf{a}) \equiv 2 \left[ (\mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{e}}) \mathbf{a} - \mathbf{e} \mathbf{a}^T + (\mathbf{e}_0 \mathbf{I} - \tilde{\mathbf{e}}) \tilde{\mathbf{a}} \right] = 2 \mathbf{G}(\mathbf{p}) \mathbf{R}(\mathbf{a}) \quad (2.6.31)$$

$$(\mathbf{A}(\mathbf{p})\mathbf{a}')_p = \mathbf{B}(\mathbf{p}, \mathbf{a}') \quad (\mathbf{A}^T(\mathbf{p})\mathbf{a})_p = \mathbf{C}(\mathbf{p}, \mathbf{a}) \quad (2.6.26) \quad (2.6.30)$$

$$\mathbf{B}(\mathbf{p}_i, \mathbf{a}')\mathbf{p}_j = \mathbf{B}(\mathbf{p}_j, \mathbf{a}')\mathbf{p}_i \quad \mathbf{C}(\mathbf{p}_i, \mathbf{a})\mathbf{p}_j = \mathbf{C}(\mathbf{p}_j, \mathbf{a})\mathbf{p}_i \quad (2.6.28) \quad (2.6.32)$$

$$\mathbf{K}(\mathbf{a}', \mathbf{b}) \equiv 2 \begin{bmatrix} \mathbf{a}'^T \mathbf{b} & \mathbf{a}'^T \tilde{\mathbf{b}} \\ \tilde{\mathbf{a}}' \mathbf{b} & \mathbf{a}' \mathbf{b}^T + \mathbf{b} \mathbf{a}'^T - \mathbf{a}'^T \mathbf{b} \mathbf{I} \end{bmatrix} \quad (\mathbf{B}^T(\mathbf{p}, \mathbf{a}') \gamma)_p = \mathbf{K}(\mathbf{a}', \gamma) \quad (2.6.37) \quad (2.6.38)$$

$$\mathbf{L}(\mathbf{a}, \mathbf{b}) \equiv 2 \begin{bmatrix} \mathbf{a}^T \mathbf{b} & -\mathbf{a}^T \tilde{\mathbf{b}} \\ -\tilde{\mathbf{a}} \mathbf{b} & \mathbf{a} \mathbf{b}^T + \mathbf{b} \mathbf{a}^T - \mathbf{a}^T \mathbf{b} \mathbf{I} \end{bmatrix} \quad (\mathbf{C}^T(\mathbf{p}, \mathbf{a}) \mathbf{b})_p = \mathbf{L}(\mathbf{a}, \mathbf{b}) \quad (2.6.41) \quad (2.6.42)$$

$$\begin{aligned}
& (\mathbf{E}(\mathbf{p})\gamma)_{\mathbf{p}} = -\mathbf{E}(\gamma) & (\mathbf{G}(\mathbf{p})\gamma)_{\mathbf{p}} = -\mathbf{G}(\gamma) & (2.6.43) & (2.6.44) \\
& (\mathbf{E}^T(\mathbf{p})\mathbf{a})_{\mathbf{p}} = \mathbf{R}(\mathbf{a}) & (\mathbf{G}^T(\mathbf{p})\mathbf{a})_{\mathbf{p}} = \mathbf{T}(\mathbf{a}) & (2.6.47) & (2.6.50) \\
& \dot{\mathbf{p}}^T \dot{\mathbf{p}} = 0 & \dot{\mathbf{A}} = 2\mathbf{E}(\mathbf{p})\dot{\mathbf{G}}^T = 2\dot{\mathbf{E}}\mathbf{G}^T(\mathbf{p}) & (2.6.52) & (2.6.54) \\
& \omega = 2\mathbf{E}(\mathbf{p})\dot{\mathbf{p}} & \omega' = 2\mathbf{G}(\mathbf{p})\dot{\mathbf{p}} & (2.6.59) & (2.6.63) \\
& \dot{\omega} = 2\mathbf{E}(\mathbf{p})\ddot{\mathbf{p}} & \dot{\omega}' = 2\mathbf{G}(\mathbf{p})\ddot{\mathbf{p}} & (2.6.76) & (2.6.77) \\
& \dot{\mathbf{p}} = \frac{1}{2}\mathbf{E}^T(\mathbf{p})\omega & \dot{\mathbf{p}} = \frac{1}{2}\mathbf{G}^T(\mathbf{p})\omega' & (2.6.61) & (2.6.64) \\
& \ddot{\mathbf{p}} = \frac{1}{2}\mathbf{E}^T\omega - \frac{1}{4}\omega^T\omega\mathbf{p} & \ddot{\mathbf{p}} = \frac{1}{2}\mathbf{G}^T\dot{\omega}' - \frac{1}{4}\omega'^T\omega'\mathbf{p} & (2.6.80) & (2.6.81) \\
& \mathbf{p}^T \mathbf{p} = 0 & \mathbf{A} = 2\mathbf{E}(\mathbf{p}) \mathbf{G}^T = 2 \mathbf{E}\mathbf{G}^T(\mathbf{p}) & (2.6.65) & (2.6.66) \\
& \pi = 2\mathbf{E}(\mathbf{p}) \mathbf{p} & \pi' = 2\mathbf{G}(\mathbf{p}) \mathbf{p} & (2.6.68) & \\
& \mathbf{p} = \frac{1}{2}\mathbf{E}^T(\mathbf{p}) \pi & \mathbf{p} = \frac{1}{2}\mathbf{G}^T(\mathbf{p}) \pi' & (2.6.69) & \\
& \mathbf{G}(\mathbf{p})\mathbf{T}^T(\mathbf{a}) = \tilde{\mathbf{a}}\mathbf{G}(\mathbf{p}) - \mathbf{a}\mathbf{p}^T & \mathbf{E}(\mathbf{p})\mathbf{R}^T(\mathbf{a}) = -\tilde{\mathbf{a}}\mathbf{E}(\mathbf{p}) - \mathbf{a}\mathbf{p}^T & (2.6.82) & \\
& \mathbf{G}(\mathbf{p})\mathbf{B}^T(\mathbf{p}, \mathbf{a}') = 2\tilde{\mathbf{a}}'\mathbf{A}^T(\mathbf{p}) & \mathbf{E}(\mathbf{p})\mathbf{C}^T(\mathbf{p}, \mathbf{a}') = -2\tilde{\mathbf{a}}'\mathbf{A}(\mathbf{p}) & (2.6.79) & \\
& (\mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i)\mathbf{p}_j)_{\mathbf{a}'_i} = 2\{\mathbf{E}(\mathbf{p}_i)\mathbf{G}^T(\mathbf{p}_j)\} \equiv \mathbf{M}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{M}(\mathbf{p}_j, \mathbf{p}_i) & & (2.6.85) & \\
& (\mathbf{C}(\mathbf{p}_i, \mathbf{a}_i)\mathbf{p}_j)_{\mathbf{a}_i} = 2\{\mathbf{G}(\mathbf{p}_i)\mathbf{E}^T(\mathbf{p}_j)\} \equiv \mathbf{N}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{N}(\mathbf{p}_j, \mathbf{p}_i) & & (2.6.86) & \\
& (\mathbf{M}(\mathbf{p}, \gamma)\mathbf{b})_{\mathbf{p}} = 2(\mathbf{E}(\mathbf{p})\mathbf{G}^T(\gamma)\mathbf{b})_{\mathbf{p}} = -2\mathbf{E}(\mathbf{G}^T(\gamma)\mathbf{b}) \equiv \mathbf{Z}(\gamma, \mathbf{b}) & & (2.6.87) & \\
& (\mathbf{N}(\mathbf{p}, \gamma)\mathbf{b})_{\mathbf{p}} = 2(\mathbf{G}(\mathbf{p})\mathbf{E}^T(\gamma)\mathbf{b})_{\mathbf{p}} = -2\mathbf{G}(\mathbf{E}^T(\gamma)\mathbf{b}) \equiv \mathbf{Y}(\gamma, \mathbf{b}) & & & \\
& (\mathbf{B}^T(\mathbf{p}, \mathbf{a}')\mathbf{b})_{\mathbf{a}'} = \mathbf{C}^T(\mathbf{p}, \mathbf{b}) & (\mathbf{C}(\mathbf{p}, \mathbf{b})^T \mathbf{a}')_{\mathbf{b}} = \mathbf{B}(\mathbf{p}, \mathbf{a}')^T & (2.6.88) & (2.6.89)
\end{aligned}$$

## Appendix 2.A Euler Angle Orientation Coordinates

A classical set of orientation parameters for bodies in space, *Euler angles*, is presented in this section. Three sequential rotations are used to determine orientation. The concept of Euler angles is an extension of the process in Section 2.3 of using a single rotation angle to define orientation in the plane. Euler angles, however, are shown to (1) suffer from having a singular orientation that must be avoided in practice and (2) involve extensive transcendental function expansion in derivative expressions, which greatly complicates kinematic and dynamic computations. While the multibody dynamics community is turning to Euler parameters treated in Sections 2.5 and 2.6, there is adequate historical experience with Euler angles to justify their introduction.

To assist in the transition from classical Euler angles for multibody kinematics and dynamics to a modern setting, consider several points from the landmark paper by Stuelpnagel (1964). He observes that in 1776, Euler showed that the orientation transformation rotation group is three dimensional. Over the intervening two- and one-half centuries, many attempts have been made to find a set of three orientation generalized coordinates that uniquely define orientation of bodies in space and is well suited to equation formulation and solution. A motivation in this search was the belief that a minimal set of three coordinates is superior to a redundant set; i.e., four or more orientation parameters that are subject to constraints.

In 1940, Hopf (1940) showed that five is the minimum number of coordinates to represent the orientation transformation group in a *globally one-to-one* manner. Stuelpnagel showed, however, that the 4-dimensional "quaternion method" (Goldstein, 1980), which is a precursor of Euler parameters presented in Sections 2.5 and 2.6, is a one-to-two parameterization that practically avoids singularities, as is shown in Section 2.5. He furthermore proved that it is impossible to have a set of three orientation coordinates that avoids singularity. Nevertheless, many in the mechanical system dynamics community have continued the search for three utopian parameters, while using Euler angles and their numerous variants well into the 20th century. Acknowledging the singularity at  $\theta = 0$  that is demonstrated herein, users of Euler angles monitor this critical condition and, as it is approached, halt simulation, define a new body reference frame, restructure the multibody system model, and restart the simulation.

Late in the twentieth century, the use of four-dimensional Euler parameters to avoid singularities became more common. It has become increasingly clear that the algebraic properties of Euler parameters, based on the quadratic character of entries in the orientation transformation matrix, rather than products of transcendental functions of Euler angles, greatly simplifies expressions for derivatives of functions that are involved in kinematics and dynamics. These and related considerations associated with computer formulation and solution of high dimensional multibody dynamics models show that the penalty associated with four dimensional Euler parameters is more than offset by their analytical simplicity and computational efficiency.

### 2.A.1 Definition of Euler Angles

The sequence of three rotations shown in Fig. 2.A.1 defines an orientation of the  $x' - y' - z'$  frame, relative to the  $x - y - z$  reference frame. With the  $x' - y' - z'$  frame initially coincident with the  $x - y - z$  reference frame, *c*ounterclockwise rotation by angle  $\phi$  about the positive  $z$  axis is implemented to form the  $x'' - y'' - z''$  frame shown in Fig. 2.A.1. Next, counterclockwise rotation by angle  $\theta$  about the positive  $x''$  axis is implemented, to obtain the

$x'' - y'' - z''$  frame. Finally, counterclockwise rotation by angle  $\psi$  about the positive  $z''$  axis is implemented to obtain the desired  $x' - y' - z'$  frame, in its general orientation shown in Fig. 2.A.1.

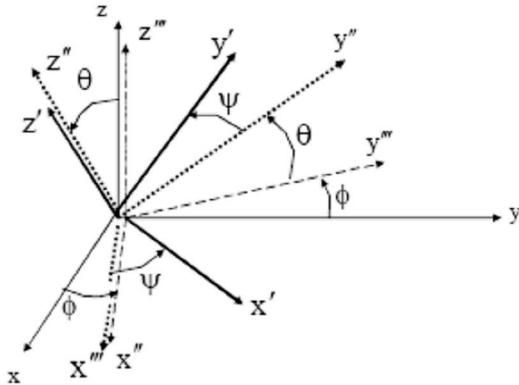


Figure 2.A.1 Euler Angle Definition

To develop analytical expressions for elements of the orientation transformation matrix **A** as functions of Euler angles, each of the rotations is considered separately. The first rotation is shown in Fig. 2.A.2. Since this is precisely the rotation defined in Section 2.3, the results of calculations carried out there; i.e., Eq. (2.3.27), yield the transformation

$$\mathbf{s} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{s}'' \equiv \mathbf{B}\mathbf{s}'' \quad (2.A.1)$$

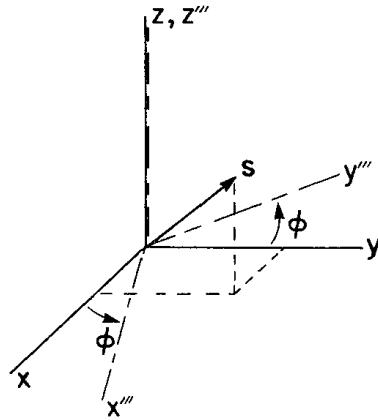


Figure 2.A.2 First Euler Angle

The second Euler angle rotation is shown in Fig. 2.A.3. Much as in the case of the first rotation treated in Section 2.3, the direction cosine transformation is

$$\mathbf{s}''' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \mathbf{s}'' \equiv \mathbf{Cs}'' \quad (2.A.2)$$

Finally, the third Euler angle rotation is shown in Fig. 2.A.4. Since it is identical in form to the first, the direction cosine transformation relation is obtained as

$$\mathbf{s}'' = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{s}' \equiv \mathbf{Ds}' \quad (2.A.3)$$

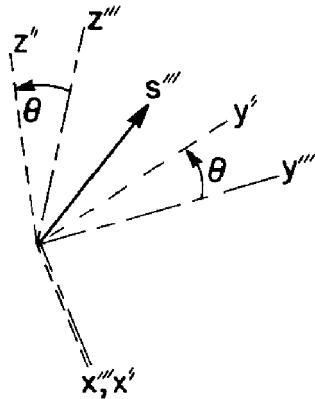


Figure 2.A.3 Second Euler Angle

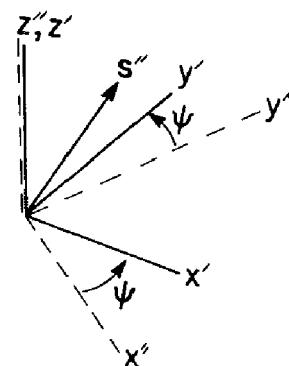


Figure 2.A.4 Third Euler Angle

Substituting the result of Eq. (2.A.3) into Eq. (2.A.2) and subsequently into Eq. (2.A.1), the transformation from the  $x' - y' - z'$  frame to the  $x - y - z$  reference frame is

$$\mathbf{s} = \mathbf{BCDs}' \equiv \mathbf{As}' \quad (2.A.4)$$

so

$$\mathbf{A} = \mathbf{BCD} \quad (2.A.5)$$

Substituting for **B**, **C**, and **D** from Eqs. (2.A.1) through (2.A.3) into Eq. (2.A.5) and carrying out the multiplications,

$$\mathbf{A} = \begin{bmatrix} \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi & -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi & \sin\theta\sin\phi \\ \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi & -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi & -\sin\theta\cos\phi \\ \sin\theta\sin\psi & \sin\theta\cos\psi & \cos\theta \end{bmatrix} \quad (2.A.6)$$

Equation (2.A.4) provides the desired Euler angle orientation transformation, where **A** is defined by Eq. (2.A.6).

## 2.A.2 Euler Angle Singularity

In order to resolve the issue of singularity of the transformation from **A** to Euler angles, consider the rotation sequence shown in Fig. 2.A.5 with  $\theta = 0$ . In this special case, the first and third rotation angles,  $\phi$  and  $\psi$ , are about the same axis. Therefore, for any  $\alpha$ ,  $\phi$  can be replaced by  $\phi + \alpha$  and  $\psi$  can be replaced by  $\psi - \alpha$  to yield the same orientation transformation

matrix. Thus, the Euler angles are not uniquely determined, even locally, by the orientation transformation matrix when  $\theta = 0$ . The transformation from  $\mathbf{A}$  to Euler angles is thus singular when  $\theta = 0$  or any multiple of  $\pi$ . This is the *Euler angle singularity*.

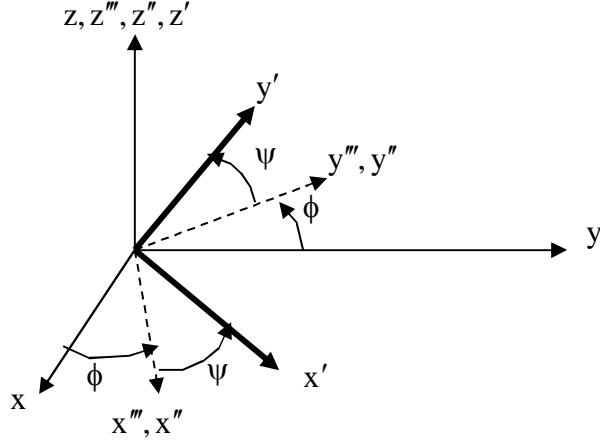


Figure 2.A.5 Euler Angle Rotations With  $\theta = 0$

Since angular velocity and angular acceleration are of concern in kinematic and dynamic analysis, it is desired to obtain a relationship between the time derivative of Euler angles and *angular velocity*. First, the time derivative of  $\mathbf{A}$  is obtained by taking the time derivative of both sides of Eq. (2.A.5), using the product rule of differentiation,

$$\dot{\mathbf{A}} = \dot{\mathbf{B}}\mathbf{C}\mathbf{D} + \mathbf{B}\dot{\mathbf{C}}\mathbf{D} + \mathbf{B}\mathbf{C}\dot{\mathbf{D}} \quad (2.A.7)$$

Equation (2.4.49), using Eqs. (2.A.5) and (2.A.7), yields

$$\begin{aligned} \tilde{\boldsymbol{\omega}}' &= \mathbf{A}^T \dot{\mathbf{A}} \\ &= \mathbf{D}^T \mathbf{C}^T \mathbf{B}^T (\dot{\mathbf{B}}\mathbf{C}\mathbf{D} + \mathbf{B}\dot{\mathbf{C}}\mathbf{D} + \mathbf{B}\mathbf{C}\dot{\mathbf{D}}) \\ &= \mathbf{D}^T \mathbf{C}^T \mathbf{B}^T \dot{\mathbf{B}}\mathbf{C}\mathbf{D} + \mathbf{D}^T \mathbf{C}^T \dot{\mathbf{C}}\mathbf{D} + \mathbf{D}^T \dot{\mathbf{D}} \end{aligned} \quad (2.A.8)$$

where the fact that  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are orthogonal matrices has been used. Beginning with the third term on the right of Eq. (2.A.8), direct calculation yields

$$\mathbf{D}^T \dot{\mathbf{D}} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\sin \psi & -\cos \psi & 0 \\ \cos \psi & -\sin \psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} \quad (2.A.9)$$

Similarly,

$$\mathbf{D}^T \mathbf{C}^T \dot{\mathbf{C}}\mathbf{D} = \mathbf{D}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{D} \dot{\theta} = \begin{bmatrix} 0 & 0 & -\sin \psi \\ 0 & 0 & -\cos \psi \\ \sin \psi & \cos \psi & 0 \end{bmatrix} \dot{\theta} \quad (2.A.10)$$

Finally, carrying out a somewhat longer expansion yields

$$\mathbf{D}^T \mathbf{C}^T \mathbf{B}^T \dot{\mathbf{B}} \mathbf{C} \mathbf{D} = \begin{bmatrix} 0 & -\cos\theta & \cos\psi\sin\theta \\ \cos\theta & 0 & -\sin\theta\sin\psi \\ -\cos\psi\sin\theta & \sin\theta\sin\psi & 0 \end{bmatrix} \dot{\phi} \quad (2.A.11)$$

Substituting from Eqs. (2.A.9) to (2.A.11) into Eq. (2.A.8) yields a  $3 \times 3$  matrix for  $\tilde{\omega}'$ ,

$$\tilde{\omega}' = \begin{bmatrix} 0 & -\cos\theta & \cos\psi\sin\theta \\ \cos\theta & 0 & -\sin\theta\sin\psi \\ -\cos\psi\sin\theta & \sin\theta\cos\psi & 0 \end{bmatrix} \dot{\phi} + \begin{bmatrix} 0 & 0 & -\sin\psi \\ 0 & 0 & -\cos\psi \\ \sin\psi & \cos\psi & 0 \end{bmatrix} \dot{\theta} + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} \quad (2.A.12)$$

Since the matrix on the right of Eq. (2.A.12) is skew symmetric, components of angular velocity  $\omega'$  are identified, using Eq. (2.1.22) as

$$\omega' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\psi} + \begin{bmatrix} \cos\psi \\ -\sin\psi \\ 0 \end{bmatrix} \dot{\theta} + \begin{bmatrix} \sin\psi\sin\theta \\ \cos\psi\sin\theta \\ \cos\theta \end{bmatrix} \dot{\phi} \quad (2.A.13)$$

To determine  $\dot{\phi}$ ,  $\dot{\theta}$ , and  $\dot{\psi}$ , when  $\omega'$  is known, Eq. (2.A.13) may be written as the matrix equation

$$\begin{bmatrix} \sin\psi\sin\theta & \cos\psi & 0 \\ \cos\psi\sin\theta & -\sin\psi & 0 \\ \cos\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \omega' \quad (2.A.14)$$

To determine whether a solution exists, the determinant of the coefficient matrix on the left may be evaluated and found to be  $-\sin\theta$ . Therefore, there exists a unique solution for Euler angle time derivatives for any angular velocity  $\omega'$ , except when  $\theta$  is equal to 0 or an integral multiple of  $\pi$ . In these singular cases, not every angular velocity can be achieved. For those that can be represented, an infinite family of Euler angle time derivatives results. For analytical purposes, Eq. (2.A.14) may be solved to obtain

$$\begin{aligned} \dot{\phi} &= \frac{\omega_y \cos\psi + \omega_x \sin\psi}{\sin\theta} \\ \dot{\theta} &= \omega_x \cos\psi - \omega_y \sin\psi \\ \dot{\psi} &= \omega'_z - \frac{(\omega_y \cos\psi + \omega_x \sin\psi) \cos\theta}{\sin\theta} \end{aligned} \quad (2.A.15)$$

which shows that a singular case occurs when  $\theta$  is an integral multiple of  $\pi$ , including 0.

### 2.A.3 Complexity of Computation with Euler Angles

Let  $\mathbf{a}'$  be a vector fixed in a rigid body from the origin of the  $x'$ - $y'$ - $z'$  frame to point P. The global vector representation of  $\mathbf{a}'$  is

$$\mathbf{a} = \mathbf{A}(\alpha) \mathbf{a}' \quad (2.A.16)$$

where  $\mathbf{a} = [\phi \quad \theta \quad \psi]^T$  is the vector of Euler angles that define orientation of the body and  $\mathbf{A}(\mathbf{a})$  is the orientation transformation matrix given by Eq. (2.A.6). For purposes of illustration, the x component of  $\mathbf{a}$  is expanded to obtain

$$a_x = (\cos \psi \cos \phi - \cos \theta \sin \phi \sin \psi) a'_x - (\sin \psi \cos \phi + \cos \theta \sin \phi \cos \psi) a'_y + \sin \theta \sin \phi a'_z \quad (2.A.17)$$

Differentiating both sides of Eq. (2.A.17) with respect to time yields the x component of velocity of point P as

$$\begin{aligned} \dot{a}_x = & [(-\cos \psi \sin \phi - \cos \theta \cos \phi \sin \psi) a'_x - (-\sin \psi \sin \phi + \cos \theta \cos \phi \cos \psi) a'_y + \sin \theta \cos \phi a'_z] \dot{\phi} \\ & + [\sin \theta \sin \phi \sin \psi a'_x + \sin \theta \sin \phi \cos \psi a'_y + \cos \theta \sin \phi a'_z] \dot{\theta} \\ & + [(-\sin \psi \cos \phi - \cos \theta \sin \phi \cos \psi) a'_x - (\cos \psi \cos \phi - \cos \theta \sin \phi \sin \psi) a'_y] \dot{\psi} \end{aligned} \quad (2.A.18)$$

Taking the time derivative of both sides of Eq. (2.A.18) yields the x component of acceleration of point P, including terms that are linear in second time derivatives of Euler angles and a quadratic form in their first derivatives,

$$\begin{aligned} \ddot{a}_x = & [(-\cos \psi \sin \phi - \cos \theta \cos \phi \sin \psi) a'_x - (-\sin \psi \sin \phi + \cos \theta \cos \phi \cos \psi) a'_y + \sin \theta \cos \phi a'_z] \ddot{\phi} \\ & + [\sin \theta \sin \phi \sin \psi a'_x + \sin \theta \sin \phi \cos \psi a'_y + \cos \theta \sin \phi a'_z] \ddot{\theta} \\ & + [(-\sin \psi \cos \phi - \cos \theta \sin \phi \cos \psi) a'_x - (\cos \psi \cos \phi - \cos \theta \sin \phi \sin \psi) a'_y] \ddot{\psi} \\ & + [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}] \mathbf{Q}(\phi, \theta, \psi, \mathbf{a}') \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \quad (2.A.19)$$

where  $\mathbf{Q}(\phi, \theta, \psi, \mathbf{a}')$  is the symmetric matrix

$$\mathbf{Q}(\phi, \theta, \psi, \mathbf{a}') = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ \text{Sym} & & Q_{33} \end{bmatrix} \quad (2.A.20)$$

with elements as follows:

$$\begin{aligned} Q_{11} &= (-\cos \psi \cos \phi + \cos \theta \sin \phi \sin \psi) a'_x + (\sin \psi \cos \phi + \cos \theta \sin \phi \cos \psi) a'_y - \sin \theta \sin \phi a'_z \\ Q_{12} &= \sin \theta \sin \phi \sin \psi a'_x + \sin \theta \cos \phi \cos \psi a'_y + \cos \theta \cos \phi a'_z \\ Q_{13} &= (\sin \psi \sin \phi - \cos \theta \cos \phi \cos \psi) a'_x + (\cos \psi \sin \phi + \cos \theta \cos \phi \sin \psi) a'_y \\ Q_{21} &= \cos \theta \sin \phi \sin \psi a'_x + \cos \theta \sin \phi \cos \psi a'_y - \sin \theta \sin \phi a'_z \\ Q_{23} &= \sin \theta \sin \phi \cos \psi a'_x - \sin \theta \sin \phi \sin \psi a'_y \\ Q_{33} &= (-\sin \psi \cos \phi + \sin \theta \sin \phi \cos \psi) a'_x + (\sin \psi \cos \phi - \cos \theta \sin \phi \cos \psi) a'_y \end{aligned} \quad (2.A.21)$$

It is instructive to compare the expressions for  $\ddot{a}_x$  in Eqs. (2.A.19) through (2.A.21) for Euler angles with corresponding expressions for Euler parameters. For  $\mathbf{a} = \mathbf{A}(\mathbf{p})\mathbf{a}'$ , using Euler parameter relations of Section 2.6,

$$\ddot{\mathbf{a}} = \mathbf{B}(\mathbf{p}, \mathbf{a}') \ddot{\mathbf{p}} + \mathbf{B}(\dot{\mathbf{p}}, \mathbf{a}') \dot{\mathbf{p}} \quad (2.A.22)$$

Expanding the first component of both sides of Eq. (2.A.22), using  $\mathbf{p} = [e_0 \ e_1 \ e_2 \ e_3]^T$  and definitions of Section 2.6, yields

$$\begin{aligned} \ddot{a}_x &= 2 \left[ \begin{array}{l} (e_0 a'_x - e_2 a'_y + e_3 a'_x) \ddot{e}_0 + (e_1 a'_x + e_2 a'_z + e_3 a'_y) \ddot{e}_1 \\ + (e_1 a'_y + e_0 a'_x - e_3 a'_x) \ddot{e}_2 + (e_1 a'_z - e_0 a'_y - e_2 a'_x) \ddot{e}_3 \end{array} \right] \\ &\quad + \dot{\mathbf{p}}^T \begin{bmatrix} 2a'_x & 0 & (a'_z - a'_y) & (a'_z - a'_y) \\ & 2a'_x & (a'_z + a'_y) & (a'_z + a'_y) \\ & & 0 & -2a'_x \\ \text{Sym} & & & 0 \end{bmatrix} \dot{\mathbf{p}} \end{aligned} \quad (2.A.23)$$

As expected, acceleration is a linear function of Euler parameter second derivatives, with coefficients in Eq. (2.A.23) that are linear in Euler parameters. In contrast, the coefficients of Euler angle second derivatives in Eq. (2.A.19) are products of numerous transcendental functions (sin and cos) of Euler angles. The contrast between coefficients of the velocity quadratic forms in Eqs. (2.A.20) and (2.A.21) for Euler angles and in Eq. (2.A.23) for Euler parameters is even more stark. The Euler parameter coefficients in Eq. (2.A.23) are constants, whereas the coefficients in Eq. (2.A.21) involve many products of two or three transcendental functions of Euler angles.

The contrast in derivative complexity between Euler angles and Euler parameters is even more extreme in equations of kinematic constraint and numerical integration in dynamics. Kinematic constraint expressions are generally far more complicated than the orientation transformation used in the above analysis, making calculation of the two derivatives required to obtain acceleration equations much more intricate. Furthermore, in implementation of implicit numerical integration methods that are required for stiff mechanical systems in dynamic simulation, a third derivative of constraint expressions is required. As shown in Sections 2.5 and 2.6, and amplified in Chapter 3, Euler parameter identities based on the quadratic dependence of orientation transformations make calculation of the numerous required derivatives efficient and practical. In contrast, calculation of the required derivatives using Euler angles is a nightmare.

## Appendix 2.B Position and Orientation Code

Brief instructions for use of Codes 2.2, 2.4. 2.5, and 2.6 that implement methods presented in corresponding sections of Chapter 2 are provided. MATLAB versions of the codes are provided and made available for use in implementations throughout the text.

### 2.B.1 Code 2.2 Newton Raphson Solution of k Equations $F(x) = 0$ for k Variables $x$

Code 2.2 for solution of k equations  $F(x) = \mathbf{0}$  for k variables  $\mathbf{x}$ , using the Newton-Raphson method of Section 2.2.7, consists of the main code, lines 1 through 24, and two functions for definition of  $F(\mathbf{x})$  and  $F_x(\mathbf{x})$  shown in Fig. 2.B.1.1. While the code can be used stand alone, it is intended for incorporation in codes focused on kinematics and dynamics applications, in which MATLAB functions for evaluation of  $F(\mathbf{x})$  and its Jacobian are generally coded. In application of the code, the user must select the error tolerance  $xtol$  in line 5, the maximum number of iterations to be allowed  $itermax$  in line 6, the function

$F(\mathbf{x}) = [F_1(\mathbf{x}) \quad \dots \quad F_k(\mathbf{x})]^T$  in function  $FEval$ , and the Jacobian  $F_x(\mathbf{x}) = [\partial F_i(\mathbf{x}) / \partial x_j]$  in function  $FxEval$ .

```
1 % Newton Raphson Solution of k Equations F(x)=0 for k Variables x
2 %Section 2.2.7
3 %Requires Functions FEval to evaluate F(x) and FxEval to evaluate
4 % Jacobian dF/dx(x)
5 xtol=0.001; % Error tolerance in satisfying F(x)=0
6 itermax=10; %Maximum number of iterations allowed
7 x0=[]; % Initial solution estimate, a k vector
8 x=x0;
9 i=1; %Iteration counter
10 err=xtol+1; %Set initial error tolerance large
11 while err > xtol %Iteration for x, through line 15
12 F=FEval(x); %Evaluate F(x)
13 Fx=FxEval(x); %Evaluate Jacobian dF/dx(x)
14 delx=-F\Ix; %Newton-Raphson iteration
15 x=x+delx;
16 err=norm(F); %Error evaluation
17 if i > itermax %Terminate if iterations exceed maximum allowed
18 warning( 'iteration limit exceeded' );
19 str = string( "iteration limit exceeded " );
20 return
21 end
22 i=i+1;
23 end
24 iter=i-1; %Report number of iterations

1 function F=FEval(x)
2 % Evaluation of F(x)
3 F=[]; %Enter k by 1 matrix [Fi(x)]
4 end
```

```

1 function Fx=FxEval(x)
2 % Evaluation of Fx(x)=partial derivative matrix of F with respect to x
3 Fx=[]; %Enter k by k matrix [dFi/dxj(x)]
4 end

```

Figure 2.B.1.1 Newton Raphson Solution of k Equations  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$  for k Variables  $\mathbf{x}$

## 2.B.2 Code 2.4 Computation of Relative Rotation About Common $\mathbf{z}'$ -Axes

Code 2.4 for computation of the relative angle of rotation  $\theta_{ij}$  between  $\mathbf{x}_i'$  and  $\mathbf{x}_j'$  axes for reference frames with common  $\mathbf{z}'$  axes in Fig. 2.4.8 in Section 2.4.4 is shown in lines 1 through 31 in the following, using MATLAB functions for evaluation of vector and matrix operations that are contained in a library provided in Code 2.6 of Section 2.B.4. The user must provide the body fixed vectors  $\mathbf{f}_i'$ ,  $\mathbf{g}_i'$ , and  $\mathbf{f}_j'$ , denoted f1pr, g1pr, and f2pr in the code and Euler parameters  $\mathbf{p}^i$  and  $\mathbf{p}^j$  for the bodies, denoted p1 and p2 in the code.

```

1 %Given unit vectors f1pr, g1pr and f2pr and Euler parameters p1 and p2
2 %for bodies 1 and 2 with common z1 and z2 axes, evaluate relative rotation
3 %Section 2.4.4
4 %Data are for Numerical Example
5 f1pr=[1;0;0];
6 g1pr=[0;1;0];
7 f2pr=[1;0;0];
8 p1=[1;0;0;0];
9 e02=cos(pi/4);
10 e2=sin(pi/4)*[0;0;1];
11 p2=[e02;e2];
12 p1d=-0.5*GT(p1)'*[0;0;1];
13 p2d=0.5*GT(p2)'*[0;0;1];
14 %Compute theta12 in Eq. (2.4.37)
15 A1=AT(p1); %AT(p) is the orientation transformation matrix as function of p
16 A2=AT(p2);
17 f1=A1*f1pr;
18 g1=A1*g1pr;
19 f2=A2*f2pr;
20 s=g1'*f2; %definition of sin and cos
21 c=f1'*f2;
22 Arcs=asin(s); %asin(s) is Arcsin(s)
23 if s<=0&&c<0
24 theta12=-pi-Arcs;
25 end
26 if c>=0
27 theta12=Arcs;
28 end
29 if s>=0&&c<0
30 theta12=pi-Arcs;
31 end

```

Figure 2.B.2.1 Computation of Relative Rotation About Common  $\mathbf{z}'$ -Axes

### 2.B.3 Code 2.5 Definition of A and Solution for p with Given A

Code 2.5 for definition of orientation transformation matrix  $\mathbf{A}$  and determining  $\mathbf{p}$  for a given  $\mathbf{A}$ , using theory presented in Section 2.5, is presented in annotated form below, using MATLAB functions for evaluation of vector and matrix operations that are contained in a library provided in Code 2.6 of Section 2.B.4. Part 1 of the code defines the orthogonal orientation transformation matrix by either direct entry, using the point definition approach presented in Section 2.5.5, or using data from Euler's theorem of Section 2.5.3. Part 2 of the code computes a vector  $\mathbf{p}$  of Euler parameters from a given orthogonal orientation transformation matrix  $\mathbf{A}$ , such that  $\mathbf{A} = \mathbf{A}(\mathbf{p})$ , using results of the Inverse Euler Parameter Mapping Theorem in Section 2.5.4. The code can be used in one call to compute  $\mathbf{A}$ , using either the point definition approach or Euler's theorem, and in a second call to evaluate  $\mathbf{p}$ .

```

1 %AA_Evaluate_Spatial_Orientation_Transformation_Matrix_and_Inverse_for_p
2 %Section 2.5
(Select option that A is to be determined in Part 1.)
3 Part=[]; %Part=1, Define orthogonal transformation matrix
4 %Part=2, Compute p for given orthogonal transformation matrix A

6 if Part==1 %Define orthogonal transformation matrix
(Select from among three modes for determination of A.)
7 mode=[]; %mode=1, enter A; mode=2, point definition of A, Section 2.5.5;
8 %mode=3, Euler'S Theorem, Section 2.5.3
9 if mode==1
10 A=[,,,...]; %Enter the 3x3 orthogonal matrix A
11 end
12 if mode==2 %Define orthogonal transformation matrix A using point
13 %definition of Section 2.5.5.
14 rO=[;;]; %Enter vector rO to origin
15 rP=[;;]; %Enter vector rP to point P
16 rQ=[;;]; %Enter vector rQ to point Q
17 f=(1/norm(rP-rO))*(rP-rO);
18 h=(1/norm(atil(f)*(rQ-rO)))*atil(f)*(rQ-rO);
19 g=-atil(f)*h;
20 A=[f,g,h];
21 end
22 if mode==3 %Define orthogonal transformation matrix A using Euler's
23 %Theorem of Section 2.5.3.
24 %Unit vector u about which rotation angle chi brings x-y-z frame into
25 %coincidence with x'-y'-z' frame
26ubar=[;;]; %Enter vector ubar about which rotation occurs,
27 %not necessarily normalized
28 chi=[]; %Enter angle chi of rotation
29 u=(1/norm(ubar))*ubar; %Normalization of ubar
30 e0=cos(chi/2);
31 e=sin(chi/2)*u;
32 A=(e0^2-e'*e)*eye(3)+2*e*e'+2*e0*atil(e);
33 end
34 end

```

**(Select option that p is to be determined in Part 1.)**

```
36 if Part==2 %Given orthogonal matrix A, compute Euler parameter vector p in  
37 %Section 2.5.4 so A=A(p)  
38 A=[7/25,24/25,0;24/25,-7/25,0;0,0,-1]; %Enter the 3x3 orthogonal matrix A  
39 [a11,a12,a13,a21,a22,a23,a31,a32,a33] = partA(A);  
40 trA=a11+a22+a33;
```

**(Determine if A is not symmetric, by far the most common case.)**

```
41 if norm(A'-A)>0.0001 %A is not symmetric  
42 e0=0.5*sqrt(trA+1);  
43 e=(1/(2*sqrt(trA+1)))*[a32-a23;a13-a31;a21-a12];  
44 end
```

**(Determine if A is symmetric, a rare case.)**

```
45 if norm(A'-A)<0.0001 %A is symmetric
```

```
46 e0=0.5*sqrt(trA+1);
```

**(Determine if e0 is unity, in which case e is zero)**

```
47 if e0==1
```

```
48 e=zeros(3,1);
```

```
49 end
```

**(Determine if e0 is zero)**

```
50 if e0==0
```

```
51 ae1=sqrt((a11+1)/2);
```

```
52 ae2=sqrt((a22+1)/2);
```

```
53 ae3=sqrt((a33+1)/2);
```

```
54 if ae1>=ae2 && ae1>=ae3
```

```
55 e1=ae1;
```

```
56 if ae2==0
```

```
57 e2=0;
```

```
58 else
```

```
59 e2=ae2*sign(a12);
```

```
60 end
```

```
61 if ae3==0
```

```
62 e3=0;
```

```
63 else
```

```
64 e3=ae3*sign(a13);
```

```
65 end
```

```
66 end
```

```
67 if ae2>=ae1 && ae2>=ae3
```

```
68 e2=ae2;
```

```
69 if ae1==0
```

```
70 e1=0;
```

```
71 else
```

```
72 e1=ae1*sign(a12);
```

```
73 end
```

```
74 if ae3==0
```

```
75 e3=0;
```

```
76 else
```

```
77 e3=ae3*sign(a23);
```

```
78 end
```

```
79 end
```

```
80 if ae3>=ae1 && ae3>=ae2
```

```
81 e3=ae3;
```

```
82 if ae1==0
```

```
83 e1=0;
```

```

84 else
85 e1=ae1*sign(a13);
86 end
87 if ae2==0
88 e2=0;
89 else
90 e2=ae2*sign(a23);
91 end
92 end
93 end
94 e=[e1;e2;e3];
95 p=[e0;e];
96 end
97 end

```

Figure 2.B.3.1 Definition of A and Solution for p with Given A

#### **2.B.4 Code 2.6 Euler Parameter Identities and Derivative Operators**

Euler parameter identities and derivative operators of Sections 2.5 and 2.6 are implemented in the form of MATLAB functions in Code 2.5, for incorporation into applications throughout kinematics and dynamics of spatial systems, as follows:

```

1 function A=AT(p)
2 %Evaluate A(p) of Eq. (2.5.24), given p
3 e0=p(1);
4 e=[p(2);p(3);p(4)];
5 I3=eye(3);
6 atil=atil(e);
7 A=(e0^2-e'*e)*I3+2*e'*e'+2*e0*ertil;
8 end

1 function atil=atil(a)
2 %Evaluate til(a) of Eq. (2.1.21), given a
3 atil=[0,-a(3),a(2);a(3),0,-a(1);-a(2),a(1),0];
4 end

1 function BT=BT(p,apr)
2 %Evaluate B(p,apr) of Eq.(2.6.25), given p and apr
3 e0=p(1);
4 e=[p(2);p(3);p(4)];
5 I3=eye(3);
6 atil=atil(e);
7 BT=2*[(e0*I3+ertil)*apr,e'*apr'-(e0*I3+ertil)*atil(apr)];
8 end

1 function CT=CT(p,a)
2 %Evaluate C(p,apr) of Eq.(2.6.31), given p and apr
3 e0=p(1);
4 e=[p(2);p(3);p(4)];
5 I3=eye(3);
6 CT=2*[(e0*I3-ertil)*a,e'*a'+(e0*I3-ertil)*atil(a)];
7 end

```

```

1 function E=ET(p)
2 %Evaluate E(p) of Eq.(2.6.1), given p
3 e0=p(1);
4 e=[p(2);p(3);p(4)];
5 E=[-e,atil(e)+e0*eye(3)];
6 end

1 function G=GT(p)
2 %Evaluate G(p) of Eq.(2.6.2), given p
3 e0=p(1);
4 e=[p(2);p(3);p(4)];
5 G=[-e,-atil(e)+e0*eye(3)];
6 end

1 function K = KT(apr,b)
2 %Evaluate K(apr,b) of Eq.(2.6.37), given apr and b
3 K=2*[apr'*b,apr'*atil(b);atil(apr)*b,apr*b'+b*apr'-apr'*b*eye(3)];
4 end

1 function L = LT(a,b)
2 %Evaluate L(a,b) of Eq.(2.6.41), given a and b
3 K=2*[a'*b,-a'*atil(b);-atil(a)*b,a*b'+b*a'-a'*b*eye(3)];
4 end

1 function M=MT(p1,p2)
2 %Evaluate M(p1,p2) of Eq.(2.6.81), given p1 and p2
3 M=2*ET(p1)*GT(p2)';
4 end

1 function N=NT(p1,p2)
2 %Evaluate N(p1,p2) of Eq.(2.6.82), given p1 and p2
3 N=2*GT(p1)*ET(p2)';
4 end

1 function R=RT(a)
2 %Evaluate R(a) of Eq.(2.6.19), given a
3 R=[0,-a';a,atil(a)];
4 end

1 function T=TT(a)
2 %Evaluate T(a) of Eq.(2.6.20), given a
3 T=[0,-a';a,-atil(a)];
4 end

1 function Y=YT(a,b)
2 %Evaluate Y(a,b) of Eq.(2.6.83), given a and b
3 Y=-2*GT(ET(a)'*b);
4 end

```

```
1 function Z=ZT(a,b)
2 %Evaluate Z(a,b) of Eq.(2.6.83), given a and b
3 Z=-2*ET(GT(a)'*b);
4 end
```

Figure 2B.4.1 Euler Parameter Identities and Derivative Operators

## Appendix 2.C Key Formulas, Chapter 2

$$\tilde{\mathbf{a}} \equiv \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (2.1.22)$$

$$\mathbf{B} = \begin{bmatrix} 0 & b_{12} & b_{13} \\ -b_{12} & 0 & b_{23} \\ -b_{13} & -b_{23} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \equiv \tilde{\mathbf{b}} = \begin{pmatrix} -b_{23} \\ b_{13} \\ -b_{12} \end{pmatrix} \quad (2.1.23)$$

$$\tilde{\mathbf{a}}^T = -\tilde{\mathbf{a}} \quad \tilde{\mathbf{a}}\tilde{\mathbf{b}} = -\tilde{\mathbf{b}}\tilde{\mathbf{a}} \quad \tilde{\mathbf{a}}\tilde{\mathbf{a}} = 0 \quad (2.1.25) \quad (2.1.28)$$

$$\tilde{\mathbf{a}}\tilde{\mathbf{b}} = \mathbf{b}\mathbf{a}^T - \mathbf{a}^T\mathbf{b}\mathbf{I} \quad \tilde{\mathbf{a}}\tilde{\mathbf{b}} = \tilde{\mathbf{a}}\tilde{\mathbf{b}} - \tilde{\mathbf{b}}\tilde{\mathbf{a}} \quad (2.1.30) \quad (2.1.32)$$

$$\tilde{\mathbf{a}}\tilde{\mathbf{b}} + \mathbf{a}\mathbf{b}^T = \tilde{\mathbf{b}}\tilde{\mathbf{a}} + \mathbf{b}\mathbf{a}^T \quad (2.1.33)$$

$$a_q \equiv \frac{\partial a}{\partial \mathbf{q}} = \left[ \frac{\partial a}{\partial q_j} \right]_{1 \times k} \quad f_q \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \left[ \frac{\partial f_i}{\partial q_j} \right]_{n \times k} \quad (2.2.34) \quad (2.2.35)$$

$$(\mathbf{g}^T \mathbf{h})_q = \mathbf{h}^T \mathbf{g}_q + \mathbf{g}^T \mathbf{h}_q \quad (\Phi(\mathbf{g}(\mathbf{q})))_q = \Phi_g \mathbf{g}_q \quad (2.2.36) \quad (2.2.37)$$

$$R^n = \left\{ \mathbf{q} = [q_1 \quad \cdots \quad q_n]^T : q_i \text{ real}, i = 1, \dots, n \right\} \quad (2.2.41)$$

$$\langle \mathbf{r}, \mathbf{s} \rangle \equiv \mathbf{r}^T \mathbf{s} = \sum_{i=1}^n r_i s_i \quad d(\mathbf{r}, \mathbf{s}) \equiv \|\mathbf{r} - \mathbf{s}\|_n \quad \|\mathbf{q}\|_n \equiv \sqrt{\langle \mathbf{q}, \mathbf{q} \rangle} \quad (2.2.43) \quad (2.2.44) \quad (2.2.45)$$

$$B_\varepsilon(\mathbf{q}^0) = \left\{ \mathbf{q} \in R^n : \|\mathbf{q} - \mathbf{q}^0\| < \varepsilon \right\} \quad (2.2.47)$$

$$\mathbf{A} = \mathbf{A}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \quad \mathbf{P} = \mathbf{A}(\pi/2) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (2.3.27) \quad (2.3.29)$$

$$\mathbf{s} = \mathbf{A}\mathbf{s}' \quad \mathbf{P}\mathbf{P} = -\mathbf{I} \quad \mathbf{P}\mathbf{A} = \mathbf{A}\mathbf{P} \quad (2.3.12) \quad (2.3.32) \quad (2.3.33)$$

$$\frac{d}{d\phi} \mathbf{A}(\phi) = \mathbf{P}\mathbf{A}(\phi) \quad \frac{d}{d\phi} \mathbf{A}^T(\phi) = -\mathbf{A}^T(\phi)\mathbf{P} \quad (2.3.31)$$

$$\mathbf{r}^P = \mathbf{r} + \mathbf{A}(\phi)\mathbf{s}'^P \quad (2.3.40)$$

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\phi}\mathbf{P}\mathbf{A}(\phi)\mathbf{s}'^P \quad \ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} + \ddot{\phi}\mathbf{P}\mathbf{A}(\phi)\mathbf{s}'^P - \dot{\phi}^2 \mathbf{A}(\phi)\mathbf{s}'^P \quad (2.3.42) \quad (2.3.43)$$

$$\delta \mathbf{A}(\phi) = \delta \phi \mathbf{P}\mathbf{A}(\phi) \quad \delta \mathbf{r}^P = \delta \mathbf{r} + \delta \mathbf{A}(\phi)\mathbf{s}'^P = \delta \mathbf{r} + \delta \phi \mathbf{P}\mathbf{A}(\phi)\mathbf{s}'^P \quad (2.3.44) \quad (2.3.45)$$

$$s = As' \quad s' = A^T s \quad A^T A = I \quad (2.4.12) \quad (2.4.20) \quad (2.4.19)$$

$$r^p = r + s^p = r + As'^p \quad (2.4.22)$$

$$\tilde{s} = A\tilde{s}'A^T \quad \tilde{s}' = A^T \tilde{s} A \quad (2.4.27) \quad (2.4.28)$$

$$\dot{r}^p = \dot{r} + \tilde{\omega}s^p \quad \dot{r}^p = \dot{r} + A\tilde{\omega}'s'^p \quad (2.4.44) \quad (2.4.47)$$

$$\dot{A} = \tilde{\omega}A \quad \dot{A} = A\tilde{\omega}' \quad (2.4.48) \quad (2.4.49)$$

$$\ddot{r}^p = \ddot{r} - \tilde{s}^p \dot{\omega} + \tilde{\omega}\tilde{\omega}s^p = \ddot{r} + A\tilde{\omega}'s'^p + A\tilde{\omega}'\tilde{\omega}'s'^p \quad (2.4.53)$$

$$\delta A = \delta \tilde{\pi} A \quad \delta A = A \delta \tilde{\pi}' \quad (2.4.58) \quad (2.4.61)$$

$$\delta r^p = \delta r + \delta \tilde{\pi} s^p = \delta r + A \delta \tilde{\pi}' s'^p \quad (2.4.63)$$

$$e_0 \equiv \cos \frac{\chi}{2} \quad e \equiv \sin \frac{\chi}{2} u \quad (2.5.21)$$

$$A = (e_0^2 - e^T e) I + 2ee^T + 2e_0 \tilde{e} \quad (2.5.22)$$

$$A = \begin{bmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_1 e_2 - e_0 e_3) & 2(e_1 e_3 + e_0 e_2) \\ 2(e_1 e_2 + e_0 e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2 e_3 - e_0 e_1) \\ 2(e_1 e_3 - e_0 e_2) & 2(e_2 e_3 + e_0 e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{bmatrix} \quad (2.5.23)$$

$$p = [e_0 \ e_1 \ e_2 \ e_3]^T = [e_0 \ e^T]^T \quad p^T p = 1 \quad p^T \dot{p} = 0 \quad (2.5.25) \quad (2.5.26)$$

$$e_0 = \pm \sqrt{\text{tr} A + 1} / 2 \quad b = \begin{bmatrix} a_{32} - a_{23} \\ a_{13} - a_{31} \\ a_{21} - a_{12} \end{bmatrix} \quad (2.5.30)$$

(Only formulas below with bold italics require  $p^T p = 1$  )

$$E(p) \equiv [-e \ \tilde{e} + e_0 I] \quad G(p) \equiv [-e \ -\tilde{e} + e_0 I] \quad (2.6.1) \quad (2.6.2)$$

$$A(p) = E(p)G(p)^T \quad E(p)p = 0 \quad G(p)p = 0 \quad (2.6.4) \quad (2.6.5) \quad (2.6.6)$$

$$E(p)E(p)^T = p^T p I \quad G(p)G(p)^T = p^T p I \quad (2.6.7) \quad (2.6.8)$$

$$E(p)^T E(p) = p^T p I_4 - pp^T \quad G(p)^T G(p) = p^T p I_4 - pp^T \quad (2.6.9) \quad (2.6.10)$$

$$E(p_i)p_j = -E(p_j)p_i \quad G(p_i)p_j = -G(p_j)p_i \quad (2.6.14) \quad (2.6.15)$$

$$R(a) \equiv \begin{bmatrix} 0 & -a^T \\ a & \tilde{a} \end{bmatrix} \quad T(a) \equiv \begin{bmatrix} 0 & -a^T \\ a & -\tilde{a} \end{bmatrix} \quad (2.6.19) \quad (2.6.20)$$

$$B(p, a') \equiv 2[(e_0 I + \tilde{e})a' - ea'^T - (e_0 I + \tilde{e})\tilde{a}'] = 2E(p)T(a') \quad (2.6.25)$$

$$C(p, a) \equiv 2[(e_0 I - \tilde{e})a - ea^T + (e_0 I - \tilde{e})\tilde{a}] = 2G(p)R(a) \quad (2.6.31)$$

$$(A(p)a')_p = B(p, a') \quad (A(p)^T a)_p = C(p, a) \quad (2.6.26) \quad (2.6.30)$$

$$B(p_i, a')p_j = B(p_j, a')p_i \quad C(p_i, a)p_j = C(p_j, a)p_i \quad (2.6.28) \quad (2.6.32)$$

$$\mathbf{K}(\mathbf{a}', \mathbf{b}) \equiv 2 \begin{bmatrix} \mathbf{a}'^T \mathbf{b} & \mathbf{a}'^T \tilde{\mathbf{b}} \\ \tilde{\mathbf{a}}' \mathbf{b} & \mathbf{a}' \mathbf{b}^T + \mathbf{b} \mathbf{a}'^T - \mathbf{a}'^T \mathbf{b} \mathbf{I} \end{bmatrix} \left( \mathbf{B}(\mathbf{p}, \mathbf{a}')^T \gamma \right)_{\mathbf{p}} = \mathbf{K}(\mathbf{a}', \gamma) \quad (2.6.37) \quad (2.6.38)$$

$$\mathbf{L}(\mathbf{a}, \mathbf{b}) \equiv 2 \begin{bmatrix} \mathbf{a}^T \mathbf{b} & -\mathbf{a}^T \tilde{\mathbf{b}} \\ -\tilde{\mathbf{a}} \mathbf{b} & \mathbf{a} \mathbf{b}^T + \mathbf{b} \mathbf{a}^T - \mathbf{a}^T \mathbf{b} \mathbf{I} \end{bmatrix} \left( \mathbf{C}(\mathbf{p}, \mathbf{a})^T \mathbf{b} \right)_{\mathbf{p}} = \mathbf{L}(\mathbf{a}, \mathbf{b}) \quad (2.6.41) \quad (2.6.42)$$

$$(\mathbf{E}(\mathbf{p})\gamma)_{\mathbf{p}} = -\mathbf{E}(\gamma) \quad (2.6.43) \quad (2.6.44)$$

$$(\mathbf{E}(\mathbf{p})^T \mathbf{a})_{\mathbf{p}} = \mathbf{R}(\mathbf{a}) \quad (2.6.47) \quad (2.6.50)$$

$$\mathbf{p}^T \dot{\mathbf{p}} = 0 \quad \dot{\mathbf{A}} = 2\mathbf{E}(\mathbf{p})\dot{\mathbf{G}}^T = 2\dot{\mathbf{E}}\mathbf{G}(\mathbf{p})^T \quad (2.6.52) \quad (2.6.54)$$

$$\omega = 2\mathbf{E}(\mathbf{p})\dot{\mathbf{p}} \quad \omega' = 2\mathbf{G}(\mathbf{p})\dot{\mathbf{p}} \quad (2.6.59) \quad (2.6.63)$$

$$\dot{\omega} = 2\mathbf{E}(\mathbf{p})\ddot{\mathbf{p}} \quad \dot{\omega}' = 2\mathbf{G}(\mathbf{p})\ddot{\mathbf{p}} \quad (2.6.76) \quad (2.6.77)$$

$$\dot{\mathbf{p}} = \frac{1}{2}\mathbf{E}(\mathbf{p})^T \omega \quad \dot{\mathbf{p}} = \frac{1}{2}\mathbf{G}(\mathbf{p})^T \omega' \quad (2.6.61) \quad (2.6.64)$$

$$\ddot{\mathbf{p}} = \frac{1}{2}\mathbf{E}^T \omega - \frac{1}{4}\omega^T \omega \mathbf{p} \quad \ddot{\mathbf{p}} = \frac{1}{2}\mathbf{G}^T \omega' - \frac{1}{4}\omega'^T \omega' \mathbf{p} \quad (2.6.80) \quad (2.6.81)$$

$$\mathbf{p}^T \delta \mathbf{p} = 0 \quad \delta \mathbf{A} = 2\mathbf{E}(\mathbf{p})\delta \mathbf{G}^T = 2\delta \mathbf{E}\mathbf{G}(\mathbf{p})^T \quad (2.6.65) \quad (2.6.66)$$

$$\delta \pi = 2\mathbf{E}(\mathbf{p})\delta \mathbf{p} \quad \delta \pi' = 2\mathbf{G}(\mathbf{p})\delta \mathbf{p} \quad (2.6.68)$$

$$\delta \mathbf{p} = \frac{1}{2}\mathbf{E}(\mathbf{p})^T \delta \pi \quad \delta \mathbf{p} = \frac{1}{2}\mathbf{G}(\mathbf{p})^T \delta \pi' \quad (2.6.69)$$

$$\mathbf{G}(\mathbf{p})\mathbf{T}(\mathbf{a})^T = \tilde{\mathbf{a}}\mathbf{G}(\mathbf{p}) - \mathbf{a}\mathbf{p}^T \quad \mathbf{E}(\mathbf{p})\mathbf{R}(\mathbf{a})^T = -\tilde{\mathbf{a}}\mathbf{E}(\mathbf{p}) - \mathbf{a}\mathbf{p}^T \quad (2.6.82)$$

$$\mathbf{G}(\mathbf{p})\mathbf{B}(\mathbf{p}, \mathbf{a}')^T = 2\tilde{\mathbf{a}}'\mathbf{A}(\mathbf{p})^T \quad \mathbf{E}(\mathbf{p})\mathbf{C}(\mathbf{p}, \mathbf{a}')^T = -2\tilde{\mathbf{a}}'\mathbf{A}(\mathbf{p}) \quad (2.6.79)$$

$$(\mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i)\mathbf{p}_j)_{\mathbf{a}'_i} = 2 \left\{ \mathbf{E}(\mathbf{p}_i)\mathbf{G}(\mathbf{p}_j)^T \right\} \equiv \mathbf{M}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{M}(\mathbf{p}_j, \mathbf{p}_i) \quad (2.6.85)$$

$$(\mathbf{C}(\mathbf{p}_i, \mathbf{a}_i)\mathbf{p}_j)_{\mathbf{a}_i} = 2 \left\{ \mathbf{G}(\mathbf{p}_i)\mathbf{E}(\mathbf{p}_j)^T \right\} \equiv \mathbf{N}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{N}(\mathbf{p}_j, \mathbf{p}_i) \quad (2.6.86)$$

$$(\mathbf{M}(\mathbf{p}, \gamma)\mathbf{b})_{\mathbf{p}} = 2 \left( \mathbf{E}(\mathbf{p})\mathbf{G}(\gamma)^T \mathbf{b} \right)_{\mathbf{p}} = -2\mathbf{E}(\mathbf{G}(\gamma)^T \mathbf{b}) \equiv \mathbf{Z}(\gamma, \mathbf{b}) \quad (2.6.87)$$

$$(\mathbf{N}(\mathbf{p}, \gamma)\mathbf{b})_{\mathbf{p}} = 2 \left( \mathbf{G}(\mathbf{p})\mathbf{E}(\gamma)^T \mathbf{b} \right)_{\mathbf{p}} = -2\mathbf{G}(\mathbf{E}(\gamma)^T \mathbf{b}) \equiv \mathbf{Y}(\gamma, \mathbf{b})$$

$$(\mathbf{B}(\mathbf{p}, \mathbf{a}')^T \mathbf{b})_{\mathbf{a}'} = \mathbf{C}(\mathbf{p}, \mathbf{b})^T \quad (\mathbf{C}(\mathbf{p}, \mathbf{b})^T \mathbf{a}')_{\mathbf{b}} = \mathbf{B}(\mathbf{p}, \mathbf{a}')^T \quad (2.6.88) \quad (2.6.89)$$

## CHAPTER 3

### Kinematics of Multibody Systems

#### 3.0 Introduction

As outlined in Chapter 1, mechanical systems are often modeled as collections of rigid bodies that are connected by joints between pairs of bodies, to form *multibody systems*. A broadly applicable kinematic constraint formulation for such systems is presented in Section 3.1, using particles as bodies to illustrate concepts that form the foundation for progressively more complex planar and spatial multibody system kinematics in later sections. Joints between pairs of bodies are characterized by *algebraic constraint equations*, called *holonomic constraints*, comprising a library that can be used to model a broad spectrum of mechanical systems. Planar and spatial joints and associated holonomic constraints are presented in Sections 3.2 and 3.3, where it is shown that constraint equations derived for each joint imply the geometry of the joint; i.e., establishing equivalence of equations used in computation and kinematics of physical systems. Examples of planar and spatial systems modeled are presented.

Configuration spaces that represent all possible positions and orientations of bodies in a mechanical system, consistent with kinematic constraints, are introduced in Section 3.4. It is shown that singular configurations may arise due to a poor design or model. This is due to nonlinearity that is unavoidable in mechanical system kinematics and is associated with configurations in which the Jacobian matrix of constraint equation derivatives fails to have full rank. The *regular configuration space* is defined that eliminates such singular behavior and is the foundation for subsequent analytical and computational developments. *Tangent space parameterization* of the *regular configuration space* is presented in Section 3.5, yielding a singularity free foundation for modern kinematic and dynamic simulation methods. Basic concepts of differential geometry in a Euclidean space setting for kinematics and dynamics of mechanical systems are presented in Section 3.6, extending local properties of the regular configuration space defined in Section 3.5 to global singularity free manifolds of system kinematic functionality. Useful theoretical results are summarized from a mathematical literature that is increasingly distant from the realities of kinematics and dynamics of mechanical systems.

*Singular configurations* of poorly designed or poorly modeled mechanisms are analyzed in Section 3.7 and the influence of singularities on system kinematics is analyzed. A systematic formulation for kinematic analysis of position, velocity, and acceleration response to time dependent drivers that are imposed on a system is presented in Section 3.8.

General purpose MATLAB computer codes for kinematic analysis of planar and spatial systems are presented in Sections 3.9 and 3.11, respectively, and included in Appendix 3.A. Examples of system kinematic analysis using these computer codes are presented in Sections 3.10 and 3.12, demonstrating the ability to create models and carry out system simulations that would be very difficult or impractical with ad-hoc equation derivation and computer coding. Readers are encouraged to use these programs to carry out numerical experiments with models presented and to formulate and analyze examples of their interest. Finally, a summary of key formulas used in the chapter is presented in Appendix 3.B.

### 3.1 Holonomic Constraints

The position and orientation of bodies in mechanical systems are represented by generalized coordinates defined in Sections 2.3 and 2.4 for planar and spatial bodies, respectively. In preparation for dealing with the intricacies of kinematic constraints that are formulated in terms of *system generalized coordinates* for rigid bodies in space, the basic form of constraints between bodies and their organization as system kinematic constraint equations are introduced in this section. Concepts are illustrated with the relatively simple form of constraints between particles, as models of rigid bodies.

#### 3.1.1 Constraints Between Pairs of Bodies

Joints in mechanical systems are physical constructions that limit relative motion of pairs of bodies identified by indices  $i$  and  $j$ , with a total number  $n_b$  of bodies in the system. For bookkeeping purposes, body  $j = 0$  is designated as the *ground body* that is fixed in a *global reference frame*; i.e., it has constant generalized coordinates relative to that frame. This numbering convention, with ground as a body, is important in accounting for geometric quantities that are related to ground attachments in constraint equations and their derivatives. Constraint equations that characterize joint  $k = k(i_k, j_k)$  between bodies  $i_k$  and  $j_k$ ,  $k = 1, \dots, n_c$ , involve only generalized coordinates  $\mathbf{q}_i \in \mathbb{R}^{n_i}$  and  $\mathbf{q}_j \in \mathbb{R}^{n_j}$  that define the position and orientation of the bodies that are connected. In particular, if  $j = 0$ , body  $i$  is connected to ground and the constraint equations involve only generalized coordinates  $\mathbf{q}_i$ .

To illustrate constraint formulation with *particles as bodies*, the generalized coordinates of particle  $i$  in the plane are  $\mathbf{q}_i = [x_i, y_i]^T$  and in space they are  $\mathbf{q}_i = [x_i, y_i, z_i]^T$ . The only constraint that acts between pairs of particles is the scalar *distance constraint* that specifies the distance  $d_k$  between them; i.e.,

$${}^k(\mathbf{q}_i, \mathbf{q}_j) = ((\mathbf{q}_j - \mathbf{q}_i)^T (\mathbf{q}_j - \mathbf{q}_i) - d_k^2) / 2 = 0 \quad (3.1.1)$$

This algebraic constraint on generalized coordinates is called a *holonomic constraint* (Pars, 1965). Defining  $\mathbf{q}_{ij} \equiv [\mathbf{q}_i^T \quad \mathbf{q}_j^T]^T$ , the *Jacobian* of the constraint of Eq. (3.1.1) is

$$\begin{matrix} {}^k \\ {}_{q_{ij}} \end{matrix}(\mathbf{q}_i, \mathbf{q}_j) \equiv \begin{bmatrix} & {}^k \\ {}^k & {}_{q_i} \end{bmatrix} = \begin{bmatrix} -(\mathbf{q}_j - \mathbf{q}_i)^T & (\mathbf{q}_j - \mathbf{q}_i)^T \end{bmatrix} \quad (3.1.2)$$

where  $\Phi_{q_i}^k \equiv \Phi(\mathbf{q}_i, \dot{\mathbf{q}}_j)_{q_i}$  and an over hat “ $\hat{\cdot}$ ” denotes a term that is held constant for the indicated differentiation.

Generalized coordinates are a function of time  $t$ , for a system in motion. Using the *chain rule of differentiation*, the time derivative of Eq. (3.1.1) yields the *kinematic velocity equation*,

$$\begin{matrix} {}^k \\ {}_{q_{ij}} \end{matrix}(\mathbf{q}_i, \mathbf{q}_j)\dot{\mathbf{q}}_{ij} = \begin{bmatrix} -(\mathbf{q}_j - \mathbf{q}_i)^T & (\mathbf{q}_j - \mathbf{q}_i)^T \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_i^T & \dot{\mathbf{q}}_j^T \end{bmatrix}^T = (\mathbf{q}_j - \mathbf{q}_i)^T (\dot{\mathbf{q}}_j - \dot{\mathbf{q}}_i) = 0 \quad (3.1.3)$$

Note that the *velocity constraint* is linear in velocity, with a coefficient matrix that depends on configuration. Differentiating Eq. (3.1.3) with respect to time yields the *kinematic acceleration equation*,

$${}^k_{q_{ij}}(\mathbf{q}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_{ij} + \left( {}^k_{q_{ij}}(\mathbf{q}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_{ij} \right)_{q_{ij}} \dot{\mathbf{q}}_{ij} = (\mathbf{q}_j - \mathbf{q}_i)^T (\ddot{\mathbf{q}}_j - \ddot{\mathbf{q}}_i) + (\dot{\mathbf{q}}_j - \dot{\mathbf{q}}_i)^T (\dot{\mathbf{q}}_j - \dot{\mathbf{q}}_i) = 0$$

or,

$$(\mathbf{q}_j - \mathbf{q}_i)^T (\ddot{\mathbf{q}}_j - \ddot{\mathbf{q}}_i) = -(\dot{\mathbf{q}}_j - \dot{\mathbf{q}}_i)^T (\dot{\mathbf{q}}_j - \dot{\mathbf{q}}_i) \equiv -\gamma(\dot{\mathbf{q}}_{ij}) \quad (3.1.4)$$

It is important to note that the acceleration constraint is linear in acceleration, with a coefficient matrix that depends on configuration, but it also contains a term that is quadratic in velocities.

More generally, a holonomic constraint is defined between planar and spatial rigid bodies  $i$  and  $j$  by  $n h_k$  algebraic equations of the form

$$\Phi^k(\mathbf{q}_i, \mathbf{q}_j) = \mathbf{0} \quad (3.1.5)$$

i.e.,  $\Phi^k(\mathbf{q}_i, \mathbf{q}_j) \in \mathbb{R}^{nh_k}$ . Families of holonomic constraints between planar and spatial bodies are presented in Sections 3.2 and 3.3, represented by equations of the form of Eq. (3.1.5), with  $\mathbf{q}_i \in \mathbb{R}^3$  for planar systems and  $\mathbf{q}_i \in \mathbb{R}^7$  for spatial systems. For a spatial body whose orientation is defined using Euler parameters, the *Euler parameter normalization condition*

$${}^p_i(\mathbf{q}_i) = (\mathbf{p}_i^T \mathbf{p}_i - 1) / 2 = 0 \quad (3.1.6)$$

is included in Eq. (3.1.5). In all cases, as in Eq. (3.1.1) for particles, it is required that the functions  $\Phi^k(\mathbf{q}_i, \mathbf{q}_j)$  have as many continuous derivatives as are needed for kinematic analysis.

The *constraint Jacobian* of Eq. (3.1.5) is comprised of subJacobians with respect to  $\mathbf{q}_i$  and  $\mathbf{q}_j$ ,

$$\Phi_{q_{ij}}^k(\mathbf{q}_i, \mathbf{q}_j) \equiv \begin{bmatrix} \Phi_{q_i}^k(\mathbf{q}_i, \mathbf{q}_j) & \Phi_{q_j}^k(\mathbf{q}_i, \mathbf{q}_j) \end{bmatrix} \quad (3.1.7)$$

where  $\Phi_{q_i}^k(\mathbf{q}_i, \mathbf{q}_j) = \Phi^k(\mathbf{q}_i, \ddot{\mathbf{q}}_j)_{q_i}$  and  $\Phi_{q_j}^k(\mathbf{q}_i, \mathbf{q}_j) = \Phi^k(\ddot{\mathbf{q}}_i, \mathbf{q}_j)_{q_j}$ . Note that if body  $j$  is ground,  $j = 0$ ,  $\mathbf{q}_j$  is constant, and  $\Phi_{q_j}^k(\mathbf{q}_i, \mathbf{q}_j) = \mathbf{0}$ . However, it is important to retain geometric parameters that define the connection of body  $i$  with ground in the constraint equation and its derivatives.

Taking the time derivative of Eq. (3.1.5), using the chain rule of differentiation, yields the *kinematic velocity equation*, or *velocity constraint*,

$$\Phi_{q_i}^k(\mathbf{q}_i, \mathbf{q}_j) \dot{\mathbf{q}}_i + \Phi_{q_j}^k(\mathbf{q}_i, \mathbf{q}_j) \dot{\mathbf{q}}_j = \mathbf{0} \quad (3.1.8)$$

Taking the time derivative of Eq. (3.1.8) yields the *kinematic acceleration equation*, or *acceleration constraint*,

$$\Phi_{q_i}^k(\mathbf{q}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_i + \Phi_{q_j}^k(\mathbf{q}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_j = -\gamma^k(\mathbf{q}_i, \mathbf{q}_j, \dot{\mathbf{q}}_i, \dot{\mathbf{q}}_j) = -\gamma^k(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \quad (3.1.9)$$

where

$$\begin{aligned} \gamma^k(\mathbf{q}_i, \mathbf{q}_j, \dot{\mathbf{q}}_i, \dot{\mathbf{q}}_j) \equiv & \left( \Phi_{q_i}^k(\mathbf{q}_i, \ddot{\mathbf{q}}_j) \ddot{\mathbf{q}}_i \right)_{q_i} \dot{\mathbf{q}}_i + \left( \Phi_{q_j}^k(\mathbf{q}_i, \ddot{\mathbf{q}}_j) \ddot{\mathbf{q}}_j \right)_{q_i} \dot{\mathbf{q}}_i \\ & + \left( \Phi_{q_i}^k(\ddot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_i \right)_{q_j} \dot{\mathbf{q}}_j + \left( \Phi_{q_j}^k(\ddot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_j \right)_{q_j} \dot{\mathbf{q}}_j \end{aligned} \quad (3.1.10)$$

**Example 3.1.1:** Particle  $p_1$  is constrained to move on a unit circle with center at point  $p_0$  that is fixed in the x-y plane, as shown in Fig. 3.1.1. Generalized coordinates for the particles are the constant vector  $\mathbf{q}_0 = [x_0 \ y_0]^T \equiv [q_{01} \ q_{02}]^T$  and the variable vector  $\mathbf{q}_1 = [x_1 \ y_1]^T \equiv [q_1^1 \ q_1^2]^T$ , subject to the holonomic constraint

$${}^1(\mathbf{q}_0, \mathbf{q}_1) = ((\mathbf{q}_1 - \mathbf{q}_0)^T (\mathbf{q}_1 - \mathbf{q}_0) - 1)/2 = 0 \quad (3.1.11)$$

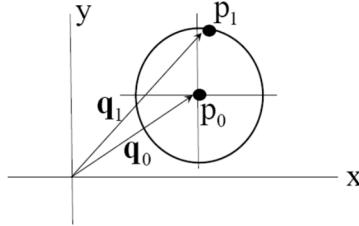


Figure 3.1.1 Particle Constrained to Move on Unit Circle with Center at  $\mathbf{p}^0$

The constraint Jacobian is

$$\Phi_{\mathbf{q}_1}^1(\mathbf{q}_1, \mathbf{q}_0) \equiv \begin{bmatrix} (\mathbf{q}_1 - \mathbf{q}_0)^T & \mathbf{0}_{1 \times 2} \end{bmatrix} \quad (3.1.12)$$

and the velocity equation of Eq. (3.1.8) is

$${}^1_{\mathbf{q}_1}(\mathbf{q}_1, \mathbf{q}_0)\dot{\mathbf{q}}_1 = (\mathbf{q}_1 - \mathbf{q}_0)^T \dot{\mathbf{q}}_1 = 0 \quad (3.1.13)$$

The term  $\gamma$  of Eq. (3.1.10) is thus

$$\gamma^1(\mathbf{q}_1, \mathbf{q}_0, \dot{\mathbf{q}}_1, \mathbf{0}) = \left( \Phi_{\mathbf{q}_1}^k(\mathbf{q}_1, \ddot{\mathbf{q}}_0) \ddot{\mathbf{q}}_1 \right)_{\mathbf{q}_1} \dot{\mathbf{q}}_1 = \left( \ddot{\mathbf{q}}_1^T (\mathbf{q}_1 - \mathbf{q}_0) \right)_{\mathbf{q}_1} = \dot{\mathbf{q}}_1^T \dot{\mathbf{q}}_1 \quad (3.1.14)$$

and the acceleration constraint of Eq. (3.1.9) is

$${}^1(\mathbf{q}_1, \mathbf{q}_0)\ddot{\mathbf{q}}_1 = (\mathbf{q}_1 - \mathbf{q}_0)^T \ddot{\mathbf{q}}_1 = -\dot{\mathbf{q}}_1^T \dot{\mathbf{q}}_1 \quad (3.1.15)$$

---

Suppressing arguments of functions and superscript k for notational simplicity, expanding matrix differentiation to component form, and changing the order of summation, multiplication of scalars, and differentiation with respect to scalar variables yields the following identity that holds for any vector function  $\mathbf{a}(\mathbf{q}_i, \mathbf{q}_j)$ :

$$\begin{aligned} \left( \mathbf{a}_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j &= \sum_{\beta} \left( \sum_{\alpha} \mathbf{a}_{q_i^{\alpha} q_j^{\beta}} \ddot{q}_i^{\alpha} \right)_{q_j^{\beta}} \dot{q}_j^{\beta} = \sum_{\beta} \sum_{\alpha} \mathbf{a}_{q_i^{\alpha} q_j^{\beta}} \dot{q}_i^{\alpha} \dot{q}_j^{\beta} = \sum_{\alpha} \sum_{\beta} \mathbf{a}_{q_j^{\beta} q_i^{\alpha}} \dot{q}_j^{\beta} \dot{q}_i^{\alpha} \\ &= \sum_{\alpha} \left( \sum_{\beta} \mathbf{a}_{q_j^{\beta} q_i^{\alpha}} \dot{q}_j^{\beta} \right)_{q_i^{\alpha}} \dot{q}_i^{\alpha} = \left( \mathbf{a}_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i \end{aligned} \quad (3.1.16)$$

where  $\mathbf{q}_i = [q_i^1 \ \dots \ q_i^{n_i}]^T$  and  $\mathbf{q}_j = [q_j^1 \ \dots \ q_j^{n_j}]^T$ . Thus, Eq. (3.1.10) reduces to

$$\gamma^k(\mathbf{q}_i, \mathbf{q}_j, \dot{\mathbf{q}}_i, \dot{\mathbf{q}}_j) = \left( \Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \dot{\mathbf{q}}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i + 2 \left( \Phi_{\mathbf{q}_i}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j + \left( \Phi_{\mathbf{q}_j}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j \quad (3.1.17)$$

To obtain an expression for  $\gamma^k$ , in a form that supports a broad spectrum of kinematic and dynamic analyses, define

$$\boldsymbol{\chi}_{ij} = \begin{bmatrix} \boldsymbol{\chi}_i^T & \boldsymbol{\chi}_j^T \end{bmatrix}^T \quad (3.1.18)$$

where  $\boldsymbol{\chi}_i$  and  $\boldsymbol{\chi}_j$  have the same dimensions as  $\mathbf{q}_i$  and  $\mathbf{q}_j$ . For joint  $k$ , define the operator

$$\begin{aligned} \mathbf{P2}^k(\mathbf{q}_{ij}, \boldsymbol{\chi}_{ij}) &\equiv \left( \Phi_{\mathbf{q}_{ij}}^k(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \ddot{\mathbf{q}}_{ij} \right)_{\mathbf{q}_{ij}} \\ &= \left[ \left( \Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \dot{\mathbf{q}}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} + \left( \Phi_{\mathbf{q}_j}^k(\mathbf{q}_i, \dot{\mathbf{q}}_j) \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} \quad \left( \Phi_{\mathbf{q}_i}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} + \left( \Phi_{\mathbf{q}_j}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \right] \end{aligned} \quad (3.1.19)$$

A property of the operator  $\mathbf{P2}(\mathbf{q}, \boldsymbol{\chi})$  that will be used in many applications is linearity in its second argument; i.e.,

$$\mathbf{P2}(\mathbf{q}, a_1 \boldsymbol{\chi}_1 + a_2 \boldsymbol{\chi}_2) = a_1 \mathbf{P2}(\mathbf{q}, \boldsymbol{\chi}_1) + a_2 \mathbf{P2}(\mathbf{q}, \boldsymbol{\chi}_2) \quad (3.1.20)$$

for arbitrary real numbers  $a_1$  and  $a_2$  and arbitrary vectors  $\boldsymbol{\chi}_1$  and  $\boldsymbol{\chi}_2$ . This property is a direct result of linearity of the derivative operator.

Evaluating  $\mathbf{P2}^k(\mathbf{q}_{ij}, \boldsymbol{\chi}_{ij}) \dot{\mathbf{q}}_{ij}$  with  $\boldsymbol{\chi}_{ij} = \dot{\mathbf{q}}_{ij}$ ,

$$\begin{aligned} \mathbf{P2}^k(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} &= \left( \Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \dot{\mathbf{q}}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i + \left( \Phi_{\mathbf{q}_j}^k(\mathbf{q}_i, \dot{\mathbf{q}}_j) \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i \\ &\quad + \left( \Phi_{\mathbf{q}_i}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j + \left( \Phi_{\mathbf{q}_j}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j \\ &= \left( \Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \dot{\mathbf{q}}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i + 2 \left( \Phi_{\mathbf{q}_i}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j + \left( \Phi_{\mathbf{q}_j}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j \end{aligned} \quad (3.1.21)$$

where Eq. (3.1.17) is used. Comparison of Eqs. (3.1.10) and (3.1.21) verifies that

$$\gamma^k(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \mathbf{P2}^k(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \quad (3.1.22)$$

### 3.1.2 Constraint, Velocity, and Acceleration Equations

For a system that is comprised of  $nb$  bodies, define *system generalized coordinates* as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \mathbf{q}_2^T & \cdots & \mathbf{q}_{nb}^T \end{bmatrix}^T \in \mathbb{R}^{ngc} \quad (3.1.23)$$

where  $\mathbf{q}_i \in \mathbb{R}^{n_i}$  and  $ngc = \sum_{i=1}^{nb} n_i$  is the total number of generalized coordinates. Numbering joints with index  $k = 1, \dots, nc$ , with joint  $k$  connecting bodies  $i_k$  and  $j_k$ , the constraint equations of Eq. (3.1.5) may be ordered as

$$\Phi(\mathbf{q}) \equiv \begin{bmatrix} \Phi^1(\mathbf{q}_{i_1}, \mathbf{q}_{j_1}) \\ \vdots \\ \Phi^{nc}(\mathbf{q}_{i_{nh}}, \mathbf{q}_{j_{nh}}) \end{bmatrix} = \mathbf{0} \in \mathbb{R}^{nhc} \quad (3.1.24)$$

where  $nhc = \sum_{k=1}^{nc} nh_k$  is the total number of holonomic constraint equations. With this ordering, the *system constraint Jacobian* is

$$\Phi_q(\mathbf{q}) \equiv \begin{bmatrix} \Phi_q^1(\mathbf{q}) \\ \vdots \\ \Phi_q^{nc}(\mathbf{q}) \end{bmatrix} \quad (3.1.25)$$

for which  $\Phi_q^k(\mathbf{q})$  contains nonzero entries only in columns  $i_k$  and  $j_k$  that correspond to generalized coordinates  $\mathbf{q}_i$  and  $\mathbf{q}$  that appear in the joint  $k$  constraint,  $k = 1, \dots, nh$ . The system constraint Jacobian is thus mostly zeros; i.e., it is a *sparse matrix*. The *system kinematic velocity equation*, made up of contributions from Eq. (3.1.8), is

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (3.1.26)$$

The derivative operator of Eq. (3.1.21) for the system is made up of contributions from Eq. (3.1.19),

$$\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \equiv (\Phi_q(\mathbf{q})\dot{\mathbf{q}})_q = \begin{bmatrix} (\Phi_q^1 \dot{\mathbf{q}})_q \\ \vdots \\ (\Phi_q^{nc} \dot{\mathbf{q}})_q \end{bmatrix} \quad (3.1.27)$$

which has the same row and column structure as the Jacobian of Eq. (3.1.25), so it is also sparse.

Finally, the *system kinematic acceleration equation* is comprised of contributions from Eqs. (3.1.21), (3.1.22), (3.1.27), and (3.1.9),

$$\Phi_q(\mathbf{q})\ddot{\mathbf{q}} = -\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = -\gamma(\mathbf{q}, \dot{\mathbf{q}}) \quad (3.1.28)$$

While the bookkeeping convention outlined above may seem to be a bit pedantic, it is an important component of computational kinematics and dynamics. If full use is to be made of the digital computer in forming and solving the equations of motion, the underlying computations must be systematically organized.

**Example 3.1.2:** As an example of a multibody system, consider the three-particle model of a planar double pendulum shown in Fig. 3.1.2. Generalized coordinates for the particles are  $\mathbf{q}_i = [x_i \ y_i]^T \equiv [q_{i1} \ q_{i2}]^T$ ,  $i = 1, 2$ , and  $\mathbf{q}_0 = \mathbf{0}$  for particle 0 that is fixed in ground. Unit distance constraints act between particle 1 and particle 0 that is fixed to the origin of the x-y frame and between particles 1 and 2, represented by the holonomic constraints of Eq. (3.1.24),

$$\Phi(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_0) = \begin{bmatrix} {}^1(\mathbf{q}_0, \mathbf{q}_1) \\ {}^2(\mathbf{q}_1, \mathbf{q}_2) \end{bmatrix} = \begin{bmatrix} (\mathbf{q}_1^T \mathbf{q}_1 - 1)/2 \\ ((\mathbf{q}_2 - \mathbf{q}_1)^T (\mathbf{q}_2 - \mathbf{q}_1) - 1)/2 \end{bmatrix} = \mathbf{0} \quad (3.1.29)$$

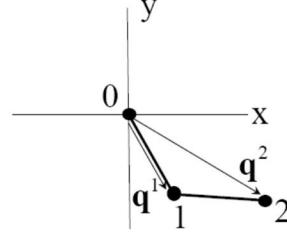


Figure 3.1.2 Three Particle Double Pendulum

With  $\mathbf{q} = [\mathbf{q}_1^T \quad \mathbf{q}_2^T]^T$  the constraint Jacobian of Eq. (3.1.25) is

$$\Phi_q(\mathbf{q}_1, \mathbf{q}_2) = \begin{bmatrix} \mathbf{q}_1^T & \mathbf{0}_{1 \times 2} \\ -(\mathbf{q}_2 - \mathbf{q}_1)^T & (\mathbf{q}_2 - \mathbf{q}_1)^T \end{bmatrix} \quad (3.1.30)$$

and the velocity equation of Eq. (3.1.26) is

$$\Phi_q(\mathbf{q}_1, \mathbf{q}_2) \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{q}_1^T & \mathbf{0}_{1 \times 2} \\ -(\mathbf{q}_2 - \mathbf{q}_1)^T & (\mathbf{q}_2 - \mathbf{q}_1)^T \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix} = \mathbf{0}_{2 \times 1} \quad (3.1.31)$$

The operator of Eq. (3.1.27) is

$$\mathbf{P}2(\mathbf{q}, \chi) = \begin{bmatrix} (\Phi_q^1 \ddot{\chi})_{\mathbf{q}} \\ (\Phi_q^2 \ddot{\chi})_{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} (\mathbf{q}_1^T \ddot{\chi}_1)_{\mathbf{q}} \\ ((-\mathbf{q}_2 + \mathbf{q}_1)^T \ddot{\chi}_1 + (\mathbf{q}_2 - \mathbf{q}_1)^T \ddot{\chi}_2)_{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \chi_1^T & \mathbf{0}_{1 \times 2} \\ \chi_1^T - \chi_2^T & -\chi_1^T + \chi_2^T \end{bmatrix} \quad (3.1.32)$$

and the acceleration constraint of Eq. (3.1.28) is

$$\Phi_q(\mathbf{q}) \ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{q}_1^T & \mathbf{0}_{1 \times 2} \\ -(\mathbf{q}_2 - \mathbf{q}_1)^T & (\mathbf{q}_2 - \mathbf{q}_1)^T \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \end{bmatrix} = - \begin{bmatrix} \dot{\mathbf{q}}_1^T \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_1^T \dot{\mathbf{q}}_1 - 2\dot{\mathbf{q}}_1^T \dot{\mathbf{q}}_2 + \dot{\mathbf{q}}_2^T \dot{\mathbf{q}}_2 \end{bmatrix} = -\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \quad (3.1.33)$$


---

Joints between pairs of bodies in a system are represented by algebraic equations that involve only generalized coordinates of the bodies connected, called holonomic constraints. The constraint equations, together with Euler parameter normalization conditions for spatial bodies, comprise the system holonomic kinematic constraints. Generalized coordinate and constraint numbering conventions that are required for generation and solution of the equations of kinematics and dynamics on a digital computer are presented and illustrated, using particle models of mechanical systems. Velocity and acceleration equations are derived, using the numbering convention presented. In general, kinematic velocity equations are linear in velocity, with matrix coefficients that depend on configuration. In contrast, acceleration equations are linear in acceleration, with matrix coefficients that depend on configuration, but they contain additional terms that are quadratic in velocities.

### ***Key Formulas***

$$\Phi^k(\mathbf{q}_i, \mathbf{q}_j) = \mathbf{0} \quad \Phi_{\mathbf{q}_{ij}}^k(\mathbf{q}_i, \mathbf{q}_j) \equiv \begin{bmatrix} \Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \mathbf{q}_j) & \Phi_{\mathbf{q}_j}^k(\mathbf{q}_i, \mathbf{q}_j) \end{bmatrix} \quad (3.1.7)$$

$$\Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \mathbf{q}_j)\dot{\mathbf{q}}_i + \Phi_{\mathbf{q}_j}^k(\mathbf{q}_i, \mathbf{q}_j)\dot{\mathbf{q}}_j = \mathbf{0} \quad (3.1.8)$$

$$\Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \mathbf{q}_j)\ddot{\mathbf{q}}_i + \Phi_{\mathbf{q}_j}^k(\mathbf{q}_i, \mathbf{q}_j)\ddot{\mathbf{q}}_j = -\gamma^k(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \quad (3.1.9)$$

$$\gamma^k(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \left( \Phi_{\mathbf{q}_i}^k(\mathbf{q}_i, \dot{\mathbf{q}}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i + 2 \left( \Phi_{\mathbf{q}_i}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j + \left( \Phi_{\mathbf{q}_j}^k(\dot{\mathbf{q}}_i, \mathbf{q}_j) \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j \quad (3.1.17)$$

$$P2^k(\mathbf{q}_{ij}, \chi_{ij}) \equiv \left[ \left( \Phi_{\mathbf{q}_i}^k \ddot{\chi}_i \right)_{\mathbf{q}_i} + \left( \Phi_{\mathbf{q}_j}^k \ddot{\chi}_j \right)_{\mathbf{q}_i} \quad \left( \Phi_{\mathbf{q}_i}^k \ddot{\chi}_i \right)_{\mathbf{q}_j} + \left( \Phi_{\mathbf{q}_j}^k \ddot{\chi}_j \right)_{\mathbf{q}_j} \right] \quad (3.1.19)$$

## 3.2 Planar Joints, Constraint Equations, and Drivers

A surprisingly modest library of kinematic joints between pairs of planar bodies, namely distance, revolute, and translational, enables modeling a wide variety of planar systems. These and related time dependent driving constraints are presented, in support of planar system kinematic and dynamic modeling and simulation. In preparation for computer implementation of the equations of kinematics, data that characterize each joint are systematically defined.

### 3.2.1 Constraints

For a joint between bodies  $i$  and  $j$ , constraint equations involve only *Cartesian generalized coordinates*  $\mathbf{q}_i = [\mathbf{r}_i^T \quad \phi_i]^T$  and  $\mathbf{q}_j = [\mathbf{r}_j^T \quad \phi_j]^T$ . To simplify notation, the *orientation transformation matrix* of Eq. (2.3.27) for body  $i$  is denoted  $\mathbf{A}_i \equiv \mathbf{A}(\phi_i)$ . To obtain expressions for  $\mathbf{P}^{2^k}(\mathbf{q}_{ij}, \chi_{ij})$  of Eq. (3.1.9), define  $\chi_i = [\chi_{r_i}^T \quad \phi_i]^T$ , where  $\chi_{r_i}$  and  $\phi_i$  have the same dimensions as  $\mathbf{r}_i$  and  $\phi_i$ , and likewise for body  $j$ .

#### 3.2.1.1 Distance Constraint

A *planar distance constraint* maintains a constant distance  $d > 0$  between points  $P_i$  and  $P_j$  on a pair of planar bodies  $i$  and  $j$  that are located by body fixed vectors  $\mathbf{s}'_i$  and  $\mathbf{s}'_j$ . It is a *massless coupler* of length  $d > 0$  shown in Fig 3.2.1. Physically, it can be envisioned as a bar of length  $d > 0$  with ball and socket joints at each end. With the vector from  $P_i$  to  $P_j$ ,

$$\mathbf{d}_{ij} \equiv \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i \quad (3.2.1)$$

the *distance constraint equation* is the scalar condition

$$\Phi^{\text{dist}}(\mathbf{q}_i, \mathbf{q}_j) \equiv (\mathbf{d}_{ij}^T \mathbf{d}_{ij} - d^2) / 2 = 0 \quad (3.2.2)$$

Data that define the distance constraint are as follows:

$$\text{dist k: } i_k, s'_{i_k}; j_k, s'_{j_k}; d \quad (3.2.3)$$

where  $k$  is the joint number in a multibody model,  $i_k$  and  $j_k$  are body numbers connected by the joint,  $s'_{i_k}$  and  $s'_{j_k}$  are body fixed vectors that locate points  $P_i$  and  $P_j$  on the bodies, and  $d > 0$  is the distance between points  $P_i$  and  $P_j$ . These data are specified for each instance of a distance constraint. Similar data sets will be defined for each type of joint treated.

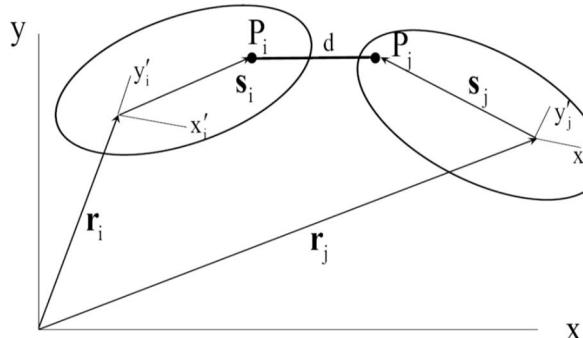


Figure 3.2.1 Distance Constraint Between Bodies i and j

If body i is held fixed, point  $P_j$  can move on a circle of radius  $d > 0$  with center  $P_i$ , and body j is free to rotate about point  $P_j$ ; i.e., body j has two degrees of freedom. Thus, a planar distance constraint eliminates one degree of freedom from the pair.

Using Eq. (2.3.31) *subJacobians* of the distance constraint are

$$\begin{aligned}\Phi_{q_i}^{\text{dist}} &= -\left[ \mathbf{d}_{ij}^T \quad \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \right] \\ \Phi_{q_j}^{\text{dist}} &= \left[ \mathbf{d}_{ij}^T \quad \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \right]\end{aligned}\quad (3.2.4)$$

and terms in the operator of Eq. (3.1.19) are

$$\begin{aligned}\left( \Phi_{q_i}^{\text{dist}} \dot{\chi}_i \right)_{q_i} &= -\left( \mathbf{d}_{ij}^T \ddot{\chi}_{r_i} + \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \ddot{\chi}_{\phi_i} \right)_{q_i} \\ &= -\left( \ddot{\chi}_{r_i}^T \mathbf{d}_{ij} + \ddot{\chi}_{\phi_i}^T \mathbf{P} \mathbf{A}_i \mathbf{s}'_i + \chi_{\phi_i} \mathbf{s}'_i^T \mathbf{A}_i^T \mathbf{P}^T \mathbf{d}_{ij} \right)_{q_i} \\ &= \mathbf{a}_i^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \right] + \mathbf{d}_{ij}^T \left[ \mathbf{0} \quad \chi_{\phi_i} \mathbf{A}_i \mathbf{s}'_i \right] \\ \left( \Phi_{q_j}^{\text{dist}} \dot{\chi}_j \right)_{q_i} &= -\mathbf{a}_j^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \right] \\ \left( \Phi_{q_i}^{\text{dist}} \dot{\chi}_i \right)_{q_j} &= -\mathbf{a}_i^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \right] \\ \left( \Phi_{q_j}^{\text{dist}} \dot{\chi}_j \right)_{q_j} &= \mathbf{a}_j^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \right] - \mathbf{d}_{ij}^T \left[ \mathbf{0} \quad \chi_{\phi_j} \mathbf{A}_j \mathbf{s}'_j \right]\end{aligned}\quad (3.2.5)$$

where  $\mathbf{a}_i = \mathbf{r}_i + \dot{\phi}_i \mathbf{P} \mathbf{A}_i \mathbf{s}'_i$  and  $\mathbf{a}_j = \mathbf{r}_j + \dot{\phi}_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j$ . Thus,

$$\mathbf{P}2^{\text{dist}}(\mathbf{q}_{ij}, \chi_{ij}) = \left[ \mathbf{P}2_i^{\text{dist}} \quad \mathbf{P}2_j^{\text{dist}} \right] \quad (3.2.6)$$

where

$$\begin{aligned}\mathbf{P}2_i^{\text{dist}} &= \left[ (\mathbf{a}_i - \mathbf{a}_j)^T \quad (\mathbf{a}_i - \mathbf{a}_j)^T \mathbf{P} \mathbf{A}_i \mathbf{s}'_i + \dot{\phi}_i \mathbf{d}_{ij}^T \mathbf{A}_i \mathbf{s}'_i \right] \\ \mathbf{P}2_j^{\text{dist}} &= \left[ (\mathbf{a}_j - \mathbf{a}_i)^T \quad (\mathbf{a}_j - \mathbf{a}_i)^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j - \dot{\phi}_j \mathbf{d}_{ij}^T \mathbf{A}_j \mathbf{s}'_j \right]\end{aligned}$$

The right side of the acceleration equation is thus

$$-\gamma^{\text{dist}} = -\mathbf{P}2^{\text{dist}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \ddot{\mathbf{q}}_{ij} \quad (3.2.7)$$

### 3.2.1.2 Revolute Constraint

A *revolute joint* between planar bodies i and j allows relative rotation about a point P that is common to bodies i and j, as shown in Fig. 3.2.2. Physically, such a joint is a rotational bearing between the bodies at point P. The *revolute joint constraint* is the vector condition

$$\Phi^{\text{rev}}(\mathbf{q}_i, \mathbf{q}_j) \equiv \mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i = \mathbf{0} \quad (3.2.8)$$

Data that define the revolute constraint as joint k are as follows:

$$\text{rev k: } i_k, s'_k; j_k, s'_k \quad (3.2.9)$$

If one body is held fixed, the other body has a single rotational degree of freedom. Thus, a revolute joint eliminates two degrees of freedom from the pair.

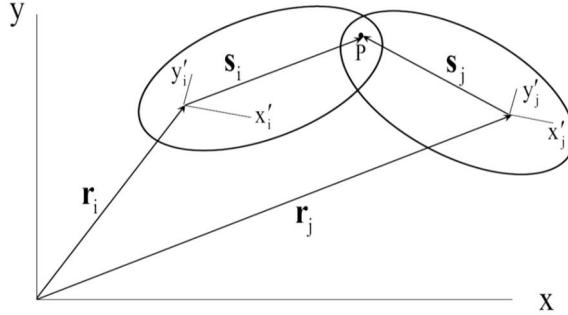


Figure 3.2.2 Revolute Constraint Between Bodies i and j

To see that Eq. (3.2.8) implies the geometry of the joint, note that vectors locating point P on bodies i and j are  $\mathbf{r}_i^P = \mathbf{r}_i + \mathbf{s}'_i = \mathbf{r}_i + \mathbf{A}_i \mathbf{s}'_i$  and  $\mathbf{r}_j^P = \mathbf{r}_j + \mathbf{s}'_j = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j$ . Thus, Eq. (3.2.8) implies  $\mathbf{r}_i^P = \mathbf{r}_j^P$ , which is the definition of the revolute joint.

*SubJacobians* of the revolute constraint are

$$\begin{aligned}\Phi_{q_i}^{\text{rev}} &= -[\mathbf{I} \quad \mathbf{P} \mathbf{A}_i \mathbf{s}'_i] \\ \Phi_{q_j}^{\text{rev}} &= [\mathbf{I} \quad \mathbf{P} \mathbf{A}_j \mathbf{s}'_j]\end{aligned}\quad (3.2.10)$$

and terms in the operator of Eq. (3.1.19) are

$$\begin{aligned}(\Phi_{q_i}^{\text{rev}} \dot{\chi}_i)_{q_i} &= [\mathbf{0} \quad \chi_{\phi_i} \mathbf{A}_i \mathbf{s}'_i] \\ (\Phi_{q_j}^{\text{rev}} \dot{\chi}_i)_{q_i} &= \mathbf{0} \\ (\Phi_{q_i}^{\text{rev}} \dot{\chi}_i)_{q_j} &= \mathbf{0} \\ (\Phi_{q_j}^{\text{rev}} \dot{\chi}_i)_{q_j} &= [\mathbf{0} \quad -\chi_{\phi_j} \mathbf{A}_j \mathbf{s}'_j]\end{aligned}\quad (3.2.11)$$

Thus,

$$\mathbf{P}^{\text{2rev}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \left[ \begin{array}{cc} \mathbf{0} & \mathbf{A}_i \mathbf{s}'_i \\ & \end{array} \right] \left[ \begin{array}{cc} \mathbf{0} & -\mathbf{A}_j \mathbf{s}'_j \\ & \end{array} \right] \quad (3.2.12)$$

and the acceleration right side is

$$-\boldsymbol{\gamma}^{\text{rev}} = -\mathbf{P}^{\text{2rev}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \ddot{\mathbf{q}}_{ij} \quad (3.2.13)$$

### 3.2.1.3 Translational Constraint

A *translational joint*, or *prismatic joint*, between planar bodies i and j is shown in Fig. 3.2.3. It allows relative translation of the bodies along a common axis that is defined by vectors  $\mathbf{v}_i$  in body i and  $\mathbf{v}_j$  in body j that emanate from points  $P_i$  and  $P_j$ , respectively, but no relative rotation of the bodies. Physically, such a joint may be defined as a straight key on one body that fits precisely in a straight slot (or keyway) in the second body and allows relative translation along their common center lines. If one body is held fixed, the other body has only a single

translational degree of freedom. Thus, a translational joint eliminates two degrees of freedom from the pair.

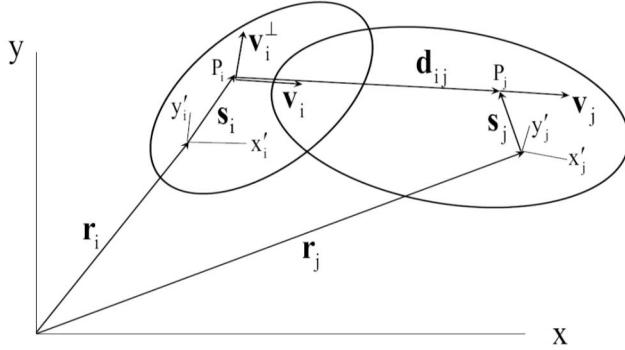


Figure 3.2.3 Translational Constraint Between Bodies i and j

For the translational joint shown in Fig. 3.2.3, points  $P_i$  and  $P_j$  are specified on the path of relative translation between bodies i and j by body fixed vectors  $s'_i$  and  $s'_j$ . Nonzero body fixed vectors  $v'_i$  and  $v'_j$  emanate from points  $P_i$  and  $P_j$  and lie on the axis of relative translation. While not required, it is recommended that  $v'_i$  and  $v'_j$  be unit vectors. The vector  $d_{ij}$  connects points  $P_i$  and  $P_j$ . The geometric definition of the translational joint is that vectors  $v_i$  and  $v_j$  must remain collinear. To have these vectors collinear, it is sufficient that  $d_{ij}$  and  $v_j$  be perpendicular to  $v_i^\perp = \mathbf{P}v_i$ , which is perpendicular to  $v_i$ . Transforming  $v_i^\perp$  to the global frame,

$$v_i^\perp = \mathbf{A}_i v_i' = \mathbf{A}_i \mathbf{P} v_i' \quad (3.2.14)$$

The translational joint conditions that  $d_{ij}$  and  $v_j$  are perpendicular to  $v_i^\perp$  are thus

$$\Phi^{\text{tran}}(\mathbf{q}_i, \mathbf{q}_j) \equiv \begin{bmatrix} \mathbf{v}_i'^T \mathbf{d}_{ij} \\ \mathbf{v}_i'^T \mathbf{v}_j \end{bmatrix} = - \begin{bmatrix} \mathbf{v}_i'^T \mathbf{P} \mathbf{A}_i^T \mathbf{d}_{ij} \\ \mathbf{v}_i'^T \mathbf{P} \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}_j' \end{bmatrix} = \mathbf{0} \quad (3.2.15)$$

Data that define the translational constraint as joint k are as follows:

$$\text{tran k: } i_k, s'_{i_k}, v'_{i_k}; j_k, s'_{j_k}, v'_{j_k} \quad (3.2.16)$$

As long as  $\mathbf{d}_{ij} \neq \mathbf{0}$ , Eq. (3.2.15) implies that the vectors  $v_i$ ,  $v_j$ , and  $\mathbf{d}_{ij}$  are parallel and, since they have points in common,  $v_i$  and  $v_j$  are collinear. If  $\mathbf{d}_{ij} = \mathbf{0}$ , which is possible,  $v_i$  and  $v_j$  emanate from the same point and, by the second of Eqs. (3.2.15), they are parallel. Thus, they are collinear and Eqs. (3.2.15) are equivalent to the geometry of the translational joint.

*SubJacobians* of the translational joint constraint are

$$\begin{aligned}\Phi_{q_i}^{\text{tran}} &= \begin{bmatrix} \mathbf{v}'^T \mathbf{P} \mathbf{A}_i^T & -\mathbf{v}'^T \mathbf{s}' - \mathbf{v}'^T \mathbf{A}_i^T \mathbf{d}_{ij} \\ \mathbf{0} & -\mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} \\ \Phi_{q_j}^{\text{tran}} &= \begin{bmatrix} -\mathbf{v}'^T \mathbf{P} \mathbf{A}_i^T & \mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{s}'_j \\ \mathbf{0} & \mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j \end{bmatrix}\end{aligned}\quad (3.2.17)$$

and terms in the operator of Eq. (3.1.19) are

$$\begin{aligned}\left(\Phi_{q_i}^{\text{tran}} \chi_i\right)_{q_i} &= \begin{bmatrix} \chi_{\phi_i} \mathbf{v}'^T \mathbf{A}_i^T & \mathbf{v}'^T \mathbf{A}_i^T \chi_{r_i} + \chi_{\phi_i} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{d}_{ij} + \chi_{\phi_i} \mathbf{v}'^T \mathbf{P} \mathbf{s}'_i \\ \mathbf{0} & \chi_{\phi_i} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} \\ \left(\Phi_{q_i}^{\text{tran}} \chi_j\right)_{q_i} &= \begin{bmatrix} \mathbf{0} & -\mathbf{v}'^T \mathbf{A}_i^T \chi_{r_j} - \chi_{\phi_j} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \\ \mathbf{0} & -\chi_{\phi_j} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} \\ \left(\Phi_{q_i}^{\text{tran}} \chi_i\right)_{q_j} &= \begin{bmatrix} -\chi_{\phi_i} \mathbf{v}'^T \mathbf{A}_i^T & -\chi_{\phi_i} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \\ \mathbf{0} & -\chi_{\phi_i} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} \\ \left(\Phi_{q_j}^{\text{tran}} \chi_j\right)_{q_j} &= \begin{bmatrix} \mathbf{0} & \chi_{\phi_j} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \\ \mathbf{0} & \chi_{\phi_j} \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{v}'_j \end{bmatrix}\end{aligned}\quad (3.2.18)$$

Thus,

$$\mathbf{P}2^{\text{tran}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \begin{bmatrix} \mathbf{P}2_i^{\text{tran}} & \mathbf{P}2_j^{\text{tran}} \end{bmatrix} \quad (3.2.19)$$

where

$$\begin{aligned}\mathbf{P}2_i^{\text{tran}} &= \begin{bmatrix} \phi_i \mathbf{v}'^T \mathbf{A}_i^T & \mathbf{v}'^T \mathbf{A}_i^T \left( \mathbf{r}_i - \mathbf{r}_j + \phi_i \mathbf{P} \mathbf{d}_{ij} - \phi_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j + \phi_i \mathbf{A}_i^T \mathbf{P} \mathbf{s}'_i \right) \\ \mathbf{0} & \left( \phi_i - \phi_j \right) \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} \\ \mathbf{P}2_j^{\text{tran}} &= \begin{bmatrix} -\phi_i \mathbf{v}'^T \mathbf{A}_i^T & \left( \phi_j - \phi_i \right) \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \\ \mathbf{0} & \left( \phi_j - \phi_i \right) \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j \mathbf{v}'_j \end{bmatrix}\end{aligned}$$

The acceleration right side is thus

$$-\gamma^{\text{tran}} = -\mathbf{P}2^{\text{tran}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \ddot{\mathbf{q}}_j \quad (3.2.20)$$

### 3.2.2 Kinematic Drivers

Kinematic constraints can become *kinematic drivers* by specifying time dependent terms that define the desired relative motion. The *distance driver* is obtained by making d a function of time,  $d(t) > 0$ , in the distance constraint of Eq. (3.2.2); i.e.,

$$\Phi^{\text{distD}} = \left( \mathbf{d}_{ij}^T \mathbf{d}_{ij} - d(t)^2 \right) / 2 = 0 \quad (3.2.21)$$

Data that define the distance driver as joint k are as with the associated distance constraint, and  $d = d(t)$ ,

$$\text{distD } k: i_k, s'_{i_k}; j_k, s'_{j_k}; d(t) \quad (3.2.22)$$

While  $\frac{\text{distD}}{q_{ij}}(q_i, q_j)$ ,  $P2^{\text{distD}}(q_{ij}, \dot{q}_{ij})$ , and  $\gamma^{\text{distD}}(q_{ij}, \ddot{q}_{ij})$  are the same as in Eqs. (3.2.4), (3.2.7), and (3.2.8), the velocity and acceleration equations of Eqs. (3.1.8) and (3.1.9) must be modified to account for the time dependent term in Eq. (3.2.21); i.e.,

$$\begin{aligned} \frac{\text{distD}}{q_i}(q_i, q_j)\dot{q}_i + \frac{\text{distD}}{q_j}(q_i, q_j)\dot{q}_j &= d(t)\dot{d}(t) \\ \frac{\text{distD}}{q_i}(q_i, q_j)\ddot{q}_i + \frac{\text{distD}}{q_j}(q_i, q_j)\ddot{q}_j &= -\gamma^{\text{distD}}(q_{ij}, \dot{q}_{ij}) + d(t)\ddot{d}(t) + \dot{d}(t)\dot{d}(t) \end{aligned} \quad (3.2.23)$$

In order to specify relative motion in a translational joint, a distance driver may be imposed between points  $P_i$  and  $P_j$  of Fig. 3.2.3.

A *relative rotation driver* associated with a revolute joint can be defined by specifying the relative rotation  $b(t)$  of bodies that are connected by the revolute joint; i.e., the constraint of Eq. (3.2.8) is augmented with

$$\text{rotD} \equiv \phi_j - \phi_i - b(t) = 0 \quad (3.2.24)$$

Data that define the relative rotation driver as joint k are as with the associated revolute constraint, with the addition of  $b(t)$ ,

$$\text{rotD } k: i_k, s'_{i_k}; j_k, s'_{j_k}; b(t) \quad (3.2.25)$$

The velocity and acceleration equations for this driver are simply

$$\begin{aligned} \dot{\phi}_j - \dot{\phi}_i &= \dot{b}(t) \\ \ddot{\phi}_j - \ddot{\phi}_i &= \ddot{b}(t) \end{aligned} \quad (3.2.26)$$

In general, when driving constraints are of the form

$$\Phi^D(q, t) = \Phi(q) + f(t) = 0 \quad (3.2.27)$$

the associated velocity and acceleration constraint equations are

$$\Phi_q^D(q, t)\dot{q} + \Phi_t^D(q, t) = \Phi_q(q)\dot{q} + \dot{f}(t) = 0 \quad (3.2.28)$$

$$\Phi_q^D(q, t)\ddot{q} + \gamma(q, \dot{q}) + \Phi_{tt}^D(q, t) = \Phi_q(q)\ddot{q} + \gamma(q, \dot{q}) + \ddot{f}(t) = 0 \quad (3.2.29)$$

Just as subroutines in kinematic analysis computer code are required for evaluation of  $\Phi_q(q)$  and  $\gamma(q, \dot{q})$  for all constraints, subroutines are required for evaluation of  $f(t)$ ,  $\dot{f}(t)$ , and  $\ddot{f}(t)$  for driving constraints.

### 3.2.3 Planar System Examples

#### 3.2.3.1 Double Pendulum

The *planar double pendulum* of Fig. 3.2.4 is made up of two bars, each of length two units, designated as bodies 1 and 2. Body 1 is connected to ground (body 0) at its left end with a

revolute joint and to body 2 at its right end with a revolute joint. Data for the joints are tabulated as follows

$$\begin{aligned} \text{rev k = 1: } i_1 &= 1, s'_1 = -\mathbf{u}_x; j_1 = 0, s'_0 = \mathbf{0} \\ \text{rev k = 2: } i_2 &= 1, s'_1 = \mathbf{u}_x; j_2 = 2, s'_2 = -\mathbf{u}_x \end{aligned} \quad (3.2.30)$$

where  $\mathbf{u}_x$  is a unit vector along the positive  $x'$  axis and  $k$  is the joint number. It is important to be consistent in use of indices  $i$  and  $j$  in the joint definitions of Eq. (3.2.30), since they define variables in the constraint equations and associated subJacobians and  $\mathbf{P}_2(\mathbf{q}, \chi)$  operators.

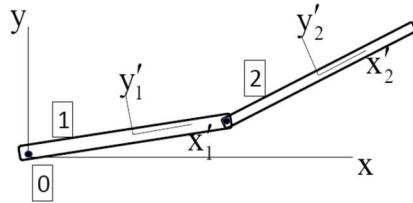


Figure 3.2.4 Planar Double Pendulum

With  $\mathbf{q} = [\mathbf{r}_1^T \quad \phi_1 \quad \mathbf{r}_2^T \quad \phi_2]^T \in \mathbb{R}^6$ , from Eq. (3.2.8), the constraint equations are

$$\Phi(\mathbf{q}) = \begin{bmatrix} \Phi_{\text{rev}1}(\mathbf{q}_1) \\ \Phi_{\text{rev}2}(\mathbf{q}_1, \mathbf{q}_2) \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_1 + \mathbf{A}_1 \mathbf{u}'_x \\ \mathbf{r}_2 - \mathbf{A}_2 \mathbf{u}'_x - \mathbf{r}_1 - \mathbf{A}_1 \mathbf{u}'_x \end{bmatrix} = \mathbf{0} \quad (3.2.31)$$

This is a system of four nonlinear equations in six generalized coordinates. If the constraint Jacobian has full rank, the system will have two degrees of freedom. From Eq. (3.2.10), the constraint Jacobian is

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} \Phi_{q_1}^{\text{rev}1} & \mathbf{0} \\ \Phi_{q_1}^{\text{rev}2} & \Phi_{q_2}^{\text{rev}2} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{PA}_1 \mathbf{u}'_x & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & -\mathbf{PA}_1 \mathbf{u}'_x & \mathbf{I} & -\mathbf{PA}_2 \mathbf{u}'_x \end{bmatrix} \quad (3.2.32)$$

Since the submatrix comprised of the first and third matrix columns has determinant 1, the Jacobian has full rank. Finally, from Eq. (3.2.13),

$$\gamma = \begin{bmatrix} \dot{\phi}_1^2 \mathbf{A}_1 \mathbf{u}'_x \\ \dot{\phi}_1^2 \mathbf{A}_1 \mathbf{u}'_x + \dot{\phi}_2^2 \mathbf{A}_2 \mathbf{u}'_x \end{bmatrix} \quad (3.2.33)$$

### 3.2.3.2 Slider-Crank, 3-Body Model

In the three-body model of the *planar slider-crank* mechanism shown in Fig. 3.2.5, body 1 is the crank of radius  $R$ . It is constrained by revolute joint 1 with ground (body 0); body 2 is the connecting rod of length 2 units, with revolute joints at the ends that are connected to bodies 1 and 3; and body 3 is the slider that translates without rotation along the global  $x$  axis in ground. Data for the joints are tabulated as follows:

$$\begin{aligned}
\text{rev k = 1: } & i_1 = 1, \mathbf{s}'_1 = \mathbf{0}; j_1 = 0, \mathbf{s}'_0 = \mathbf{0} \\
\text{tran k = 2: } & i_2 = 3, \mathbf{s}'_3 = \mathbf{0}, \mathbf{v}'_3 = \mathbf{u}'_x; j_2 = 0, \mathbf{s}'_0 = \mathbf{0}, \mathbf{v}'_0 = \mathbf{u}'_x \\
\text{rev k = 3: } & i_3 = 1, \mathbf{s}'_1 = R\mathbf{u}'_x; j_3 = 2, \mathbf{s}'_2 = -\mathbf{u}'_x \\
\text{rev k = 4: } & i_4 = 2, \mathbf{s}'_2 = \mathbf{u}'_x; j_4 = 3, \mathbf{s}'_3 = \mathbf{0}
\end{aligned} \tag{3.2.34}$$

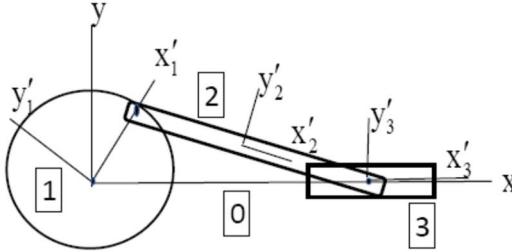


Figure 3.2.5 Planar Slider-Crank, Three Body Model

With  $\mathbf{q} = [\mathbf{r}_1^T \quad \phi_1 \quad \mathbf{r}_2^T \quad \phi_2 \quad \mathbf{r}_3^T \quad \phi_3]^T \in \mathbb{R}^9$ , from Eqs. (3.2.8) and (3.2.15), the constraint equations are

$$\Phi(\mathbf{q}) = \begin{bmatrix} \Phi^1(\mathbf{q}_1) \\ \Phi^2(\mathbf{q}_3) \\ \Phi^3(\mathbf{q}_1, \mathbf{q}_2) \\ \Phi^4(\mathbf{q}_2, \mathbf{q}_3) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ -\mathbf{u}'_x^T \mathbf{P} \mathbf{r}_3 \\ -\mathbf{u}'_x^T \mathbf{P} \mathbf{A}_3 \mathbf{u}'_x \\ \mathbf{r}_2 - \mathbf{A}_2 \mathbf{u}'_x - \mathbf{r}_1 - R \mathbf{A}_1 \mathbf{u}'_x \\ \mathbf{r}_3 - \mathbf{r}_2 - \mathbf{A}_2 \mathbf{u}'_x \end{bmatrix} = \mathbf{0} \tag{3.2.35}$$

This is a system of eight nonlinear equations in nine generalized coordinates. If the constraint Jacobian has full rank, the system will have one degree of freedom.

While the Jacobian and matrix  $\mathbf{P}(\mathbf{q}, \chi)$  for the planar double pendulum have some zero entries, sparsity is more significant for the three-body slider-crank. Block submatrices in Eq. (3.2.36) that are left blank are zeros. From Eqs. (3.2.10) and (3.2.17), the constraint Jacobian is

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} \Phi_{q_1}^1 \\ \Phi_{q_3}^2 \\ \Phi_{q_{12}}^3 \\ \Phi_{q_{23}}^4 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & & \\ & & -\mathbf{u}'_x^T \mathbf{P} & \mathbf{0} \\ & & \mathbf{0} & \mathbf{u}'_x^T \mathbf{A}_3 \mathbf{u}'_x \\ -\mathbf{I} & -R \mathbf{P} \mathbf{A}_1 \mathbf{u}'_x & \mathbf{I} & -\mathbf{P} \mathbf{A}_2 \mathbf{u}'_x \\ & & -\mathbf{I} & -\mathbf{P} \mathbf{A}_2 \mathbf{u}'_x & \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{3.2.36}$$

Finally, from Eqs. (3.2.13) and (3.2.20),

$$\gamma(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \gamma^1(\mathbf{q}_1, \dot{\mathbf{q}}_1) \\ \gamma^2(\mathbf{q}_3, \dot{\mathbf{q}}_3) \\ \gamma^3(\mathbf{q}_1, \dot{\mathbf{q}}_1, \mathbf{q}_2, \dot{\mathbf{q}}_2) \\ \gamma^4(\mathbf{q}_2, \dot{\mathbf{q}}_2, \mathbf{q}_3, \dot{\mathbf{q}}_3) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ \dot{\phi}_3^2 \mathbf{u}'_x^T \mathbf{P} \mathbf{A}_3 \mathbf{u}'_x \\ \dot{\phi}_1^2 R \mathbf{P} \mathbf{A}_1 \mathbf{u}'_x + \dot{\phi}_2^2 \mathbf{A}_2 \mathbf{u}'_x \\ \dot{\phi}_2^2 \mathbf{A}_2 \mathbf{u}'_x \end{bmatrix} \tag{3.2.37}$$

### 3.2.3.3 Slider-Crank, 2-Body Model with Rotation Driver

In the two-model of the *planar slider-crank* mechanism shown in Fig. 3.2.6, body 1 is the crank of radius R. It is constrained by revolute joint 1 with ground (body 0). Body 2 is the slider that translates along the global x axis in ground. The connecting rod is modeled as a distance constraint between bodies one and two of length two units. A *rotation driver* that specifies the angle of rotation  $\theta(t)$  of the crank relative to ground is imposed to specify the motion of the system. Data for the joints and driver are tabulated as follows:

$$\begin{aligned} \text{rev k = 1: } i_1 &= 1, s'_1 = \mathbf{0}; j_1 = 0, s'_0 = \mathbf{0} \\ \text{tran k = 2: } i_2 &= 2, s'_2 = \mathbf{0}, v'_2 = \mathbf{u}'_x; j_2 = 0, s'_0 = \mathbf{0}, v'_0 = \mathbf{u}'_x \\ \text{dist k = 3: } i_3 &= 1, s'_1 = R\mathbf{u}'_x; j_3 = 2, s'_2 = \mathbf{0}; d = 2 \\ \text{rotD k = 4: } i_4 &= 1, s'_1 = \mathbf{0}; j_4 = 0, s'_0 = \mathbf{0}; \phi_1 - \theta(t) = 0 \end{aligned} \quad (3.2.38)$$

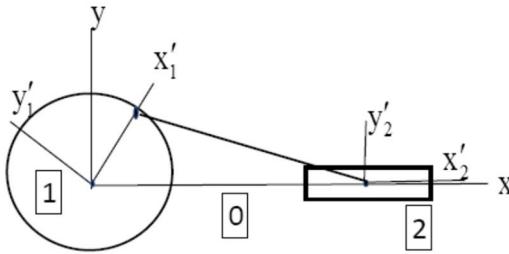


Figure 3.2.6 Planar Slider-Crank, Two Body Model

With  $\mathbf{q} = [\mathbf{r}_1^T \quad \phi_1 \quad \mathbf{r}_2^T \quad \phi_2]^T \in \mathbb{R}^6$ , from Eqs. (3.2.2), (3.2.8) and (3.2.15), the constraint equations are

$$\Phi(\mathbf{q}, t) = \begin{bmatrix} \Phi^1(\mathbf{q}_1) \\ \Phi^2(\mathbf{q}_2) \\ \Phi^3(\mathbf{q}_1, \mathbf{q}_2) \\ \Phi^4(\mathbf{q}_1, t) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ -\mathbf{u}'_x^T \mathbf{P} \mathbf{r}_2 \\ -\mathbf{u}'_x^T \mathbf{P} \mathbf{A}_2 \mathbf{u}'_x \\ (\mathbf{d}_{12}^T \mathbf{d}_{12} - 4)/2 \\ \phi_1 - \theta(t) \end{bmatrix} = \mathbf{0} \quad (3.2.39)$$

where  $\mathbf{d}_{12} = \mathbf{r}_2 - \mathbf{r}_1 - R\mathbf{A}_1 \mathbf{u}'_x$ . This is a system of six nonlinear equations in six generalized coordinates and time. If the constraint Jacobian is nonsingular, it will have a unique solution for the generalized coordinates as functions of time. From Eqs. (3.2.4), (3.2.10), and (3.2.17), the constraint Jacobian is

$$\Phi_q(\mathbf{q}, t) = \begin{bmatrix} \Phi_{q_1}^1 \\ \Phi_{q_2}^2 \\ \Phi_{q_{12}}^3 \\ \Phi_{q_1}^4 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & -\mathbf{u}'_x^T \mathbf{P} & 0 \\ \mathbf{0} & 0 & \mathbf{0} & \mathbf{u}'_x^T \mathbf{A}_2 \mathbf{u}'_x \\ -\mathbf{d}_{12}^T & -R\mathbf{d}_{12}^T \mathbf{P} \mathbf{A}_1 \mathbf{u}'_x & \mathbf{d}_{12}^T & 0 \\ \mathbf{0} & 1 & \mathbf{0} & 0 \end{bmatrix} \quad (3.2.40)$$

Finally, from Eqs. (3.2.7), (3.2.13), and (3.2.20),

$$\gamma(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \gamma^1(\mathbf{q}_1, \dot{\mathbf{q}}_1) \\ \gamma^2(\mathbf{q}_2, \dot{\mathbf{q}}_2) \\ \gamma^3(\mathbf{q}_1, \dot{\mathbf{q}}_1, \mathbf{q}_2, \dot{\mathbf{q}}_2) \\ \gamma^4(\mathbf{q}_1, \dot{\mathbf{q}}_1) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ \dot{\phi}_2 \mathbf{u}'_x^T \mathbf{P} \mathbf{A}_2 \mathbf{u}'_x \\ b \\ 0 \end{bmatrix} \quad (3.2.41)$$

where  $b = \dot{\mathbf{r}}_1^T \dot{\mathbf{r}}_1 + \dot{\phi}_2^2 (R^2 + R \mathbf{d}_{12}^T \mathbf{A}_1 \mathbf{u}'_x) + \dot{\mathbf{r}}_2^T \dot{\mathbf{r}}_2 - 2R \dot{\phi}_1 \mathbf{u}'_x^T \mathbf{A}_1^T \mathbf{P} \dot{\mathbf{r}}_1 + 2R \dot{\phi}_1 \mathbf{u}'_x^T \mathbf{A}_1^T \mathbf{P} \dot{\mathbf{r}}_2 - 2\dot{\mathbf{r}}_1^T \dot{\mathbf{r}}_2$ .

### 3.2.3.4 Windshield Wiper with Rotation Driver

The *windshield wiper* mechanism of Fig. 1.2.3 is modeled in the plane, with bodies and constraints shown in Fig. 3.2.7. Three bodies are pivoted in body 0 (ground), which is the vehicle chassis, and rotate about revolute joints, as shown. Body 1 is the crank that rotates relative to ground, with  $\dot{\phi}_1 - b(t) = 0$ . Bodies two and three are rocker arms that control rotation of the pair of windshield wipers shown in Fig. 1.2.3. Connecting rods of lengths 70 cm and 100 cm are modeled as distance constraints between bodies connected. Data for the joints and driver are as follows:

$$\begin{aligned} \text{rev k = 1: } & i_1 = 1, \mathbf{s}'_1 = \mathbf{0}; j_1 = 0, \mathbf{s}'_0 = \mathbf{0} \\ \text{rev k = 2: } & i_2 = 2, \mathbf{s}'_2 = \mathbf{0}; j_2 = 0, \mathbf{s}'_0 = -70\mathbf{u}_x \\ \text{rev k = 3: } & i_3 = 3, \mathbf{s}'_3 = \mathbf{0}; j_3 = 0, \mathbf{s}'_0 = 30\mathbf{u}_x \\ \text{dist k = 4: } & i_4 = 1, \mathbf{s}'_1 = -5\mathbf{u}'_y; j_4 = 2, \mathbf{s}'_2 = -8\mathbf{u}'_y, d = 70 \\ \text{dist k = 5: } & i_5 = 2, \mathbf{s}'_2 = 5\mathbf{u}'_y; j_5 = 3, \mathbf{s}'_3^P = 5\mathbf{u}'_y, d = 100 \\ \text{revD k = 6: } & i_6 = 1, \mathbf{s}'_1 = \mathbf{0}; j_6 = 0, \mathbf{s}'_0 = \mathbf{0}; \dot{\phi}_1 - b(t) = 0 \end{aligned} \quad (3.2.42)$$

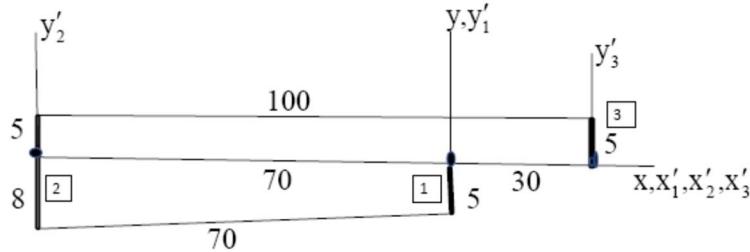


Figure 3.2.7 Windshield Wiper Mechanism

With  $\mathbf{q} = [\mathbf{r}_1^T \quad \dot{\phi}_1 \quad \mathbf{r}_2^T \quad \dot{\phi}_2 \quad \mathbf{r}_3^T \quad \dot{\phi}_3]^T \in \mathbb{R}^9$ , the constraint equations are

$$\Phi(\mathbf{q}) = \begin{bmatrix} \Phi^1(\mathbf{q}_1) \\ \Phi^2(\mathbf{q}_2) \\ \Phi^3(\mathbf{q}_3) \\ {}^4(\mathbf{q}_1, \mathbf{q}_2) \\ {}^5(\mathbf{q}_2, \mathbf{q}_3) \\ {}^6(\mathbf{q}_1) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 + 70\mathbf{u}_x \\ \mathbf{r}_3 - 30\mathbf{u}_x \\ (\mathbf{d}_{12}^T \mathbf{d}_{12} - 70^2)/2 \\ (\mathbf{d}_{23}^T \mathbf{d}_{23} - 100^2)/2 \\ \dot{\phi}_1 - b(t) \end{bmatrix} = \mathbf{0} \quad (3.2.43)$$

This is a system of nine nonlinear equations in nine generalized coordinates. If the constraint Jacobian is nonsingular, the system will have a unique solution as a function of time.

While Eqs. (3.2.4) and (3.2.10) can be used to expand the  $9 \times 9$  Jacobian, the explicit form of the matrix is not needed. The constraint Jacobian is numerically evaluated using computer routines in Code 3.9 of Appendix 3.A. Similarly, the 9-vector  $\gamma$  could be written, using Eqs. (3.2.7) and (3.2.13), as in the prior examples. It would be rather complex and is not needed in explicit form, since it is evaluated numerically using routines in Code 3.9.

Numerical examples of kinematic analysis using these and other models are presented in Section 3.10, using Code 3.9 of Appendix 3.A.

A family of three kinematic constraints and associated time dependent drivers provide the foundation for modeling and kinematic analysis of a broad spectrum of planar multibody systems. The associated holonomic constraint equations and derivatives derived are implemented in MATLAB Code 3.9 of Appendix 3.A and used in examples presented in Section 3.10. Models of four planar systems are presented to illustrate use of the library of constraints and provide kinematic equations that are used later in the chapter for numerical examples.

## Key Formulas

$$\Phi^{\text{dist}} \equiv (\mathbf{d}_{ij}^T \mathbf{d}_{ij} - d^2) / 2 = 0 \quad \mathbf{d}_{ij} \equiv \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i \quad (3.2.2) \quad (3.2.1)$$

$$\Phi^{\text{rev}} \equiv \mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i = \mathbf{0} \quad \Phi^{\text{tran}} \equiv - \begin{bmatrix} \mathbf{v}'_i^T \mathbf{P} \mathbf{A}_i^T \mathbf{d}_{ij} \\ \mathbf{v}'_i^T \mathbf{P} \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} = \mathbf{0} \quad (3.2.8) \quad (3.2.15)$$

### 3.3 Spatial Joints, Constraint Equations, and Drivers

In contrast with the modest complexity of planar system kinematic modeling, a wide variety of joint designs are available for use in spatial systems. The library of spatial kinematic joints is thus large and constraint equations that define their geometry are functions of seven generalized coordinates per body, in contrast to just three for planar bodies. The algebra and calculus involved in formulating spatial constraint equations and their derivatives is intricate and can only be carried out using the identities and derivatives derived in Section 2.6 and computer code that implements them in MATLAB Code 2.6 of Appendix 2.C. Fortunately, complexity can be managed using a library of four *building block constraints* that support the breadth of spatial kinematic constraints. Even more important than for planar joints, data that characterize each spatial joint are systematically defined for use in computation.

#### 3.3.1 Structure of Constraint Equations

Body fixed  $x'-y'-z'$  and  $x''-y''-z''$  reference frames shown in Fig. 3.3.1 are used to locate points such as  $P_i$  and  $P_j$  on bodies i and j and vectors  $s_i^P$ ,  $s_j^P$ ,  $a_i$ , and  $a_j$  that are fixed in bodies i and j and used to formulate kinematic constraint equations. The orientation of the  $x'_i-y'_i-z'_i$  frame relative to the global  $x-y-z$  frame is specified by the Euler parameter vector  $p_i$ , and similarly for body j. A constant *orthogonal transformation matrix*  $C_i$  transforms a vector  $a_i''$  that is represented in the  $x''_i-y''_i-z''_i$  frame to the  $x'_i-y'_i-z'_i$  frame in body i, and similarly for body j. Generalized coordinate vectors that define the location and orientation of bodies i and j, relative to the global  $x-y-z$  frame, are the *Cartesian generalized coordinates*

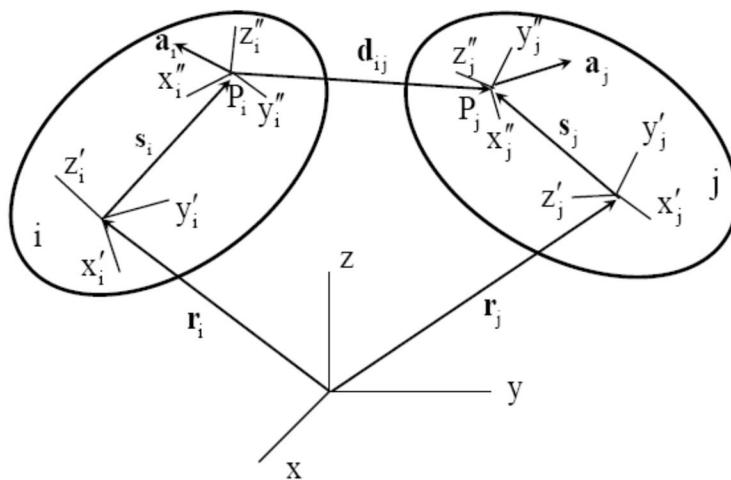
$$q_i = [r_i^T \quad p_i^T]^T \text{ and } q_j = [r_j^T \quad p_j^T]^T, \text{ where } p_i \text{ and } p_j \text{ are Euler parameters for bodies i and j, respectively.}$$


Figure 3.3.1 Geometry of Bodies in Space

The organization of spatial constraint equations and their derivatives is as presented in Section 3.1, with the exception that vector  $\chi_i$  is partitioned as  $\chi_i = [\chi_{r_i}^T \quad \chi_{p_i}^T]^T$ , where the dimensions of  $\chi_{r_i}$  and  $\chi_{p_i}$  are the same as  $r_i$  and  $p_i$ , and likewise for body j. For calculation of

derivatives of constraints that act on bodies  $i$  and  $j$  with respect to generalized coordinates, define  $\mathbf{q}_{ij} \equiv [\mathbf{q}_i^T \quad \mathbf{q}_j^T]^T$ . To simplify notation, the orientation transformation matrix of Eq. (2.5.22) for body  $i$  is denoted  $\mathbf{A}_i \equiv \mathbf{A}(\mathbf{p}_i)$ .

### 3.3.2 Building Block Constraints

Four *building block constraints* provide the foundation for representing a variety of spatial joints between pairs of bodies. Based on vectors shown in Fig. 3.3.1, they are as follows:

- (1) *distance constraint* specifies the positive distance  $d > 0$  between points  $P_i$  and  $P_j$
- (2) *spherical constraint* specifies that points  $P_i$  and  $P_j$  coincide
- (3) *dot1 constraint* specifies that nonzero body fixed vectors  $\mathbf{a}_i$  and  $\mathbf{a}_j$  are orthogonal
- (4) *dot2 constraint* specifies that nonzero body fixed vector  $\mathbf{a}_j$  and vector  $\mathbf{d}_{ij}$  are orthogonal

#### 3.3.2.1 Distance Constraint

The *distance constraint* shown in Fig. 3.3.2 consists of a *massless coupler* between bodies  $i$  and  $j$  that contains spherical joints at each end, attached to bodies  $i$  and  $j$  at points  $P_i$  and  $P_j$ . The joint is defined by locating joint definition points  $P_i$  and  $P_j$  in bodies  $i$  and  $j$  with body fixed vectors  $\mathbf{s}'_i$  and  $\mathbf{s}'_j$ , and by specifying the distance  $d > 0$  between points  $P_i$  and  $P_j$  on the coupler, as shown in Fig. 3.3.2.

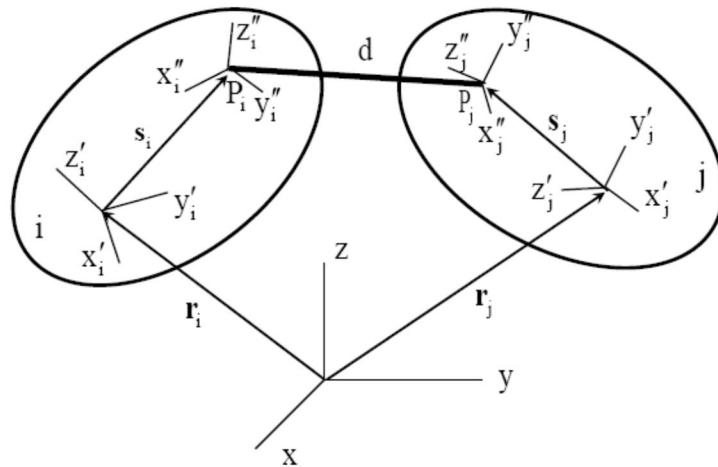


Figure 3.3.2 Geometry of the Spatial Distance Constraint

The geometry of the *spatial distance constraint* is simply that the distance between points  $P_i$  and  $P_j$  is equal to  $d > 0$ ; i.e.,

$$\Phi^{\text{dist}}(P_i, P_j, d) \equiv (\mathbf{d}_{ij}^T \mathbf{d}_{ij} - d^2)/2 = 0 \quad (3.3.1)$$

where

$$\mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i \quad (3.3.2)$$

Data that define distance constraint k are as follows:

$$\text{dist } k: i_k, \mathbf{s}'_{i_k}; j_k, \mathbf{s}'_{j_k}; d \quad (3.3.3)$$

Where k is the joint number in a multibody system, as in Section 3.1.1.

Equation (3.3.1) guarantees that the distance between points  $P_i$  and  $P_j$  is equal to  $d > 0$ , so it implies the geometry of the joint. The only danger in using this constraint is that it will introduce serious singularities in computation if the user specifies  $d = 0$ .

Since this joint is characterized by a scalar constraint equation, it should allow five relative degrees of freedom between the bodies connected. From Fig. 3.3.2, if body j is fixed in space, then point  $P_i$  on body i can move on a two-dimensional spherical surface of radius d, with center at point  $P_j$ . In addition, body i has three orientation degrees of freedom, for a total of five *relative degrees of freedom* of bodies connected by the joint.

Differentiating Eq. (3.3.1) yields the constraint *subJacobians*

$$\begin{aligned} \mathbf{q}_i^{\text{dist}} &= \left( (\hat{\mathbf{d}}_{ij}^T \mathbf{d}_{ij})_{\mathbf{q}_i} + (\mathbf{d}_{ij}^T \hat{\mathbf{d}}_i)_{\mathbf{q}_i} \right) / 2 = \left( (\hat{\mathbf{d}}_{ij}^T \mathbf{d}_{ij})_{\mathbf{q}_i} + (\hat{\mathbf{d}}_{ij}^T \mathbf{d}_{ij})_{\mathbf{q}_i} \right) / 2 \\ &= \mathbf{d}_{ij}^T (\mathbf{d}_{ij})_{\mathbf{q}_i} = -\mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_i, \mathbf{s}')] \\ \mathbf{q}_j^{\text{dist}} &= \left( (\hat{\mathbf{d}}_{ij}^T \mathbf{d}_{ij})_{\mathbf{q}_j} + (\mathbf{d}_{ij}^T \hat{\mathbf{d}}_i)_{\mathbf{q}_j} \right) / 2 = \left( (\hat{\mathbf{d}}_{ij}^T \mathbf{d}_{ij})_{\mathbf{q}_j} + (\hat{\mathbf{d}}_{ij}^T \mathbf{d}_{ij})_{\mathbf{q}_j} \right) / 2 \\ &= \mathbf{d}_{ij}^T (\mathbf{d}_{ij})_{\mathbf{q}_j} = \mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}')] \end{aligned} \quad (3.3.4)$$

where the derivative operator  $\mathbf{B}(\mathbf{p}, \mathbf{s}')$  is given by Eq. (2.6.25) and terms in Eq. (3.1.19) are

$$\begin{aligned} (\Phi_{\mathbf{q}_i}^{\text{dist}} \hat{\boldsymbol{\chi}}_i)_{\mathbf{q}_i} &= -\left( \mathbf{d}_{ij}^T \boldsymbol{\chi}_{r_i} + \mathbf{d}_{ij}^T \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \boldsymbol{\chi}_{p_i} \right)_{\mathbf{q}_i} = -\left( \boldsymbol{\chi}_{r_i}^T \mathbf{d}_{ij} + \boldsymbol{\chi}_{p_i}^T \mathbf{B}^T(\hat{\mathbf{p}}_i, \mathbf{s}'_i) \mathbf{d}_{ij} + \hat{\mathbf{d}}_{ij}^T \mathbf{B}(\boldsymbol{\chi}_{p_i}, \mathbf{s}'_i) \mathbf{p}_i \right)_{\mathbf{q}_i} \\ &= \left[ \left( \boldsymbol{\chi}_{r_i}^T + \boldsymbol{\chi}_{p_i}^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \right) \quad \left( \boldsymbol{\chi}_{r_i}^T + \boldsymbol{\chi}_{p_i}^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \right) \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) - \mathbf{d}_{ij}^T \mathbf{B}(\boldsymbol{\chi}_{p_i}, \mathbf{s}'_i) \right] \\ &= \left[ \mathbf{b}_i^T \quad \mathbf{b}_i^T \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) - \mathbf{d}_{ij}^T \mathbf{B}(\boldsymbol{\chi}_{p_i}, \mathbf{s}'_i) \right] \\ (\Phi_{\mathbf{q}_i}^{\text{dist}} \hat{\boldsymbol{\chi}}_i)_{\mathbf{q}_j} &= -\left[ \left( \boldsymbol{\chi}_{r_i}^T + \boldsymbol{\chi}_{p_i}^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \right) \quad \left( \boldsymbol{\chi}_{r_i}^T + \boldsymbol{\chi}_{p_i}^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \right) \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) \right] \\ &= -\left[ \mathbf{b}_i^T \quad \mathbf{b}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) \right] \\ (\Phi_{\mathbf{q}_j}^{\text{dist}} \hat{\boldsymbol{\chi}}_j)_{\mathbf{q}_i} &= -\left[ \left( \boldsymbol{\chi}_{r_j}^T + \boldsymbol{\chi}_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \right) \quad \left( \boldsymbol{\chi}_{r_j}^T + \boldsymbol{\chi}_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \right) \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \right] \\ &= -\left[ \mathbf{b}_j^T \quad \mathbf{b}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \right] \\ (\Phi_{\mathbf{q}_j}^{\text{dist}} \hat{\boldsymbol{\chi}}_j)_{\mathbf{q}_j} &= \left[ \left( \boldsymbol{\chi}_{r_j}^T + \boldsymbol{\chi}_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \right) \quad \left( \boldsymbol{\chi}_{r_j}^T + \boldsymbol{\chi}_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \right) \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) + \mathbf{d}_{ij}^T \mathbf{B}(\boldsymbol{\chi}_{p_j}, \mathbf{s}'_j) \right] \\ &= \left[ \mathbf{b}_j^T \quad \mathbf{b}_j^T \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) + \mathbf{d}_{ij}^T \mathbf{B}(\boldsymbol{\chi}_{p_j}, \mathbf{s}'_j) \right] \end{aligned} \quad (3.3.5)$$

where  $\mathbf{b}_i \equiv \mathbf{r}_i + \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)$  and  $\mathbf{b}_j \equiv \mathbf{r}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)$ . Substituting from Eqs. (3.3.5) into Eq. (3.1.19),

$$\mathbf{P2}^{\text{dist}}(\mathbf{q}_{ij}, \chi_{ij}) = \begin{bmatrix} \mathbf{P2}_i^{\text{dist}} & \mathbf{P2}_j^{\text{dist}} \end{bmatrix} \quad (3.3.6)$$

where

$$\begin{aligned} \mathbf{P2}_i^{\text{dist}} &= \left( \frac{\text{dist}}{\mathbf{q}_i} \right)_{\mathbf{q}_i} + \left( \frac{\text{dist}}{\mathbf{q}_j} \right)_{\mathbf{q}_j} = \left[ \mathbf{b}_i^T \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) - \mathbf{d}_{ij}^T \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \right] + \left[ \mathbf{b}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \right] \\ &= \left[ (\mathbf{b}_i^T - \mathbf{b}_j^T) \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) - \mathbf{d}_{ij}^T \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \right] \\ \mathbf{P2}_j^{\text{dist}} &= \left[ (\mathbf{b}_j^T - \mathbf{b}_i^T) \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) + \mathbf{d}_{ij}^T \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) \right] \end{aligned} \quad (3.3.7)$$

The right side of the acceleration constraint is thus

$$-\ddot{\mathbf{q}}_{ij} = -\mathbf{P2}^{\text{dist}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \quad (3.3.8)$$

This is typical of simplifications that are made possible through use of derivative operators of Section 2.6.

### 3.3.2.2 Spherical Constraint

The *spherical joint* (or *ball and socket joint*) shown in Fig. 3.3.3 is defined by the geometric condition that the center of the ball at point  $P_i$ , located by the body fixed vector  $\mathbf{s}'_i$  on body i, coincides with the center of the socket at  $P_j$  on body j, located by  $\mathbf{s}'_j$ . This condition is the vector equation

$$\Phi^{\text{sph}}(P_i, P_j) \equiv \mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i = \mathbf{0} \quad (3.3.9)$$

Data that define the spherical constraint are as follows:

$$\text{sph k: } i_k, \mathbf{s}'_{i_k}; j_k, \mathbf{s}'_{j_k} \quad (3.3.10)$$

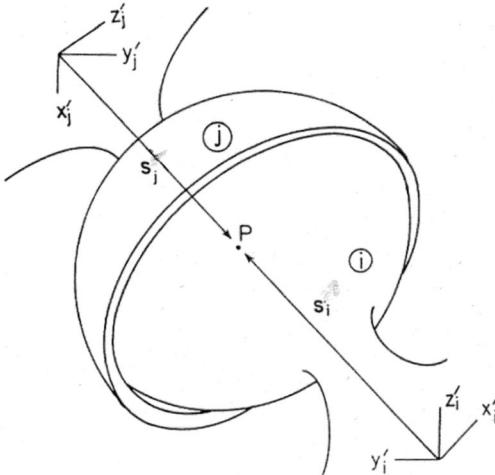


Figure 3.3.3 Geometry of the Spherical Joint

Equation (3.3.9), which is equivalent to the spherical joint, can be viewed as three scalar equations in the three components of the vectors involved. It thus eliminates three relative degrees of freedom between the bodies connected. This can be seen from Fig. 3.3.3. If body j is fixed in space, body i can rotate freely about the common point P, hence it has three degrees of orientation freedom relative to body j.

Differentiating Eq. (3.3.9) yields the constraint subJacobians

$$\begin{aligned}\Phi_{q_i}^{sph} &= -[\mathbf{I} \quad \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] \\ \Phi_{q_j}^{sph} &= [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)]\end{aligned}\quad (3.3.11)$$

and terms in Eq. (3.1.19) are

$$\begin{aligned}(\Phi_{q_i}^{sph} \dot{\chi}_i)_{q_i} &= [\mathbf{0} \quad -\mathbf{B}(\chi_{p_i}, \mathbf{s}'_i)] \\ (\Phi_{q_i}^{sph} \dot{\chi}_i)_{q_j} &= \mathbf{0} \\ (\Phi_{q_j}^{sph} \dot{\chi}_j)_{q_i} &= \mathbf{0} \\ (\Phi_{q_j}^{sph} \dot{\chi}_j)_{q_j} &= [\mathbf{0} \quad \mathbf{B}(\chi_{p_j}, \mathbf{s}'_j)]\end{aligned}\quad (3.3.12)$$

Thus,

$$\mathbf{P2}^{sph}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \left[ \begin{bmatrix} \mathbf{0} & -\mathbf{B}(\chi_{p_i}, \mathbf{s}'_i) \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} & \mathbf{B}(\chi_{p_j}, \mathbf{s}'_j) \end{bmatrix} \right] \quad (3.3.13)$$

and the right side of the acceleration equation is

$$-\boldsymbol{\gamma}^{sph} = -\mathbf{P2}^{sph}(\mathbf{q}_{ij}, \dot{\mathbf{p}}_{ij}) \ddot{\mathbf{p}}_{ij} \quad (3.3.14)$$

### 3.3.2.3 Dot 1 Constraint

The *dot1 constraint* is that nonzero vectors  $\mathbf{a}_i$  and  $\mathbf{a}_j$  fixed in bodies i and j are orthogonal; i.e.,

$$\Phi^{\text{dot1}}(\mathbf{a}_i, \mathbf{a}_j) = \mathbf{a}_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{a}'_j = 0 \quad (3.3.15)$$

Differentiating Eq. (3.3.15) yields the constraint subJacobians

$$\begin{aligned}\Phi_{q_i}^{\text{dot1}} &= [\mathbf{0} \quad \mathbf{a}'_j^T \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i)] \\ \Phi_{q_j}^{\text{dot1}} &= [\mathbf{0} \quad \mathbf{a}'_i^T \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{a}'_j)]\end{aligned}\quad (3.3.16)$$

and terms in Eq. (3.1.19) are

$$\begin{aligned}
\left(\Phi_{q_i}^{\text{dot1}} \ddot{\chi}_i\right)_{q_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{a}'^T \mathbf{A}_j^T \mathbf{B}(\chi_{p_i}, \mathbf{a}') \end{bmatrix} \\
\left(\Phi_{q_i}^{\text{dot1}} \ddot{\chi}_i\right)_{q_j} &= \begin{bmatrix} \mathbf{0} & \chi_{p_i}^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{a}') \mathbf{B}(\mathbf{p}_j, \mathbf{a}') \end{bmatrix} \\
\left(\Phi_{q_j}^{\text{dot1}} \ddot{\chi}_j\right)_{q_i} &= \begin{bmatrix} \mathbf{0} & \chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}') \mathbf{B}(\mathbf{p}_i, \mathbf{a}') \end{bmatrix} \\
\left(\Phi_{q_j}^{\text{dot1}} \ddot{\chi}_j\right)_{q_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{a}'^T \mathbf{A}_i^T \mathbf{B}(\chi_{p_j}, \mathbf{a}') \end{bmatrix}
\end{aligned} \tag{3.3.17}$$

Thus,

$$\mathbf{P}2^{\text{dot1}}(q_{ij}) = \begin{bmatrix} \mathbf{P}2_i^{\text{dot1}} & \mathbf{P}2_j^{\text{dot1}} \end{bmatrix} \tag{3.3.18}$$

where

$$\begin{aligned}
\mathbf{P}2_i^{\text{dot1}} &= \begin{bmatrix} \mathbf{0} & \mathbf{a}'^T \mathbf{A}_j^T \mathbf{B}(\chi_{p_i}, \mathbf{a}') + \chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}') \mathbf{B}(\mathbf{p}_i, \mathbf{a}') \end{bmatrix} \\
\mathbf{P}2_j^{\text{dot1}} &= \begin{bmatrix} \mathbf{0} & \mathbf{a}'^T \mathbf{A}_i^T \mathbf{B}(\chi_{p_j}, \mathbf{a}') + \chi_{p_i}^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{a}') \mathbf{B}(\mathbf{p}_j, \mathbf{a}') \end{bmatrix}
\end{aligned}$$

and the right side of the acceleration equation is

$$-\boldsymbol{\gamma}^{\text{dot1}} = -\mathbf{P}2^{\text{dot1}}(q_{ij}, \dot{\mathbf{p}}_{ij}) \dot{\mathbf{p}}_{ij} \tag{3.3.19}$$

### 3.3.2.4 Dot 2 Constraint

The *dot2 constraint* is that the nonzero vector  $\mathbf{a}_j$  fixed in body j and vector  $\mathbf{d}_{ij}$  are orthogonal; i.e.,

$$\Phi^{\text{dot2}}(\mathbf{a}_j, \mathbf{d}_{ij}) = \mathbf{a}'^T \mathbf{A}_j^T \mathbf{d}_{ij} = 0 \tag{3.3.20}$$

Differentiating Eq. (3.3.20) yields the constraint subJacobians

$$\begin{aligned}
\left.\frac{\partial}{\partial q_i}\right|_{q_j}^{\text{dot2}} &= \mathbf{a}'^T \mathbf{A}_j^T [-\mathbf{I}_3 \quad -\mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] \\
\left.\frac{\partial}{\partial q_j}\right|_{q_i}^{\text{dot2}} &= \mathbf{a}'^T \mathbf{A}_j^T [\mathbf{I}_3 \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)] + \mathbf{d}_{ij}^T [\mathbf{0} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{a}')]
\end{aligned} \tag{3.3.21}$$

and terms in Eq. (3.1.19) are

$$\begin{aligned}
\left(\Phi_{q_i}^{\text{dot2}} \ddot{\chi}_i\right)_{q_i} &= \begin{bmatrix} \mathbf{0} & -\mathbf{a}'^T \mathbf{A}_j^T \mathbf{B}(\chi_{p_i}, \mathbf{s}'_i) \end{bmatrix} \\
\left(\Phi_{q_i}^{\text{dot2}} \ddot{\chi}_i\right)_{q_j} &= \begin{bmatrix} \mathbf{0} & -(\chi_{r_i}^T + \chi_{p_i}^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i)) \mathbf{B}(\mathbf{p}_j, \mathbf{a}') \end{bmatrix} \\
\left(\Phi_{q_j}^{\text{dot2}} \ddot{\chi}_j\right)_{q_i} &= -\chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}') [\mathbf{I}_3 \quad \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] \\
\left(\Phi_{q_j}^{\text{dot2}} \ddot{\chi}_j\right)_{q_j} &= [\chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}') \quad \mathbf{c}]
\end{aligned} \tag{3.3.22}$$

where  $\mathbf{c} = \chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}') \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) + \mathbf{d}_{ij}^T \mathbf{B}(\chi_{p_j}, \mathbf{a}') + \mathbf{a}'^T \mathbf{A}_j^T \mathbf{B}(\chi_{p_j}, \mathbf{s}'_j) + (\chi_{r_j}^T + \chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j)) \mathbf{B}(\mathbf{p}_j, \mathbf{a}')$ .

Thus,

$$\mathbf{P}2^{\text{dot2}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \begin{bmatrix} \mathbf{P}2_i^{\text{dot2}} & \mathbf{P}2_j^{\text{dot2}} \end{bmatrix} \quad (3.3.23)$$

where

$$\mathbf{P}2_i^{\text{dot2}} = -[\chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}'_j) \quad \mathbf{a}'_j^T \mathbf{A}_j^T \mathbf{B}(\chi_{p_i}, \mathbf{s}'_i) + \chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}'_j) \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)]$$

$$\mathbf{P}2_j^{\text{dot2}} = [\chi_{p_j}^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}'_j) \quad \mathbf{c} - (\chi_{p_i}^T + \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i)) \mathbf{B}(\mathbf{p}_j, \mathbf{a}'_j)]$$

and the right side of the acceleration equation is

$$\ddot{\mathbf{r}}^{\text{dot2}} = -\mathbf{P}2^{\text{dot2}}(\mathbf{q}_{ij}, \dot{\mathbf{p}}_{ij}) \ddot{\mathbf{p}}_{ij} \quad (3.3.24)$$

Expressions for *building block constraints* and their derivatives that are derived in this section have been coded in Code 3.11 in Appendix 3.A. They are used to assemble constraint equations of joints and their derivatives that are derived in subsequent sections for use in kinematic and dynamic analysis.

### 3.3.3 Cylindrical, Revolute, and Translational Joints

At joint definition points  $P_i$  and  $P_j$  on bodies  $i$  and  $j$ , respectively, right hand triads of unit vectors  $\mathbf{u}^i$ - $\mathbf{v}^i$ - $\mathbf{w}^i$  and  $\mathbf{u}^j$ - $\mathbf{v}^j$ - $\mathbf{w}^j$  fixed in bodies  $i$  and  $j$  are used to enforce cylindrical, revolute, and translational joints.

#### 3.3.3.1 Cylindrical Joint

The geometry of the joint shown schematically in Fig. 3.3.4 enables relative translation along and rotation about colinear body fixed unit vectors  $\mathbf{w}^i$  and  $\mathbf{w}^j$ ; i.e., a *cylindrical joint*. If constraints are added to prevent relative translation or relative rotation, revolute or translational joints, respectively, are obtained.

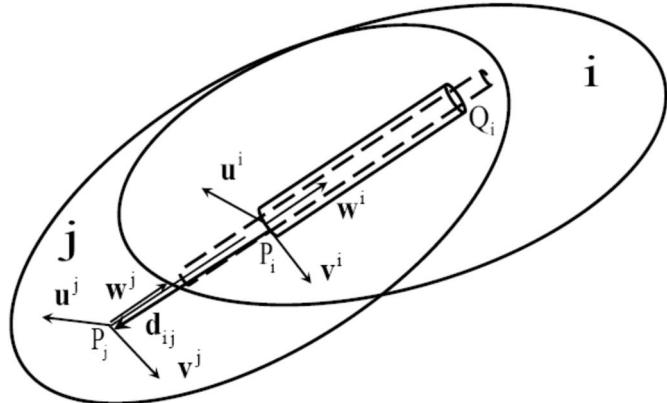


Figure 3.3.4. Geometry of Cylindrical, Revolute, and Translational Joints

The body fixed orthogonal unit vectors  $\mathbf{u}'^i$ ,  $\mathbf{v}'^i$ , and  $\mathbf{w}'^i$  shown in Fig. 3.3.4 may be represented in the global x-y-z frame, to assist in formulation of constraint equations, as

$$\begin{aligned} \mathbf{u}^i &= \mathbf{A}_i \mathbf{u}'^i \\ \mathbf{v}^i &= \mathbf{A}_i \mathbf{v}'^i \\ \mathbf{w}^i &= \mathbf{A}_i \mathbf{w}'^i \end{aligned} \quad (3.3.25)$$

and likewise for body j. Points  $P_i$  and  $P_j$  are fixed in bodies i and j, located as shown in Fig. 3.3.1 by constant body fixed vectors  $s'_i$  and  $s'_j$ . Thus,  $P_i$  is located in the global frame by the vector  $\mathbf{r}^{P_i} = \mathbf{r}_i + \mathbf{A}_i s'_i$ , and  $P_j$  by the vector  $\mathbf{r}^{P_j} = \mathbf{r}_j + \mathbf{A}_j s'_j$ .

With the vector  $\mathbf{d}_{ij}$  from  $P_i$  to  $P_j$  of Eq. (3.3.2),  $\mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j s'_j - \mathbf{r}_i - \mathbf{A}_i s'_i$ , the following four *cylindrical joint constraint equations* specify that  $\mathbf{w}^i$  and  $\mathbf{w}^j$  are collinear:

$$\Phi_{cyl}(\mathbf{q}_i, \mathbf{q}_j) \equiv \begin{bmatrix} \mathbf{u}^{jT} \mathbf{d}_{ij} \\ \mathbf{v}^{jT} \mathbf{d}_{ij} \\ \mathbf{u}^{jT} \mathbf{w}^i \\ \mathbf{v}^{jT} \mathbf{w}^i \end{bmatrix} = \begin{bmatrix} \Phi^{\text{dot2}}(\mathbf{u}^j, \mathbf{d}_{ij}) \\ \Phi^{\text{dot2}}(\mathbf{v}^j, \mathbf{d}_{ij}) \\ \Phi^{\text{dot1}}(\mathbf{w}^i, \mathbf{u}^j) \\ \Phi^{\text{dot1}}(\mathbf{w}^i, \mathbf{v}^j) \end{bmatrix} = \mathbf{0} \quad (3.3.26)$$

If  $\mathbf{d}_{ij} \neq \mathbf{0}$  these equations imply that  $\mathbf{w}^i$ ,  $\mathbf{d}_{ij}$ , and  $\mathbf{w}^j$  are parallel and, since they have points in common, they are collinear. If  $\mathbf{d}_{ij} = \mathbf{0}$ , the third and fourth equations imply that  $\mathbf{w}^i$  and  $\mathbf{w}^j$  are parallel and, since they have a point in common, they are collinear.

Since  $\mathbf{v}'^i = \tilde{\mathbf{w}}'^i \mathbf{u}'^i$  and  $\mathbf{v}'^j = \tilde{\mathbf{w}}'^j \mathbf{u}'^j$ , only  $\mathbf{u}'^i \cdot \mathbf{w}'^i$  and  $\mathbf{u}'^j \cdot \mathbf{w}'^j$  fixed in bodies i and j, respectively, need be specified. Data that define cylindrical joint k are as follows:

$$\text{cyl k: } i_k, s'_{i_k}, \mathbf{u}'^{i_k}, \mathbf{w}'^{i_k}; j_k, s'_{j_k}, \mathbf{u}'^{j_k}, \mathbf{w}'^{j_k} \quad (3.3.27)$$

where k is the joint number in a multibody system, as in Section 3.2.1.

The subJacobians and operators of Eq. (3.1.19) for the cylindrical constraint are obtained from Eqs. (3.3.16), (3.3.17), (3.3.21), and (3.3.22) for the dot1 and dot2 building block constraints. Rather than writing explicit expressions for these intricate terms, they are assembled in MATLAB Code 3.11 of Appendix 3.A, using subroutines that evaluate terms in Eqs. (3.3.16), (3.3.17), (3.3.21), and (3.3.22).

### 3.3.3.2 Revolute Joint

Adding the constraint that the scalar product of  $\mathbf{A}_j \mathbf{w}'^j$  and  $\mathbf{d}_{ij}$  is zero to the cylindrical constraints of Eq. (3.3.26) implies that points  $P_i$  and  $P_j$  are coincident and yields the *revolute joint* that satisfies Eqs (3.3.26) and the fifth revolute joint constraint equation,

$$\Phi^{\text{rev5}}(\mathbf{q}_i, \mathbf{q}_j) \equiv \mathbf{w}'^{jT} \mathbf{A}_j^T \mathbf{d}_{ij} = \Phi^{\text{dot2}}(\mathbf{w}^j, \mathbf{d}_{ij}) = 0 \quad (3.3.28)$$

The full set of *revolute joint constraint equations* is

$$\Phi^{\text{rev}}(\mathbf{q}_i, \mathbf{q}_j) \equiv [\Phi^{\text{cylT}}(\mathbf{q}_i, \mathbf{q}_j) \quad \Phi^{\text{rev5}}(\mathbf{q}_i, \mathbf{q}_j)]^T = \mathbf{0} \quad (3.3.29)$$

Data that define revolute joint k are as follows:

$$\text{rev k: } i_k, s'_{i_k}, \mathbf{u}'^{i_k}, \mathbf{w}'^{i_k}; j_k, s'_{j_k}, \mathbf{u}'^{j_k}, \mathbf{w}'^{j_k} \quad (3.3.30)$$

The subJacobians and terms in Eq. (3.1.19) for the revolute constraint are obtained from Eqs. (3.3.16), (3.3.17), (3.3.21), and (3.3.22) for the dot1 and dot2 building block constraints.

### 3.3.3.3 Translational Joint

Adding the constraint that the scalar product of  $\mathbf{u}^i$  and  $\mathbf{v}^i$  is zero to the cylindrical constraints of Eq. (3.3.26) implies that there can be no relative rotation about the common  $\mathbf{A}_i \mathbf{w}^i$  axis and yields the *translational joint* that satisfies Eq. (3.3.26) and the fifth translational joint constraint equation,

$$\Phi^{\text{tran}5}(\mathbf{q}_i, \mathbf{q}_j) \equiv \Phi^{\text{dot1}}(\mathbf{v}^i, \mathbf{u}^j) = 0 \quad (3.3.31)$$

The full set of *translational joint constraint equations* is

$$\Phi^{\text{tran}}(\mathbf{q}_i, \mathbf{q}_j) \equiv \left[ \Phi^{\text{cylT}}(\mathbf{q}_i, \mathbf{q}_j) \quad \Phi^{\text{tran}5}(\mathbf{q}_i, \mathbf{q}_j) \right]^T = \mathbf{0} \quad (3.3.32)$$

Data that define translational joint k are as follows:

$$\text{tran k: } i_k, s'_{i_k}, \mathbf{u}'^{i_k}, \mathbf{w}'^{i_k}; j_k, s'_{j_k}, \mathbf{u}'^{j_k}, \mathbf{w}'^{j_k} \quad (3.3.33)$$

The subJacobians and operators of Eq. (3.1.19) for the translational constraint are obtained from Eqs. (3.3.16), (3.3.17), (3.3.21), and (3.3.22) for the dot1 and dot2 building block constraints.

Note that the guide in body i and translating bar in body j of Fig. 3.3.4 for the translational joint need not be cylinders. In reality, and consistent with the present model, they are rectangular in character.

### 3.3.4 Composite Joints

Joints between pairs of bodies may often be modeled using an *intermediate element*, thought of as a *massless body*, that serves only to define kinematic relations between the bodies. A substantial library of *composite joints* for spatial systems is presented in Vol I of this series (Haug, 1989). Three commonly encountered composite joints are presented in this section.

#### 3.3.4.1 Universal Joint

A *universal joint* between bodies i and j, shown in Fig. 3.3.5, is constructed with an intermediate body, or cross, that is pivoted in bodies i and j. The geometric definition of the universal joint is that the center of the cross is fixed in bodies i and j, defined by points  $P_i$  and  $P_j$  fixed in the respective bodies, and that the axes of the cross are orthogonal. As with many mechanical joints, a universal joint that connects a pair of bodies can be assembled in different ways. For example, body j in Fig. 3.3.5 could be held in the same orientation shown and the joint could be assembled with body i inverted relative to the cross; i.e., with the vector  $\mathbf{w}'^i$  pointing upward. In order to define vector equations that characterize this joint, unit vectors  $\mathbf{w}'^i$  and  $\mathbf{w}'^j$  on bars of the cross in bodies i and j, respectively, as shown in Fig 3.3.5. Since points  $P_i$  and  $P_j$  must coincide, the spherical constraint equation of Eq. (3.3.9) must hold, where  $P_i$  and  $P_j$  are located on the respective bodies by vectors  $s_i$  and  $s_j$ ,

$$\Phi^{\text{sph}}(P_i, P_j) \equiv d_{ij} = r_j + A_j s'_j - r_i - A_i s'_i = 0 \quad (3.3.34)$$

Since vectors  $\mathbf{w}'^i$  and  $\mathbf{w}'^j$  lie on the bars of the cross, which must be orthogonal, they must satisfy the dot1 constraint,

$$\text{dot1}(\mathbf{w}'^i, \mathbf{w}'^j) = \mathbf{w}'^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{w}'^j = 0 \quad (3.3.35)$$

Data that define universal joint k are as follows:

$$\text{univ k: } i_k, s'_{i_k}, w'^{i_k}; j_k, s'_{j_k}, w'^{j_k} \quad (3.3.36)$$

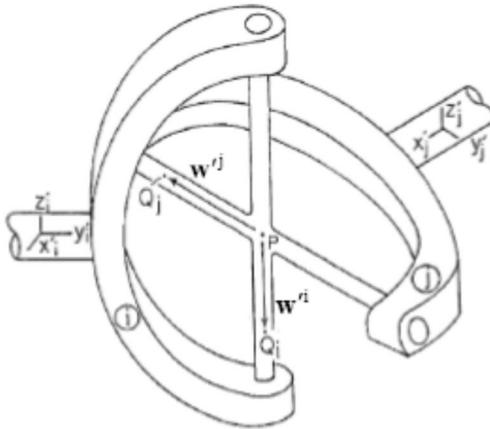


Figure 3.3.5 Geometry of the Universal Joint

In order to show that Eqs. (3.3.34) and (3.3.35) imply the geometry of the universal joint, note first that Eq. (3.3.34) implies coincidence of points  $P_i$  and  $P_j$ , just as was shown in the case of the spherical joint. Since vectors  $\mathbf{w}'^i$  and  $\mathbf{w}'^j$  are unit vectors, neither is zero and Eq. (3.3.35) implies that the two vectors are orthogonal. Thus, the bars of the cross, which intersect at point P, are orthogonal. These are the geometric conditions for the joint, which are implied by Eqs. (3.3.34) and (3.3.35).

Note that Eq. (3.3.35) is satisfied if either of the joint definition vectors is reversed in sign. For example, if  $\mathbf{w}'^i$  is replaced by its negative, which for this joint is equivalent to rotating the cross about vector  $\mathbf{w}'^j$  by an angle  $\pi$ , the constraint equation is still satisfied; i.e.,

$\text{dot1}(\mathbf{w}'^i, \mathbf{w}'^j) = -\mathbf{w}'^{iT} \mathbf{w}'^j = \mathbf{w}'^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{w}'^j = 0$ . This is consistent with the observation made in geometric definition of the joint that it can be assembled in two distinct configurations. Since it takes engineering judgment to determine which configuration is to be selected, it is not surprising that mathematical equations that characterize the joint cannot make the distinction between the two configurations. As is shown in Chapter 2, however, once a configuration is selected, if only small variations in position and orientation of the bodies are permitted in an increment of time, enforcement of Eqs. (3.3.34) and (3.3.35) will maintain the configuration of the joint.

Since Eqs. (3.3.34) and (3.3.35) comprise four scalar equations, they remove four relative degrees of freedom between the bodies connected. This can be seen from Fig. 3.3.5, by holding body j fixed in space and investigating the admissible motion of body i. Body i is free to rotate

about each of the orthogonal axes of the cross, but cannot rotate about the axis orthogonal to the cross. It thus has two rotational degrees of freedom, relative to body j.

**Example 3.3.1:** Consider the *universal joint* shown in Fig 3.3.6. The  $x'_1$  and  $x'_2$  axes are constrained to lie in the x-y plane. Let the driving angle of body 1 be  $\theta_1$  and the output angle of body 2 be  $\theta_2$ . Both bodies are constrained to rotate in the x-y plane, so  $\phi = 0$  when  $\theta_1 = 0$ .

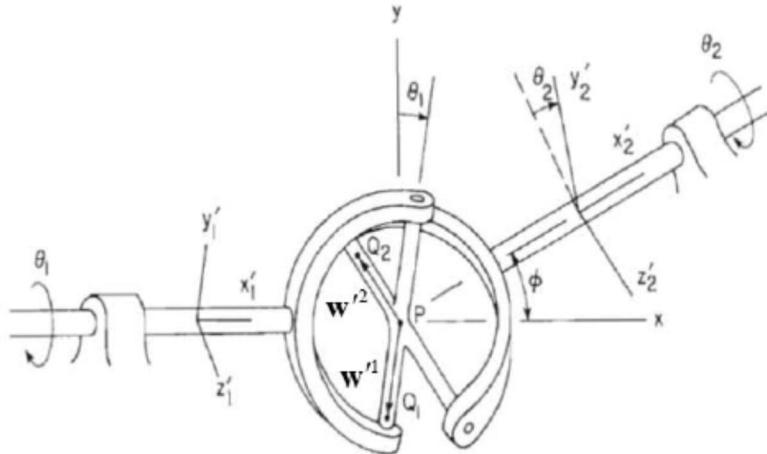


Figure 3.3.6 Universal Joint Example

From Fig. 3.3.6,

$$\begin{aligned}\mathbf{w}'^1 &= -\mathbf{u}_y = [0 \quad -1 \quad 0]^T \\ \mathbf{w}'^2 &= -\mathbf{u}_z = [0 \quad 0 \quad -1]^T \\ \mathbf{A}_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 \\ 0 & \sin \theta_1 & \cos \theta_1 \end{bmatrix} \\ \mathbf{A}_2 &= \begin{bmatrix} \cos \phi & -\cos \theta_2 \sin \phi & \sin \theta_2 \sin \phi \\ \sin \phi & \cos \theta_2 \cos \phi & -\sin \theta_2 \cos \phi \\ 0 & \sin \theta_2 & \cos \theta_2 \end{bmatrix}\end{aligned}$$

From Eq. (3.3.15),

$$\begin{aligned}{}^{d1}(\mathbf{w}^1, \mathbf{w}^2) &= \mathbf{w}'^{1T} \mathbf{A}_1^T \mathbf{A}_2 \mathbf{w}'^2 \\ &= [0 \quad -1 \quad 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & \sin \theta_1 \\ 0 & -\sin \theta_1 & \cos \theta_1 \end{bmatrix} \begin{bmatrix} \cos \phi & -\cos \theta_2 \sin \phi & \sin \theta_2 \sin \phi \\ \sin \phi & \cos \theta_2 \cos \phi & -\sin \theta_2 \cos \phi \\ 0 & \sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (3.3.37) \\ &= -\cos \theta_1 \sin \theta_2 \cos \phi + \sin \theta_1 \cos \theta_2 = 0\end{aligned}$$

Dividing by  $\cos\theta_1\cos\theta_2$ ,

$$\tan\theta_1 = \tan\theta_2 \cos\phi \quad (3.3.38)$$

Hence,  $\theta_2 = \text{Arctan}(\tan\theta_1 / \cos\phi)$ .

Taking the differential of both sides of Eq. (3.3.37),

$$-(\cos_1 \cos\theta_2 \cos\phi + \sin_1 \sin\theta_2) \delta\theta_2 + (\sin_1 \sin\theta_2 \cos\phi + \cos_1 \cos\theta_2) \delta\theta_1 = 0 \quad (3.3.39)$$

Similarly, taking the time derivative of Eq. (3.3.37),

$$-(\cos_1 \cos\theta_2 \cos\phi + \sin_1 \sin\theta_2) \dot{\theta}_2 + (\sin_1 \sin\theta_2 \cos\phi + \cos_1 \cos\theta_2) \dot{\theta}_1 = 0 \quad (3.3.40)$$

Note that when  $\phi = \pi/2$ ,  $\cos\phi = 0$  and for any  $0 < \theta_2 < \pi/2$ , Eq. (3.3.38) yields  $\tan\theta_1 = 0$ , which shows that body 1 is locked. Similarly, from Eq. (3.3.39), with  $\cos\phi = 0$ ,  $\tan\theta_1 \tan\theta_2 \delta\theta_2 = \delta\theta_1$ . Since  $\tan\theta_1 = 0$ , this equation indicates that for any variation  $\delta\theta_2$ ,  $\delta\theta_1 = 0$ . This singular behavior is interpreted as *lock-up* of the driving shaft, as illustrated in Fig. 3.3.7. Thus, the universal joint has a physically *singular configuration*, which must be avoided in applications.

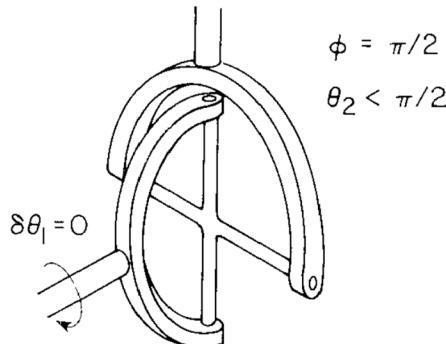


Figure 3.3.7 Singular Behavior of Universal Joint

In most applications of the universal joint; e.g., in the driveline of a vehicle, the angle  $\phi$  in Fig. 3.3.6 is near zero. If  $\phi = 0$ ,  $\cos\phi = 1$  and Eq. (3.3.37) shows that  $\theta_1 = \theta_2$ , so the input and output angles are the same. Also, Eq. (3.3.40) shows that  $\dot{\theta}_2 = \dot{\theta}_1$ . If  $\phi$  is small, which is the case in most applications of the universal joint, Taylor's theorem yields  $\cos\phi \approx 1 - \phi^2/2$ . Thus, Eq. (3.3.40) yields

$$\dot{\theta}_2 \approx \dot{\theta}_1 + \frac{(\sin_1 \sin\theta_2 - \phi^2 \cos_1 \cos\theta_2 / 2)}{(\cos_1 \cos\theta_2 - \phi^2 \sin_1 \sin\theta_2 / 2)} \dot{\theta}_1 \quad (3.3.41)$$

and the output velocity varies somewhat from the input velocity. If two universal joints are used in a driveline, with an intermediate shaft, this variation can be cancelled.

This example clearly shows that there may be *singular configurations* of mechanisms. It should however be taken with a grain of salt. The configuration shown in Fig 3.3.6 is typical of

the use of the universal joint, but only when the angle  $\phi$  is small. While the configuration shown in Fig. 3.3.7 is not practical in an engineering sense, it does illustrate that singularities can occur in mechanisms if one is careless in design, modeling, or application.

### 3.3.4.2 Strut Composite Joint

The *strut composite joint* shown in Fig. 3.3.8 consists of a coupler that has a cylindrical joint about the body fixed vector  $\mathbf{w}'^j$  in body  $j$  and a spherical joint at point  $P_i$  that is fixed in body  $i$ . This is the so-called *McPherson strut* suspension shown in Fig. 1.2.6, in the vehicle model of Section 1.1.5.

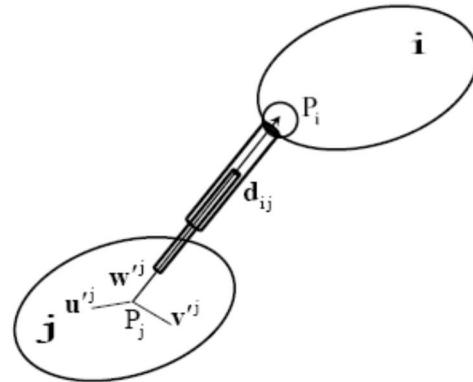


Figure 3.3.8 Geometry of the Strut Composite Joint

The geometric definition of the strut composite joint is that vector  $\mathbf{w}^j$  passes through point  $P_i$ . This requires that  $\mathbf{w}^j$  and  $\mathbf{d}_{ij}$  are parallel, hence collinear. Equations intended to imply this condition are

$$\begin{aligned}\overset{\text{dot2}}{\left( \mathbf{u}^j, \mathbf{d}_{ij} \right)} &= \mathbf{u}'^{jT} \mathbf{A}_j^T \mathbf{d}_{ij} = 0 \\ \overset{\text{dot2}}{\left( \mathbf{v}^j, \mathbf{d}_{ij} \right)} &= \mathbf{v}'^{jT} \mathbf{A}_j^T \mathbf{d}_{ij} = 0\end{aligned}\quad (3.3.42)$$

Data that define strut composite joint  $k$  are as follows:

$$\text{str } k: i_k, s'_{i_k}; j_k, s'_{j_k}, \mathbf{u}'^{j_k}, \mathbf{w}'^{j_k} \quad (3.3.43)$$

To show that Eqs. (3.3.42) imply the geometry of the strut composite joint, note first that if  $\mathbf{d}_{ij} \neq \mathbf{0}$ , these equations imply that  $\mathbf{w}^j$  and  $\mathbf{d}_{ij}$  are parallel. Since they have a point in common, they are collinear and  $\mathbf{w}^j$  passes through point  $P_j$ . In the special case  $\mathbf{d}_{ij} = \mathbf{0}$ ,  $P_i$  coincides with  $P_j$ , and  $\mathbf{w}^j$  passes through point  $P_j$ . In either case, Eqs. (3.3.42) imply the geometry of the strut composite joint.

Since the strut composite joint is defined by two scalar constraint equations, it should allow four relative degrees of freedom between the bodies connected. To see that this is true, consider body  $j$  as fixed. Point  $P_i$  on body  $i$  can translate along the  $\mathbf{w}^j$  axis and body  $i$  can rotate freely in space about point  $P_i$ , for a total of four relative degrees of freedom.

### 3.3.4.3 Revolute-Spherical Composite Joint

The *revolute-spherical composite joint* shown in Fig. 3.3.9 is comprised of a massless coupler with a spherical joint connected to body i at point  $P_i$  and a revolute joint connected to body j, with  $\mathbf{w}^j$  as its centerline. Joint definition points  $P_i$  and  $P_j$  define the centers of the respective joints on the bodies. It is required that the vector  $\mathbf{d}_{ij}$  has constant length equal to  $d > 0$  and that  $\mathbf{d}_{ij}$  is orthogonal to  $\mathbf{w}^j$ .

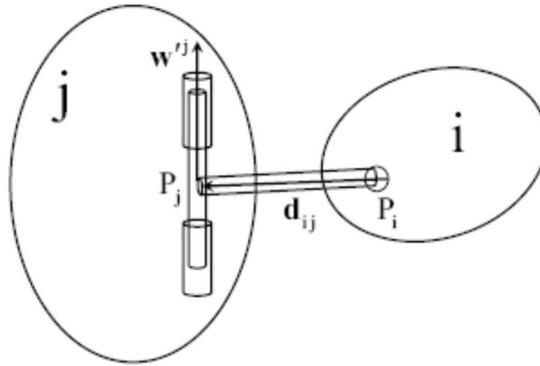


Figure 3.3.9 Geometry of the Revolute-Spherical Composite Joint

Equations that imply the geometry of the joint are simply the distance constraint between points  $P_i$  and  $P_j$  and the dot2 orthogonality condition for vectors  $\mathbf{w}^j$  and  $\mathbf{d}_{ij}$ ; i.e.,

$$\begin{aligned} \text{dist}\left(P_i, P_j, d\right) &= \left(\mathbf{d}_{ij}^T \mathbf{d}_{ij} - d^2\right) / 2 = 0 \\ \text{dot2}\left(\mathbf{w}^j, \mathbf{d}_{ij}\right) &= \mathbf{w}'^{jt} \mathbf{A}_j^T \mathbf{d}_{ij} = 0 \end{aligned} \quad (3.3.44)$$

Since  $d \neq 0$ , the first equation implies  $\mathbf{d}_{ij} \neq \mathbf{0}$  and the second equation implies  $\mathbf{w}^j$  and  $\mathbf{d}_{ij}$  are orthogonal. Data that define revolute-spherical composite joint k are as follows:

$$\text{revsph } k: i_k, s'_{i_k}; j_k, s'_{j_k}, \mathbf{w}'^{j_k}; d \quad (3.3.45)$$

Since two constraint equations characterize this joint, it should allow four relative degrees of freedom between the bodies connected. If body j is fixed in space, point  $P_i$  on body i is free to move on a one-dimensional circle of radius d in the plane perpendicular to vector  $\mathbf{w}^j$ . In addition, body i has three rotational degrees of freedom, for a total of four relative degrees of freedom.

### 3.3.5 Kinematic Drivers

In case the distance d defined in Fig. 3.3.5 and Eq. (3.3.1) is specified as a function of time,  $d(t) > 0$ ; e.g., by a hydraulic or electrical actuator, a time dependent *distance driver* is defined in the form

$$\text{distD}\left(P_i, P_j, d\right) \equiv \left(\mathbf{d}_{ij}^T \mathbf{d}_{ij} - d^2(t)\right) / 2 = 0 \quad (3.3.46)$$

Data that define distance driver k are as with the associated distance constraint, with the addition of  $d(t)$ ,

$$\text{distD } k: i_k, s'_{i_k}; j_k, s'_{j_k}; d(t) \quad (3.3.47)$$

This scalar constraint equation eliminates one relative degree of freedom between bodies  $i$  and  $j$ . Derivatives of this driver with respect to generalized coordinates and  $\gamma$  are the same as for the distance constraint of Section 3.3.2. Time derivatives for this driver are  $\frac{\text{distD}}{t} = -\dot{d}$ ,  $\frac{\text{distD}}{tt} = -\ddot{d} - \dot{d}\dot{d}$ , and  $\frac{\text{distD}}{qt} = 0$ .

In a cylindrical or revolute joint of Fig. 3.3.4, a relative rotation  $\theta_{ij}(t)$  may be specified as a function of time, as shown in Fig 3.3.10, where  $\dot{\theta}_{ij}(q_i, q_j)$  is defined by Eq. (2.4.37). Rather than using this complex relation, Eqs. (2.4.35) and (2.4.36) are more practical. Even these equations raise intricate issues. Two alternative forms of *relative rotation driver* may be used, with the notation of Fig. 3.3.10. By definition of scalar product of unit vectors,  $\cos(\theta_{ij}) = \mathbf{u}^{iT} \mathbf{u}^j$  and  $\cos(\pi/2 - \theta_{ij}) = \mathbf{v}^{iT} \mathbf{u}^j = \cos(\pi/2) \cos(\theta_{ij}) + \sin(\pi/2) \sin(\theta_{ij}) = \sin(\theta_{ij})$ . In order to avoid singularities, the relative rotation driver is enforced as follows:

$$\begin{aligned} \text{rotD} &\equiv \begin{cases} \mathbf{v}^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}^j - \sin(\theta_{ij}(t)) = 0, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \mathbf{u}^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}^j - \cos(\theta_{ij}(t)) = 0, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \\ &= \begin{cases} \text{dotl}(\mathbf{v}^i, \mathbf{u}^j) - \sin(\theta_{ij}(t)) = 0, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \text{dotl}(\mathbf{u}^i, \mathbf{u}^j) - \cos(\theta_{ij}(t)) = 0, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \end{aligned} \quad (3.3.48)$$

where  $s = \mathbf{v}^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}^j$  and  $c = \mathbf{u}^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}^j$ . Data that define revolute and cylindrical relative rotation driver  $k$  are as follows:

$$\begin{aligned} \text{revD } k: &i_k, s'_{i_k}, \mathbf{u}^{i_k}, \mathbf{w}^{i_k}; j_k, s'_{j_k}, \mathbf{u}'^{j_k}, \mathbf{w}'^{j_k}; \theta_{ij}(t) \\ \text{cylD } k: &i_k, s'_{i_k}, \mathbf{u}^{i_k}, \mathbf{w}^{i_k}; j_k, s'_{j_k}, \mathbf{u}'^{j_k}, \mathbf{w}'^{j_k}; \theta_{ij}(t) \end{aligned} \quad (3.3.49)$$

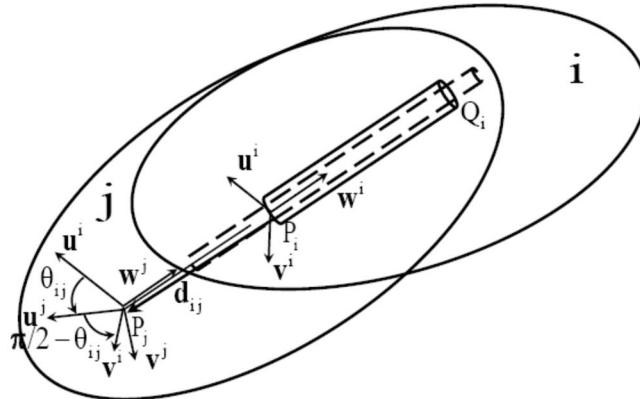


Figure 3.3.10 Relative Rotation Driver

SubJacobians of this driver with respect to generalized coordinates are

$$\begin{aligned} \dot{\mathbf{q}}_i^{\text{rotD}} &= \begin{cases} \frac{\text{dotl}(\mathbf{v}^i, \mathbf{u}^j)}{q_i}, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \frac{\text{dotl}(\mathbf{u}^i, \mathbf{u}^j)}{q_i}, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \\ \dot{\mathbf{q}}_j^{\text{rotD}} &= \begin{cases} \frac{\text{dotl}(\mathbf{v}^i, \mathbf{u}^j)}{q_j}, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \frac{\text{dotl}(\mathbf{u}^i, \mathbf{u}^j)}{q_j}, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \end{aligned} \quad (3.3.50)$$

and time derivatives for this driver are

$$\begin{aligned} \dot{\mathbf{q}}_i^{\text{rotD}} &\equiv \begin{cases} -\dot{\theta}_{ij}(t) \cos(\theta_{ij}(t)), & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \dot{\theta}_{ij}(t) \sin(\theta_{ij}(t)), & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \\ \ddot{\mathbf{q}}_i^{\text{rotD}} &\equiv \begin{cases} -\ddot{\theta}_{ij}(t) \cos(\theta_{ij}(t)) + \dot{\theta}_{ij}^2(t) \sin(\theta_{ij}(t)), & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \ddot{\theta}_{ij}(t) \sin(\theta_{ij}(t)) + \dot{\theta}_{ij}^2(t) \cos(\theta_{ij}(t)), & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \end{aligned} \quad (3.3.51)$$

and  $\dot{\mathbf{q}}_i^{\text{rotD}} = \mathbf{0}$ . Using the subJacobians of Eq. (3.3.50), terms in Eq. (3.1.19) are

$$\begin{aligned} \left( \frac{\text{rotD}}{q_i} \dot{\chi}_{q_i} \right)_{q_i} &= \begin{cases} \left( \frac{\text{dotl}(\mathbf{v}^i, \mathbf{u}^j)}{q_i} \dot{\chi}_{q_i} \right)_{q_i}, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \left( \frac{\text{dotl}(\mathbf{u}^i, \mathbf{u}^j)}{q_i} \dot{\chi}_{q_i} \right)_{q_i}, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \\ \left( \frac{\text{rotD}}{q_i} \ddot{\chi}_{q_i} \right)_{q_j} &= \begin{cases} \left( \frac{\text{dotl}(\mathbf{v}^i, \mathbf{u}^j)}{q_i} \ddot{\chi}_{q_i} \right)_{q_j}, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \left( \frac{\text{dotl}(\mathbf{u}^i, \mathbf{u}^j)}{q_i} \ddot{\chi}_{q_i} \right)_{q_j}, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \\ \left( \frac{\text{rotD}}{q_j} \dot{\chi}_{q_j} \right)_{q_i} &= \begin{cases} \left( \frac{\text{dotl}(\mathbf{v}^i, \mathbf{u}^j)}{q_j} \dot{\chi}_{q_j} \right)_{q_i}, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \left( \frac{\text{dotl}(\mathbf{u}^i, \mathbf{u}^j)}{q_j} \dot{\chi}_{q_j} \right)_{q_i}, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \\ \left( \frac{\text{rotD}}{q_j} \ddot{\chi}_{q_j} \right)_{q_j} &= \begin{cases} \left( \frac{\text{dotl}(\mathbf{v}^i, \mathbf{u}^j)}{q_j} \ddot{\chi}_{q_j} \right)_{q_j}, & \text{if } \text{abs}(c) \geq \text{abs}(s) \\ \left( \frac{\text{dotl}(\mathbf{u}^i, \mathbf{u}^j)}{q_j} \ddot{\chi}_{q_j} \right)_{q_j}, & \text{if } \text{abs}(s) > \text{abs}(c) \end{cases} \end{aligned} \quad (3.3.52)$$

The P2 operator and right side of the acceleration equation are thus

$$\begin{aligned} \mathbf{P2}^{\text{rotD}}(\mathbf{q}_{ij}, \chi_{ij}) &= \left[ \left( \frac{\text{rotD}}{q_i} \dot{\chi}_{q_i} \right)_{q_i} + \left( \frac{\text{rotD}}{q_j} \dot{\chi}_{q_j} \right)_{q_i} \quad \left( \frac{\text{rotD}}{q_i} \ddot{\chi}_{q_i} \right)_{q_j} + \left( \frac{\text{rotD}}{q_j} \ddot{\chi}_{q_j} \right)_{q_j} \right] \\ -\gamma^{\text{rotD}} &= -\mathbf{P2}^{\text{rotD}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \end{aligned} \quad (3.3.53)$$

While the expressions for this driver involve two forms of equation and logic to select the proper form, computer implementation is routine.

In general, when driving constraints are of the form

$$\Phi^D(\mathbf{q}, t) = \Phi(\mathbf{q}) + \mathbf{f}(t) = \mathbf{0} \quad (3.3.54)$$

the associated velocity and acceleration constraint equations are

$$\Phi_q^D(\mathbf{q}, t)\dot{\mathbf{q}} + \Phi_t^D(\mathbf{q}, t) = \Phi_q(\mathbf{q})\dot{\mathbf{q}} + \dot{\mathbf{f}}(t) = \mathbf{0} \quad (3.3.55)$$

$$\Phi_q^D(\mathbf{q}, t)\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) + \Phi_{tt}^D(\mathbf{q}, t) = \Phi_q(\mathbf{q})\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) + \ddot{\mathbf{f}}(t) = \mathbf{0} \quad (3.3.56)$$

Just as subroutines in kinematic analysis computer code are required for evaluation of  $\Phi_q(\mathbf{q})$  and  $\gamma(\mathbf{q}, \dot{\mathbf{q}})$  for all constraints, subroutines are required for evaluation of  $\mathbf{f}(t)$ ,  $\dot{\mathbf{f}}(t)$ , and  $\ddot{\mathbf{f}}(t)$  for driving constraints.

### 3.3.6 Euler Parameter Normalization Constraints

In addition to kinematic constraints and drivers, for spatial systems, *Euler parameter normalization constraints*

$$p_i(\mathbf{p}_i) = (\mathbf{p}_i^T \mathbf{p}_i - 1) / 2 = 0, \quad i=1, \dots, nb \quad (3.3.57)$$

must be satisfied. The Jacobians of these constraints are simply

$$p_i(\mathbf{p}_i)_{\mathbf{q}_i} = \begin{bmatrix} \mathbf{0} & \mathbf{p}_i^T \end{bmatrix} \quad (3.3.58)$$

and P2 is

$$P2^{p_i}(\mathbf{p}_i, \chi_{p_i}) = \left( p_i(\mathbf{p}_i)_{\mathbf{q}_i} \chi_{p_i} \right)_{\mathbf{q}_i} = \left( \mathbf{p}_i^T \chi_{p_i} \right)_{\mathbf{q}_i} = \begin{bmatrix} \mathbf{0} & \chi_{p_i}^T \end{bmatrix} \quad (3.3.59)$$

Differentiating the velocity equation  $\mathbf{p}_i^T \dot{\mathbf{p}}_i = 0$ , the acceleration equation is  $\mathbf{p}_i^T \ddot{\mathbf{p}}_i = -\dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i$ , so the acceleration right side is

$$-\gamma^{p_i} = -\dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i \quad (3.3.60)$$

### 3.3.7 Spatial System Examples

#### 3.3.7.1 Two Body Spatial Pendulum

Two spheres of unit radius are shown in Fig. 3.3.12, constrained to form a *two-body spatial pendulum*. The origin of the body 1 reference frame is fixed to the origin of the x-y-z global frame by a spherical joint, so body 1 is free to rotate, but cannot translate. A unit distance constraint acts between points  $P_1$  on body 1 at  $-\mathbf{u}'_{1z}$  and  $P_2$  on body 2 at  $\mathbf{u}'_{2z}$ . No drivers are imposed on this system, as it is prepared only for dynamic simulation in Chapter 5.

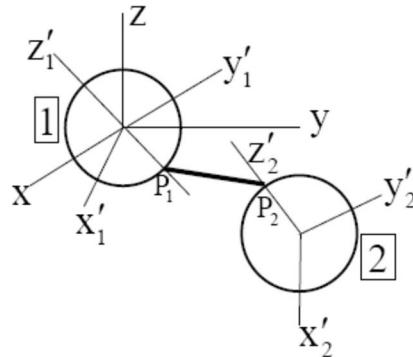


Figure 3.3.12 Two Body Spatial Pendulum

Data for the spherical joint between body 1 and ground and the distance constraint between bodies 1 and 2 are

$$\begin{aligned} \text{sph k = 1: } & i_1 = 1, \mathbf{s}'^P_1 = \mathbf{0}; j_1 = 0, \mathbf{s}'^P_0 = \mathbf{0} \\ \text{dist k = 2: } & i_2 = 1, \mathbf{s}'^P_1 = -\mathbf{u}_z; j_1 = 2, \mathbf{s}'^P_2 = \mathbf{u}_z \end{aligned}$$

The spherical constraint is simply

$$\Phi^{sph}(\mathbf{q}) = \mathbf{r}_1 = \mathbf{0}$$

with the vector from point  $P_1$  to  $P_2$ ,  $\mathbf{d}_{12} = \mathbf{r}_2 + \mathbf{A}(\mathbf{p}_2)\mathbf{u}'_z + \mathbf{A}(\mathbf{p}_1)\mathbf{u}'_z$ , and the distance constraint is

$$\Phi^{dist}(\mathbf{q}) = (\mathbf{d}_{12}^T \mathbf{d}_{12} - 1)/2 = 0$$

The Euler parameter normalization constraints are

$$\Phi^{p_i}(\mathbf{q}) = (\mathbf{p}_i^T \mathbf{p}_i - 1)/2 = 0, i=1,2$$

The vector of four kinematic constraints is thus

$$\Phi(\mathbf{q}) = \begin{bmatrix} \mathbf{r}_1 \\ (\mathbf{d}_{12}^T \mathbf{d}_{12} - 1)/2 \\ (\mathbf{p}_1^T \mathbf{p}_1 - 1)/2 \\ (\mathbf{p}_2^T \mathbf{p}_2 - 1)/2 \end{bmatrix} = \mathbf{0}_{5 \times 1}$$

Since this is a system of five nonlinear constraint equations in eleven generalized coordinates, if the constraint Jacobian has full rank, the system should have eight degrees of freedom. These are three degrees of rotation freedom for each body and two degrees of freedom of point  $P_2$  on the surface of a unit sphere with center at point  $P_1$

From Eqs. (3.3.4) and (3.3.58), the constraint Jacobian is

$$\Phi_q = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{d}_{12}^T \mathbf{B}(\mathbf{p}_1, \mathbf{u}'_z) & \mathbf{d}_{12}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_1^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{p}_2^T \end{bmatrix}$$

From Eq. (3.3.5), with  $\chi_{r_i} = \mathbf{0}$ , derivatives associated with the distance constraint are

$$\begin{aligned} (\Phi_{q_1}^{dist} \dot{\chi}_1)_{q_1} &= [\chi_{p_1}^T \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}_z) \mathbf{B}(\mathbf{p}_1, \mathbf{u}_z) + \mathbf{d}_{12}^T \mathbf{B}(\chi_{p_1}, \mathbf{u}_z)] \\ (\Phi_{q_1}^{dist} \dot{\chi}_1)_{q_2} &= [\chi_{p_1}^T \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}_z) \quad \chi_{p_1}^T \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}_z) \mathbf{B}(\mathbf{p}_2, \mathbf{u}_z)] \\ (\Phi_{q_2}^{dist} \dot{\chi}_2)_{q_1} &= [(\chi_{r_2}^T + \chi_{p_2}^T \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}_z)) \mathbf{B}(\mathbf{p}_1, \mathbf{u}_z)] \\ (\Phi_{q_2}^{dist} \dot{\chi}_2)_{q_2} &= [(\chi_{r_2}^T + \chi_{p_2}^T \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}_z)) \quad (\chi_{r_2}^T + \chi_{p_2}^T \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}_z)) \mathbf{B}(\mathbf{p}_2, \mathbf{u}_z) + \mathbf{d}_{12}^T \mathbf{B}(\chi_{p_2}, \mathbf{u}_z)] \end{aligned}$$

From Eqs. (3.1.9) and (3.3.59),

$$\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{a}\mathbf{B}(\mathbf{p}_1, \mathbf{u}'_z) + \mathbf{d}_{12}^T \mathbf{B}(\boldsymbol{\chi}_{\mathbf{p}_1}, \mathbf{u}'_z) & \mathbf{a} & \mathbf{a}\mathbf{B}(\mathbf{p}_2, \mathbf{u}'_z) + \mathbf{d}_{12}^T \mathbf{B}(\boldsymbol{\chi}_{\mathbf{p}_2}, \mathbf{u}'_z) \\ \mathbf{0} & \boldsymbol{\chi}_{\mathbf{p}_1}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\chi}_{\mathbf{p}_2}^T \end{bmatrix}$$

where  $\mathbf{a} = (\boldsymbol{\chi}_{\mathbf{p}_1}^T \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}'_z) + \boldsymbol{\chi}_{\mathbf{r}_2}^T + \boldsymbol{\chi}_{\mathbf{p}_2}^T \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}'_z))$ . Finally, from Eq. (3.1.17),

$$\boldsymbol{\gamma} = \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$$

### 3.3.7.2 Two-Bar Spatial Mechanism with Two Drivers

Bar 1 of the two-bar spatial mechanism shown in Fig. 3.3.13 moves in the x-y plane, with rotation angle  $\theta_{01}$  about the z axis from the x axis. Bar 2 rotates in a plane that is orthogonal to the  $x'_1$  axis fixed in body 1, with rotation angle  $\theta_{12}$  from an axis at the end of body 1 that is parallel to the  $-z$  axis.

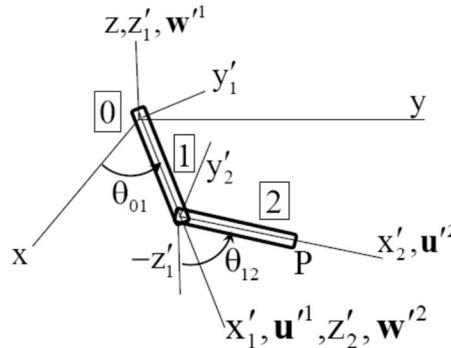


Figure 3.3.13 Two-Bar Spatial Mechanism

The revolute joints provide five constraint equations each and there are two Euler parameter normalization constraints, for a total of twelve nonlinear constraint equations in fourteen generalized coordinates. If the constraint Jacobian has full rank, the system will have two degrees of freedom. Adding two rotational drivers should yield a unique solution as a function of time.

Vectors used in definition of the revolute joints and drivers are as follows:

$$\begin{aligned} \text{rev k = 1: } & i_1 = 1, \mathbf{s}'_1 = \mathbf{0}, \mathbf{w}'^1 = \mathbf{u}_z, \mathbf{u}'^1 = \mathbf{u}_x; \\ & j_1 = 0, \mathbf{s}'_0 = \mathbf{0}, \mathbf{w}'^0 = \mathbf{u}_z, \mathbf{u}'^0 = \mathbf{u}_x \\ \text{rev k = 2: } & i_2 = 1, \mathbf{s}'_1 = \mathbf{u}_x, \mathbf{w}'^1 = \mathbf{u}_x, \mathbf{u}'^1 = \mathbf{u}_z; \\ & j_2 = 2, \mathbf{s}'_2 = \mathbf{0}, \mathbf{w}'^2 = \mathbf{u}_z, \mathbf{u}'^2 = \mathbf{u}_x \\ \text{rotD k = 3: } & i_3 = 1; j_3 = 0, \text{ data same as rev1; } \theta_{01}(t) = \omega_1 t \\ \text{rotD k = 4: } & i_4 = 1; j_4 = 0, \text{ data same as rev2; } \theta_{12}(t) = \omega_2 t \end{aligned} \tag{3.3.61}$$

Two Euler parameter normalization constraints are automatically implemented.

While explicit constraint expressions and their derivatives could be written, as in Example 3.3.7.1, it is more convenient and efficient to use computer routines in Code 3.11 of Appendix 3.A to form and solve kinematic equations.

### 3.3.7.3 Spatial Slider-Crank

The *spatial slider-crank* mechanism shown in Fig. 3.3.14 has been used as a computational example in dynamics with friction (Marques, Flores, Pimenta Claro, and Lankarani, 2016) and without friction (Haug, 1989). It involves potentially large *constraint reaction forces* and *near singular configurations* that may dominate friction effects. Body 1 rotates about an axis that is parallel to and offset from the global x axis in ground (body 0), as a crank in the y-z plane of radius of 0.08 m. The slider (body 2) translates in a guide along the global x axis in ground. A distance constraint of length  $d > 0$  connects the centroid of body 2 with the end of the crank in body 1, as a model of the connecting rod. A rotation driver for the crank about the revolute joint axis is  $\theta_{10}(t) = \omega t$ , where  $\omega$  is constant.

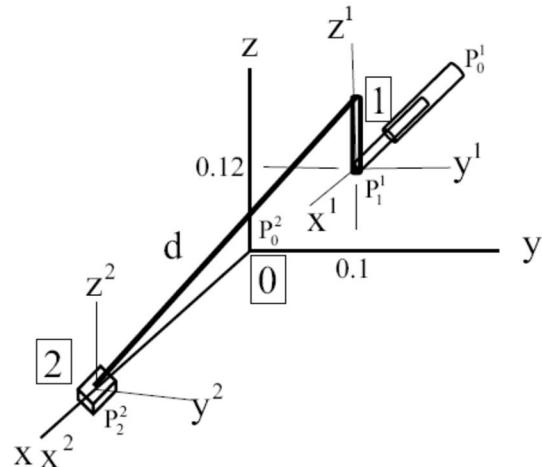


Figure 3.3.14 Spatial Slider-Crank

The revolute and translational joints provide five constraint equations each, the distance constraint provides one, and there are two Euler parameter normalization constraints, for a total of thirteen nonlinear constraint equations in fourteen generalized coordinates. If the constraint Jacobian has full rank, the system will have one degree of freedom. Adding the rotational driver should yield a unique solution as a function of time.

Vectors used in definition of the revolute joint are  $\mathbf{u}'^1 = \mathbf{u}'^0 = \mathbf{u}_z$  and  $\mathbf{w}'^1 = \mathbf{w}'^0 = \mathbf{u}_x$ .

Other data that define the joints are tabulated as follows:

$$\begin{aligned}
 \text{rev k = 1: } & i_1 = 1, \mathbf{s}'_1 = \mathbf{0}, \mathbf{w}'^1 = \mathbf{u}_x, \mathbf{u}'^1 = \mathbf{u}_z; \\
 & j_1 = 0, \mathbf{s}'_0 = [0 \ 0.1 \ 0.12]^T, \mathbf{w}'^0 = \mathbf{u}_x, \mathbf{u}'^0 = \mathbf{u}_z \\
 \text{tran k = 2: } & i_2 = 2, \mathbf{s}'_2 = \mathbf{0}, \mathbf{w}'^2 = \mathbf{u}_x, \mathbf{u}'^2 = \mathbf{u}_z; \\
 & j_2 = 0, \mathbf{s}'^2_0 = \mathbf{0}, \mathbf{w}'^0 = \mathbf{u}_x, \mathbf{u}'^0 = \mathbf{u}_z \\
 \text{dist k = 3: } & i_3 = 1, \mathbf{s}'_1 = 0.08\mathbf{u}_z; j_3 = 2, \mathbf{s}'_2 = \mathbf{0}; d = \text{dist} \\
 \text{rotD k = 4: } & i_4 = 1; j_4 = 0, \text{ data same as rev1; } \theta_{10}(t) = \omega t
 \end{aligned} \tag{3.3.62}$$

Two Euler parameter normalization constraints are automatically implemented.

While explicit constraint expressions and their derivatives could be written, as in Example 3.3.7.1, it is more convenient and efficient to use computer routines in MATLAB Code 3.11 of Appendix 3.A to form and solve kinematic equations.

### 3.3.7.4 Fly-Ball Governor

The *fly-ball governor* of Fig. 3.3.15 is comprised of four moving bodies and ground. Body 1 is the rotor in a revolute joint with ground. Body 2 is the collar that translates on and rotates with the rotor. Bodies 3 and 4 contain heavy balls at their ends and rotate in revolute joints at the top of the rotor. Two distance constraints connect bodies 3 and 4 with the collar, to control fuel feed. The revolute constraints with bodies 3 and 4 on body 1 are offset 0.02 m from its centerline, as are the distance constraint attachment points on body 2. With three revolute and one translational joints, two distance constraints and four Euler parameter normalization constraints, there are a total of 26 constraint equations in 28 generalized coordinates. If the constraint Jacobian has full rank, the system has two degrees of freedom.

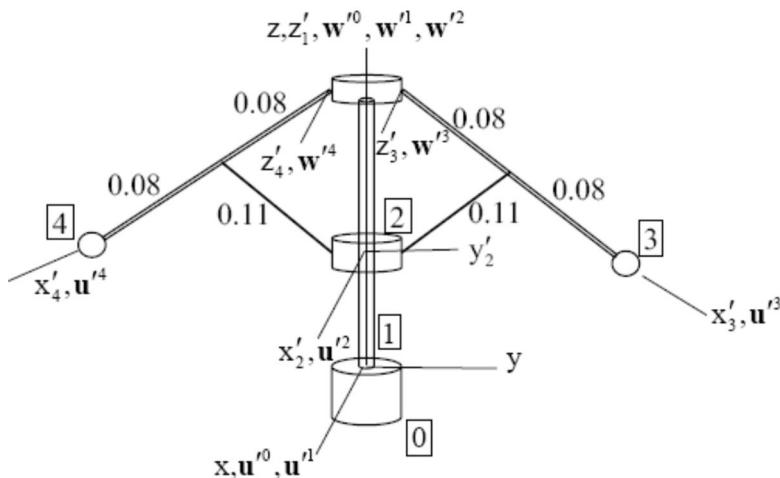


Figure 3.3.15 Fly-Ball Governor

Two drivers are imposed. The first specifies the angle of rotation of the shaft relative to ground and the second specifies the distance between the collar and the base of the shaft. Data that define the kinematics of the mechanism are as follows:

$$\begin{aligned}
& \text{rev k = 1: } i_1 = 1, \mathbf{s}'_1 = \mathbf{0}, \mathbf{w}'^1 = \mathbf{u}_z, \mathbf{u}'^1 = \mathbf{u}_x; \\
& \quad j_1 = 0, \mathbf{s}'_0 = \mathbf{0}, \mathbf{w}'^0 = \mathbf{u}_z, \mathbf{u}'^0 = \mathbf{u}_x \\
& \text{tran k = 2: } i_2 = 2, \mathbf{s}'_2 = \mathbf{0}, \mathbf{w}'^2 = \mathbf{u}_z, \mathbf{u}'^2 = \mathbf{u}_x; \\
& \quad j_2 = 1, \mathbf{s}'_0^2 = \mathbf{0}, \mathbf{w}'^0 = \mathbf{u}_z, \mathbf{u}'^0 = \mathbf{u}_x \\
& \text{rev k = 3: } i_3 = 1, \mathbf{s}'_1 = 0.2\mathbf{u}_z + 0.02\mathbf{u}_y, \mathbf{w}'^1 = \mathbf{u}_x, \mathbf{u}'^1 = \mathbf{u}_z; \\
& \quad j_3 = 3, \mathbf{s}'_3 = \mathbf{0}, \mathbf{w}'^3 = \mathbf{u}_z, \mathbf{u}'^3 = \mathbf{u}_x \\
& \text{rev k = 4: } i_1 = 1, \mathbf{s}'_1 = 0.2\mathbf{u}_z - 0.02\mathbf{u}_y, \mathbf{w}'^1 = \mathbf{u}_x, \mathbf{u}'^1 = \mathbf{u}_z; \\
& \quad j_1 = 4, \mathbf{s}'_4 = \mathbf{0}, \mathbf{w}'^4 = \mathbf{u}_z, \mathbf{u}'^4 = \mathbf{u}_x \\
& \text{dist k = 5: } i_5 = 2, \mathbf{s}'_2 = 0.02\mathbf{u}_y; j_5 = 3, \mathbf{s}'_3 = 0.08\mathbf{u}_x; d = 0.11 \\
& \text{dist k = 6: } i_6 = 1, \mathbf{s}'_2 = -0.02\mathbf{u}_y; j_6 = 4, \mathbf{s}'_4 = 0.08\mathbf{u}_x; d = 0.11 \\
& \text{rotD k=7: } i_7 = 1, \mathbf{s}'_1 = \mathbf{0}; j_7 = 0, \mathbf{s}'_0 = \mathbf{0}; \theta_{10}(t) = \omega \times t \\
& \text{distD k = 8: } i_8 = 1, \mathbf{s}'_1 = \mathbf{0}; j_8 = 2, \mathbf{s}'_0 = \mathbf{0}; d(t) = 0.1 + 0.02 \sin(\omega \times t)
\end{aligned} \tag{3.3.63}$$

In addition to these kinematic constraints and drivers, two Euler parameter normalization constraints are automatically implemented.

While explicit constraint expressions and their derivatives could be written, as in Example 3.3.7.1, it is more convenient and efficient to use computer routines in MATLAB Code 3.11 of Appendix 3.A to form and solve kinematic equations.

### 3.3.7.5 Automobile Suspension

The *automobile suspension* outlined in Figs. 1.2.5 to 1.2.7 of Section 1.1 is modeled using the schematic of Fig. 3.3.16 and building block and composite constraints. The chassis (body 1) and front wheel spindle assemblies (bodies 2 and 3) are represented by seven generalized coordinates each. A revolute-spherical composite joint is pivoted at point A and has a spherical connection to body 2 at point B. A strut composite constraint acts between body 2 at point B and body 1 at point C. Identical constraints act on body 3. Front wheel assemblies are steered with tie rods (distance constraints) that act between the steering rack (body 6), whose position in the chassis is specified as a function of time, and points on bodies 2 and 3. The right rear wheel assembly (body 4) is pivoted in a revolute joint with body 1 at point H, using relative coordinate  $\theta_{14}$ . The left rear wheel assembly (body 5) is likewise modeled with a relative coordinate  $\theta_{15}$ . Constraint connectivity is as follows:

rev-sph k=1:  $i_1 = 1, j_1 = 2$ ; 2 constraints  
 strut k = 2 :  $i_2 = 2, j_2 = 1$ ; 2 constraints  
 rev-sph k=3:  $i_3 = 1, j_3 = 3$ ; 2 constraints  
 strut k = 4 :  $i_4 = 2, j_4 = 1$ ; 2 constraints  
 dist k = 5 :  $i_5 = 2, j_5 = 6$ ; 1 constraint  
 dist k = 6 :  $i_6 = 3, j_6 = 6$ ; 1 constraint  
 Eulerparnorm: bodies 1,2,3; 3 constraints  
 Total; 13 constraints

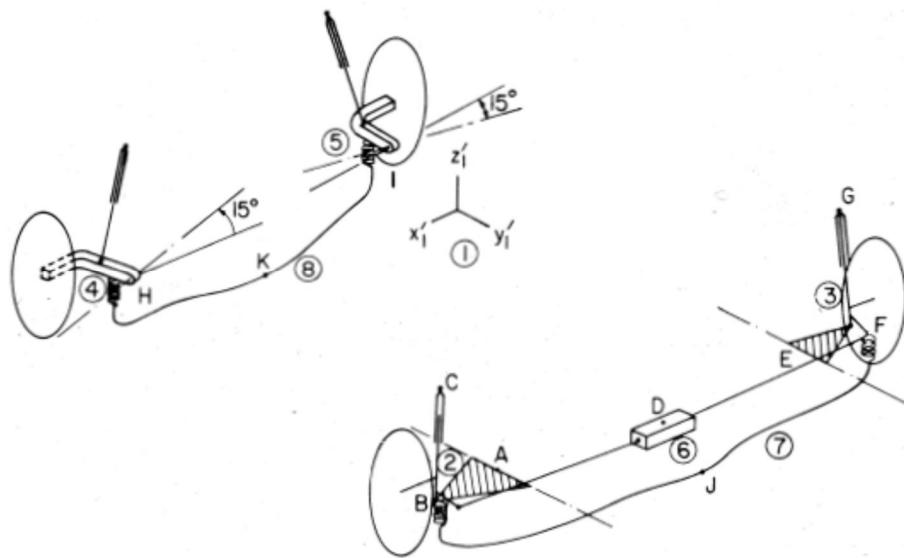
(3.3.64)


Figure 3.3.16 Automobile Suspension Schematic

Generalized coordinates are seven vectors  $\mathbf{q}_i = [\mathbf{r}_i^T \quad \mathbf{p}_i^T]^T$ ,  $i = 1, 2, 3$ ,  $\theta_{14}$ , and  $\theta_{15}$ , for a total of 23 generalized coordinates. If the constraint Jacobian has full rank, the model has 10 degrees of freedom. The 10 degrees of freedom are as follows: 6 dof of chassis (body 1) and one wheel suspension travel degree of freedom for each of four wheel assemblies. Steering angles of the front wheels are controlled by a specified rack displacement history and the associated tie rod constraints. This is an example whose dynamics are treated in detail in the automotive industry. Kinematic analysis is less meaningful for the entire system.

While the spectrum of joints in spatial systems is much broader than that for planar systems, a library of four building block constraints and associated derivatives provides the foundation for representing a substantial number of constraints between pairs of bodies. Joints that directly connect pairs of bodies and joints that involve intermediate couplers, called composite joints, are presented using the building block constraints. Spatial multibody systems are modeled using the constraint library for numerical kinematic analysis in Section 3.12.

It is important to note that the library of derivative operators derived in Section 2.6 enables systematic computer construction of large numbers of nonlinear constraint equations and their derivatives that are required for kinematic and dynamic simulation. Numerical solution methods presented in Section 2.2 may thus be brought to bear. Without these systematic tools and their computer implementation, simulation of the vehicle of Section 3.3.7.3, with its 13 highly nonlinear constraint equations and 23 generalized coordinates, would be unthinkable.

## Key Formulas

$$\Phi^{\text{dist}}(P_i, P_j, d) \equiv (\mathbf{d}_{ij}^T \mathbf{d}_{ij} - d^2) / 2 = 0 \quad \mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i \quad (3.3.1)$$

$$\Phi^{\text{sph}}(P_i, P_j) \equiv \mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i = \mathbf{0} \quad (3.3.9)$$

$$\Phi^{\text{dot1}}(\mathbf{a}_i, \mathbf{a}_j) = \mathbf{a}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{a}_j = 0 \quad \Phi^{\text{dot2}}(\mathbf{a}_j, \mathbf{d}_{ij}) = \mathbf{a}'_j^T \mathbf{A}_j^T \mathbf{d}_{ij} = 0 \quad (3.3.15) \quad (3.3.20)$$

### 3.4 Configuration Spaces

The multibody planar and spatial constraint formulations of Sections 3.2 and 3.3 provide a uniform means of modeling a broad spectrum of mechanical systems. The kinematic models, however, are highly nonlinear and, for multibody systems with several bodies, constraint equations must be computer generated, precluding any possibility of analytical solution. This dilemma is addressed in the following sections, using a practical implementation of mathematical concepts of differential geometry in Euclidean space  $\mathbf{R}^{\text{ngc}}$ , in which mechanical systems function. In this section, configuration space concepts are introduced using analytical models of slider-crank mechanisms.

Configurations  $\mathbf{q} \in \mathbf{R}^{\text{ngc}}$  of a mechanical system that satisfy nhc holonomic constraint equations of Eq. (3.1.22),

$$\Phi(\mathbf{q}) = \mathbf{0} \in \mathbf{R}^{\text{nnc}} \quad (3.4.1)$$

comprise the *configuration space* of the system,

$$C \equiv \{\mathbf{q} : \Phi(\mathbf{q}) = \mathbf{0}\} \subset \mathbf{R}^{\text{ngc}} \quad (3.4.2)$$

This is a subset of Euclidean space  $\mathbf{R}^{\text{ngc}}$  that defines all configurations of the system that satisfy the constraints of Eq. (3.4.1). If the constraints are independent, to be defined, the system has  $\text{DOF} = \text{ngc} - \text{nnc}$  *degrees of freedom*.

A mechanical system configuration space may contain configurations for which the constraint Jacobian fails to have full rank, called *singular configurations*, and analytical complexity occurs. Generalized coordinates in the configuration space for which the constraint Jacobian has full rank comprise the *regular configuration space*,

$$\tilde{C} = \{\mathbf{q} \in C : \text{rank}(\Phi_q(\mathbf{q})) = \text{nnc}\} = \{\mathbf{q} \in C : |\Phi_q(\mathbf{q})\Phi_q^T(\mathbf{q})| > 0\} \quad (3.4.3)$$

The condition  $\text{rank}(\Phi_q(\mathbf{q})) = \text{nnc}$  is equivalent to the inequality  $|\Phi_q(\mathbf{q})\Phi_q^T(\mathbf{q})| > 0$ , since the matrix  $\Phi_q(\mathbf{q})\Phi_q^T(\mathbf{q})$  is positive semidefinite. Thus, the complement of  $\tilde{C}$  in  $C$  is the closed set in which  $|\Phi_q(\mathbf{q})\Phi_q^T(\mathbf{q})| = 0$  and  $\tilde{C}$  is open in  $C$ . If there are no singular configurations,  $\tilde{C} = C$ .

With the induced topology defined in Section 2.2.4, the configuration space  $C$  is a topological space, so continuity of functions of generalized coordinates in  $C$  is defined. Since the function  $\Phi(\mathbf{q})$  is generally nonlinear, however,  $C$  is generally not a vector space. To see that this is the case, let  $\mathbf{q}^1$  and  $\mathbf{q}^2$  belong to  $C$ ; i.e.,  $\Phi(\mathbf{q}^1) = \mathbf{0} = \Phi(\mathbf{q}^2)$ . Due to nonlinearity of the function  $\Phi(\mathbf{q})$ , it is expected that  $\Phi(\mathbf{q}^1 + \mathbf{q}^2) \neq \Phi(\mathbf{q}^1) + \Phi(\mathbf{q}^2) = \mathbf{0}$  and  $\mathbf{q}^1 + \mathbf{q}^2$  is not in  $C$ , in which case  $C$  is not a vector space.

The subset  $C$  of  $\mathbf{R}^{\text{ngc}}$  may be comprised of a single connected set of configurations that can be smoothly traversed in a way that any pair of configurations in the set can be connected by a continuous trajectory in the set. To be more precise, consider configurations  $\mathbf{q}^1$  and  $\mathbf{q}^2$  in  $C$ ; i.e., for which  $\Phi(\mathbf{q}^1) = \mathbf{0}$  and  $\Phi(\mathbf{q}^2) = \mathbf{0}$ . The space  $C$  is *path connected* if for any

$\mathbf{q}^1$  and  $\mathbf{q}^2$  in  $C$  there is a continuous function  $\mathbf{u}(t) : \mathbb{R}^1 \rightarrow \mathbb{R}^{ngc}$  such that (1)

(1)  $\mathbf{q}^1 = \mathbf{u}(0)$  and  $\mathbf{q}^2 = \mathbf{u}(1)$  and (2)  $\mathbf{q}(t) \in C$  for all  $0 \leq t \leq 1$ . Criterion for  $C$  or  $\tilde{C}$  to be path connected require results of differential geometry that are summarized in Sections 3.5 and 3.6.

Based on Cartesian coordinate kinematics of Sections 3.2 and 3.3, four systems used in illustrations in Section 1.1 have the numbers of bodies (nb), generalized coordinates (ngc), holonomic constraints (nhc), and degrees of freedom (DOF) shown in Table 3.5.1. The dimension ngc of configuration space  $C$  of Eq. (3.4.2) for each of these systems clearly precludes drawing pictures in  $\mathbb{R}^{ngc}$  to gain insight into kinematics of even the most modest of these systems. This is motivation for the differential geometric approach employed in this and the following sections and used throughout the text.

Table 3.5.1 Dimension of Coordinates, Constraints, and Degrees of Freedom

System	Section	nb	ngc	nhc	DOF
Planar Slider-Crank	3.2.3.2	3	9	8	1
Planar Windshield Wiper	3.2.3.4	3	9	8	1
Spatial Slider-Crank	3.3.7.2	2	14	13	1
Spatial Automobile Suspension	3.3.7.3	6	23	13	10

As defined in Section 2.2.4, endowed with open sets of the form  $\Omega \cap C$ , where  $\Omega$  is open in  $\mathbb{R}^{ngc}$ ,  $C$  is a topological space. This simple construction on subset  $C$  of Euclidean space  $\mathbb{R}^{ngc}$ , using open sets that are unions of balls  $B_\epsilon(\mathbf{q}^0)$  defined in Eq. (2.2.46), enables parameterizations; i.e., changes of variable, that support analytical and computational methods in mechanical system kinematics and dynamics. The purpose of this and the following two sections is to provide a self-contained introduction to differential geometry in Euclidean configuration space  $\mathbb{R}^{ngc}$  that underpins mechanical system kinematics and dynamics. This is in contrast to mathematical literature on *differential geometry* and *differential topology* that consists of hundreds of thousands of pages focused on abstract spaces and topologies that may or may not provide practical tools to deal with mechanical system kinematics and dynamics in Euclidean space.

To identify some issues that must be dealt with in this development, planar slider-crank models are studied in this section, with two generalized coordinates that enable drawings in the plane  $\mathbb{R}^{ngc} = \mathbb{R}^2$ .

**Example 3.4.1** The two-body slider-crank model of Fig. 3.4.1 is comprised of a crank of unit radius and a connecting rod of length 2 units, modeled as a distance constraint between the crank and slider. With generalized coordinates  $\mathbf{q} = [q_1 \quad q_2]^T$  shown, the single constraint equation is

$$\Phi(\mathbf{q}) = ((q_2 - \cos q_1)^2 + \sin^2 q_1 - 4)/2 = 0 \quad (3.4.4)$$

with Jacobian

$$\Phi_q(\mathbf{q}) = [q_2 \sin q_1 \quad q_2 - \cos q_1] \quad (3.4.5)$$

The configuration space is thus

$$C = \{q \in R^2 : \Phi(q) = 0\} \quad (3.4.6)$$

Since the Jacobian of Eq. (3.4.5) is a row vector, it has full row rank if and only if  $\Phi_q(q) \neq \mathbf{0}$ ; i.e., the regular configuration space is

$$\tilde{C} = \{q \in C : \Phi_q(q) = [q_2 \sin q_1 \quad q_2 - \cos q_1] \neq [0 \quad 0]\} \quad (3.4.7)$$

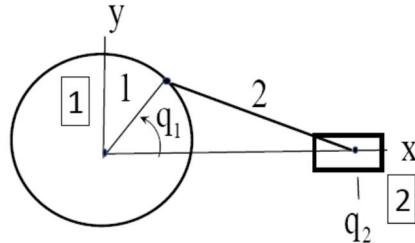


Figure 3.4.1 Slider-Crank with Two Unit Connecting Rod

The only possibility that  $[q_2 \sin q_1 \quad q_2 - \cos q_1] = [0 \quad 0]$  in  $C$  is  $q_2 - \cos q_1 = 0$  and  $q_2 \sin q_1 = 0$ . Using the first condition in Eq. (3.4.4),  $\sin^2 q_1 = 4$  which is not possible. Thus, the mechanism has no singular configurations and  $\tilde{C} = C$ .

From Eq. (3.4.4),  $q_2 = \cos q_1 \pm \sqrt{4 - (\sin q_1)^2}$ . These two solutions define two disjoint subsets of  $\tilde{C}$ ,

$$\begin{aligned} \tilde{C}_{\text{plus}} &= \left\{ q : q_2 = \cos q_1 + \sqrt{4 - (\sin q_1)^2} \right\} \\ \tilde{C}_{\text{minus}} &= \left\{ q : q_2 = \cos q_1 - \sqrt{4 - (\sin q_1)^2} \right\} \end{aligned} \quad (3.4.8)$$

shown in Fig. 3.4.2 on the interval  $0 \leq q_1 \leq 2\pi$ , whose union is  $\tilde{C}$ . Since  $\text{ngc} = 2$ , these subsets of  $\tilde{C}$  are one-dimensional curves that can be drawn in  $R^2$ . Since the sets are continued periodically on the  $q_1$ -axis, neither set has a boundary. This is important, since the literature on mechanism singularities suggests that characteristics of subsets of  $\tilde{C}$  are determined by singular configurations. Even though there are no singular configurations in this example, there are multiple disjoint connected subsets of  $\tilde{C}$ , associated with two distinct assembled configurations, the plus sign with  $q_2 > 0$  and the minus sign with  $q_2 < 0$ . Clearly, there is no continuous nonsingular trajectory in configuration space between these disjoint sets. In both subsets, the crank can be rotated throughout the range  $0 \leq q_1 \leq 2\pi$ . This simple example shows that reliance on singular configurations to determine characteristics of the configuration space is fatally flawed.

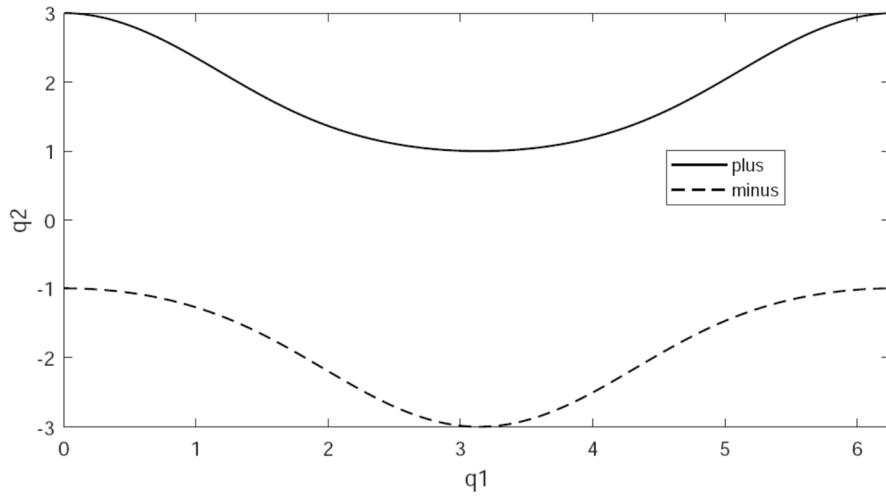


Figure 3.4.2 Disjoint Subsets of  $\tilde{C}$ , Slider-Crank with Two Unit Length Connecting Rod

**Example 3.4.2** The two-body slider-crank model of Fig. 3.4.3 is comprised of a crank of unit radius and a connecting rod of length  $1/2$  unit, modeled as a distance constraint between the crank and slider. The constraint equation is

$$\Phi(\mathbf{q}) = ((q_2 - \cos q_1)^2 + \sin^2 q_1 - 1/4)/2 = 0 \quad (3.4.9)$$

The Jacobian is the same as in Eq. (3.4.5) and the regular configuration space is as in Eq. (3.4.7).

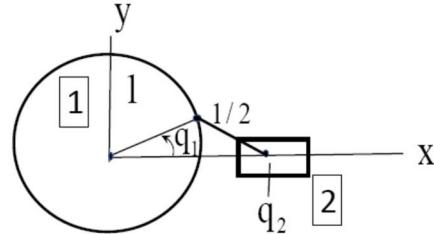


Figure 3.4.3 Slider-Crank with  $1/2$  Unit Connecting Rod

As in Example 3.4.1, the only possibility in which the Jacobian is zero is  $q_2 - \cos q_1 = 0$  and  $q_2 \sin q_1 = 0$ . Since the geometry of the mechanism implies  $q_2 \geq 1/2$  or  $q_2 \leq -1/2$ ,  $\sin q_1 = 0$  is required for a singular configuration; i.e.,  $q_1 = 0$  or  $\pi$ . In these configurations, however,  $q_2 = \pm 1.5$  and  $q_2 - \cos q_1 \neq 0$ . Thus, there are no singular configurations for this mechanism in  $C$ , so  $\tilde{C} = C$ . Solving Eq. (3.4.9),  $q_2 = \cos q_1 \pm \sqrt{1/4 - (\sin q_1)^2}$ , so  $-1/2 \leq \sin q_1 \leq 1/2$  and  $-\text{Arcsin}(1/2) \leq q_1 \leq \text{Arcsin}(1/2)$  or  $-\text{Arcsin}(1/2) \leq q_1 \leq +\text{Arcsin}(1/2)$ .

The regular configuration space in this case is comprised of two disjoint subsets,

$$\begin{aligned}\tilde{C}_R &= \left\{ \mathbf{q} \in C : q_2 = \cos q_1 \pm \sqrt{1/4 - (\sin q_1)^2}, -\text{Arcsin}(1/2) \leq q_1 \leq \text{Arcsin}(1/2) \right\} \\ \tilde{C}_L &= \left\{ \mathbf{q} \in C : q_2 = \cos q_1 \pm \sqrt{1/4 - (\sin q_1)^2}, -\pi - \text{Arcsin}(1/2) \leq q_1 \leq -\pi + \text{Arcsin}(1/2) \right\}\end{aligned}\quad (3.4.10)$$

that are shown as closed curves in Fig. 3.4.4. The curve at the upper right is  $\tilde{C}_R$  and that to the lower left is  $\tilde{C}_L$ . While these sets have very different characteristics than those in Fig. 3.4.2, they are continuous and have no boundaries. Clearly, there is no continuous nonsingular path in configuration space between configurations in the two sets. Again, mechanism singularity theory sheds no light on characteristics of these sets. Note that, unlike the slider-crank mechanism of Example 3.4.1, the crank cannot be rotated through a full revolution.

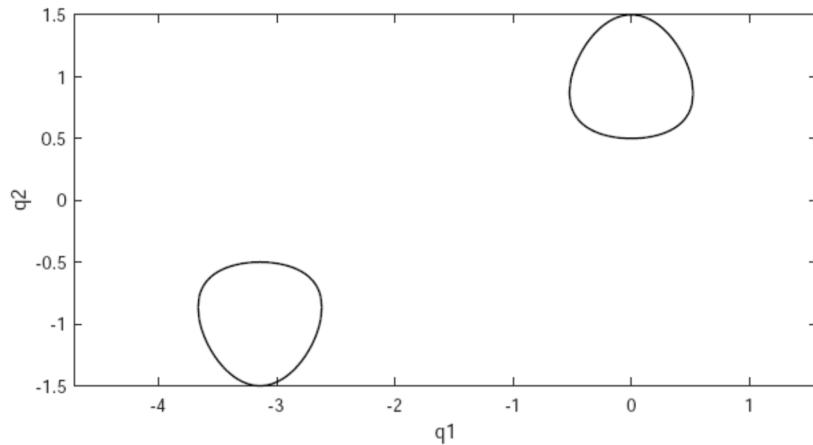


Figure 3.4.4 Disjoint Subsets of  $\tilde{C}$ , Slider-Crank with 1/2 Unit Length Connecting Rod

**Example 3.4.3** The two-body slider-crank mechanism model of Fig. 3.4.5 is comprised of a crank of unit radius and a connecting rod of unit length, modeled as a distance constraint between the crank and slider. With the generalized coordinates  $\mathbf{q} = [q_1 \ q_2]^T$  shown, the single constraint equation is

$$\Phi(\mathbf{q}) = ((q_2 - \cos q_1)^2 + \sin^2 q_1 - 1)/2 = q_2(q_2 - 2\cos q_1)/2 = 0 \quad (3.4.11)$$

with Jacobian

$$\Phi_q(\mathbf{q}) = [q_2 \sin q_1 \quad q_2 - \cos q_1] \quad (3.4.12)$$

The configuration space is thus

$$C = \left\{ \mathbf{q} \in R^2 : \Phi(\mathbf{q}) = 0 \right\} = C_1 \cup C_2 \quad (3.4.13)$$

where

$$\begin{aligned} C_1 &= \left\{ \mathbf{q} \in \mathbb{R}^2 : q_2 = 2 \cos q_1, -\pi/2 \leq q_1 \leq 3\pi/2 \right\} \\ C_2 &= \left\{ \mathbf{q} \in \mathbb{R}^2 : q_2 = 0, -\pi/2 \leq q_1 \leq 3\pi/2 \right\} \end{aligned} \quad (3.4.14)$$

The interval on  $q_1$  has been selected to avoid problems of periodic wrap-around of variables.

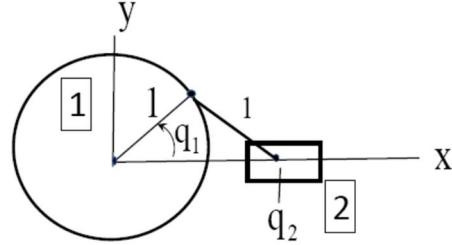


Figure 3.4.5 Slider-Crank with Unit Length Connecting Rod

To obtain  $\tilde{C}$ , subsets of  $C_1$  and  $C_2$  must be found for which  $\Phi_q(\mathbf{q}) \neq \mathbf{0}$ ; i.e.,  $q_2 \sin q_1 \neq 0$  or  $q_2 - \cos q_1 \neq 0$ . In  $C_1$ , since  $q_2 - \cos q_1 = \cos q_1$ ,  $q_2 - \cos q_1 \neq 0$  if  $q_1 \neq -\pi/2, \pi/2$ , or  $3\pi/2$ . Thus, from  $C_1$ ,

$$\begin{aligned} \tilde{C}_1 &= \left\{ \mathbf{q} \in \mathbb{R}^2 : q_2 = 2 \cos q_1, -\pi/2 < q_1 < \pi/2 \right\} \\ \tilde{C}_2 &= \left\{ \mathbf{q} \in \mathbb{R}^2 : q_2 = 2 \cos q_1, \pi/2 < q_1 < 3\pi/2 \right\} \end{aligned} \quad (3.4.15)$$

From  $C_2$ , since  $q_2 = 0, -\cos q_1 \neq 0$  is required, so  $q_1 \neq -\pi/2, \pi/2$ , or  $3\pi/2$  and

$$\begin{aligned} \tilde{C}_3 &= \left\{ \mathbf{q} \in \mathbb{R}^2 : q_2 = 0, -\pi/2 < q_1 < \pi/2 \right\} \\ \tilde{C}_4 &= \left\{ \mathbf{q} \in \mathbb{R}^2 : q_2 = 0, \pi/2 < q_1 < 3\pi/2 \right\} \end{aligned} \quad (3.4.16)$$

Configurations of the mechanism in the four disjoint subsets of  $\tilde{C}$  are shown in Fig. 3.4.6. Plotted in the  $\mathbf{q}$  plane, the  $\tilde{C}_i$  are the four one dimensional subsets of  $\mathbb{R}^2$  shown in Fig. 3.4.7. Note that they have boundaries at singular configurations of the mechanism, at which the configuration may be discontinuously transformed from one of the configurations shown in Fig. 3.4.6 to another. In this example, mechanism singularity theory yields useful information. Within each of the four open subsets  $\tilde{C}_i$  of  $\tilde{C}$ , motion is continuous and any configuration in a  $\tilde{C}_i$  can be transformed to any other configuration in the same set via a singularity free path. For configurations in  $\tilde{C}_i$  and  $\tilde{C}_j$ ,  $i \neq j$ , however, there is no singularity free path. These properties are shown in the next two sections to be universally valid.

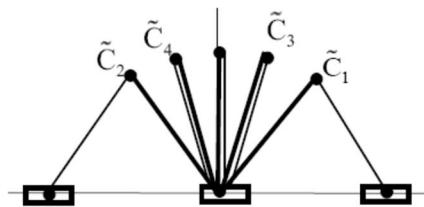


Figure 3.4.6 Configurations  $\tilde{C}_i$ , Slider-Crank with Unit Length Connecting Rod

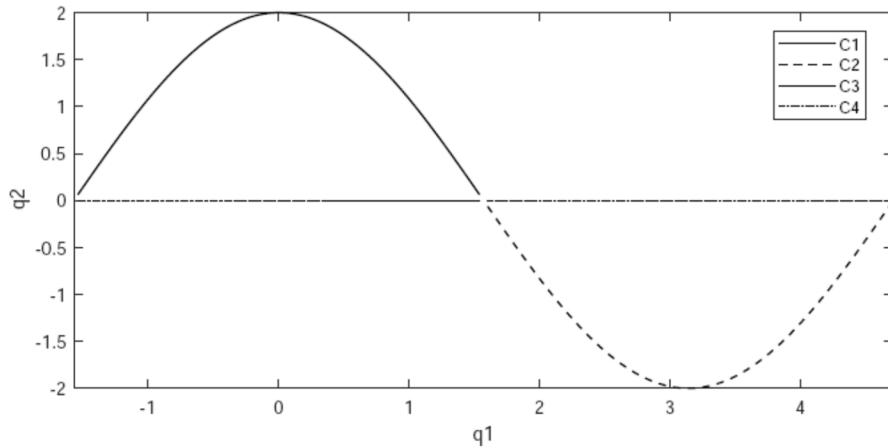


Figure 3.4.7 Disjoint Subsets of  $\tilde{C}$ , Tw Slider-Crank with Unit Length Connecting Rod

---

While analytical manipulations in the relatively simple Examples 3.4.1 through 3.4.3 shed light on the nature of singular behavior of mechanisms, in more complex applications a deeper mathematical theory and computational methods are needed.

Three planar slider-crank models yield analytical definition of their regular configuration spaces. More general mechanisms, especially in spatial systems, are highly nonlinear and kinematic constraint equations must be computer generated, precluding analytical treatment. This dilemma is addressed in the following sections, using a practical implementation of mathematical concepts of differential geometry in the Euclidean space  $R^{ngc}$ , in which mechanical systems function.

### 3.5 Constraint Manifolds

Rather than giving an *abstract definition of manifolds*, as is done in the mathematical literature (Spivak, 1999; Lee, 2013; Dundas, 2018), it is instructive to introduce the concept using properties of nonlinear equations of kinematics derived in Sections 3.1 through 3.4. A family of *parameterizations* of subsets of the regular configuration space  $\tilde{C}$  of Eq. (3.4.3) is defined by invertible mappings from  $\mathbb{R}^{n_{\text{gc}}-n_{\text{hc}}}$  to  $\tilde{C}$  that locally represent  $\tilde{C}$ . Such parameterizations are first introduced, using Example 1.2.1 of Section 1.2.2 for a particle that is constrained to move on the unit sphere. The concept is extended to a broad spectrum of mechanical systems in the following section.

#### 3.5.1 Parameterization of a Particle on a Unit Sphere

Consider the *particle on a unit sphere* shown in Fig. 3.5.1 that was introduced in Example 1.2.1 of Section 1.2.2. Vectors  $\mathbf{q}$  to points on the unit sphere satisfy the *holonomic kinematic constraint*,

$$(\mathbf{q}) = (\mathbf{q}^T \mathbf{q} - 1) / 2 = 0 \quad (3.5.1)$$

Points on the unit sphere in  $\mathbb{R}^3$  define the *configuration space* of Eq. (3.4.2),

$$C = \{\mathbf{q} \in \mathbb{R}^3 : \mathbf{q}^T \mathbf{q} = 1\} \quad (3.5.2)$$

Since  $\Phi_q(\mathbf{q}) = \mathbf{q}^T \neq \mathbf{0}$  for  $\mathbf{q} \in C$ , the constraint Jacobian has full rank throughout  $C$  and the *regular configuration space* of Eq. (3.4.3) is

$$\tilde{C} = \{\mathbf{q} \in C : q(\mathbf{q}) = \mathbf{q}^T \neq \mathbf{0}\} = C \quad (3.5.3)$$

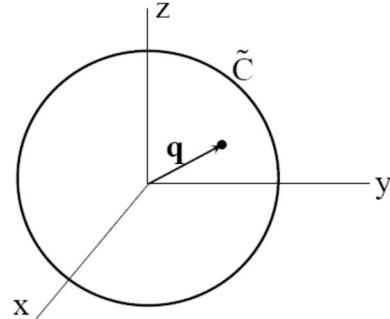


Figure 3.5.1 Particle on Unit Sphere

At a point  $\mathbf{q}^0$  on the unit sphere, the vector  $\mathbf{q}^0$  from the origin of the x-y-z frame is orthogonal to the surface of the sphere and vectors  $\mathbf{w}$  tangent to the surface of the sphere satisfy  $\Phi_q(\mathbf{q}^0)\mathbf{w} = \mathbf{q}^{0T}\mathbf{w} = 0$ . Since  $\mathbf{q}^0 \neq \mathbf{0}$  cannot coincide with all three of the unit coordinate vectors  $\mathbf{u}_x$ ,  $\mathbf{u}_y$ , and  $\mathbf{u}_z$ , say  $\mathbf{q}^0 \neq \alpha \mathbf{u}_x$ , define the unit vector  $\mathbf{V}^1 = (1/\|\tilde{\mathbf{u}}_x \mathbf{q}^0\|)\tilde{\mathbf{u}}_x \mathbf{q}^0$  that is orthogonal to  $\mathbf{u}_x$  and  $\mathbf{q}^0$ . Next, define the unit vector  $\mathbf{V}^2 = (1/\|\tilde{\mathbf{q}}^0 \mathbf{V}^1\|)\tilde{\mathbf{q}}^0 \mathbf{V}^1$  that is orthogonal to  $\mathbf{q}^0$  and  $\mathbf{V}^1$ . Since the unit vectors  $\mathbf{V}^1$  and  $\mathbf{V}^2$  are orthogonal and both are orthogonal to  $\mathbf{q}^0$ , they form a basis for the *tangent plane*, or *tangent space*, to the sphere at point  $\mathbf{q}^0$ . Define

$$\mathbf{U} = \begin{smallmatrix} {}^T \\ \mathbf{q} \end{smallmatrix} (\mathbf{q}^0) = \mathbf{q}^0 \quad (3.5.4)$$

and

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}^1 & \mathbf{V}^2 \end{bmatrix} \quad (3.5.5)$$

where, by construction,

$$\begin{aligned} \mathbf{V}^T \mathbf{V} &= \mathbf{I} \\ \mathbf{V}^T \mathbf{U} &= \mathbf{0} \\ \mathbf{U}^T \mathbf{U} &= 1 \end{aligned} \quad (3.5.6)$$

Since the unit vector  $\mathbf{U}$  is orthogonal to the linearly independent columns of  $\mathbf{V}$ , the columns of  $\mathbf{V}$  and  $\mathbf{U}$  form a basis for  $\mathbb{R}^3$ .

Since Euclidean space  $\mathbb{R}^3$  is a vector space of dimension 3, any vector  $\mathbf{a} \in \mathbb{R}^3$  can be uniquely represented as  $\mathbf{a} = \mathbf{Vv} - \mathbf{Uu}$ . Multiplying both sides of this equation by  $\mathbf{V}^T$  and  $\mathbf{U}^T$ ,  $\mathbf{v} = \mathbf{V}^T \mathbf{a}$  and  $\mathbf{u} = -\mathbf{U}^T \mathbf{a}$ , so  $\mathbf{a} = \mathbf{V}\mathbf{V}^T \mathbf{a} + \mathbf{U}\mathbf{U}^T \mathbf{a}$ . Thus, for all  $\mathbf{a} \in \mathbb{R}^3$ ,  $(\mathbf{I} - \mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T)\mathbf{a} = \mathbf{0}$  and  $\mathbf{U}$  and  $\mathbf{V}$  are related by the condition

$$\mathbf{I} = \mathbf{V}\mathbf{V}^T + \mathbf{U}\mathbf{U}^T \quad (3.5.7)$$

Every vector  $\mathbf{q}$  in  $\mathbb{R}^3$  can be uniquely written in terms of *tangent space generalized coordinates*  $\mathbf{v} \in \mathbb{R}^2$  and  $\mathbf{u} \in \mathbb{R}^1$ , defined as

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{u}\mathbf{U} \quad (3.5.8)$$

where the sign of the third term on the right is selected to represent a projection onto the regular configuration space from the tangent space, as shown schematically in Fig. 3.5.2. It is important to note that this representation of configurations is possible because Euclidean space  $\mathbb{R}^3$  is a vector space of dimension 3.

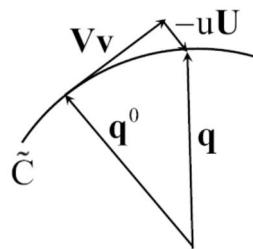


Figure 3.5.2 Orthogonal Projection onto Regular Configuration Space

Multiplying Eq. (3.5.8) on the left by  $\mathbf{V}^T$  and  $\mathbf{U}^T$  and using Eqs. (3.5.6) yields the inverse relations

$$\begin{aligned} \mathbf{v} &= \mathbf{V}^T(\mathbf{q} - \mathbf{q}^0) \\ \mathbf{u} &= -\mathbf{U}^T(\mathbf{q} - \mathbf{q}^0) \end{aligned} \quad (3.5.9)$$

In particular, Eq. (3.5.9) implies that for a time dependent parameterization,  $\mathbf{v}(t^0) = \mathbf{v}^0 = \mathbf{0}$  and  $\mathbf{u}(t^0) = \mathbf{u}^0 = 0$ .

Requiring the configuration  $\mathbf{q}$  of Eq. (3.5.8) to satisfy the constraint of Eq. (3.5.1) yields

$$(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{u}\mathbf{U}) = (u^2 - 2u + \mathbf{v}^T\mathbf{v}) / 2 = 0 \quad (3.5.10)$$

Using the quadratic formula,  $u = 1 \pm \sqrt{1 - \mathbf{v}^T\mathbf{v}}$ . Since  $u = 0$  if  $\mathbf{v} = \mathbf{0}$ , the minus sign must be selected, so

$$u = 1 - \sqrt{1 - \mathbf{v}^T\mathbf{v}} \equiv h(\mathbf{v}) \quad (3.5.11)$$

and Eq. (3.5.8) is

$$\mathbf{q} \equiv \psi(\mathbf{v}) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \left(1 - \sqrt{1 - \mathbf{v}^T\mathbf{v}}\right)\mathbf{q}^0 = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}h(\mathbf{v}) \quad (3.5.12)$$

Clearly, this parameterization is only valid if  $\mathbf{v}^T\mathbf{v} < 1$ . Geometrically, this means that if  $\mathbf{v}^T\mathbf{v} > 1$ , the vector perpendicular to the tangent space through the point  $\mathbf{q}^0 + \mathbf{V}\mathbf{v}$  in Fig. 3.5.4 does not intersect the unit sphere in  $\mathbb{R}^3$ .

Substituting  $\mathbf{q}$  of Eq. (3.5.12) into Eq. (3.5.1),

$$(\mathbf{q}) = (\mathbf{q}^T\mathbf{q} - 1) / 2 = (\mathbf{v}^T\mathbf{V}^T\mathbf{V}\mathbf{v} + (1 - \mathbf{v}^T\mathbf{v})\mathbf{q}^{0T}\mathbf{q}^0 - 1) / 2 = (\mathbf{v}^T\mathbf{v} - \mathbf{v}^T\mathbf{v}) / 2 = 0$$

which verifies that the holonomic constraint of Eq. (3.5.1) is satisfied by the parameterization for arbitrary  $\mathbf{v}$  in a neighborhood  $\mathbf{V}^0$  of  $\mathbf{v}^0 = \mathbf{0}$ . The representation of Eq. (3.5.12) is thus a *local parameterization* of the regular configuration space  $\tilde{\mathcal{C}}$  of Eq. (3.5.3), with  $\mathbf{v} \in \mathbb{R}^2$  as independent coordinates.

To see that the mapping  $\psi$  is locally one-to-one, set  $\psi(\mathbf{v}^1) = \psi(\mathbf{v}^2)$  for  $\mathbf{v}^1$  and  $\mathbf{v}^2$  in  $\mathbf{V}^0$ ,

$$\psi(\mathbf{v}^1) = \mathbf{q}^0 + \mathbf{V}\mathbf{v}^1 - \mathbf{U}h(\mathbf{v}^1) = \mathbf{q}^0 + \mathbf{V}\mathbf{v}^2 - \mathbf{U}h(\mathbf{v}^2) = \psi(\mathbf{v}^2)$$

Multiplying the equation on the left by  $\mathbf{V}^T$  and using Eqs. (3.5.6),  $\mathbf{v}^1 = \mathbf{v}^2$ . Thus, the mapping is *locally one-to-one*; i.e., one-to-one in the neighborhood  $\mathbf{V}^0$ .

Multiplying both sides of Eq. (3.5.12) by  $\mathbf{V}^T$  yields

$$\mathbf{v} = \mathbf{V}^T(\mathbf{q} - \mathbf{q}^0) = \mathbf{V}^T\mathbf{q} \equiv \phi(\mathbf{q}) \quad (3.5.13)$$

where Eqs. (3.5.6) have been used. From Eqs. (3.5.12) and (3.5.13),

$$\phi(\psi(\mathbf{v})) = \mathbf{V}^T\left(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\left(1 - \sqrt{1 - \mathbf{v}^T\mathbf{v}}\right)\mathbf{q}^0\right) = \mathbf{v}, \text{ for all } \mathbf{v} \in \mathbf{V}^0; \text{ i.e. } \phi \text{ is the inverse of } \psi,$$

$$\phi^{-1} = \psi \quad (3.5.14)$$

The mappings  $\psi$  and  $\phi$  are continuously differentiable in the induced topology; i.e., the metric topology on  $\mathbb{R}^3$ , so they are *diffeomorphisms* (Lee, 2010; Schlichtkrull, 2013) that define  $\tilde{\mathcal{C}}$  as approximately Euclidean in a neighborhood of  $\mathbf{v} = \mathbf{0}$ . An important property of these mappings is that  $\psi : \mathbb{R}^2 \rightarrow \tilde{\mathcal{C}} \subset \mathbb{R}^3$  maps open sets of  $\mathbb{R}^2$  onto open sets of  $\tilde{\mathcal{C}}$ , which are the intersection of open sets in  $\mathbb{R}^3$  with  $\tilde{\mathcal{C}}$ , so  $\tilde{\mathcal{C}}$  is an open set of dimension 2. This defines the number of *degrees of freedom* of the system in  $\psi(\mathbf{V}^0)$ .

To more explicitly verify that  $\mathbf{q}$  of Eq. (3.5.12), for  $\mathbf{v}$  of Eq. (3.5.13), satisfies Eq. (3.5.12), using Eq. (3.5.13),

$$\mathbf{v}^T \mathbf{v} = (\mathbf{q}^T \mathbf{V}) \mathbf{V}^T \mathbf{q} = \mathbf{q}^T (\mathbf{V} \mathbf{V}^T) \mathbf{q} = \mathbf{q}^T (\mathbf{I} - \mathbf{U} \mathbf{U}^T) \mathbf{q} = \mathbf{q}^T \mathbf{q} - (\mathbf{q}^{0T} \mathbf{q})^2 = 1 - (\mathbf{q}^{0T} \mathbf{q})^2$$

Substituting this result into the right side of Eq. (3.5.12) and using Eq. (3.5.7) yields the desired result,  $\psi(\phi(\mathbf{q})) = \mathbf{q}^0 + \mathbf{V} \mathbf{V}^T \mathbf{q} - \mathbf{q}^0 \left( 1 - \sqrt{1 - 1 + (\mathbf{q}^{0T} \mathbf{q})^2} \right) = (\mathbf{V} \mathbf{V}^T + \mathbf{U} \mathbf{U}^T) \mathbf{q} = \mathbf{q}$ . Thus,

$$\Psi^{-1} = \phi \quad (3.5.15)$$

With domain  $U^0 = \Psi(V^0)$ ,  $\phi$  is a differentiable map from an open set  $U^0 \subset \tilde{C}$  onto an open set  $V^0 \subset R^2$ , with continuous inverse  $\Psi$ . In the language of *differential geometry* (Spivak, 1999), or *differential topology* (Lee, 2010; Schlichtkrull, 2013), the pair  $(\phi, U^0)$  is a *chart* on  $\tilde{C}$ .

With a family of configurations  $\mathbf{q}^i \in \tilde{C}$ , the associated neighborhoods  $U^i = \Psi(V^i) \subset \tilde{C}$  define a family of *charts*  $(\phi^i, U^i)$  on  $\tilde{C}$ . If enough configurations  $\mathbf{q}^i$  and associated charts are defined so that  $\tilde{C} = \cup_i U^i$ , the family of charts is called an *atlas* for  $\tilde{C}$ . This terminology is motivated by a family of maps of the surface of the earth that are representations of regions  $U^i$  on the surface of the earth such that the maps agree on overlapping regions. These maps are then bound into a book (atlas) that represents a large region; e.g., a country.

The mapping of Eq. (3.5.12) is a *parameterization* of the nonlinear regular configuration space  $\tilde{C}$  in terms of the variable  $\mathbf{v}$  on open subsets  $V^0$  of  $R^2$ . This provides a local mathematical representation of kinematics of the system on the linear space of real variables  $R^2$ . The essential nonlinearities of the kinematic system are thus embedded in the function  $\Psi(\mathbf{v})$ . It is important to note that  $\tilde{C}$  cannot be parameterized by a single mapping from  $R^2$ . It is only fully represented if enough charts  $(\phi_i, U^i)$  are defined such that  $\phi_i(\mathbf{v})$  are local parameterizations of  $\tilde{C}$  on  $U^i$  and  $\tilde{C} = \cup_i U^i$ . The reason multiple charts are required for the unit sphere is explained later, when a bit more mathematical machinery is available.

Since the transformation of Eq. (3.5.12) is valid for  $\mathbf{v}^T \mathbf{v} < 1$ , with the restriction  $\mathbf{v}^T \mathbf{v} \leq 0.8$ , the chart domains  $U^i$  comprise approximately 80% of the area of the hemisphere that has  $\mathbf{q}^0$  as pole. Therefore, the union of a small number of such charts covers the unit sphere.

While all the machinery used in the foregoing is not required to analyze the kinematics of a particle on the unit sphere in  $R^3$ , it is presented as an introduction to concepts of *differential geometry* that are required for analysis of the kinematics and dynamics of a broad spectrum of mechanical systems in Euclidean space  $R^{ngc}$ ,  $ngc > 3$ , where drawings are not possible. Fortunately, a rich body of theory of differential geometry that has potential to positively impact the theory and practice of kinematics and dynamics of mechanical systems has emerged in the mathematical literature. Unfortunately, much of the mathematical literature is clothed in *abstract mathematical garb* that makes it difficult to access by we mere mortal engineers. For this reason, a subset of this theory that is applicable for kinematics and dynamics of mechanical systems in

Euclidean space  $\mathbf{R}^{\text{ngc}}$  that is developed in (Schlichtkrull, 2013) is summarized in Section 2.2.4 and used in this and the following sections.

As an illustration of this theory, no chart can parameterize the entire surface of the unit sphere. The reason is that the unit sphere is closed and bounded in the topology induced by  $\mathbf{R}^3$ , hence *compact*. Any single homeomorphism that would map it onto a set in  $\mathbf{R}^2$  would require that the image would be compact (Mendelson, 1962). However, the image of a chart in  $\mathbf{R}^2$  is open, yielding a contradiction. Thus, no single chart can parameterize the entire unit sphere. This is the mathematical reason for the dilemma encountered in Example 1.2.1 and is analogous to the result of (Stuelpnagel, 1964), cited in Section 2.4.3, that there exists no singularity free set of three orientation generalized coordinates for a body in space. It is these theoretical results that require use of *local parameterizations*; i.e., *charts*, to characterize the regular configuration space. Unfortunately, this reality has not yet permeated the literature of mechanical system kinematics and dynamics.

### 3.5.2 Tangent Space Parameterization of the Regular Configuration Space

For a general mechanical system with generalized coordinates  $\mathbf{q} \in \mathbf{R}^{\text{ngc}}$  and  $\text{nhc}$  holonomic kinematic constraints,

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (3.5.16)$$

the configuration space is

$$C = \left\{ \mathbf{q} \in \mathbf{R}^{\text{ngc}} : \Phi(\mathbf{q}) = \mathbf{0} \right\} \quad (3.5.17)$$

and the regular configuration space is

$$\begin{aligned} \tilde{C} &= \left\{ \mathbf{q} \in C : \text{rank}(\Phi_q(\mathbf{q})) = \text{nhc} \right\} \\ &= \left\{ \mathbf{q} \in \mathbf{R}^{\text{ngc}} : \Phi(\mathbf{q}) = \mathbf{0} \text{ and } \text{rank}(\Phi_q(\mathbf{q})) = \text{nhc} \right\} \end{aligned} \quad (3.5.18)$$

The geometry of the sphere in Section 3.5.1 may be generalized, by observing that kinematically admissible velocities  $\dot{\mathbf{q}}$  of Eq. (3.1.23); i.e.,

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (3.5.19)$$

at  $\mathbf{q} \in \tilde{C}$  are by definition tangent to the *regular configuration space* at  $\mathbf{q}$ . Geometrically, this says that the *tangent space* of the regular configuration space at  $\mathbf{q}$  is comprised of vectors in the *null space*  $N_q = \left\{ \mathbf{y} \in \mathbf{R}^n : \Phi_q(\mathbf{q})\mathbf{y} = \mathbf{0} \right\}$  of the *constraint Jacobian*  $\Phi_q(\mathbf{q})$  at configuration  $\mathbf{q}$ .

Hence, vectors that are orthogonal to the tangent space at  $\mathbf{q} \in \tilde{C}$  are linear combinations of columns of the transpose  $\Phi_q^T(\mathbf{q})$  of the constraint Jacobian  $\Phi_q(\mathbf{q})$ . Note that this requires that the Jacobian has full rank, as is embedded in the definition of  $\tilde{C}$ .

The constraint Jacobian  $\Phi_q(\mathbf{q})$  at a configuration  $\mathbf{q}^0 \in \tilde{C}$  defines an  $\text{ngc} \times \text{nhc}$  matrix  $\mathbf{U}$  whose column vectors are orthogonal to the tangent space of  $\tilde{C}$  at  $\mathbf{q}^0$ ; i.e.,

$$\mathbf{U} \equiv \Phi_q^{0T} = \Phi_q^T(\mathbf{q}^0) \quad (3.5.20)$$

For  $\mathbf{q}^0 \in \tilde{\mathcal{C}}$ , the constraint Jacobian has full row rank, so the columns of  $\mathbf{U}$  are *linearly independent* and  $\mathbf{U}^T \mathbf{U}$  is positive definite, hence nonsingular. An  $n_{\text{gc}} \times (n_{\text{gc}} - n_{\text{hc}})$  matrix  $\mathbf{V}$  in the null space of  $\Phi_q(\mathbf{q}^0)$  is defined as the solution of

$$\begin{aligned}\Phi_q^0 \mathbf{V} &= \mathbf{U}^T \mathbf{V} = \mathbf{0} \\ \mathbf{V}^T \mathbf{V} &= \mathbf{I}\end{aligned}\tag{3.5.21}$$

using *singular value decomposition* (Strang, 1980). The second of Eqs. (3.5.21) implies that  $\mathbf{V}$  has full rank. Since its columns are orthogonal to the columns of  $\mathbf{U}$ , the columns of  $\mathbf{U}$  and  $\mathbf{V}$  span the  $n_{\text{gc}}$ -dimensional Euclidean vector space  $\mathbb{R}^{n_{\text{gc}}}$ .

Matrices  $\mathbf{U}$  and  $\mathbf{V}$  are further related. Any vector  $\mathbf{a} \in \mathbb{R}^{n_{\text{gc}}}$  is uniquely represented as

$$\mathbf{a} = \mathbf{Vv} - \mathbf{Uu}\tag{3.5.22}$$

where  $\mathbf{v} \in \mathbb{R}^{n_{\text{gc}}-n_{\text{hc}}}$  and  $\mathbf{u} \in \mathbb{R}^{n_{\text{hc}}}$ . Multiplying on the left by  $\mathbf{V}^T$  and using the second of Eqs. (3.5.21),  $\mathbf{v} = \mathbf{V}^T \mathbf{a}$ . Similarly, multiplying on the left by  $\mathbf{U}^T$  and using the first of Eqs. (3.5.21),  $\mathbf{u} = -(\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{a}$ . Substituting these relations into Eq. (3.5.22),  $\mathbf{a} = \mathbf{V} \mathbf{V}^T \mathbf{a} + \mathbf{U} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{a}$ , or  $(\mathbf{V} \mathbf{V}^T + \mathbf{U} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T - \mathbf{I}) \mathbf{a} = \mathbf{0}$ , for arbitrary  $\mathbf{a}$ . Thus,

$$\mathbf{V} \mathbf{V}^T + \mathbf{U} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T = \mathbf{I}\tag{3.5.23}$$

A *tangent space parameterization* of the regular configuration space  $\tilde{\mathcal{C}}$ , in a neighborhood  $\mathbf{U}^0$  of  $\mathbf{q}^0 = \mathbf{q}(t^0)$ , is defined as

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{Vv} - \mathbf{Uu}\tag{3.5.24}$$

The variables  $\mathbf{v}$  and  $\mathbf{u}$  comprise a set of *local generalized coordinates* that are in one-to-one correspondence with  $\mathbf{q} \in \mathbf{U}^0$ . To see this, multiply Eq. (3.5.24) by  $\mathbf{V}^T$  and  $\mathbf{U}^T$  and use Eqs. (3.5.21) to obtain unique values of  $\mathbf{v}$  and  $\mathbf{u}$ ,

$$\begin{aligned}\mathbf{v} &= \mathbf{V}^T (\mathbf{q} - \mathbf{q}^0) \\ \mathbf{u} &= -(\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T (\mathbf{q} - \mathbf{q}^0)\end{aligned}\tag{3.5.25}$$

There is thus a one-to-one correspondence between  $\mathbf{q}$  and  $(\mathbf{u}, \mathbf{v})$  in a neighborhood of  $\mathbf{q}^0$ .

Further, from Eq. (3.5.25) at  $\mathbf{q}(t^0) = \mathbf{q}^0$ ,

$$\begin{aligned}\mathbf{v}^0 &= \mathbf{0} \\ \mathbf{u}^0 &= \mathbf{0}\end{aligned}\tag{3.5.26}$$

To assure that  $\mathbf{q}$  of Eq. (3.5.24) satisfies the constraints of Eq. (3.5.16), it is required that

$$\Phi(\mathbf{q}^0 + \mathbf{Vv} - \mathbf{Uu}) = \mathbf{0}\tag{3.5.27}$$

The derivative of the left side of this equation with respect to  $\mathbf{u}$ , with  $\mathbf{v}$  held constant, is

$$\left( \Phi(\dot{\mathbf{q}}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}) \right)_{\mathbf{u}} = -\Phi_q(\mathbf{q})\mathbf{U}. \text{ Evaluated at } \mathbf{q}^0, \text{ hence } \mathbf{v}^0 = 0 \text{ and } \mathbf{u}^0 = 0, \text{ this is}$$

$$\Phi_u(\mathbf{q}^0) = -\Phi_q(\mathbf{q}^0)\mathbf{U} = -\mathbf{U}^T\mathbf{U} \equiv -(\mathbf{B}^0)^{-1} \quad (3.5.28)$$

which is nonsingular, since the Jacobian has full rank at  $\mathbf{q}^0$ . Since  $\Phi_q(\mathbf{q})$  is a continuously differentiable function of  $\mathbf{q}$ ,  $\Phi_q(\mathbf{q})\mathbf{U} = \Phi_q(\mathbf{q})\Phi_q^T(\mathbf{q}^0)$  is nonsingular in a neighborhood  $U^0$  of  $\mathbf{q}^0$  and

$$\mathbf{B}(\mathbf{q}) \equiv (\Phi_q(\mathbf{q})\mathbf{U})^{-1} \quad (3.5.29)$$

is continuously differentiable with respect to  $\mathbf{q}$  in  $U^0$ .

Since  $\Phi_u(\mathbf{q}) = -\Phi_q(\mathbf{q})\mathbf{U} = -\mathbf{B}^{-1}(\mathbf{q})$  is nonsingular in a neighborhood of  $\mathbf{q}^0$  and  $\mathbf{u} = \mathbf{0}$  is a solution of Eq. (3.5.27) with  $\mathbf{v} = \mathbf{0}$ , the *implicit function theorem* of Section 2.2.5 guarantees that Eq. (3.5.27) has a unique continuously differentiable solution

$$\mathbf{u} = \mathbf{h}(\mathbf{v}) \quad (3.5.30)$$

in a neighborhood  $V^0$  of  $\mathbf{v} = \mathbf{0}$ . Thus, Eq. (3.5.24) becomes

$$\mathbf{q} \equiv \psi(\mathbf{v}) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}) \quad (3.5.31)$$

which satisfies Eq. (3.5.16), for all  $\mathbf{v}$  in  $V^0$  and  $\psi: U^0 \rightarrow \psi(V^0) \equiv U^0 \subset \tilde{C}$ . It is important to note that the right side of Eq. (3.5.24) is linear in  $\mathbf{v}$  and  $\mathbf{u}$ , but the right side of Eq. (3.5.31) is a nonlinear function of  $\mathbf{v}$ . This is because the nonlinear function  $\mathbf{h}(\mathbf{v})$  is defined to satisfy the nonlinear constraint equation of Eq. (3.5.16). The nonlinearity of the constraint equations is thus embedded in the function  $\mathbf{h}(\mathbf{v})$ .

To see geometrically what Eq. (3.5.31) means and why the extent of the neighborhood  $V^0$  is important, the geometry of choosing  $\mathbf{u}$  to stay in  $\tilde{C}$  of Eq. (3.5.18) is shown schematically in Fig. 3.5.3. The vector  $\mathbf{q}^0 + \mathbf{V}\mathbf{v}$  may be viewed as movement in the *tangent plane* of  $\tilde{C}$ , which must be modified by vector  $-\mathbf{U}\mathbf{u}$  to yield  $\mathbf{q}(\mathbf{v})$  on  $\tilde{C}$ . Equation (3.5.31) consolidates the vectors into the continuously differentiable function  $\mathbf{q} = \psi(\mathbf{v})$ . It is clear from Fig. 3.5.3, however, that if  $\mathbf{v}$  is large, the vector  $-\mathbf{U}\mathbf{u}$  that is perpendicular to the tangent space to  $\tilde{C}$  at  $\mathbf{q}^0$  may not intersect  $\tilde{C}$ , in which case the parameterization fails; i.e.,  $\mathbf{v}$  is no longer a valid vector of generalized coordinates. Fortunately, a large *condition number* of  $\Phi_q(\mathbf{q})\mathbf{U}$ , whose inverse is  $\mathbf{B}(\mathbf{q})$ , as defined in Section 2.2.7, is an effective warning that such a situation is eminent, in which case a new vector  $\bar{\mathbf{q}}^0$  is defined as the current value of  $\mathbf{q}$ , new bases  $\bar{\mathbf{U}}$  and  $\bar{\mathbf{V}}$  are computed, and analysis is continued with a new mapping  $\bar{\psi}(\bar{\mathbf{v}})$ .

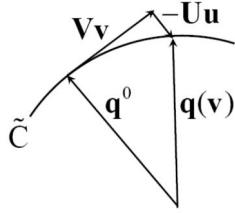


Figure 3.5.3 Projection onto  $\tilde{C}$

### 3.5.3 The Regular Configuration Space is a Differentiable Manifold

Note that Fig. 3.5.3 is drawn in a two-dimensional plane, even though the vectors involved reside in  $R^{ngc}$ . The beauty of *differential geometry* is that its analytical constructs may be viewed as abstractions of geometric concepts in  $R^2$  and  $R^3$ . It is important, however, that care be taken to be rigorous in the analytical setting, lest what may seem to be obvious in  $R^2$  or  $R^3$  does not generalize to high dimensional spaces.

To see that the differentiable mapping  $\psi: V^0 \rightarrow U^0$  of Eq. (3.5.31) is one-to-one, for  $v^1$  and  $v^2$  in  $V^0$ , set

$$\psi(v^1) = q^0 + Vv^1 - Uh(v^1) = q^0 + Vv^2 - Uh(v^2) = \psi(v^2)$$

Multiplying the equation on the left by  $V^T$  and using Eq. (3.5.21),  $V^T q^0 + v^1 = V^T q^0 + v^2$ , or  $v^1 = v^2$ , so the mapping  $\psi$  is *one-to-one* on  $V^0$ .

Multiplication of Eq. (3.5.31) on the left by  $V^T$  yields the differentiable mapping

$$v \equiv \phi(q) = V^T (q - q^0) \quad (3.5.32)$$

To see that  $\phi^{-1}$  is  $\psi$ ,

$$\phi(\psi(v)) = V^T (\psi(v) - q^0) = V^T (q^0 + Vv - Uh(v) - q^0) = v \quad (3.5.33)$$

for all  $v \in V^0$ ; i.e.,

$$\phi^{-1} = \psi \quad (3.5.34)$$

This shows that  $\psi$  is a *diffeomorphism* (Lee, 2010; Schlichtkrull, 2013) that maps open sets of  $R^{ngc-nhc}$  onto open sets of  $\tilde{C} \subset R^{ngc}$ . Thus,  $\tilde{C}$  is an  $ngc - nhc$  dimensional subset of  $R^{ngc}$  that is open in the induced topology of  $\tilde{C} \subset R^{ngc}$ ; i.e., for  $N$  open in  $R^{ngc}$ ,  $N \cap \tilde{C}$  is open in  $\tilde{C}$ .

To show that  $\psi^{-1} = \phi$ , using Eqs. (3.5.31) and (3.5.32),

$$\begin{aligned} \psi(\phi(q)) &= q^0 + VV^T (q - q^0) - Uh(V^T (q - q^0)) \\ &= (I - VV^T) q^0 + VV^T q - Uh(V^T (q - q^0)) \end{aligned} \quad (3.5.35)$$

Since  $h(V^T (q - q^0))$  is the unique differentiable solution of Eq. (3.5.27) with  $v = V^T (q - q^0)$  for  $q \in U^0$ ,

$$\begin{aligned} & \Phi\left(\mathbf{q}^0 + \mathbf{V}\mathbf{V}^T(\mathbf{q} - \mathbf{q}^0) - \mathbf{U}\mathbf{h}(\mathbf{V}^T(\mathbf{q} - \mathbf{q}^0))\right) \\ &= \Phi\left((\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{q}^0 + \mathbf{V}\mathbf{V}^T\mathbf{q} - \mathbf{U}\mathbf{h}(\mathbf{V}^T(\mathbf{q} - \mathbf{q}^0))\right) = \mathbf{0} \end{aligned} \quad (3.5.36)$$

Since  $\mathbf{q}$  of Eq. (3.5.31) satisfies  $\Phi(\mathbf{q}) = \mathbf{0}$ , a solution for  $\mathbf{h}(\mathbf{V}^T(\mathbf{q} - \mathbf{q}^0))$  is obtained by setting the argument of Eq. (3.5.36) equal to  $\mathbf{q}$ ; i.e.,

$$\mathbf{q} = (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{q}^0 + \mathbf{V}\mathbf{V}^T\mathbf{q} - \mathbf{U}\mathbf{h}(\mathbf{V}^T(\mathbf{q} - \mathbf{q}^0))$$

Multiplying this relation on the left by  $\mathbf{U}^T$  and using the relation  $\mathbf{U}^T\mathbf{V} = \mathbf{0}$ ,

$$\mathbf{h}(\mathbf{V}^T(\mathbf{q} - \mathbf{q}^0)) = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\left\{(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{q}^0 + \mathbf{V}\mathbf{V}^T\mathbf{q} - \mathbf{q}\right\} = -(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T(\mathbf{q} - \mathbf{q}^0)$$

This is the unique solution of Eq. (3.5.36). Substituting this result into Eq. (3.5.35), and using Eq. (3.5.23),

$$\psi(\phi(\mathbf{q})) = \left(\mathbf{I} - \mathbf{V}\mathbf{V}^T - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\right)\mathbf{q}^0 + \left(\mathbf{V}\mathbf{V}^T + \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\right)\mathbf{q} = \mathbf{q} \quad (3.5.37)$$

which shows that

$$\psi^{-1} = \phi \quad (3.5.38)$$

With domain  $\mathbf{U}^0 = \psi(V^0)$ ,  $\phi : \mathbf{U}^0 \rightarrow V^0$  is a differentiable map from an open set  $\mathbf{U}^0 \subset \tilde{\mathcal{C}}$  onto an open set  $V^0 \subset \mathbb{R}^{n_{hc}}$ , with differentiable inverse  $\psi$ . A countable collection of charts  $(\phi^i, U^i)$ ,  $i = 1, 2, \dots$ , such that the union of the  $U^i$  covers  $\tilde{\mathcal{C}}$ ; i.e.,  $\tilde{\mathcal{C}} = \cup_i U^i$  is called an *atlas* and provides a basis for characterizing  $\tilde{\mathcal{C}}$  as a *differentiable manifold* (Lee, 2010; Schlichtkrull, 2013). It is important to note that a set is a differentiable manifold if and only if there exist an atlas of differentiable charts that covers the set; i.e., a set without an atlas is not necessarily a manifold.

The mapping of Eq. (3.5.31) is a *parameterization* of open subsets  $U^i \subset \tilde{\mathcal{C}}$  of the *constraint manifold*, equivalently the *regular configuration space*, by the variable  $\mathbf{v}$  on open subsets  $V^i \subset \mathbb{R}^{n_{gc-nhc}}$ . This provides a mathematical representation of the kinematics of mechanical systems on the Euclidean space  $\mathbb{R}^{n_{gc-nhc}}$ , establishing the fact that the system has  $n_{gc} - n_{hc}$  *degrees of freedom* in  $V^0$ . The essential nonlinearities of the kinematic system are embedded in the function  $\psi(\mathbf{v})$ . While the extent (size) of  $\tilde{\mathcal{C}}$  is not explicitly defined, an important benefit of the fact that it is open is that once computation begins in  $\tilde{\mathcal{C}}$ , it can be continued with confidence until a boundary of  $\tilde{\mathcal{C}}$ , if one exists, is encountered.

### 3.5.4 Computation of $\mathbf{B}(\mathbf{q})$ and $\mathbf{h}(\mathbf{v})$

While the functions  $\mathbf{B}(\mathbf{q})$  and  $\mathbf{h}(\mathbf{v})$  of Eqs. (3.5.29) and (3.5.30) are shown to exist and be differentiable functions of  $\mathbf{q}$  and  $\mathbf{v}$ , the derivation does not show how they may be evaluated. Since they play a pivotal role in implementing the parameterization defined by Eqs. (3.5.31) and (3.5.32), numerical methods for their evaluation are needed. This is critical in computational

kinematics and dynamics of mechanical systems. If an abstract mathematical approach to manifold theory were employed, no insight into computation would be obtained.

At  $\mathbf{q}^0$ ,  $\mathbf{B}(\mathbf{q}^0) = (\Phi_q(\mathbf{q}^0)\mathbf{U})^{-1} = (\mathbf{U}^T\mathbf{U})^{-1}$  of Eq. (3.5.29) may be numerically evaluated.

For  $\mathbf{q}$  in a neighborhood of  $\mathbf{q}^0$ ,  $\mathbf{B}(\mathbf{q})$  must satisfy Eq. (3.5.29), written in the form

$\bar{\mathbf{R}} = (\Phi_q(\mathbf{q})\mathbf{U})\mathbf{B}(\mathbf{q}) - \mathbf{I} = \mathbf{0}$ . With an approximation  $\mathbf{B}^1 = \mathbf{B}(\mathbf{q}^0) \approx \mathbf{B}(\mathbf{q})$  of the solution and suppressing the argument  $\mathbf{q}$ , since it does not change in the iterative process for  $\mathbf{B}(\mathbf{q})$ , the *matrix version of Newton-Raphson iteration* is defined by  $(\Phi_q\mathbf{U})\Delta\mathbf{B}^1 = -\bar{\mathbf{R}}^1 = -\Phi_q\mathbf{U}\mathbf{B}^1 + \mathbf{I}$ . Since the matrix  $(\Phi_q\mathbf{U})$  need not be inverted with great precision for use in the Newton-Raphson process and  $\mathbf{B}^1 \approx (\Phi_q\mathbf{U})^{-1}$ ,  $\Delta\mathbf{B}^1 = -\mathbf{B}^1\Phi_q\mathbf{U}\mathbf{B}^1 + \mathbf{B}^1$  and  $\mathbf{B}^2 = \mathbf{B}^1 + \Delta\mathbf{B}^1 = 2\mathbf{B}^1 - \mathbf{B}^1\Phi_q\mathbf{U}\mathbf{B}^1$ . This yields the iterative algorithm

$$\mathbf{B}^{i+1} = 2\mathbf{B}^i - \mathbf{B}^i\Phi_q\mathbf{U}\mathbf{B}^i, \quad i = 1, 2, \dots, \text{until } \|\Phi_q\mathbf{U}\mathbf{B}^{i+1} - \mathbf{I}\| \leq \text{Btol} \quad (3.5.39)$$

where Btol is a specified error tolerance. This is an efficient computation, requiring only matrix multiplication.

While the function  $\mathbf{h}(\mathbf{v})$  of Eq. (3.5.30) cannot be analytically determined, it can be evaluated as accurately as desired using Newton-Raphson iteration to solve Eq. (3.5.27) for  $\mathbf{u} = \mathbf{h}(\mathbf{v})$ , with a given  $\mathbf{v}$ ; i.e.,  $\Phi_u\Delta\mathbf{u}^i = -\Phi_q\mathbf{U}\Delta\mathbf{u}^i = -\mathbf{B}^{-1}\Delta\mathbf{u}^i = -\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i)$ . The solution is  $\Delta\mathbf{u}^i = \mathbf{B}\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i)$  and  $\mathbf{u}^{i+1} = \mathbf{u}^i + \Delta\mathbf{u}^i$ . This yields the iterative algorithm

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \mathbf{B}\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i), \quad i = 1, 2, \dots, \text{until } \|\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^{i+1})\| \leq \text{utol} \quad (3.5.40)$$

where utol is a specified error tolerance. Since the Newton-Raphson method does not require an exact Jacobian, the matrix  $\mathbf{B}$  is held constant throughout the process. This is an efficient computation, requiring only matrix multiplication. With  $\mathbf{q}$  extended to  $\mathbf{q}^j$ , an associated estimate of  $\mathbf{B}^j \approx \mathbf{B}^{j-1}$  is used with Eq. (3.5.39) to compute  $\mathbf{B}^j$ , continuing the process.

These results provide practical tools for *local parameterization* of constraint manifolds for kinematics and dynamics of mechanical systems. A summary of the differential geometry of constraint manifolds is presented in Section 3.6, as the basis for extension of these locally defined tools to globally defined properties for rigorous kinematic and dynamic analysis of mechanical systems.

The system configuration space is defined as the set of all configurations that satisfy holonomic equations of constraint. It is shown that this space may contain singular configurations that can be avoided by requiring the constraint Jacobian to have full rank on a subset, called the regular configuration space.

A tangent space parameterization is defined for the regular configuration space, which is shown to be a differentiable manifold. Tangent space parameterization of the constraint manifold provides a constructive computational tool for kinematic and dynamic analysis and an introduction to the underlying theory of differentiable manifolds. Iterative numerical methods are presented for efficient construction of the tangent space parameterization.

## Key Formulas

$$C = \{q : \Phi(q) = \mathbf{0}\} \subset R^{ngc} \quad (3.5.17)$$

$$\tilde{C} = \left\{ q \in C : \text{rank}(\Phi_q(q)) = \text{nhc} \right\} \subset R^{ngc} \quad (3.5.18)$$

$$U = \Phi_q^T(q^0) \quad \Phi_q^0 V = U^T V = \mathbf{0}, V^T V = I \quad (3.5.20) \quad (3.5.21)$$

$$q = q^0 + Vv - Uh(v) \quad \Phi(q^0 + Vv - Uh(v)) = \mathbf{0} \quad u = h(v) \quad (3.5.31) \quad (3.5.27) \quad (3.5.30)$$

$$u^{i+1} = u^i + B\Phi(q^0 + Vv - Uh(v)), \quad i = 1, 2, \dots \text{ until } \|\Phi(q^0 + Vv - Uh^{i+1})\| \leq utol \quad (3.5.40)$$

$$B(q) \equiv (\Phi_q(q)U)^{-1} \quad (3.5.29)$$

$$B^{i+1} = 2B^i - B^i \Phi_q U B^i, \quad i = 1, 2, \dots, \text{ until } \|\Phi_q U B^{i+1} - I\| \leq Btol \quad (3.5.39)$$

### 3.6 Differential Geometry of Constraint Manifolds

Characterization of constraint manifolds in Section 3.5 are local, in the sense that they are assured to hold only in neighborhoods of configurations involved. A differentiable manifold  $\tilde{C}$  looks like the Euclidean space  $R^{n_{hc}}$  on a chart  $(\phi, U)$ , where  $\phi: U \subset R^{n_{gc}} \rightarrow R^{n_{gc}-n_{hc}}$ ,  $n_{gc} > n_{hc}$ , and  $\phi$  is a diffeomorphism; i.e., a differentiable mapping with a differentiable inverse. A metaphor is that a nearsighted observer sees the neighborhood  $U$  as Euclidean, even though  $\tilde{C}$  is not Euclidean. As outlined in Section 2.2.4, differential geometry is a tool, analogous to corrective lenses, that provide a global view of the manifold by the nearsighted observer. The basics of differential geometry in Euclidean space  $R^{n_{gc}}$ , where mechanical systems function, are applied in this section to establish global properties of constraint manifolds and to demonstrate their use in dynamics of a model problem that provides insight into developments to come.

#### 3.6.1 Components of Constraint Manifolds

A consequence of the fact that the *regular configuration space*  $\tilde{C}$  of Eq. (3.5.18), with the *tangent space parameterization* of Eq. (3.5.31), is a *differentiable manifold* is that it is a *topological space* in the *topology induced by*  $R^{n_{gc}}$  and, as summarized in Section 2.2.4, can be partitioned into *maximal, singularity free, path connected, components*  $\tilde{C}_\alpha$ ,  $\alpha \in I$ , such that  $\tilde{C}_\alpha \cap \tilde{C}_\beta = \emptyset$ , for all  $\alpha \neq \beta$ , and  $\tilde{C} = \cup_{\alpha \in I} \tilde{C}_\alpha$ . In particular, this shows that the number of *degrees of freedom* of a multibody system is constant throughout each  $\tilde{C}_\alpha$ . These facts are true in Euclidean space, but do not necessarily hold in abstract topological spaces. The components  $\tilde{C}_\alpha$ , abbreviated *domains of functionality*, are maximal, path connected, open subsets of  $\tilde{C}$ , throughout each of which the mechanism is locally parameterized by Eq. (3.5.34); i.e., no singular configurations occur in  $\tilde{C}_\alpha$ .

Extension of the local parameterization of Eq. (3.5.31) to a *global setting* on each  $\tilde{C}_\alpha$  is a powerful result that *differential geometry* brings to kinematics. While such results are consequences of *differential geometry*, a deep study of the abstract theory of differential geometry is not required for their use. Once established, implementation of these results requires only use of linear algebra and multivariable calculus.

The components  $\tilde{C}_\alpha$  are disjoint in  $\tilde{C}$ , hence also disjoint in the *ambient space*  $R^{n_{gc}}$ . A continuous transition of mechanism configuration from the closure of one such domain to another requires crossing the boundaries of each in  $R^{n_{gc}}$ , if they have boundaries, in which case a continuous *one-dimensional configuration trajectory*  $\mathbf{q}(t)$ ,  $0 \leq t \leq 1$ , in  $R^{n_{gc}}$  with  $\mathbf{q}(0) \in \tilde{C}_\alpha$  and  $\mathbf{q}(1) \in \tilde{C}_\beta$ ,  $\alpha \neq \beta$ , must pass through  $\partial \tilde{C}_\alpha$  and  $\partial \tilde{C}_\beta$ , hence encountering a singularity. This is illustrated schematically in Fig. 3.6.1. The closures of some domains  $\tilde{C}_\alpha$ ; e.g.,  $\tilde{C}_1$ , may not intersect others; they may intersect at a single point; e.g.  $\tilde{C}_2$  and  $\tilde{C}_3$ ; or they may intersect along common boundaries; e.g.,  $\tilde{C}_4$  and  $\tilde{C}_5$ . In any case, a continuous transition from one domain of functionality to another cannot occur without encountering a singular configuration or going outside  $\tilde{C}$ . The latter case is illustrated by Examples 3.4.1 and 3.4.2 in which  $\tilde{C}$  is partitioned into a pair of components whose closures do not intersect. In such cases, to transform the

mechanism configuration from one component to another, it must be disassembled and reassembled in the desired component. In this sense, such disjoint components are characterized by associated *mechanism assembled configurations*. If closures of a pair of domains intersect, external influence may be exerted to make a continuous transition across the common singular boundary, so the mechanism does not need to be disassembled and reassembled to make the transition, but it must encounter a singularity. This behavior is illustrated by example 3.4.3.

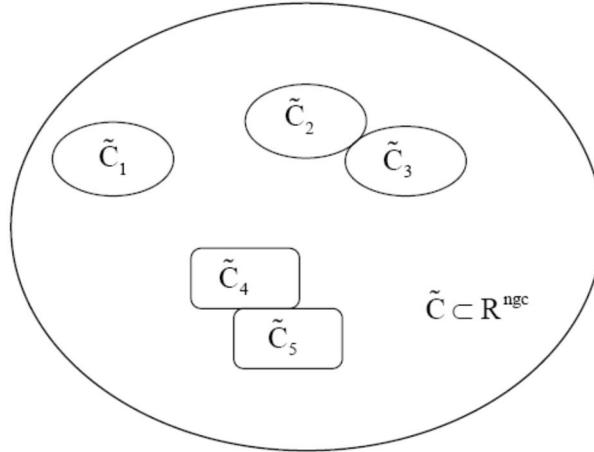


Figure 3.6.1 Closures of Domains of Functionality

If two *charts* defined in  $\tilde{C}_\alpha$  intersect, as shown schematically in Fig. 3.6.2, then there is a differentiable mapping between intersecting domains of the charts and they are called *compatible charts* (Lee, 2010; Schlichtkrull, 2013), much as two planar maps of the surface of the earth that cover the same locale are compatible. A family of compatible charts whose images cover the entire manifold is called an *atlas*. The atlas provides a structure upon which to characterize *trajectories in a constraint manifold*, such as a solution curve for the equations of constrained dynamics, as shown schematically in Fig. 3.6.2. The charts provide for *changes in local coordinates* along a trajectory  $\mathbf{q}(v(t))$  in  $\tilde{C}_\alpha$ , including differentiability of coordinates and tangent vectors. The theory of *differentiable manifolds* (Lee, 2010; Schlichtkrull, 2013) assures that a solution trajectory can be continued in the open component  $\tilde{C}_\alpha$  until the constraint Jacobian  $\Phi_q(\mathbf{q})$  approaches rank deficiency. While the theory does not explicitly determine the *extent of components* of the constraint manifold, it does assure that the *parameterization by charts* that has been presented is valid on the largest possible subset of regular configuration space. Nothing more is possible.

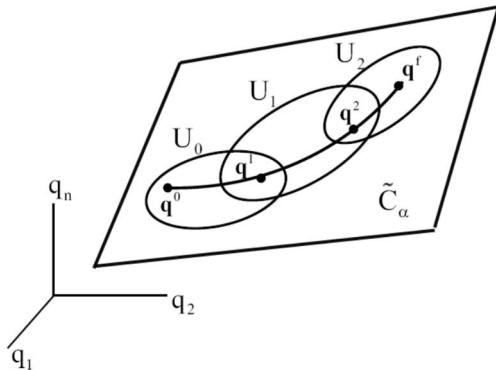


Figure 3.6.2 Continuation of Solution Trajectory over Charts

Planar two-body slider-crank mechanisms treated in Sections 3.4 provide specific instances of the foregoing concepts, as does the spatial particle on a unit sphere treated in Section 3.5.1. Kinematic analysis of more complex mechanisms is carried out in the remainder of this chapter and their dynamics is treated in Chapter 5.

A few observations are in order. While an enterprising person might draw pictures of manifolds for mechanisms in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , drawings of manifolds and in higher dimensional spaces are out of the question. The theoretical existence of maximal path connected components of manifolds in Euclidean space is realized in examples treated, even for mechanisms with embedded singular configurations. As observed in Chapter one, mechanical system kinematics is rife with nonlinearity. The only hope to deal effectively with this challenge is the differential geometric construct that enables dealing with applications of moderate to large dimension, without relying on pictures, but with insight based on two- and three-dimensional experience. Some aspects of defining the extent (size) of manifold components as subsets of  $\mathbb{R}^{n_{\text{gc}}}$  remain unresolved.

### 3.6.2 Dynamics of a Particle on a Unit Sphere

Consider the *particle on a unit sphere* shown in Fig. 3.6.4 that was introduced in Example 1.2.1 of Section 1.2. *Tangent space parameterization* of the *constraint manifold*  $\tilde{C}$  is presented in Section 3.5.1, summarized in this section and extended to obtain ODE of motion for the constrained system.

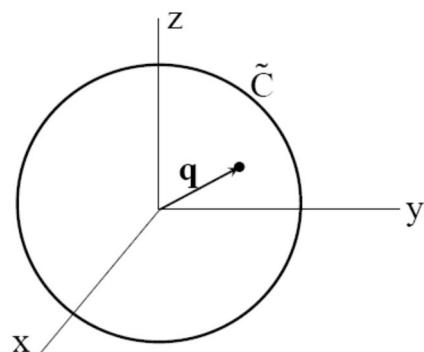


Figure 3.6.4 Particle on Unit Sphere

### 3.6.2.1 Tangent Space Generalized Coordinates

Vectors  $\mathbf{q}$  to points on the unit sphere must satisfy the holonomic kinematic constraint,

$$(\mathbf{q}) = (\mathbf{q}^T \mathbf{q} - 1) / 2 = 0 \quad (3.6.1)$$

At a point  $\mathbf{q}(t^0) = \mathbf{q}^0$  on the unit sphere, the vector  $\mathbf{q}^0$  is normal to the surface of the sphere and a basis  $\mathbf{V}$  for the tangent space is obtained by solving

$$\begin{aligned} {}_{\mathbf{q}}(\mathbf{q}^0) \mathbf{V} &= \mathbf{q}^{0T} \mathbf{V} = 0 \\ \mathbf{V}^T \mathbf{V} &= \mathbf{I} \end{aligned} \quad (3.6.2)$$

using *singular-value decomposition* (Strang, 1980). Since the columns of  $\mathbf{V}$  are orthogonal and both are orthogonal to  $\mathbf{q}^0$ , they form a basis for the *tangent plane*, or *tangent space*, to the sphere at point  $\mathbf{q}^0$ . Define

$$\mathbf{U} = {}_{\mathbf{q}}^T(\mathbf{q}^0) = \mathbf{q}^0 \quad (3.6.3)$$

where  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$ ,  $\mathbf{V}^T \mathbf{U} = \mathbf{0}$ , and  $\mathbf{U}^T \mathbf{U} = 1$ .

Since the unit vector  $\mathbf{U}$  is orthogonal to the linearly independent columns of  $\mathbf{V}$ , the columns of  $\mathbf{V}$  and  $\mathbf{U}$  form a basis for  $\mathbb{R}^3$ . Every vector  $\mathbf{q}$  in  $\mathbb{R}^3$  can thus be uniquely written in terms of *tangent space generalized coordinates*  $\mathbf{v} \in \mathbb{R}^2$  and  $u \in \mathbb{R}^1$  as

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - u\mathbf{U} \quad (3.6.4)$$

where the sign of the third term on the right is selected to represent a projection onto the regular configuration space from the tangent space, as shown schematically in Fig. 3.6.5. Multiplying Eq. (3.6.4) on the left by  $\mathbf{V}^T$  and  $\mathbf{U}^T$  and using the relations  $\mathbf{U}^T \mathbf{V} = \mathbf{0}$  and  $\mathbf{V}^T \mathbf{U} = \mathbf{0}$  yields the inverse relations

$$\begin{aligned} \mathbf{v} &= \mathbf{V}^T(\mathbf{q} - \mathbf{q}^0) \\ u &= -\mathbf{U}^T(\mathbf{q} - \mathbf{q}^0) \end{aligned} \quad (3.6.5)$$

In particular,  $\mathbf{v}(t^0) = \mathbf{v}^0 = \mathbf{0}$  and  $u(t^0) = u^0 = 0$ .

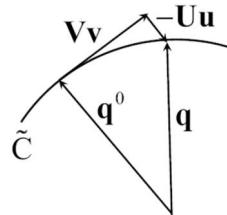


Figure 3.6.5 Projection onto Regular Configuration Space

Requiring that  $\mathbf{q}$  of Eq. (3.6.4) satisfies the holonomic constraint of Eq. (3.6.1); i.e., substituting  $\mathbf{q}$  of Eq. (3.6.4) into Eq. (3.6.1) yields  $\mathbf{q}^T \mathbf{q} - 1 = u^2 - 2u + \mathbf{v}^T \mathbf{v} = 0$ . Using the quadratic formula,  $u = 1 \pm \sqrt{1 - \mathbf{v}^T \mathbf{v}}$ . Clearly, this parameterization is valid only if  $\mathbf{v}^T \mathbf{v} < 1$ . Geometrically, this means that if  $\mathbf{v}^T \mathbf{v} > 1$ , the line perpendicular to the tangent space through the

point  $\mathbf{q}^0 + \mathbf{V}\mathbf{v}$  does not intersect the unit sphere in  $\mathbb{R}^3$ . Since  $u = 0$  if  $\mathbf{v} = \mathbf{0}$ , the minus sign must be selected, so

$$u = 1 - \sqrt{1 - \mathbf{v}^T \mathbf{v}} \equiv h(\mathbf{v}) \quad (3.6.6)$$

and Eq. (3.6.4) is

$$\mathbf{q} \equiv \psi(\mathbf{v}) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \left(1 - \sqrt{1 - \mathbf{v}^T \mathbf{v}}\right)\mathbf{q}^0 = \mathbf{V}\mathbf{v} + \sqrt{1 - \mathbf{v}^T \mathbf{v}}\mathbf{q}^0 \quad (3.6.7)$$

The vector  $\mathbf{q}$  of Eq. (3.6.7) is shown in Section 3.5.1 to satisfy Eq. (3.6.1), for all  $\mathbf{v}$  in a neighborhood of  $\mathbf{v}^0 = \mathbf{0}$ .

Since  $\mathbf{V}$  and  $\mathbf{q}^0$  are constant, differentiation of Eq. (3.6.4) yields

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{V}\dot{\mathbf{v}} - \dot{u}\mathbf{q}^0 \\ \ddot{\mathbf{q}} &= \mathbf{V}\ddot{\mathbf{v}} - \ddot{u}\mathbf{q}^0 \end{aligned} \quad (3.6.8)$$

Multiplying Eqs. (3.6.8) on the left by  $\mathbf{V}^T$  and  $\mathbf{q}^{0T}$  and manipulating yields the inverse relations

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{V}^T \dot{\mathbf{q}} \\ \dot{u} &= -\mathbf{q}^{0T} \dot{\mathbf{q}} \\ \ddot{\mathbf{v}} &= \mathbf{V}^T \ddot{\mathbf{q}} \\ \ddot{u} &= -\mathbf{q}^{0T} \ddot{\mathbf{q}} \end{aligned} \quad (3.6.9)$$

In particular, the first of Eqs. (3.6.9) implies  $\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0$ .

Substituting  $\dot{\mathbf{q}}$  of Eq. (3.6.8) into the kinematic velocity constraint,  $\mathbf{q}^T \dot{\mathbf{q}} = \mathbf{q}^T \mathbf{q} = 0$ , yields the relation  $\mathbf{q}^T \mathbf{V} \dot{\mathbf{v}} - \dot{u} \mathbf{q}^T \mathbf{q}^0 = 0$ . Since  $\mathbf{q}^{0T} \mathbf{q}^0 = 1$ ,  $\mathbf{q}^T \mathbf{q}^0 \neq 0$  in a neighborhood of  $\mathbf{q}^0$  and

$$\dot{u} = \left( \frac{1}{\mathbf{q}^T \mathbf{q}^0} \right) \mathbf{q}^T \mathbf{V} \dot{\mathbf{v}} \quad (3.6.10)$$

in that neighborhood. Substituting this relation into the first of Eqs. (3.6.8),

$$\dot{\mathbf{q}} = \mathbf{V} \dot{\mathbf{v}} - \left( \frac{1}{\mathbf{q}^T \mathbf{q}^0} \right) \mathbf{q}^0 \mathbf{r}^T \mathbf{V} \dot{\mathbf{v}} = \left[ \mathbf{I} - \left( \frac{1}{\mathbf{q}^T \mathbf{q}^0} \right) \mathbf{q}^0 \mathbf{q}^T \right] \mathbf{V} \dot{\mathbf{v}} = \mathbf{D}(\mathbf{q}) \dot{\mathbf{v}} \quad (3.6.11)$$

where

$$\mathbf{D}(\mathbf{q}) \equiv \left[ \mathbf{I} - \left( \frac{1}{\mathbf{q}^T \mathbf{q}^0} \right) \mathbf{q}^0 \mathbf{q}^T \right] \mathbf{V} \quad (3.6.12)$$

Substituting  $\mathbf{q}$  of Eq. (3.6.7) and  $\dot{\mathbf{q}}$  of Eq. (3.6.11) into the kinematic velocity constraint verifies that it is satisfied for arbitrary  $\mathbf{v}$  in a neighborhood of  $\mathbf{v}^0$  and for all  $\dot{\mathbf{v}}$ . A result identical in form to Eq. (3.6.11) holds for differentials of variables; i.e.,

$$\delta \mathbf{q} = \mathbf{D}(\mathbf{q}) \delta \mathbf{v} \quad (3.6.13)$$

Substituting from the second of Eqs. (3.6.8) into the kinematic acceleration constraint,  $\ddot{\mathbf{q}}(\mathbf{q})\ddot{\mathbf{q}} = \dot{\mathbf{q}}^T\ddot{\mathbf{q}} + \ddot{\mathbf{q}}^T\dot{\mathbf{q}} = 0$ , yields the relation  $\mathbf{q}^T\mathbf{V}\ddot{\mathbf{v}} - \ddot{\mathbf{u}}\mathbf{q}^T\mathbf{q}^0 + \dot{\mathbf{q}}^T\dot{\mathbf{q}} = 0$ . In a neighborhood of  $\mathbf{q}^0$ ,

$$\ddot{\mathbf{u}} = \left( \frac{1}{\mathbf{q}^T\mathbf{q}^0} \right) (\mathbf{q}^T\mathbf{V}\ddot{\mathbf{v}} + \dot{\mathbf{q}}^T\dot{\mathbf{q}}) \quad (3.6.14)$$

Substituting this relation into the second of Eqs. (3.6.8) and manipulating,

$$\ddot{\mathbf{q}} = \mathbf{D}(\mathbf{q})\ddot{\mathbf{v}} - \left( \frac{\dot{\mathbf{q}}^T\dot{\mathbf{q}}}{\mathbf{q}^T\mathbf{q}^0} \right) \mathbf{q}^0 \quad (3.6.15)$$

Substituting  $\mathbf{q}$  of Eq. (3.6.7),  $\dot{\mathbf{q}}$  of Eq. (3.6.11), and  $\ddot{\mathbf{q}}$  of Eq. (3.6.15) into the kinematic acceleration constraint verifies that it is satisfied for arbitrary  $\mathbf{v}$  in a neighborhood of  $\mathbf{v}^0$  and for all  $\dot{\mathbf{v}}$  and  $\ddot{\mathbf{v}}$ . Equations (3.6.7), (3.6.11), and (3.6.15) thus yield position, velocity, and acceleration that satisfy all three forms of kinematic constraint.

### 3.6.2.2 ODE of Motion

With gravitational acceleration  $\mathbf{g}$  acting in the negative  $\mathbf{z}$  direction on the particle of Fig. 3.6.4, the variational equation of motion of Eq. (1.2.5) is

$$\delta\mathbf{q}^T [m\ddot{\mathbf{q}} + mg\mathbf{u}_z] = 0 \quad (3.6.16)$$

which must hold for all  $\mathbf{q}$  such that  $\Phi_q(\mathbf{q}) \mathbf{q} = 0$ . Substituting  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  from Eqs. (3.6.13) and (3.6.15) into Eq. (3.6.16),

$$\delta\mathbf{v}^T \mathbf{D}(\mathbf{q})^T \left[ m\mathbf{D}(\mathbf{q})\ddot{\mathbf{v}} - m \left( \frac{\dot{\mathbf{q}}^T\dot{\mathbf{q}}}{\mathbf{q}^T\mathbf{q}^0} \right) \mathbf{q}^0 + mg\mathbf{u}_z \right] = 0 \quad (3.6.17)$$

Since  $\Phi_q(\mathbf{q}) \mathbf{q} = \mathbf{q}^T \mathbf{D}(\mathbf{q}) \mathbf{v} = \left[ \mathbf{q}^T - \frac{\mathbf{q}^T\mathbf{q}^0}{\mathbf{q}^T\mathbf{q}^0} \mathbf{q}^T \right] \mathbf{V} \mathbf{v} = 0$ , Eq. (3.6.17) holds for arbitrary  $\mathbf{v}$  and,

$$m\mathbf{D}^T(\mathbf{q})\mathbf{D}(\mathbf{q})\ddot{\mathbf{v}} = m\mathbf{D}^T(\mathbf{q}) \left( \frac{\dot{\mathbf{q}}^T\dot{\mathbf{q}}}{\mathbf{q}^T\mathbf{U}} \right) \mathbf{U} - mg\mathbf{D}^T(\mathbf{q})\mathbf{u}_z \quad (3.6.18)$$

which, with Eqs. (3.6.7) and (3.6.11) is a second order ODE in  $\mathbf{v}$ . At  $\mathbf{q} = \mathbf{q}^0$ ,  $\mathbf{q}^{0T}\mathbf{V} = \mathbf{0}$  and Eq. (3.6.12) is  $\mathbf{D}(\mathbf{q}^0) = \mathbf{V}$ . Since  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ , the matrix  $m\mathbf{D}^T(\mathbf{q})\mathbf{D}(\mathbf{q})$  on the left of Eq. (3.6.18) at  $\mathbf{q}^0$  equals  $m\mathbf{I}$ , which is positive definite, hence nonsingular. Since  $\mathbf{D}(\mathbf{q})$  is a continuous function of  $\mathbf{q}$ , in a neighborhood of  $\mathbf{q}^0$ , the symmetric matrix  $m\mathbf{D}^T(\mathbf{q})\mathbf{D}(\mathbf{q})$  is positive definite, hence nonsingular in that neighborhood.

The first of Eqs. (3.6.5) and Eq. (3.6.9) yield initial conditions

$$\begin{aligned} \mathbf{v}^0 &= \mathbf{0} \\ \dot{\mathbf{v}}^0 &= \mathbf{V}^T \dot{\mathbf{q}}^0 \end{aligned} \quad (3.6.19)$$

Finally, Eq. (3.6.18) is a *second order ODE*,

$$\mathbf{M}(\mathbf{v})\ddot{\mathbf{v}} = \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}}, t) \quad (3.6.20)$$

where, from Eqs. (3.6.7) and (3.6.11),  $\mathbf{q} = \mathbf{q}(\mathbf{v}) = \mathbf{V}\mathbf{v} + \sqrt{1 - \mathbf{v}^T \mathbf{v}} \mathbf{q}^0$  and

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}) = \mathbf{D}(\mathbf{q}(\mathbf{v}))\dot{\mathbf{v}} = \left[ \mathbf{I} - \left( \frac{1}{\mathbf{q}^T(\mathbf{v})\mathbf{q}^0} \right) \mathbf{q}^0 \mathbf{q}^T(\mathbf{v}) \right] \mathbf{V}\dot{\mathbf{v}}, \text{ so}$$

$$\begin{aligned} \mathbf{M}(\mathbf{v}) &= m\mathbf{D}^T(\mathbf{q}(\mathbf{v}))\mathbf{D}(\mathbf{q}(\mathbf{v})) \\ \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) &= m\mathbf{D}^T(\mathbf{q}(\mathbf{v})) \left( \frac{\dot{\mathbf{q}}^T(\mathbf{v}, \dot{\mathbf{v}})\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}})}{\mathbf{q}^T(\mathbf{v})\mathbf{q}^0} \right) \mathbf{q}^0 - mg\mathbf{D}^T(\mathbf{q}(\mathbf{v}))\mathbf{u}_z \end{aligned} \quad (3.6.21)$$

The arguments of all functions of  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  involved are shown, to emphasize that they are functions of  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ . Since  $\mathbf{M}(\mathbf{v})$  is nonsingular in a neighborhood of  $\mathbf{v}^0$ , the initial-value problem of Eqs. (3.6.20) and (3.6.19) has a unique solution in a neighborhood of  $\mathbf{v}^0$  (Teschl, 2012). If Eq. (3.6.20), with initial conditions of Eq. (3.6.19), is numerically integrated for  $\mathbf{v}$  and its derivatives, using an ODE solver,  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  of Eqs. (3.6.7), (3.6.11), and (3.6.15) satisfy all three forms of kinematic constraint, to within the accuracy of the integrator.

Since  $\mathbf{v}$  is only a valid parameterization of the constraint manifold for  $\mathbf{v}^T \mathbf{v} < 1$ , bounds on the magnitude of  $\mathbf{v}$  must be monitored during numerical integration. Further, the *condition number*, defined in Section 2.2.8, of  $\mathbf{M}(\mathbf{v})$  in Eq. (3.6.21) must be bounded to avoid singularity in Eq. (3.6.18). If an assigned tolerance on the norm of  $\mathbf{v}$  or condition number of  $\mathbf{M}(\mathbf{v})$  is exceeded, the current value of  $\mathbf{q}$  is designated  $\bar{\mathbf{q}}^0$ , and  $\bar{\mathbf{V}}$  is redefined. With this *reparameterization*, the integration process is continued with new initial conditions on  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ , namely from Eqs. (3.6.19),  $\bar{\mathbf{v}}^0 = \mathbf{0}$  and  $\bar{\mathbf{V}}^0 = \mathbf{V}^T \bar{\mathbf{q}}^0$ . As shown in Fig. 3.6.2, this process amounts to moving from one chart to another and continuing to a specified final time, with results reported in terms of  $\mathbf{q}$  and its derivatives, using Eqs. (3.6.7), (3.6.11), and (3.6.15). The user of the process need not know values of  $\mathbf{v}$  and its derivatives that have been computed. This process is guaranteed to be continued, since with  $\mathbf{q}^T \mathbf{q} = 1$ ,  $\mathbf{q}(\mathbf{q}) = \mathbf{q}^T$  never fails to have full rank and the regular configuration space  $\tilde{\mathcal{C}}$  is the entire unit sphere. This manifold continuation process resolves the difficulties encountered with conventional methods of dynamics in Example 1.2.1.

### 3.6.2.3 Numerical Solution

Inertia properties used for simulation are  $m = 1$  kg and  $g = 9.8$  m/sec<sup>2</sup>. The initial configuration is with the mass at the pole on the z axis,  $\mathbf{q}^0 = \mathbf{u}_z = [0 \ 0 \ 1]^T$ , and initial velocity is  $\dot{\mathbf{q}}^0 = \mathbf{u}_y = [0 \ 1 \ 0]^T$ . This begins motion at the singularity in the  $\theta - \phi$  generalized coordinates of Example 1.2.1 and necessarily passes through the singular point  $\mathbf{q} = \mathbf{u}_y$  for the  $\alpha - \beta$  generalized coordinates of the same example. As a check on conservation of energy, total energy is  $TE = (1/2)m\dot{\mathbf{q}}^T \dot{\mathbf{q}} + mgq_z$ , where  $q_z$  is the z component of the vector  $\mathbf{q}$ . Code 3.6.1 of Appendix 3.A implements the foregoing formulation and solution process.

Plots in Fig. 3.6.5 show the y and z coordinates of the particle as time progresses over a 10 sec interval. As required, the x coordinate of the particle remained zero and *total energy* was

constant for this *conservative system*, both within computer precision. Numerical results reported were obtained using the *Nystrom 4 explicit integrator* (Hairer, Norsett, and Wanner, 1993) with constant step size 0.001 sec. All three forms of kinematic constraint were satisfied, with maximum error equal to zero to computer precision. Over the 10 sec simulation, with 10,000 time steps, 18 reparameterizations were required (555 time steps per *reparameterization*), all due to a bound of 0.75 on the norm of  $\mathbf{v}$ . This means that 18 charts in Fig. 3.6.2 were used to compute the entire solution. No singularities were encountered, even though the simulation passed through the y and z poles several times.

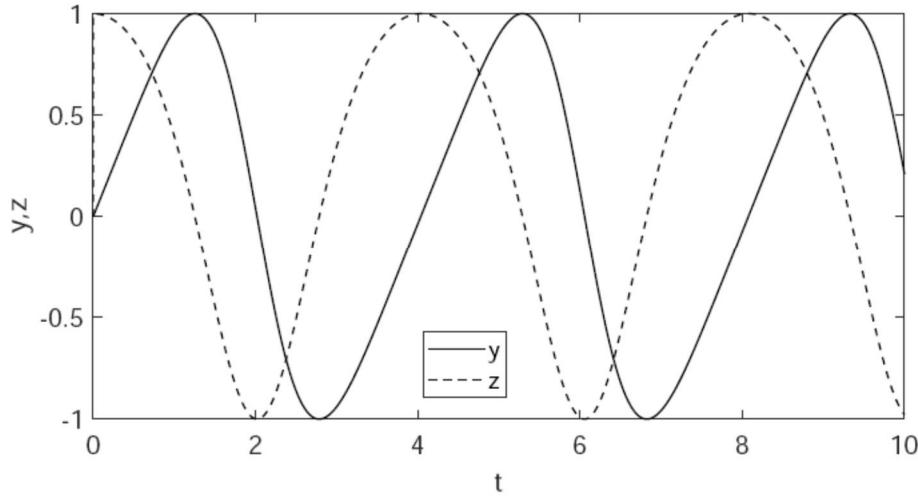


Figure 3.6.5 y and z Coordinates of Particle on Unit Sphere

As an excursion, gravity was set as  $-mg[0.5 \ 0 \ 1]^T$  and a 40 second simulation was run with the same data and initial conditions. The trajectory obtained is plotted on the unit sphere in Fig. 3.6.6. In 40,000 time steps, 156 reparameterizations were algorithmically carried out (256 time steps per reparameterization). Geometrically, this amounts to using 156 charts in Fig. 3.6.2 to continue the solution trajectory over a 40 seconds period of time. No singularities were encountered and configuration, velocity, and acceleration constraints were satisfied to computer precision.

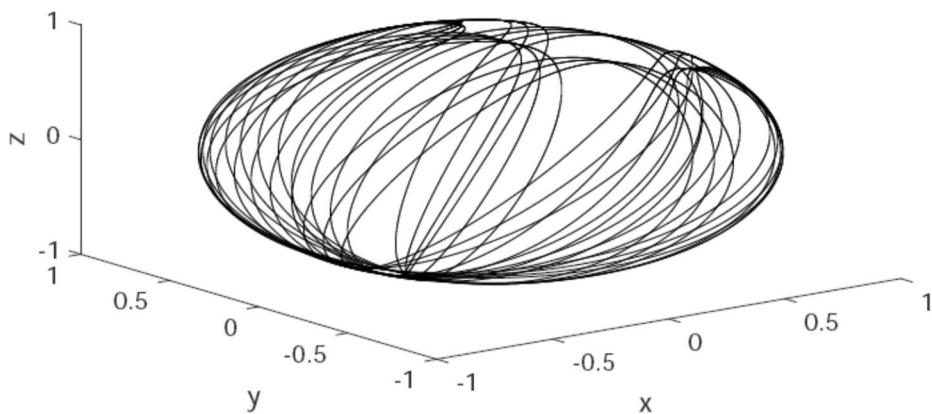


Figure 3.6.6 Particle on Unit Sphere with Gravity  $mg[0.5 \ 0 \ 1]^T$

### 3.6.2.4 A Roadmap for Subsequent Development of Methods of Multibody Dynamics

The model problem of a particle constrained to move on a unit sphere served as a warning in Section 1.2 that subtle issues lurk in the field of mechanical system dynamics. While there exists no globally valid parameterization of even the unit sphere, much less constraint manifolds for general multibody systems, the tools of differential geometry emerge as a practical foundation for multibody system kinematics and dynamics. Following completion of the treatment of multibody system kinematics in this chapter and derivation of broadly applicable variational equations of dynamics in Chapter 4, the differential geometric approach used in this section to treat the particle on the unit sphere is extended in Chapter 5 to provide a comprehensive theory and numerical methods *for multibody dynamics on differentiable manifolds*. This formulation is then be used to provide theoretical and computational methods for *nonholonomic systems, systems with friction, and manipulators* in Chapters 6, 8, and 9.

### 3.6.3 Differentiable Manifolds of a Triple Slider-Crank in $\mathbb{R}^4$

Most well-designed mechanisms encounter no singular configurations, in which case the regular configuration space, which is shown in Section 3.5.3 to be a differentiable manifold, is equal to the entire configuration space. This is the case for planar slider-crank Examples 3.4.1 and 3.4.2, the planar double pendulum and windshield wiper of Section 3.2, and the spatial two-body pendulum, fly-ball governor, and automotive suspension of Section 3.3. The planar and spatial slider-crank mechanisms of Sections 3.2 and 3.3 encounter isolated singular configurations for certain values of connecting rod length. A simple mechanism with configuration space in  $\mathbb{R}^4$  is considered, in which set valued singular configurations occur.

The *triple slider-crank* planar mechanism shown in Fig. 3.6.7 is comprised of three sliders that translate along coordinate axes with generalized coordinates  $q_1, q_3$ , and  $q_4$  and a crank (body 2) of unit radius that is pivoted in slider 1 and rotates with generalized coordinate  $q_2$  relative to slider 1. The configuration space is thus a subset of  $\mathbb{R}^4$ , where pictures may not be easily drawn. A distance constraint of length  $a > 0$  acts between the point on the  $x'_2$ -axis on the crank periphery and slider 3. Finally, a distance constraint of length 2 units acts between slider 1 and slider 4.

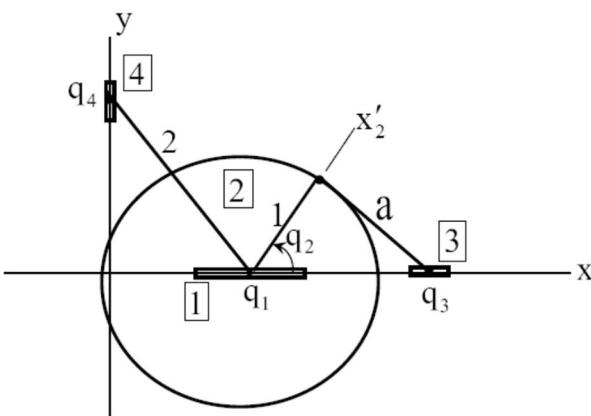


Figure 3.6.7 Triple Slider-Crank

Kinematic constraints for this mechanism and the associated constraint Jacobian are

$$\Phi(\mathbf{q}) = (1/2) \begin{bmatrix} (q_3 - q_1 - \cos q_2)^2 + \sin^2 q_2 - 1 \\ q_1^2 + q_4^2 - 4 \end{bmatrix} \quad (3.6.22)$$

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} -(q_3 - q_1 - \cos q_2) & (q_3 - q_1) \sin q_2 & (q_3 - q_1 - \cos q_2) & 0 \\ q_1 & 0 & 0 & q_4 \end{bmatrix} \quad (3.6.23)$$

**Example 3.6.1** For the mechanism of Fig. 3.6.7 with  $a = 2$ , Eq. (3.6.22) yields

$q_3 - q_1 - \cos q_2 = \pm \sqrt{4 - \sin^2 q_2} \neq 0$ . To consider the rank of  $\Phi_q(\mathbf{q})$  of Eq. (3.6.23), the equation  $\Phi_q^T(\mathbf{q})\mathbf{a} = 0$  yields  $\pm \sqrt{4 - \sin^2 q_2} a_1 = 0$ , so  $a_1 = 0$ , and the equations  $q_1 a_2 = 0 = q_4 a_2$ , so  $a_2 = 0$ . Thus, the Jacobian has full rank for all  $\mathbf{q}$ . From Eq. (3.6.22), four solution sets are

$$C_1 = \left\{ \mathbf{q} \in \mathbb{R}^4 : q_3 = q_1 + \cos q_2 + \sqrt{4 - \sin^2 q_2} \text{ and } q_4 = \sqrt{4 - \sin^2 q_2}, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\}$$

$$C_2 = \left\{ \mathbf{q} \in \mathbb{R}^4 : q_3 = q_1 + \cos q_2 + \sqrt{4 - \sin^2 q_2} \text{ and } q_4 = -\sqrt{4 - \sin^2 q_2}, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\}$$

$$C_3 = \left\{ \mathbf{q} \in \mathbb{R}^4 : q_3 = q_1 + \cos q_2 - \sqrt{4 - \sin^2 q_2} \text{ and } q_4 = \sqrt{4 - \sin^2 q_2}, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\}$$

$$C_4 = \left\{ \mathbf{q} \in \mathbb{R}^4 : q_3 = q_1 + \cos q_2 - \sqrt{4 - \sin^2 q_2} \text{ and } q_4 = -\sqrt{4 - \sin^2 q_2}, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\}$$

At  $q_1 = \pm 2$ ,  $q_4 = 0$  and sets  $C_1$  and  $C_2$  have configurations in common. Likewise for  $C_3$  and  $C_4$ . Thus, there are only two disjoint components in  $\tilde{C}$ ,

$$\tilde{C}_1 = \left\{ \mathbf{q} \in \mathbb{R}^4 : q_3 = q_1 + \cos q_2 + \sqrt{4 - \sin^2 q_2} \text{ and } q_4 = \pm \sqrt{4 - \sin^2 q_2}, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\}$$

$$\tilde{C}_2 = \left\{ \mathbf{q} \in \mathbb{R}^4 : q_3 = q_1 + \cos q_2 - \sqrt{4 - \sin^2 q_2} \text{ and } q_4 = \pm \sqrt{4 - \sin^2 q_2}, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\}$$

where  $q_3 - q_1 > 0$  throughout  $\tilde{C}_1$  and  $q_3 - q_1 < 0$  throughout  $\tilde{C}_2$ . The closures of these components do not intersect.

**Example 3.6.2** For the mechanism of Fig. 3.6.7 with  $a = 1$ , Eq. (3.6.22) reduces to

$$\Phi(\mathbf{q}_1) = (1/2) \begin{bmatrix} (q_3 - q_1)(q_3 - q_1 - 2 \cos q_2) \\ q_1^2 + q_4^2 - 4 \end{bmatrix} = \mathbf{0}$$

and the Jacobian remains as in Eq. (3.6.23). This yields solutions  $q_3 - q_1 = 0$  and  $q_3 - q_1 = 2 \cos q_2$ , which are subsets of the configuration space,

$$\begin{aligned} C_1 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_1 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\} \\ C_2 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_1 + 2\cos q_2 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 \leq q_1 \leq 2, 0 \leq q_2 \leq 2\pi \right\} \end{aligned} \quad (3.6.24)$$

and  $C = C_1 \cup C_2$ . These subsets are not components of  $C$ , however, since

$$C_1 \cap C_2 = \left\{ \mathbf{q} = \begin{bmatrix} q_1 & \pm\pi/2 & q_1 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 \leq q_1 \leq 2 \right\} \neq \emptyset. \text{ Thus, the configuration space}$$

for this example is connected and consists of only a single component. That component, however contains sets of singular configurations with  $q_2 = \pm\pi/2$ , where  $\text{rank}(\Phi_q(\mathbf{q})) = 1 < \text{nhc} = 2$ .

Eliminating these singular configurations from  $C_1$  and  $C_2$  yields four disjoint components of the regular configuration space  $\tilde{C}$ ,

$$\begin{aligned} \tilde{C}_1 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_1 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 < q_1 < 2, -\pi/2 < q_2 < \pi/2 \right\} \\ \tilde{C}_2 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_1 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 < q_1 < 2, \pi/2 < q_2 < 3\pi/2 \right\} \\ \tilde{C}_3 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_1 + 2\cos q_2 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 < q_1 < 2, -\pi/2 < q_2 < \pi/2 \right\} \\ \tilde{C}_4 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_1 + 2\cos q_2 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 < q_1 < 2, \pi/2 < q_2 < 3\pi/2 \right\} \end{aligned} \quad (3.6.25)$$

Closures of these components intersect at  $q_2 = \pm\pi/2$  in the following pair of one dimensional sets of singularities in  $C \subset \mathbb{R}^4$ :

$$\begin{aligned} S_1 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & -\pi/2 & q_1 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 \leq q_1 \leq 2 \right\} \\ S_2 &= \left\{ \mathbf{q} = \begin{bmatrix} q_1 & \pi/2 & q_1 & \pm\sqrt{4-q_1^2} \end{bmatrix}^T, -2 \leq q_1 \leq 2 \right\} \end{aligned}$$

Since no rank deficiency of the Jacobian occurs in any of the disjoint open sets of Eq. (3.6.25), they are components of  $\tilde{C}$ . These are two-dimensional subsets of  $\mathbb{R}^4$ , so drawing pictures of these components, as was possible in the case of  $\mathbb{R}^2$  for the two-body slider-crank examples in Section 3.4, is not feasible.

---

Basic results of differential geometry show that open sets defined by local relations in  $\mathbf{R}^{n_{\text{gc}}}$  can be extended to maximal sets that have properties such as being singularity free and path connected. It is especially important that strong forms of these properties hold in Euclidean space, in which mechanical system kinematics and dynamics occur.

The theory of differential geometry shows that the regular configuration space is a manifold that is comprised of maximal, singularity free, path connected components on which continuation of numerical solution of the nonlinear equations of kinematics and dynamics can be carried out until fundamental system singularities, if any exist, block progress. These properties are illustrated for solution of the model problem of dynamics of a particle on a unit sphere that was introduced in Chapter 1. Results for this modest model problem using differential geometry provide a roadmap for subsequent developments.

Finally, an example with configuration space in  $\mathbf{R}^4$  shows that one dimensional sets of singular configurations occur, in contrast to the isolated singular configurations encountered in prior examples.

### 3.7 Singular Configurations

Constraint equations are used for both kinematic and dynamic analysis of mechanical systems in which motion is intended to proceed smoothly with time. For trial designs, however, singular configurations may be encountered as boundaries of constraint manifolds such as the two body slider-crank of Example 3.4.3 and the four body double slider-crank model of Example 3.6.2, beyond which motion cannot continue, more than one possible motion can occur, or discontinuous motion can occur. Singularities due to imposition of time dependent drivers are analyzed in Section 3.7.2, and criteria for identifying singular behavior are discussed in Section 3.7.3. Finally, the relationship between differentiable manifold components and kinematic singularities is discussed in Section 3.7.4

#### 3.7.1 Kinematic Singularities

To see that such behavior is not restricted to pathological cases that are invented by a mathematician, several simple examples are studied. Kinematic singularities that have nothing to do with the intended functionality of the mechanism are studied in Sections 3.4, 3.5, and 3.7.1.

**Example 3.7.1:** A two body model of a *parallelogram mechanism* is shown in Fig. 3.7.1. Bodies one and two have unit length and are pivoted in ground, with angles of rotation  $q_1$  and  $q_2$  relative to the horizontal axis. Their outboard ends are connected by a distance constraint of length  $\ell$ . The distance between pivot points of bodies one and two on the horizontal axis is  $\ell$ .

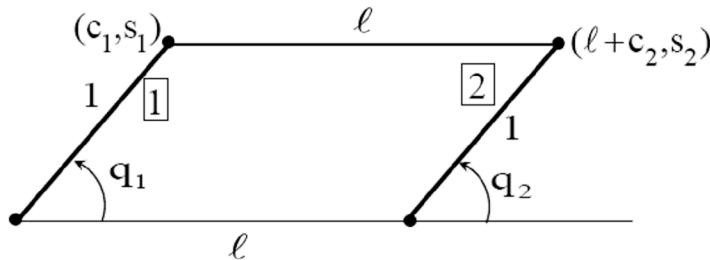


Figure 3.7.1 Parallelogram Mechanism

With the notation  $c_i \equiv \cos q_i$  and  $s_i \equiv \sin q_i$ ,  $i = 1, 2$ , the condition that the length of the top bar is  $\ell$  is the holonomic constraint

$$(\mathbf{q}) = ((\ell + c_2 - c_1)^2 + (s_2 - s_1)^2 - \ell^2) / 2 = 0 \quad (3.7.1)$$

with Jacobian

$$\begin{aligned} {}_{\mathbf{q}}(\mathbf{q}) &= [(\ell + c_2 - c_1)s_1 \quad (\ell + c_2 - c_1)(-s_2)] \\ &= [\ell s_1 + (s_1 c_2 - c_1 s_2) \quad -\ell s_2 - ((s_1 c_2 - c_1 s_2))] \\ &= [\ell s_1 - \sin(q_2 - q_1) \quad -\ell s_2 + \sin(q_2 - q_1)] \end{aligned} \quad (3.7.2)$$

In any kinematically singular configuration, the constraint Jacobian is zero; i.e.,  $\ell s_1 = \sin(q_2 - q_1)$  and  $\ell s_2 = \sin(q_2 - q_1)$ . Thus,  $s_1 = s_2$ , so  $q_1 = q_2$  or  $q_2 = \pi - q_1$ . The singular configuration with  $q_1 = q_2$ ,  $s_1 = s_2$ , and  $q_1 = q_2 = 0$  is shown in Fig. 3.7.2. Admissible

configurations in a neighborhood of this singularity are shown in Fig. 3.7.3. If  $\ell = 1$  a more pathological singularity may occur, as shown in Fig. 3.7.4. In this singularity,  $q_1$  is locked at  $q_1 = 0$  and  $q_2$  is able to rotate freely, with body 2 and the coupler colinear.



Figure 3.7.2 Singular Configuration with  $q_1 = q_2 = 0$

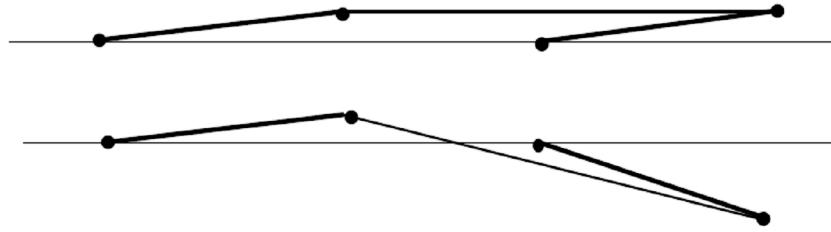


Figure 3.7.3 Configurations in Neighborhood of Singularity with  $q_1 = q_2 = 0$

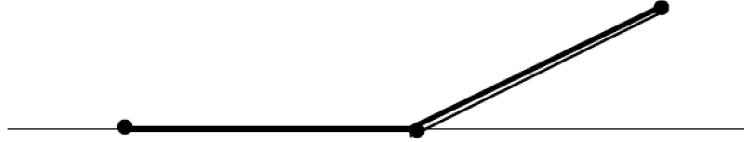


Figure 3.7.4 Locked Singularity with  $q_1 = 0$  and  $\ell = 1$

### 3.7.2 Singularities Due to Time Dependent Drivers

If time dependent drivers are specified, so that the constraint Jacobian matrix is square, singularities that are not purely kinematic may arise. To see this, the slider-crank of Example 3.4.1 is revisited.

**Example 3.7.2:** Imposing the driving constraint  ${}^D(\mathbf{q}, t) = q_1 - \omega t = 0$  on the crank of Example 3.4.1 with  $\ell = 2$ , the combined kinematic and driving constraints are

$$\Phi(\mathbf{q}, t) = \begin{bmatrix} ((q_2 - c)^2 + s^2 - \ell^2)/2 \\ q_1 - \omega t \end{bmatrix} = \mathbf{0} \quad (3.7.3)$$

whose Jacobian is

$$\Phi_q(\mathbf{q}, t) = \begin{bmatrix} sq_2 & q_2 - c \\ 1 & 0 \end{bmatrix} \quad (3.7.4)$$

Thus, a singularity occurs if  $|\Phi_q(\mathbf{q}, t)| = -(q_2 - c) = 0$ ; i.e., the connecting rod is vertical. If  $\ell > 1$ , this cannot occur and there is no singularity. This is the case for practically designed slider-crank mechanisms.

If  $\ell = 1$  in Example 3.4.3, the singularity shown in Fig. 3.4.6 will arise and motion corresponding to the left configurations of Fig. 3.4.3 or to the locked configuration with rotation  $q_1 \neq \pi/2$  will occur. If  $\ell < 1$ , the connecting rod will be vertical in the singular configuration shown in Fig. 3.7.5. While this is not a kinematic singularity, in this configuration,  $\Phi_q$  of Eq. (3.7.4) is singular and the velocity equation  $\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} = -\Phi_t(\mathbf{q}, t)$  will generally yield infinite velocity. Even more likely, the acceleration equation  $\Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} = -\gamma(\mathbf{q}, t)$  will yield infinite acceleration.

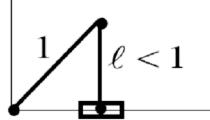


Figure 3.7.5 Driver Singularity with  $\ell < 1$

### 3.7.3 Criteria for Singular Behavior in Kinematic Analysis

Physical insight into *singular behavior* can be gained by studying the constraint velocity equation for a system with square Jacobian,

$$\Phi_q \dot{\mathbf{q}} = -\Phi_t \quad (3.7.5)$$

If  $\Phi_q$  is singular at a time  $t^*$ , Eq. (3.7.5) has a solution if and only if, by the *theorem of the alternative* (Strang, 1980),

$$\boldsymbol{\beta}^T \Phi_t = \mathbf{0} \quad (3.7.6)$$

for all solutions  $\boldsymbol{\beta}$  of

$$\Phi_q^T \boldsymbol{\beta} = \mathbf{0} \quad (3.7.7)$$

This result shows that Eq. (3.7.5) has a finite solution for velocity only if  $\Phi_q$  and  $\Phi_t$  are properly related.

Consider next a small *virtual displacement*  $\delta\mathbf{q}$  that satisfies constraints to *first order*, with time held fixed. Expanding the constraint equations to first order,  $\delta\mathbf{q}$  satisfies

$$\Phi_q \delta\mathbf{q} = \mathbf{0} \quad (3.7.8)$$

Since  $\Phi_q$  is singular at  $t^*$ , its columns are linearly dependent and there exists a solution  $\delta\mathbf{q} \neq \mathbf{0}$  of Eq. (3.7.8) that may be viewed as tangent to solution trajectories that emanate from  $\mathbf{q}(t^*)$ .

Note that if a finite  $\dot{\mathbf{q}}$  satisfies Eq. (3.7.5) and  $\delta\mathbf{q}$  satisfies Eq. (3.7.8), then

$$\Phi_q(\dot{\mathbf{q}} \pm \delta\mathbf{q}) = \Phi_q \dot{\mathbf{q}} \pm \Phi_q \delta\mathbf{q} = -\Phi_t \quad (3.7.9)$$

Thus, two distinct velocities may occur after  $t^*$ , which characterizes a *bifurcation point*.

Since the determinant  $|\Phi_q|$  is zero at  $t^*$ , except in cases for which the rank deficiency of  $\Phi_q$  is greater than 1, its sign must change as a bifurcation point is passed. This can serve as a

computational test for a bifurcation configuration. Only if no solution of Eq. (3.7.5) exists beyond  $t^*$ ; i.e., only if  $\dot{\mathbf{q}}$  approaches infinity as  $t$  approaches  $t^*$ , is there no possibility of continuing a solution. This characterizes a *lock-up configuration*. Thus,  $\dot{\mathbf{q}}$  and/or  $\ddot{\mathbf{q}}$  approaching infinity serves as criteria for lock-up.

A final note on the effect of a *design variation* on performance of a mechanism near a singular point may help in evaluating the engineering feasibility of a design. Let  $\mathbf{b} = [b_1 \cdots b_k]^T$  be a vector of design parameters, such as dimensions. Since these parameters arise in kinematic constraint equations, the constraint equation of the mechanism is a function of design; i.e.,

$$\Phi(\mathbf{q}, t; \mathbf{b}) = 0 \quad (3.7.10)$$

There is a relationship between admissible virtual displacements  $\delta\mathbf{q}$  and small variations  $\delta\mathbf{b}$  in design. Equation (3.7.10) may be expanded to first order (linearized) as

$$\Phi_q \delta\mathbf{q} = -\Phi_b \delta\mathbf{b} \quad (3.7.11)$$

With a singular configuration at time  $t^*$ ,  $\Phi_q$  is singular. Thus, for some  $\delta\mathbf{b}$ , Eq. (3.7.11) may have multiple solutions for  $\delta\mathbf{q}$ , and for other  $\delta\mathbf{b}$  it may have no solutions. Physically, this may be interpreted as indicating that, for some design variations, multiple changes in configurations are possible, as in bifurcation, and for other design variations the mechanism cannot be assembled.

---

**Example 3.7.3:** Consider the slider-crank model of Example 3.4.2 with  $\ell < 1$ , in the *lock-up configuration* shown in Fig. 3.7.5. If  $\ell$  is a design parameter, a positive  $\delta\ell$  leads to two values of  $x_2$ , as shown in Fig. 3.7.6(a). A negative  $\delta\ell$ , however, leads to a design that cannot be assembled with  $\phi_1 = \sin^{-1} \ell$ , as shown in Fig. 3.7.6(b).

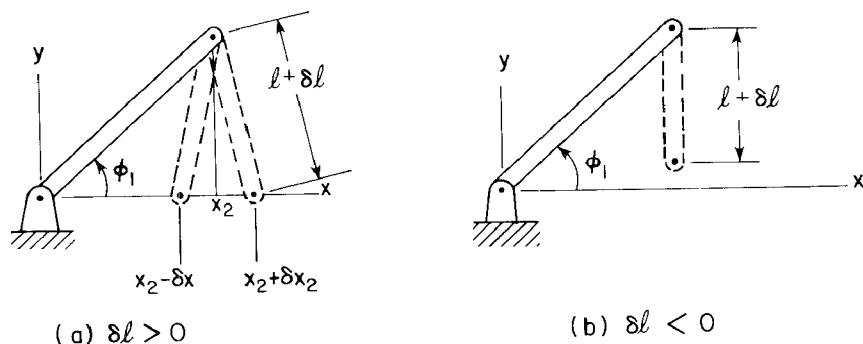


Figure 3.7.6 Effect of Design Variation on Slider-Crank Behavior

---

*Singular points* that define lock-up or bifurcation for kinematically driven systems are isolated; i.e., there are intervals of time in which a unique solution of the equations of kinematics occurs on one or both sides of an *isolated singular point*  $t^*$ . Thus, except at  $t^*$ , if the mechanism

can be assembled, it is well behaved in a neighborhood of the assembled configuration. Another form of singularity that is commonly encountered is a *redundant constraint*. If a mechanism is modeled using standard joints, it is possible that one or more of the kinematic constraint equations is identically satisfied if the remaining constraints are satisfied, in which case the constraints are redundant and consistent.

**Example 3.7.4:** The *five bar parallelogram* mechanism of Fig. 3.7.7 is modeled as a single bar with three distance constraints that represent bars of unit length that are pivoted in ground. With  $\mathbf{q} = [x_1 \ y_1 \ \phi_1]^T = [x \ y \ \phi]^T$ , the kinematic constraint equations are

$$\Phi(\mathbf{q}) = \begin{bmatrix} x^2 + y^2 - 1 \\ (x + \cos \phi - 1)^2 + (y + \sin \phi - 1)^2 \\ (x + 2\cos \phi - 2)^2 + (y + 2\sin \phi)^2 \end{bmatrix} = \mathbf{0} \quad (3.7.12)$$

and the constraint Jacobian is

$$\Phi_q = \begin{bmatrix} 2x & 2y & 0 \\ 2(x + \cos \phi - 1) & 2(y + \sin \phi - 1) & -2(x - 1)\sin \phi + 2y \cos \phi \\ 2(x + 2\cos \phi - 2) & 2(y + 2\sin \phi) & -4(x - 2)\sin \phi + 4y \cos \phi \end{bmatrix} \quad (3.7.13)$$

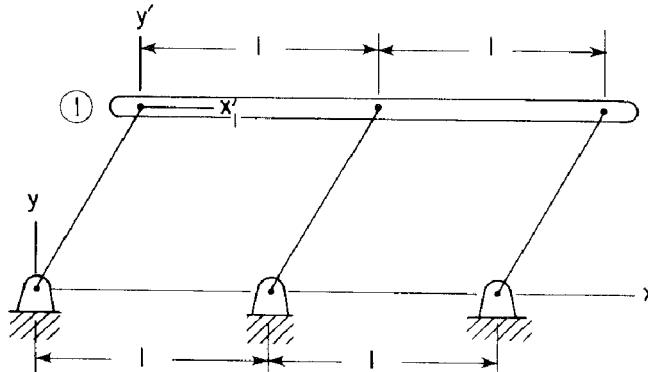


Figure 3.7.7 Five Bar Parallelogram Mechanism Model

Subtracting the first row of the Jacobian from two times the second row and subtracting the result from the third row leads to a matrix with a zero third row, for all values of  $\mathbf{q}$ . Thus, the rank of the constraint Jacobian is only two, and the third of the constraint equations of Eq. (3.7.12) is redundant. This is clearly true from Fig. 3.7.7, since the third distance constraint is automatically satisfied by the parallelogram that is defined with the first two distance constraints.

### 3.7.4 Differentiable Manifold Components and Mechanism Singularities

A distinction between the approach presented herein and much of the literature on mechanism kinematics is the treatment of *singular configurations*. The focus herein is on *regular mechanism configuration components*, which are disjoint, maximal, singularity free, path connected sets in which the mechanism can be effectively modeled. As a biproduct, boundaries of closures of these components, if they exist, are *sets of singularities*. A tenant of the

mechanism singularity literature is that a knowledge of sets of singularities enables their avoidance in mechanism operation. Even if successful, this *singularity avoidance approach* fails to characterize the differentiable manifold structure presented in Sections 3.5 and 3.6, particularly parameterizations that are inherent in the *differentiable manifold formulation* and enable quantitative kinematic analysis.

The foregoing analysis and examples verify theoretical predictions regarding properties of differentiable manifold components, but suggests difficulties in implementation using only *singularity information*. The intricate process of computing sets of singular configurations and piecing them together in  $n_{gc}$ -dimensional configuration space to define boundaries of components  $\tilde{C}_i$  is problematic, for even mechanisms of modest dimension. With evaluation of a large number of  $(n_{gc} - 1)$ -dimensional sets of singular configurations in  $C \subset R^{n_{gc}}$ , it is not clear how to identify and characterize the likewise large number of *singularity free components*  $\tilde{C}_i$  in which the manipulator will function reliably. Piecing these sets of singularities together to define manifold components is akin to solving a jigsaw puzzle in  $n_{gc}$ -dimensional space, a process that is unlikely to succeed. Even if successful, such a process defines only boundaries of some mechanism components, not their structure nor parameterizations that are key to mechanism analysis. In spite of the potential offered by singularity free manifold components and deficiencies associated with relying only on a knowledge of singularities, Web of Science literature searches using keywords “singularity free kinematics” yielded 316 papers and “singularity free workspace” yielded 347 papers, both containing many of the same papers. In contrast a search using keywords “manipulator singularities” yielded 4,787 papers. Achieving the potential for definition and computation of singularity free mechanism regular configuration components would appear to justify a higher research priority.

Singular configurations of mechanisms arise as a result of trial designs that lead to lock-up or bifurcation, or poorly constructed models that have similar behavior. Examples studied show that constraint Jacobian matrices are rank deficient at kinematically singular configurations. In addition, imposing time dependent driving constraints can lead to singular behavior at configurations that are not kinematic singularities. Singular behavior is shown to manifest itself in velocities and/or accelerations diverging to infinity as singularities are approached. Design variations are shown to enable assessment of design deficiency in neighborhoods of singular configurations. Finally, there appears to be an imbalance in the mechanisms literature toward the study of singularities, rather than on defining singularity free components of configuration differentiable manifolds in which mechanisms can reliably function, free of singularities.

### 3.8 Kinematic Position, Velocity, and Acceleration Analysis

A kinematic analysis formulation is presented that includes time dependent drivers, in number equal to the number of degrees of freedom of the system. Derivatives are taken to obtain velocity and acceleration equations. Iterative methods are presented for solution of nonlinear kinematic equations for configurations on a time grid. Linear solution methods are presented for velocities and accelerations on the same time grid. Numerical criteria are presented for identification of singular behavior.

#### 3.8.1 Generalized Coordinates and Constraint Equations

For a multibody system that is made up of nb bodies that are subjected to *holonomic constraints*, a set of ngc *system generalized coordinates* is defined to locate and orient each of the bodies in the system. The *generalized coordinate vector*

$$\mathbf{q} = \left[ \mathbf{q}_1^T \cdots \mathbf{q}_{nb}^T \right]^T \in \mathbb{R}^{ngc} \quad (3.8.1)$$

defines the positions and orientations of all bodies in the system. For planar body i,

$\mathbf{q}_i = \begin{bmatrix} \mathbf{r}_i^T & \phi_i \end{bmatrix}^T$ , and for spatial body i,  $\mathbf{q}_i = \begin{bmatrix} \mathbf{r}_i^T & \mathbf{p}_i^T \end{bmatrix}^T$ , where the Euler parameter vector must satisfy the condition  $\mathbf{p}_i^T \mathbf{p}_i = 1$ .

The kinematic model of a mechanism that is comprised of nb bodies is defined by constraint equations of the form

$$\Phi(\mathbf{q}, t) = \begin{bmatrix} \Phi^K(\mathbf{q}) \\ \Phi^D(\mathbf{q}, t) \\ \Phi^P(\mathbf{q}) \end{bmatrix} = \mathbf{0} \quad (3.8.2)$$

where *Euler parameter normalization constraints* for nsb spatial bodies are nsb equations of the form

$$\Phi^P(\mathbf{q}) = \begin{bmatrix} \mathbf{p}_1^T \mathbf{p}_1 - 1 \\ \vdots \\ \mathbf{p}_{nsb}^T \mathbf{p}_{nsb} - 1 \end{bmatrix} = \mathbf{0} \quad (3.8.3)$$

and nhc *holonomic kinematic constraints* derived in Sections 3.2 and 3.3 are written in vector form as

$$\Phi^K(\mathbf{q}) = \mathbf{0} \quad (3.8.4)$$

The total number of constraints in Eqs. (3.8.3) and (3.8.4) is nsb + nhc. In order that Eq. (3.8.2) is comprised of ngc equations in ngc variables, nd = ngc - nsb - nhc *driving constraints* of Sections 3.1 to 3.3 are defined in the form

$$\Phi^D(\mathbf{q}, t) = \mathbf{0} \quad (3.8.5)$$

and the time dependent driving term Pf(t) relative to a constant value in a constraint, and its first and second time derivatives Pf<sub>t</sub>(t) and Pf<sub>tt</sub>(t) are entered in the P5 Function of MATLAB code.

The organization of constraint equations in Eq. (3.8.2) is important for computer implementation in Sections 9 and 11. Kinematic constraints are entered before driving constraints in the MATLAB code. The constraint function is organized likewise. The code automatically creates the Euler parameter normalization constraints in the last nsb rows of the constraint function.

Let each of the constraint equations of Eq. (3.8.2) have m continuous derivatives with respect to both  $\mathbf{q}$  and t and define the  $n_{\text{gc}} \times n_{\text{gc}}$  system constraint Jacobian as

$$\Phi_q(\mathbf{q}, t) = \begin{bmatrix} \Phi_q^K(\mathbf{q}) \\ \Phi_q^D(\mathbf{q}, t) \\ \Phi_q^P(\mathbf{q}) \end{bmatrix} \quad (3.8.6)$$

At initial time  $t^0$ , let  $\mathbf{q}^0$  denote an *assembled configuration* of the mechanism; i.e.,

$$\Phi(\mathbf{q}^0, t^0) = \mathbf{0} \quad (3.8.7)$$

Under the foregoing conditions, if the system constraint Jacobian is nonsingular at the initial time  $t^0$  and associated assembled configuration  $\mathbf{q}^0$  that satisfy  $\Phi(\mathbf{q}^0, t^0) = \mathbf{0}$ ; i.e.,

$$|\Phi_q(\mathbf{q}^0, t^0)| \neq 0 \quad (3.8.8)$$

the *implicit function theorem* of Section 2.2.5 guarantees that Eq. (3.8.2) has a unique, m-times continuously differentiable solution  $\mathbf{q}(t)$ ; i.e.,  $\Phi(\mathbf{q}(t), t) = \mathbf{0}$ , for all t in a neighborhood of  $t^0$ .

Differentiating Eq. (3.8.2) with respect to time yields the *system velocity equations*,

$$\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} = -\Phi_t(\mathbf{q}, t) \equiv \mathbf{v}(\mathbf{q}, t) \quad (3.8.9)$$

Since the kinematic and Euler parameter constraint functions of Eq. (3.8.2) do not depend explicitly on time,

$$\mathbf{v}(\mathbf{q}, t) = \begin{bmatrix} \mathbf{0} \\ -\Phi_t^D(\mathbf{q}, t) \\ \mathbf{0} \end{bmatrix} \quad (3.8.10)$$

Under the assumption that the system constraint Jacobian is square and nonsingular, Eq. (3.8.9) has an  $(m-1)$ -times continuously differentiable solution for the generalized velocity  $\dot{\mathbf{q}}(t)$ , for all t in a neighborhood of  $t^0$ .

Differentiating Eq. (3.8.9) with respect to time, using the chain rule of differentiation, yields the *system acceleration equations*

$$\begin{aligned} \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} &= -\left(\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - 2\Phi_{tq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \\ &= -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) - 2\Phi_{tq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \end{aligned} \quad (3.8.11)$$

Under the assumption that the system constraint Jacobian is nonsingular for all time, Eq. (3.8.11) has an  $(m - 2)$ -times continuously differentiable solution for the generalized acceleration  $\ddot{\mathbf{q}}(t)$ , for all  $t$  in a neighborhood of  $t^0$ .

### 3.8.2 System Assembly

With a reasonable estimate of an *assembled configuration*  $\mathbf{q}^0$ , the *Newton-Raphson method* of Section 2.2.6 may be used for position analysis, hopefully converging to an accurate solution for  $\mathbf{q}^0$ . If that process fails, a method that is less dependent on a good estimate of  $\mathbf{q}^0$  is needed. One such approach is a *numerical minimization method* applied to the sum of squares of constraint errors,

$$\Psi(\mathbf{q}) \equiv \Phi(\mathbf{q}, t^0)^T \Phi(\mathbf{q}, t^0) \quad (3.8.12)$$

Modern numerical minimization methods are available, implemented in robust computer codes, to begin with even a poor estimate of  $\mathbf{q}^0$  and converge to an accurate solution.

### 3.8.3 Position Analysis

Equation (3.8.2) is a system of  $n_{gc}$  nonlinear equations in a vector  $\mathbf{q} \in \mathbb{R}^{n_{gc}}$  that is to be determined in *position analysis*. Even if the constraint Jacobian is nonsingular, there is little hope of finding an analytical solution  $\mathbf{q}(t)$ , in spite of the fact that the implicit function theorem guarantees existence of a unique solution with  $m$  continuous derivatives. It is necessary, therefore, to resort to construction of a numerical approximate solution at discrete points in time. At  $t = t_i$  of a *time grid*  $t_0, t_1, \dots, t_i$  on which solutions of kinematic equations are to be obtained, let an estimate of the solution of Eq. (3.8.2) be  $\mathbf{q}^j \approx \mathbf{q}_i = \mathbf{q}(t_i)$ . Using the *Newton-Raphson method* of Section 2.2.7, the following linear equations are solved:

$$\begin{aligned} \Phi_q(\mathbf{q}^j, t_i) \Delta \mathbf{q}^j &= -\Phi(\mathbf{q}^j, t_i) \\ \mathbf{q}^{j+1} &= \mathbf{q}^j + \Delta \mathbf{q}^j \end{aligned} \quad (3.8.13)$$

The process is continued until the following convergence criteria are met:

$$\begin{aligned} \|\Phi(\mathbf{q}^{j+1}, t_i)\| &\leq \varepsilon \\ \|\Delta \mathbf{q}^{j+1}\| &\leq \beta \end{aligned} \quad (3.8.14)$$

where  $\varepsilon$  and  $\beta$  are solution error tolerances.

If the Jacobian and right side of the first of Eqs. (3.8.13) are evaluated accurately and the Jacobian is nonsingular at all iterations, then Newton's method has attractive convergence properties. In particular, for sufficiently small error  $\|\mathbf{q}^j - \mathbf{q}(t_i)\|$ , the method will be *quadratically convergent*; i.e., there exists a constant  $a > 0$  such that for iteration number  $j$  sufficiently large,

$$\|\mathbf{q}^{j+1} - \mathbf{q}(t_i)\| \leq a \|\mathbf{q}^j - \mathbf{q}(t_i)\|^2 \quad (3.8.15)$$

If the *initial estimate* of the solution is substantially inaccurate, Newton's method may diverge.

### 3.8.4 Velocity and Acceleration Analysis

Velocities  $\dot{\mathbf{q}}^i \approx \dot{\mathbf{q}}(t_i)$ ,  $i = 1, 2, \dots$ ; i.e., at specified time steps, are determined by solving the *velocity equations*

$$\Phi_q(\mathbf{q}^i, t_i) \dot{\mathbf{q}}^i = -\Phi_t(\mathbf{q}^i, t_i) \equiv v(\mathbf{q}^i, t_i) \quad (3.8.16)$$

Note that no iterative computation is required, since the equations are linear in velocities.

Much as in *velocity analysis*, in *acceleration analysis*,  $\ddot{\mathbf{q}}^i \approx \ddot{\mathbf{q}}(t_i)$ ,  $i = 1, 2, \dots$ , are determined by solving the *acceleration equations*

$$\begin{aligned} \Phi_q(\mathbf{q}^i, t_i) \ddot{\mathbf{q}}^i &= -\left(\Phi_q(\mathbf{q}^i, t_i) \dot{\mathbf{q}}^i\right)_q \dot{\mathbf{q}}^i - 2\Phi_{tq}(\mathbf{q}^i, t_i) \dot{\mathbf{q}}^i - \Phi_{tt}(\mathbf{q}^i, t_i) \\ &= -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) - 2\Phi_{tq}(\mathbf{q}, t) \dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \end{aligned} \quad (3.8.17)$$

As in velocity analysis, no iterative computation is required, since the equations are linear in accelerations.

### 3.8.5 Criteria for Singular Behavior

A time grid is normally defined with  $t_{i+1} = t_i + h$ ,  $i = 0, 1, 2, \dots$ , where  $h$  is the *time step*, generally a small positive number. Given the numerical solution  $\mathbf{q}^i$ ,  $\dot{\mathbf{q}}^i$ , and  $\ddot{\mathbf{q}}^i$  at  $t_i$ , an approximate solution at  $t_{i+1}$  is estimated as

$$\mathbf{q}(t_{i+1}) \approx \mathbf{q}^i + h\dot{\mathbf{q}}^i + \frac{1}{2}h^2\ddot{\mathbf{q}}^i \quad (3.8.18)$$

and Newton iteration for  $\mathbf{q}^{i+1}$  is carried out using Eq. (3.8.13). In this way, approximate solutions for  $\mathbf{q}$  are found at each specified time step, as long as the constraint Jacobian is nonsingular. Manifold theoretic results of section 3.6, illustrated in Fig. 3.6.2, show that this will be the case until the boundary of the regular constraint manifold is approached.

As noted in Section 3.7, a *trial design* may have singular configurations. Such configurations may be identified during the foregoing kinematic analysis on the time grid  $t_0, t_1, t_2, \dots$ . The most obvious indicator of an impending *singular configuration* is a precipitous increase in  $\|\dot{\mathbf{q}}^i\|$  and  $\|\ddot{\mathbf{q}}^i\|$ , especially the latter. Another indicator is an increase in the number of iterations required for convergence in Eq. (3.8.13). A third criteria is an increase in the *condition number*, defined in Section 2.2.7, of the Jacobian  $\Phi_q(\mathbf{q}^i, t_i)$  that is factored in solution of the first of Eqs. (3.8.13).

For holonomically constrained systems, time dependent drivers are defined to create the same number of constraint equations as generalized coordinates. Kinematic analysis is then carried out, using iterative methods to solve nonlinear constraint equations for an assembled configuration and for kinematically admissible configurations on a specified time grid. Linear velocity and acceleration equations are solved on the same time grid for dynamic response. Criteria for singular behavior are monitored, including the condition number of the Jacobian, rapid changes in velocity and acceleration, and divergence in position analysis.

### Key Formulas

$$\Phi_q(q^i, t_i) \Delta q^i = -\Phi(q^i, t_i); \quad q^{i+} = q^i + \Delta q^i \quad (3.8.13)$$

$$\Phi_q(q^i, t_i) \dot{q}^i = -\Phi_t(q^i, t_i) \equiv v(q^i, t_i) \quad (3.8.16)$$

$$\Phi_q(q^i, t_i) \ddot{q}^i = -(\Phi_q(q^i, t_i) \dot{q}^i)_q \dot{q}^i - 2\Phi_{tq}(q^i, t_i) \dot{q}^i - \Phi_{tt}(q^i, t_i) \equiv -\gamma(q^i, \dot{q}^i, t_i) \quad (3.8.17)$$

### 3.9 Code 3.9 for Planar System Kinematic Analysis

The general-purpose MATLAB computer Code 3.9 of Appendix 3.A implements the *kinematic analysis* formulation of Section 3.8 for planar multibody systems. Revolute, translational, and distance kinematic constraints and rotational and relative distance drivers of Section 3.2 are included, with functions and derivatives presented in Section 3.2 that are used to implement position, velocity, and acceleration analysis.

Following an explanation of Code 3.9 in this section, numerical examples are presented in Section 3.10, including those treated with detailed derivations in Sections 3.2. The essence of *computational kinematics* is to enable *computer formulation and solution* of the kinematic equations, without the painful detail of ad-hoc derivation of equations of motion and explicit coding of numerical solution algorithms. The computer implementation presented in this section and applied in Section 3.10 is intended to introduce the reader to methods that are available in commercial kinematic and dynamic simulation software and provide a tool for analysis of mechanisms of the reader's choice.

Components of Code 3.9 that interface with the user are presented in Section 3.9.1, followed by an outline of the body of the code in Section 3.9.2, with which the user need not interact.

#### 3.9.1 User Components of Code

The initial segment of code involves definition of numerical solution parameters that underlie the kinematic analysis formulation and associated numerical solution methods presented in prior sections. Data included in the following discussion are for the two-body slider-crank mechanism model of Fig. 3.2.6. Lines 3 to 7 of Fig. 3.9.1 define parameters that are used to define time step and error control in the solution. Application data sets in the AppData function of Fig. 3.9.2 are defined in lines 9 through 13 and selected for analysis in line 14.

1 %AA Planar Kinematic Analysis

```
2 %Error Control Parameters
3 qtol=10^-6; %Tolerance in solving for q
4 maxiter=25; %Maximum number of iterations in Newton-Raphson iteration
5 maxCond=10^4; %Maximum condition number for Jacobian
6 h=0.001; %Time step
7 tfinal=2*pi; %Final simulation time
8 %Application Data
9 %app=1, Slider-Crank, crank angle driver
10 %app=2, Quick Return
11 %app=3, Windshield Wiper
12 %app=4 Double Slider-Crank
13 %app=5 Triple Slider-Crank
14 app=1; %Select application for analysis
```

Figure 3.9.1 Analysis Parameters

The second component of user entered data is the *AppData function* that is shown for the slider-crank in Fig. 3.9.2. The AppData function defines data that are entered for each application, which are passed to the main code and functions executed in calls throughout the code. Lines 10 through 14 define the dimension of variables that characterize the model. If these data are entered incorrectly, the code will fail in ways that are difficult to understand from MATLAB error messages.

The *Planar Joint Data Table* (PJDT) is defined in lines 16 through 24 of Fig. 3.9.2. Table data for revolute, translational, and distance constraints and rotational and relative distance drivers are defined in lines 17 through 20, with explanation expanded below. Line 17 defines elements of data in each column of the table, with the function of each element defined in Lines 18 through 20, repeated as follows for explanation:

16 %PJDT(12,nh): Planar Joint Data Table (First nh joints not time dependent)

17 %PJDT(:,k)=[T;i;j;si;sj;dr;vi;vj]

18 %k=joint No., T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD), i&j=bodies connected,

19 %si;sj=vectors to Pi&Pj in joint definition frame, dr=dist.,

20 %vi;vj=vectors along translational axis

Time dependent drivers must occupy the last columns of the PJDT. Elements of data not required for a joint are entered as zeroes.

To be more specific, a column of the PJDT that defines joint k as a *revolute joint* between bodies i and j shown in Fig. 3.2.2, repeated here for clarity, is as follows:

$$\text{PJDT}(:,k)=[1;i;j;\mathbf{s}'_i;\mathbf{s}'_j;0;\text{zer};\text{zer}]$$

	<b>Fig. 3.2.2 Revolute Joint</b> Elements of data not required for the revolute joint are set to zero, where $\text{zer} = [0 \ 0]^T$ . The user need enter only values k, i, j, $\mathbf{s}'_i$ , and $\mathbf{s}'_j$ in this template.
--	---

A column of the PJDT that defines joint k as a *translational joint* between bodies i and j shown in Fig. 3.2.3, repeated here for clarity, is as follows:

$$\text{PJDT}(:,k)=[2;i;j;\mathbf{s}'_i;\mathbf{s}'_j;0;\mathbf{v}'_i;\mathbf{v}'_j]$$

	<b>Fig. 3.2.3 Translational Joint</b> The user need enter only values k, i, j, $\mathbf{s}'_i$ , $\mathbf{s}'_j$ , $\mathbf{v}'_i$ , and $\mathbf{v}'_j$ in this template. While not required, it is recommended that $\mathbf{v}'_i$ and $\mathbf{v}'_j$ be unit vectors.
--	---

A column of the PJDT that defines joint k as a *distance constraint* between bodies i and j shown in Fig. 3.2.1, repeated here for clarity, is as follows:

$$\text{PJDT}(:,k)=[3;i;j;\mathbf{s}'_i;\mathbf{s}'_j;d;\text{zer};\text{zer}]$$

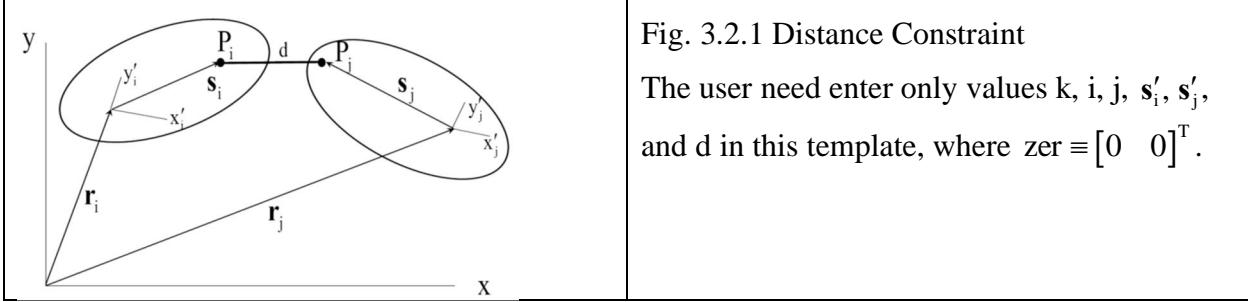


Fig. 3.2.1 Distance Constraint

The user need enter only values  $k$ ,  $i$ ,  $j$ ,  $\mathbf{s}'_i$ ,  $\mathbf{s}'_j$ , and  $d$  in this template, where  $\text{zer} \equiv [0 \ 0]^T$ .

A column of the PJDT that defines joint  $k$  as a *relative angle driver* between bodies  $i$  and  $j$  that are connected by a revolute joint, which defines the constraint equation Eq. (3.2.20),

$$\Phi^{\text{rotD}} \equiv \phi_j - \phi_i - b(t) = 0, \text{ is as follows:}$$

$$\text{PJDT}(:,k)=[4;i;j;\text{zer};\text{zer};0;\text{zer};\text{zer}], \text{ where } \text{zer} \equiv [0 \ 0]^T.$$

The user need enter only  $k$ ,  $i$ , and  $j$  in this template. The time dependent term  $-b(t)$  and its first and second time derivatives are entered in function P5.

A column of the PJDT that defines joint  $k$  as a *relative distance driver* between bodies  $i$  and  $j$  that are connected by a revolute joint, which defines the constraint equation in Eq. (3.2.17),

$$\Phi^{\text{distD}} = \left( \mathbf{d}_{ij}^T \mathbf{d}_{ij} - d(t)^2 \right) / 2 = 0, \text{ is as follows:}$$

$$\text{PJDT}(:,k)=[5;i;j;\mathbf{s}'_i;\mathbf{s}'_j;0;\text{zer};\text{zer}], \text{ where } \text{zer} \equiv [0 \ 0]^T.$$

The user need enter only  $k$ ,  $i$ ,  $j$ ;  $\mathbf{s}'_i$ , and  $\mathbf{s}'_j$  in this template. The time dependent term  $-d(t)^2 / 2$  and its first and second time derivatives are entered in function P5.

It is **critical** that driving constraints be the last entered into the PJDT.

Finally, initial generalized coordinate estimates are defined in line 26.

$$\mathbf{q0e}=[\mathbf{r}_1^0; \phi_1^0; \dots; \mathbf{r}_{nb}^0; \phi_{nb}^0]$$

Data for the slider-crank of Eq. (3.2.33) in Section 3.2.3.3 are entered in lines 21 through 24 of Fig. 3.9.2.

```

8 if app==1      %Slider-Crank
10 nb=2;        %Number of bodies
11 ngc=3*nb;    %number of generalized coordinates
12 nh=3;        %Number of holonomic constraints
13 nhc=5;       %Number of constraint equations
14 nd=ngc-nhc; %Number of drivers
16 %PJDT(12,nh): Planar Joint Data Table (First nh joints not time dependent)
17 %PJDT(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
18 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD), i&j=bodies connected,
19 %sipr&sjpr=vectors to Pi&Pj in joint definition frame, d=dist.,
20 %vipr&vjpr=vectors along translational axis
21 PJDT(:,1)=[1;1;0;zer;zer;0;zer;zer];      %Revolute-crank to ground
22 PJDT(:,2)=[2;2;0;zer;zer;0;ux;ux];        %Trans.-slider2 to ground
23 PJDT(:,3)=[3;1;2;ux;zer;1.01;zer;zer];   %Dist.-crank to slider2

```

```

24 PJDT(:,4)=[4;1;0;zer;zer;0;zer;zer];      %RelRotDrver-crank to ground
25 q0e=[0;0;0;2.2*ux;0];      %Initial generalized coordinate estimate

```

Figure 3.9.2 AppData Function

The third component of user entered code is the P5 function of Fig. 3.9.3 that defines *time dependent driver* data, illustrated by the slider-crank. For each application, the position driver vector Pf that is the of time dependent input over any constant value entered in the AppData Function, as a function of  $t_n$ , is entered on line 9. Its first and second time derivatives are entered on lines 11 and 13, respectively. These data are transferred by the code to the P5 vector in lines 10, 12, and 14, for use throughout the code.

```

7 if app==1      %Slider-Crank
8 omega=1;
9 PDf=[omega*tn];      %Enter nd driver functions of tn minus any constant value entered in AppData
10 Pf=zeros(nhc,1);PDf; %Time dependent driver increment vector
11 PDfd=[omega];      %Enter nd first derivatives of driver functions of tn
12 Pfst=zeros(nhc,1);PDfd; %Time dependent driver increment velocity vector
13 PDFdd=[0];      %Enter nd second derivatives of driver functions of tn
14 Pfstt=zeros(nhc,1);PDFdd; %Time dependent driver increment acceleration vector
15 end

```

Figure 3.9.3 Time Dependent Driver Data in Function P5

### 3.9.2 Computational Components of Code

Computational flow in the main program, which requires little or no input from the user, is presented in Fig. 3.9.4. Data for initiation of kinematic analysis are defined in lines 31 through 37. Iterative solution of position constraint equations and drivers, using the Newton-Raphson method, is implemented in lines 41 through 66. Checks on condition number of the constraint Jacobian and the number of iterations compared to the upper limit allowed are initiated in lines 52 and 60, respectively. If either test fails, the analysis is terminated with a warning. If the user wishes to continue the analysis, limits maxCond and maxiter may be increased. Once the configuration is known, velocity is calculated in lines 68 through 72. With configuration and velocity known, acceleration is calculated in lines 74 through 78. Finally, output for each application is defined, as for the slider-crank in lines 82 through 89. Code that define desired output data are entered by the user for each application.

```

26 %Kinematic Analysis
27 n=1; %Index n
28 tn=0;
29 t(1)=0; %Initial time
30 while tn<tfinal %Loop through line 128
31 t(n)=tn;
32 %Evaluate time dependent terms in driver
33 [Pf,Pfst,Pfstt]=P5Eval(tn,par);
34 %Evaluate q
35 %q-estimate
36 if n==1
37 q=q0e;
38 else
39 q=Q(:,n-1);

```

```

40 end
41 if n-1>1
42 q=q+h*Qd(:,n-1);
43 end
44 i=1; %Position iteration counter
45 err=qtol+1;
46 while err >qtol %Iteration for q, through line 65
47 Phi=PhiEval(q,PJDT,par);
48 Phiq=PhiqEval(q,PJDT,par);
49 PhiqCond=cond(Phiq);
50 if PhiqCond>maxCond %Check for ill conditioned constraint Jacobian
51 fprintf('Warning: Constraint Jacobian Ill Conditioned |n')
52 break
53 end
54 delq=-Phiq\(\Phii+Pf); %Newton-Raphson increment
55 q=q+delq;
56 err=norm(Phi+Pf);
57 if i==1
58 Phi1norm(n)=norm(Phi);
59 err1(n)=err;
60 end
61 i=i+1;
62 if i>maxiter %Check for failure to converge in Newton-Raphson
63 fprintf('Warning: Newton-Raphson convergende failure')
64 break
65 end
66 end
67 iter(n)=i-1; %Report number of iterations
68 Q(:,n)=q; %Record q
69 %Evaluate qd
70 Phiq=PhiqEval(q,PJDT,par);
71 CondPhiq(n)=cond(Phiq); %Record condition number of Jacobian
72 qd=-Phiq\Pfst; %Solution of velocity equation
73 Qd(:,n)=qd; % Record qd
74 % Evaluate qdd
75 P2=P2Eval(q,qd,PJDT,par);
76 Gam=P2*qd;
77 qdd=-Phiq\(\Gam+pfstt); %Solution of acceleration equation
78 Qdd(:,n)=qdd; %record qdd
79 %Calculate output data of interest (Enter for each application)
80 if app==1 %Slider-Crank, crank rel rot driver
81 phi1(n)=q(3);
82 phi1d(n)=qd(3);
83 phi1dd(n)=qdd(3);
84 x2(n)=q(4);
85 x2d(n)=qd(4);
86 x2dd(n)=qdd(4);
87 end

```

Figure 3.9.4 Main Code Computational Flow

Computing functions that support the main code of Fig. 3.9.4 are identified in Fig. 3.9.5. Computing functions include the *Add function* that adds nonzero submatrices to sparse matrices, below and to the right of the address of the (1,1) term in the submatrix added to the underlying

matrix that was initialized to zeroes. ATran evaluates the orientation transformation matrix. The user input function AppData is as presented in Fig. 3.9.2. Vector partition functions parPart and qPart support partitioning of vectors involved into components that are used in kinematic computations.

#### Computing Functions

Add

ATran

#### User Input Function

AppData

#### Vector Partition Functions

parPart

qPart

#### Constraint and Driver Partition Functions

DistPart

RevPart

TranPart

RelRotPart

#### Constraint and Constraint Derivative Evaluation Functions

PhiEval

PhiqEval

P2Eval

P5Eval

Figure 3.9.5 Computing Functions

The Constraint and Constraint Derivative Evaluation Functions and Constraint and Driver Partition Functions shown in Fig. 3.9.5 are at the heart of kinematic and dynamic simulation. Even though the user need not interact with these functions, it is important to understand how they work.

The Constraint and Constraint Derivative Evaluation Functions listed in Fig. 3.9.5 evaluate kinematic constraint expressions and their derivatives that are defined in Section 3.2 and are used in implementing the main code of Fig 3.9.4. As an illustration, the PhiEval function is presented in Fig. 3.9.6. The vector Phi of constraint functions is initially set to zero in line 5. The index k introduced in line 11 is varied over the range of holonomic and driving constraints in the system, nh+nd. For a value of k, line 13 determines if the associate constraint is a revolute joint, as defined in the AppData function and represented in the Planar Joint Data Table (PJDT). If so, line 14 calls the associated Constraint and Data Table Partition Function of Fig 3.9.7 and enters i and j as the bodies connected and joint data  $s_i^{tp}$  and  $s_j^{tp}$ . Lines 15 through 22 define body generalized coordinates and orientation transformation matrices.

Care is taken to account for the fact that ground is designated by index  $j = 0$  and its generalized coordinates are constant, yielding no contribution in derivative evaluation functions, but including geometric quantities that define the constraint of body i with ground. The revolute joint constraint expression of Eq. (3.2.8) is evaluated in line 23. The Add function is then used in line 24 to insert the revolute joint constraint value into the proper location in the vector Phi and the address parameter m is indexed by the number of constraint expressions added in line 25. The process is repeated in lines 28, 44, and 60 to determine if k represents a translational constraint, distance constraint, or relative rotation driver. Associated time independent constraint

or driver values are entered in the constraint vector Phi. Index k is increased by one in line 73 and the process is repeated until all constraints and drivers are accounted for. If there are drivers, the associated time dependent terms are entered in the P5 function.

```

1 function Phi=PhiEval(q,PJDT,par)
2 % par is short for the parameters assigned in AA_Planar.. code which takes
3 % variables assigned in AppData to evaluate constraint vector
4 [nb,ngc,nh,nhc,nd,qtol,appl]=parPart(par);
5 % nb = Number of bodies; ngc = number of generalized coordinates; nh =
6 % Number of holonomic constraints; nhc = Number of constraint equations;
7 % nd = Number of drivers; qtol = tolerance assigned in AA_Planar...; app =
8 % application choice made in APData
9 Phi=zeros(ngc,1); %Set system constraint vector to zero
10 I2=eye(2);
11 P=[0,-1;1,0];
12 k=1;
13 m=0;
14 while k<=nh+nd %Cycle through nh holonomic and nd driver constraints
15 % Initial steps of pulling data from the PJDT related specifically to
16 % the constraint at each joint and assigning body 1(i) with r1 (2,1)
17 % position matrix and ph1 angle, then set body 2(j) as ground as that
18 % position will always be at origin and angle 0. If j is not ground,
19 % then the position of the second body joint is pulled from the qPart
20 % function
21 if PJDT(1,k)==1 %Revolute constraint
22 [i,j,s1pr,s2pr]=RevPart(k,PJDT);
23 [r1,ph1]=qPart(q,i);
24 r2=[0;0];
25 ph2=0;
26 if j>=1 %If body j is not ground
27 [r2,ph2]=qPart(q,j);
28 end
29 A1=ATran(ph1);
30 A2=ATran(ph2);
31 PhiRev=r2+A2*s2pr-r1-A1*s1pr; %Evaluate revolute constraint
32 Phi=Add(Phi,PhiRev,m,0); %Add revolute contribution to Phi
33 m=m+2;
34 end
35 if PJDT(1,k)==2 %Translational constraint
36 [i,j,s1pr,s2pr,v1pr,v2pr]=TranPart(k,PJDT);
37 [r1,ph1]=qPart(q,i);
38 r2=[0;0];
39 ph2=0;
40 if j>=1 %If body j is not ground
41 [r2,ph2]=qPart(q,j);
42 end
43 A1=ATran(ph1);
44 A2=ATran(ph2);
45 d12=r2+A2*s2pr-r1-A1*s1pr;
46 PhiTran=-[v1pr'*P*A1'*d12;v1pr'*P*A1'*A2*v2pr];
47 Phi=Add(Phi,PhiTran,m,0);
48 m=m+2;
49 end

```

```

50 if PJDT(1,k)==3 %Distance constraint
51 [i,j,s1pr,s2pr,d]=DistPart(k,PJDT);
52 [r1,ph1]=qPart(q,i);
53 r2=[0;0];
54 ph2=0;
55 if j>=1 %If body j is not ground
56 [r2,ph2]=qPart(q,j);
57 end
58 A1=ATran(ph1);
59 A2=ATran(ph2);
60 d12=r2+A2*s2pr-r1-A1*s1pr;
61 PhiDist=(d12'*d12-d^2)/2;
62 Phi=Add(Phi,PhiDist,m,0);
63 m=m+1;
64 end
65 if PJDT(1,k)==4 %Relative rotation driver
66 [i,j]=RelRotPart(k,PJDT);
67 [r1,ph1]=qPart(q,i);
68 r2=[0;0];
69 ph2=0;
70 if j>=1
71 [r2,ph2]=qPart(q,j);
72 end
73 PhiRelRot=ph2-ph1;
74 Phi=Add(Phi,PhiRelRot,m,0);
75 m=m+1;
76 end
77 if PJDT(1,k)==5 %Relative Distance driver
78 [i,j,s1pr,s2pr]=RelDistPart(k,PJDT);
79 [r1,ph1]=qPart(q,i);
80 r2=[0;0];
81 ph2=0;
82 if j>=1 %If body j is not ground
83 [r2,ph2]=qPart(q,j);
84 end
85 A1=ATran(ph1);
86 A2=ATran(ph2);
87 d12=r2+A2*s2pr-r1-A1*s1pr;
88 PhiDist=(d12'*d12)/2;
89 Phi=Add(Phi,PhiDist,m,0);
90 m=m+1;
91 end
92 k=k+1;
93 end

```

Figure 3.9.6 Constraint Evaluation Function, PhiEval

The foregoing process is repeated in the PhiqEval, P2Eval, and P5Eval functions. The sequence of calculation is identical for each, with the function Phi replaced by Phiq, P2, and PDf.

Finally, the Constraint and Driver Partition Functions provide access to elements of the Planar Joint Data Table, PJDT, that is entered in the AppData function to control computation, as illustrated in Fig. 3.9.6. To illustrate use of these functions, the RevPart function that appears in line 14 of Fig. 3.9.6 is presented in Fig. 3.9.7. Once line 13 of Fig 3.9.6 determines that the value

$k$  designates a revolute joint, the function of Fig. 3.9.7 is called in line 14 to provide data needed on bodies connected and connection data that are defined in the Planar Joint Data Table of the AppData function. Comparable functions DistPart, TranPart, and RelRotPart define data for the associated constraints and drivers.

```

1 function [i,j,s1pr,s2pr]=RevPart(k,PJDT)
2 i=PJDT(2,k);
3 j=PJDT(3,k);
4 s1pr=[PJDT(4,k);PJDT(5,k)];
5 s2pr=[PJDT(6,k);PJDT(7,k)];
6 end

```

Figure 3.9.7 Constraint Partition Function, RevPart

### 3.9.3 Code Output

In addition to output defined for each application in the AppData function, the code reports the following arrays that shed light on the application and performance of the code:

CondPhi $q$ : the condition number of the Jacobian  $\Phi_q$  at each time step  
 iter; the number of Newton-Raphson iterations required at each time step  
 $Q$ ; the array of values of  $q$  at each time step  
 $Qd$ ; the array of values of  $\dot{q}$  at each time step  
 $Qdd$ ; the array of values of  $\ddot{q}$  at each time step

MATLAB Code 3.9 of Appendix 5.A is a tool for numerical simulation and experimentation with the kinematic analysis of planar multibody systems. It is typical of code that enables the engineer to create models of mechanical systems, carry out simulations, and investigate the effects of model and computational variations, without investing extensive time and frustration in ad-hoc derivation and coding of specific applications. The structure of the kinematic analysis code is identical to more extensive codes for system dynamics. In fact, it provides the kinematic underpinning of those codes.

### 3.10 Planar System Kinematic Analysis Using Code 3.9

The equations derived in Section 3.2 and implemented in Code 3.9 of Section 3.9 form the foundation for *computer aided kinematic analysis* of planar systems. A slider-crank mechanism that is subject to near singular conditions, a double slider-crank, a moderate dimensional quick return mechanism, and a windshield wiper mechanism are studied in the following subsections.

#### 3.10.1 Planar Slider-Crank Mechanisms

Two planar slider-crank mechanisms are analyzed using Code 3.9. The first is the conventional mechanism of Section 3.2.3.3 and the second is a double slider-crank.

##### 3.10.1.1 Two Body Slider-Crank

The *two-body slider-crank* model of Section 3.2.3.3, shown in Fig. 3.10.1, is analyzed using Code 3.9 of Appendix 3.A, with data presented in Eq. (3.2.33) and the AppData function of Fig. 3.10.2. The angle driver that defines constant angular velocity of the crank is  $\dot{\phi}_0 - \dot{\phi}_1 + \omega \times t \equiv \dot{\phi}_0 - \dot{\phi}_1 + PDf(t) = 0$ , with  $PDf(t)$  and its derivatives entered in function P5Eval of Fig. 3.10.3. To study the effect of varying crank angular velocity, crank radius = 1 m and variable connecting rod length, initially  $a = 1.2$  m. Plots of position, velocity, and acceleration ( $x_2$ ,  $x_{2d}$ , and  $x_{2dd}$ ) of the slider for crank angular velocities  $\omega = 1, 2$ , and  $4$  rad/sec are presented in Fig. 3.10.4, over the time interval  $[0 \quad 2\pi]$ ; i.e., 1, 2, and 4 revolutions of the crank. As expected, velocity varies linearly with input angular velocity and acceleration varies with the square of velocity.

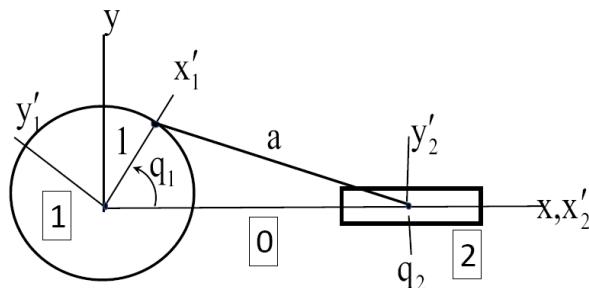


Figure 3.10.1 Two Body Slider-Crank Mechanism

```

8 if app==1 %Slider-Crank, crank rel rot driver
9 nb=2; %Number of bodies
10 ngc=3*nb; %number of generalized coordinates
11 nh=3; %Number of holonomic constraints
12 nhc=5; %Number of constraint equations
13 nd=ngc-nhc; %Number of drivers
14 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
15 %PJDT(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
16 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD), i&j=bodies connected,
17 %sipr&sjpr=vectors to Pi&Pj in joint definition, d=dist.,
18 %vipr&vjpr=vectors along translational axis
19 PJDT(:,1)=[1;1;0;zer;zer;0;zer;zer]; %Revolute-crank to ground
20 PJDT(:,2)=[2;2;0;zer;zer;0;ux;ux]; %Trans.-slider2 to ground

```

```

21 PJDT(:,3)=[3;1;2;ux;zer;1.1;zer;zer]; %Dist.-crank to slider2
22 PJDT(:,4)=[4;1;0;zer;zer;0;zer;zer]; %RelRotDrver-crank to ground
23 q0e=[0;0;0;1.5*ux;0]; %Initial generalized coordinate estimate
24 end

```

Figure 3.10.2 AppData Function for Two Body Slider-Crank

```

6 if app==1 %Slider-Crank, crank rel rot driver
7 omega=4;
8 PDf=[omega*tn]; %Enter nd driver functions of tn
9 Pf=[zeros(nhc,1);PDf]; %Time dependent driver vector
10 PDfd=[omega]; %Enter nd first derivatives of driver functions of tn
11 Pfst=[zeros(nhc,1);PDfd]; %Time dependent driver velocity vector
12 PDfdd=[0]; %Enter nd second derivatives of driver functions of tn
13 Pfstt=[zeros(nhc,1);PDfdd]; %Time dependent driver acceleration vector
14 end

```

Figure 3.10.3 P5Eval Function for Two Body Slider-Crank

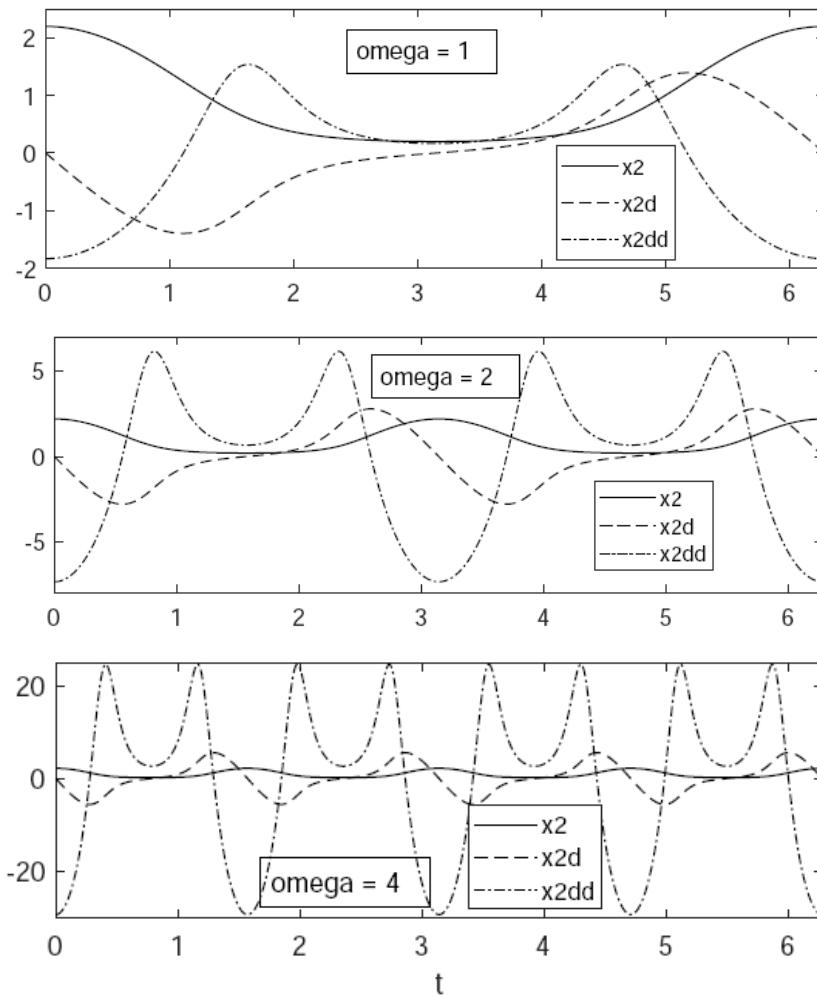


Figure 3.10.4 Position, Velocity, and Acceleration of Slider,  $a = 1.2 \text{ m}$

To study the influence of variable connecting rod length, the crank is driven at  $\omega_{\text{crank}} = 4 \text{ rad/sec}$  and connecting rod lengths are selected as  $a = 1.1$  and  $1.01 \text{ m}$ . Plots of position, velocity, and acceleration of the slider are presented in Fig. 3.10.5. Since a connecting rod length of  $1 \text{ m}$  is a singular configuration, it is not surprising that accelerations, hence constraint reaction forces, become extreme with a connecting rod length of  $1.01 \text{ m}$ .

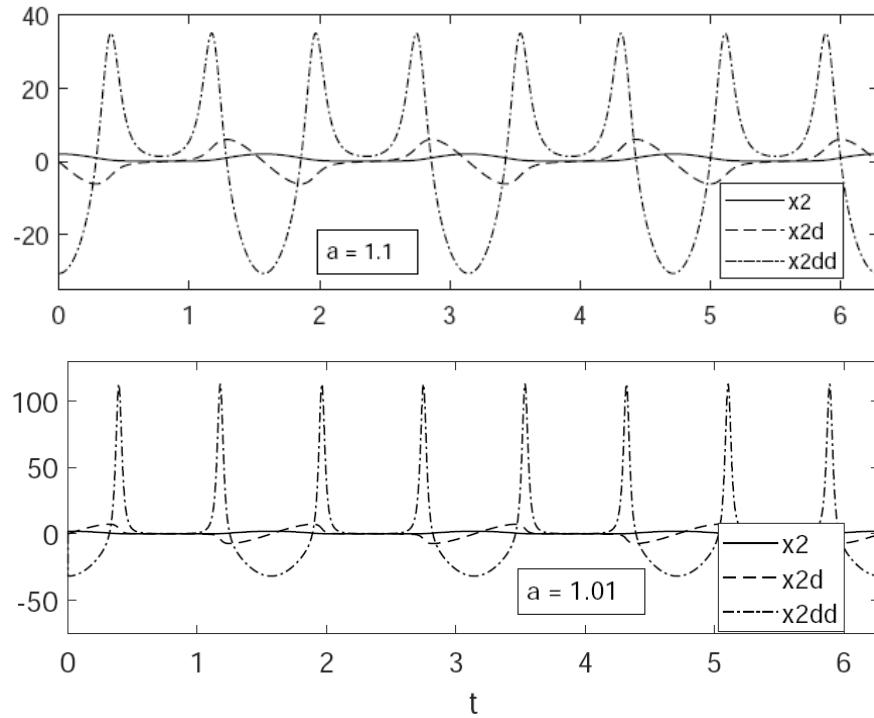


Figure 3.10.5 Near Singular Behavior of Slider-Crank,  $\omega_{\text{crank}} = 4 \text{ rad/sec}$

### 3.10.1.2 Three Body Double Slider-Crank

The two degree of freedom *three-body double slider-crank* of Fig. 3.10.6 is driven by a rotation driver between bodies 1 and 2 and a distance driver between ground and body 1. Data for the mechanism are presented in the *AppData Function* of Fig. 3.10.7, where  $a = 1.5$  is assigned. The rotation driver is defined in the P5 Function of Fig. 3.10.8, where a constant angular velocity of body 2 relative to body 1 is specified as  $\omega = 10 \text{ rad/sec}$ ; i.e.,  $\phi_2 - \phi_1 = \omega \times t$ , or

$\phi_2 - \phi_1 - \omega \times t \equiv \phi_2 - \phi_1 + PDf_1(t) = 0$  and the distance driver between point  $-\mathbf{u}_x$  in ground and the origin of body 1 is specified as  $d(t) = 1 + 0.5 \sin(t)$ , so  $PDf_2(t) = -0.5(1 + 0.5 \sin(t))^2$ .

These time dependent drivers are entered in the P5Eval function of Fig. 3.10.8. Numerical results of simulation using Code 3.9 are presented in Fig. 3.10.9.

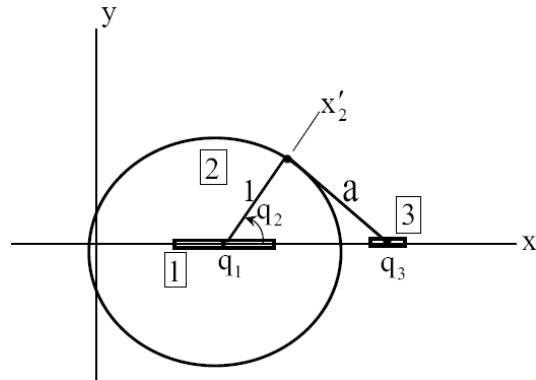


Figure 3.10.6 Three Body Double Slider-Crank

```

74 if app==4 %Double Slider-Crank
75 nb=3;
76 ngc=3*nb;
77 nh=4;
78 nhc=7;
79 nd=ngc-nhc;
80 %PJDT(12,nh); Planar Joint Data Table (First nh not time dependent)
81 %PJTd(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
82 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD), i&j=bodies connected,
83 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis
84 PJDT(:,1)=[1;1;2;zer;zer;0;zer;zer]; %Rev-bod 1 to body 2
85 PJDT(:,2)=[2;1;0;zer;zer;0;ux;ux]; %Tran-bod 1 to ground
86 PJDT(:,3)=[2;3;0;zer;zer;0;ux;ux]; %Tran-bod 3 to ground
87 PJDT(:,4)=[3;2;3;ux;zer;1.5;zer;zer]; %Dist.-bod 2 to bod 3
88 PJDT(:,5)=[4;1;2;zer;zer;0;zer;zer]; %RotD-bod 1 to bod 2
89 PJDT(:,6)=[5;1;0;zer;-1*ux;0;zer;zer]; %DistD-bod 1 to ground
90 %Initial generalized coordinate estimate
91 q10=[0;0;0];
92 q20=[0;0;0];
93 q30=[2.1;0;0];
94 q0e=[q10;q20;q30];
95 end

```

Figure 3.10.7 AppData Function for Double Slider-Crank

```

39 if app==4 %Double Slider-Crank
40 omega=10; %Angular velocity in RotD
41 PDF1=-omega*tn; %Ervier function
42 PDF1d=-omega; %First derivative of driver function
43 PDF1dd=0; %Second derivative of driver function
44 d=1+0.5*sin(tn); %Length of DistD
45 PDF2=-0.5*d^2; %Driver function
46 PDF2d=-0.5*d*cos(tn); %First derivative of driver function
47 PDF2dd=0.5*d*sin(tn)-0.25*cos(tn)^2; %Second derivative of driver function
48 Pf=[zeros(nhc,1);PDF1;PDF2]; %Driver vector
49 Pfst=[zeros(nhc,1);PDF1d;PDF2d]; %Driver velocity vector
50 Pfstt=[zeros(nhc,1);PDF1dd;PDF2dd]; %Driver acceleration vector

```

Figure 3.10.8 P5Eval Function for Double Slider-Crank

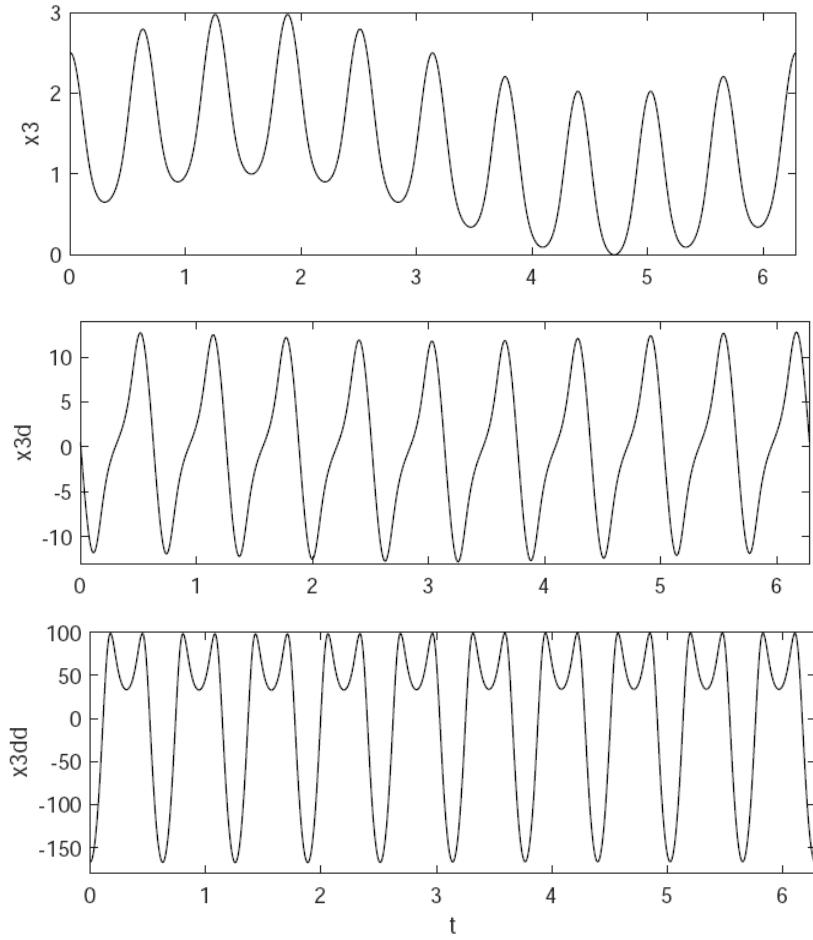


Figure 3.10.9 Slider Position, Velocity, and Acceleration

### 3.10.2 Quick Return Mechanism

The *quick return mechanism* shown in Fig. 3.10.10 is driven by counterclockwise rotation of the flywheel of body 2 to cause oscillation of body 1, via the body 3 key that translates in the slot shown in body 1. The connecting rod between point  $P_1$  on body 1 and the origin of the cutting tool reference frame on body 4 causes body 4 to move slowly to the left in a cutting stroke and to return more quickly to the right for the next cutting stroke. The radius of the flywheel is 1.5 m, the distance between points O and  $P_1$  on body 1 is 4 m, and the length of the connecting rod that is modeled as a distance constraint is 2.5298 m. These data and the initial generalized coordinate estimate for the mechanism are given in the *AppData Function* of Fig. 3.10.11. A rotation driver is applied to the flywheel as  $\phi_0 - \phi_2 + \omega \times t_n \equiv \phi_0 - \phi_2 + PDf(t) = 0$ , with the time dependent driving function and its derivatives entered in the P5Eval function of Fig. 3.10.12.

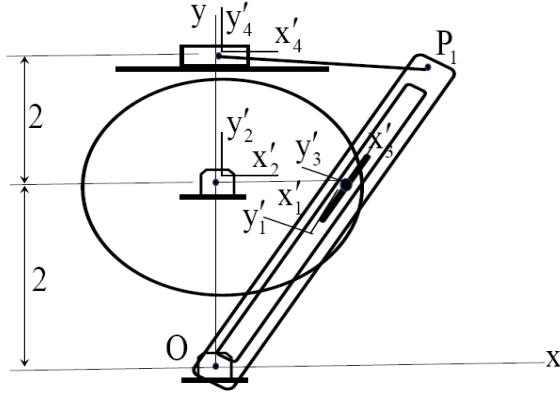


Figure 3.10.10 Quick Return Mechanism

```

26 if app==2 %Quick Return
27 nb=4; %Number of bodies
28 ngc=3*nb; %number of generalized coordinates
29 nh=6; %Number of time independent holonomic constraints
30 nhc=11; %Number of time independent holonomic constraint equations
31 nd=ngc-nhc; %Number of time dependent driving constraint equations
32 %PJDT(12,nh): Planar Joint Data Table (First nh not time dependent)
33 %PJTd(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
34 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD), i&j=bodies conn.,
35 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis
36 PJDT(:,1)=[1;1;0;-2*ux;zer;0;zer;zer]; %Revolute-bar to ground
37 PJDT(:,2)=[1;2;0;zer;2*uy;0;zer;zer]; %Revolute-crank to ground
38 PJDT(:,3)=[1;2;3;1.5*ux;zer;0;zer;zer]; %Revolute-crank to key
39 PJDT(:,4)=[2;1;3;zer;zer;0;ux;ux]; %Trans.-bar to key
40 PJDT(:,5)=[2;4;0;zer;4*uy;0;ux;ux]; %Trans.-cutter to ground
41 PJDT(:,6)=[3;1;4;2*ux;zer;2.5298;zer;zer]; %Dist.-bar to cutter
42 PJDT(:,7)=[4;2;0;zer;2*uy;0;zer;zer]; %RotD-crank to ground
43 %Initial generalized coordinate estimate
44 q10=[1.2;1.6;0.9273];
45 q20=[0;2;0];
46 q30=[1.5;2;0.9273];
47 q40=[0;4;0];
48 q0e=[q10;q20;q30;q40];
49 end

```

Figure 3.10.11 AppData Function for Quick Return Mechanism

```

17 if app==2 %Quick Return
18 omega=4;
19 PDf=[omega*tn]; %Enter driver functions of tn
20 Pf=[zeros(nhc,1);PDf];
21 PDfd=[omega]; %Enter first derivatives of driver functions of tn
22 Pfst=[zeros(nhc,1);PDfd];
23 PDfdd=[0]; %Enter second derivatives of driver functions of tn
24 Pfstt=[zeros(nhc,1);PDfdd];
25 end

```

Figure 3.10.12 P5Eval Function for Quick Return Mechanism

Plots of position, velocity, and acceleration of the slider,  $x4$ ,  $x4d$  and  $x4dd$ , for angular velocity  $\omega = 1, 2$ , and  $4 \text{ rad/sec}$  of the crank are shown in Fig. 3.10.13. The initial slow motion of the cutter and the quick return are clearly shown. As the crank angular velocity increases, cutter velocity increases linearly.

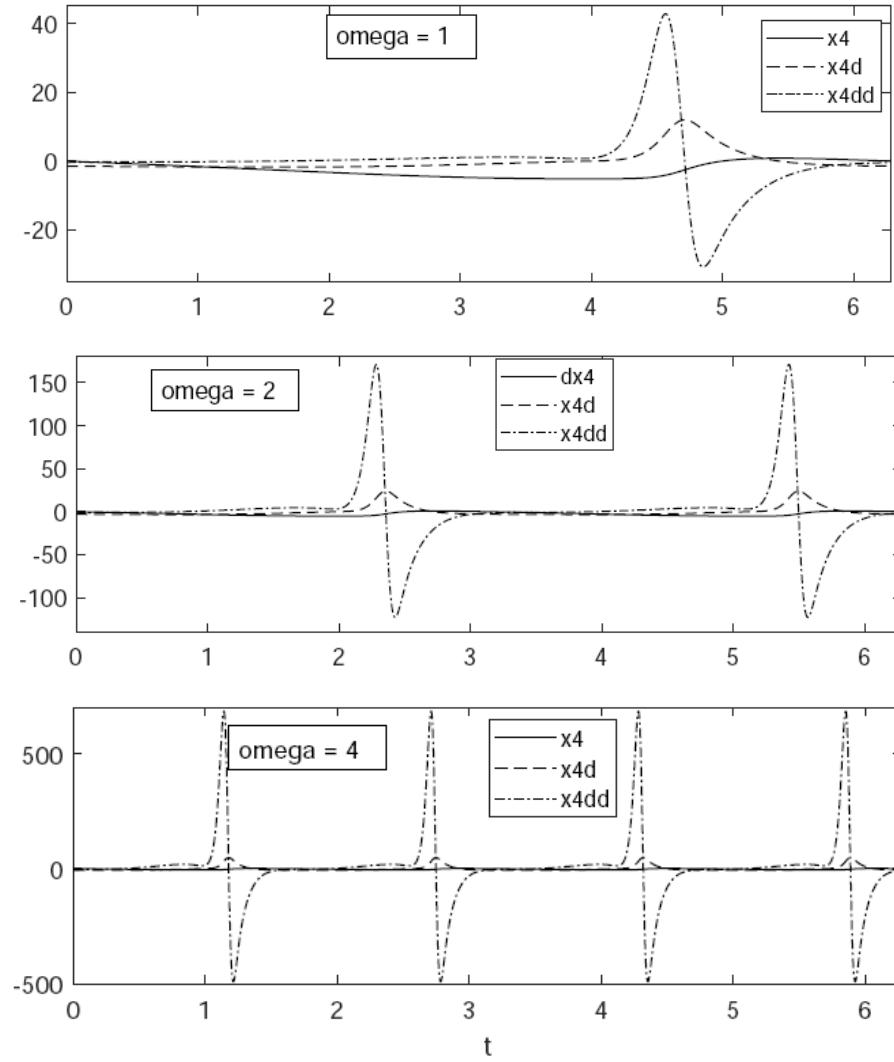


Figure 3.10.13 Cutter Position and Velocity, and Acceleration

### 3.10.3 Windshield Wiper

The *windshield wiper* model of Section 3.2.3.4 shown in Fig. 3.10.14 is analyzed using Code 3.9. Data presented in Eq. (3.2.37) are implemented with a relative rotation driver  $\phi_0 - \phi_1 + \omega \times t \equiv \phi_0 - \phi_1 + \text{PDf}(t) = 0$  and initial generalized coordinate estimates in the *AppData Function* of Fig. 3.10.15. The time dependent driving function and its derivatives are entered in the P5Eval function of Fig 3.10.16.

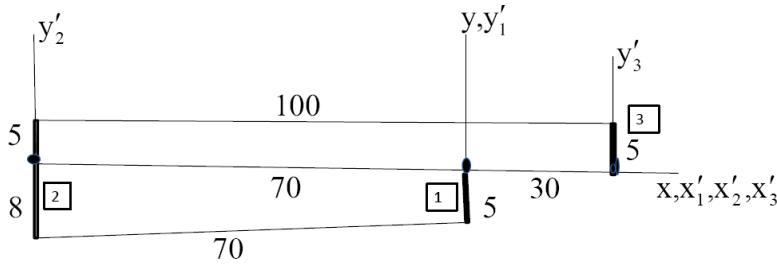


Figure 3.10.14 Windshield Wiper Mechanism

```

51 if app==3 %Windshield Wiper
52 nb=3; %Number of bodies
53 ngc=3*nb; %number of generalized coordinates
54 nh=5; %Number of time independent holonomic constraints
55 nhc=8; %Number of time independent holonomic constraint equations
56 nd=ngc-nhc; %Number of time dependent driving constraint equations
57 %PJDT(12,nh): Planar Joint Data Table (First nh not time dependent)
58 %PJTd(:,k)=[T;i;j;sipr;sjpr;d;vipl;vjpl]; k=joint No.,
59 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD), i&j=bodies connected,
60 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis
61 PJDT(:,1)=[1;1;0;zer;zer;0;zer;zer]; %Rev-bod 1 to ground
62 PJDT(:,2)=[1;2;0;zer;-70*ux;0;zer;zer]; %Rev-bod 2 to ground
63 PJDT(:,3)=[1;3;0;zer;30*ux;0;zer;zer]; %Rev-bod 3 to ground
64 PJDT(:,4)=[3;1;2;-5*uy;-8*uy;70;zer;zer]; %Dist.-bod 1 to bod 2
65 PJDT(:,5)=[3;2;3;5*uy;5*uy;100;zer;zer]; %Dist.-bod 2 to bod 3
66 PJDT(:,6)=[4;1;0;zer;zer;0;zer;zer]; %RotD-bod 1 to ground
67 %Initial generalized coordinate estimate
68 q10=[0;0;0];
69 q20=[-70;0;0];
70 q30=[30;0;0];
71 q0e=[q10;q20;q30];
72 end

```

Figure 3.10.15 AppData Function for Windshield Wiper

```

28 if app==3 %Windshield Wiper
29 omega=2;
30 PDf=[omega*tn]; %Enter driver functions of tn
31 Pf=zeros(nhc,1);PDf;
32 PDfd=[omega]; %Enter first derivatives of driver functions of tn
33 Pfst=zeros(nhc,1);PDfd;
34 PDfdd=[0]; %Enter second derivatives of driver functions of tn
35 Pfstt=zeros(nhc,1);PDfdd;
36 end

```

Figure 3.10.16 P5Eval Function for Windshield Wiper

Plots of left wiper angle, angular velocity, and angular acceleration ( $\phi_2$ ,  $\dot{\phi}_2$ , and  $\ddot{\phi}_2$ ) on the time interval  $[0 \quad 2\pi]$  are presented in Fig. 3.10.17 for input angular velocities  $\omega = 1$  and  $2$  rad/sec. These inputs correspond roughly to a low- and high-speed setting during light and heavy rain, respectively. The nonlinear character of the kinematic equations, in

particular the quadratic dependence of the right side of the kinematic acceleration equations on velocity, is reflected in significantly higher accelerations in the case  $\omega = 2 \text{ rad/sec}$ . Plots of right wiper angle, angular velocity, and angular acceleration are essentially identical, as expected due to symmetry of the coupling between left and right wipers. As with the slider-crank example, velocity varies linearly with input angular velocity and acceleration varies with the square of velocity.

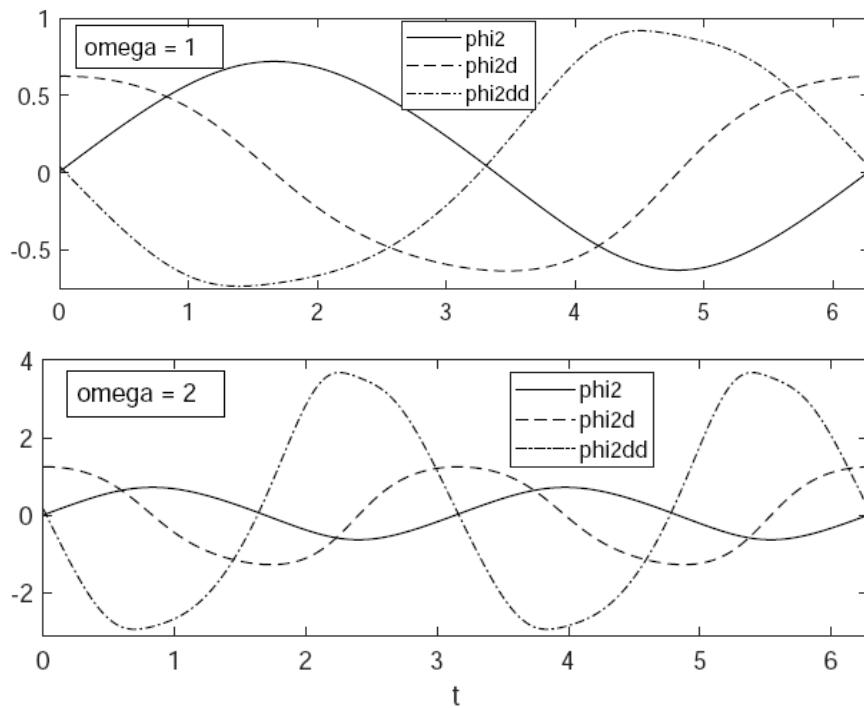


Figure 3.10.17 Left Wiper Angle, Angular Velocity, and Angular Acceleration

For system models with time dependent drivers that create the same number of constraint equations as generalized coordinates, kinematic analysis is carried out using iterative methods to solve nonlinear constraint equations for kinematically admissible configurations on a time grid that is specified for analysis. Linear velocity and acceleration equations are solved on the same time grid for dynamic response. Criteria for singular behavior are monitored, including the condition number of the Constraint Jacobian, rapid changes in acceleration, and loss of existence in position analysis.

Numerical analyses using Code 3.9 of Appendix 3.A are carried out with planar examples. Results confirm the criteria for onset of singular behavior and quadratic dependence of system acceleration on velocities.

### 3.11 Code 3.11 for Spatial System Kinematic Analysis

MATLAB Code 3.11 of Appendix 3.A implements the *kinematic analysis formulation* of Sections 3.3 and 3.8 for spatial multibody systems. Building block distance, spherical, dot1, and dot2 constraints are implemented and serve as the foundation for cylindrical, revolute, translational, distance, universal, strut, and revolute-spherical kinematic constraints of Section 3.3. Equations of Section 3.3 are used to implement position, velocity, and acceleration analysis. Numerical examples using Code 3.11 are presented in Section 3.12.

Components of Code 3.11 that interface with the user are presented in Section 3.11.1, followed by an outline of the body of the code, with which the user need not interact, in Section 3.11.2. The structure of Code 3.11 is identical to that of Code 3.9 presented in Section 3.9. References are made to that code, where appropriate, to avoid unnecessary repetition.

#### 3.11.1 User Components of Code

The initial segment of code involves definition of numerical solution parameters that underlie the kinematic analysis formulation and associated numerical solution. Since it is identical to the code of Fig 3.9.1, the reader is referred to Section 3.9.1 for discussion of its use.

*Application data* are indexed in lines 9 to 14 of Fig. 3.11.1 to specific applications that are defined in the AppData Function presented in the following. The declaration in line 13 defines which application is implemented in the analysis, the spatial slider-crank in this section. Data that are returned from the AppData Function in line 14 and parameters in the par vector of line 14 are used throughout the code to pass data for each application. Data and storage arrays that are used throughout the code are defined in lines 22 through 24.

```
1 %AA Spatial Kinematic Analysis
2 % User Input
3 qtol=10^-6; %Tolerance in solving for q
4 maxiter=25; %Maximum number of iterations in Newton-Raphson iteration
5 maxCond=10^4; %Maximum condition number for Jacobian
6 h=0.001; %Time step
7 tfinal=1;
8 %Application Data
9 %app=1, Spatial 2-Bar with 2 rotational drivers
10 %app=2, 4-Bar with 1 rotational driver
11 %app=3, Spatial Slider-Crank, RotDr Bod1-grnd
12 %app=4, Fly-Ball Governor
13 app=4;
14 [nb,ngc,nh,nhc,nc,nd,SJDT,q0e]=AppData(app); %Data from AppData function
15 par=[nb;ngc;nh;nhc;nd;qtol;app]; %Parameter vector for analysis control
16 global ux uy uz z3 %Make vectors available to all functions
21 % Data Storage Arrays
22 Q=zeros(ngc,10);
23 Qd=zeros(ngc,10);
24 Qdd=zeros(ngc,10);
```

Figure 3.11.1 Analysis Parameters and Data Storage

The third component of user entered code is the *AppData Function* of Fig. 3.11.2 for the spatial slider-crank mechanism of Section 3.3.7.3 in Fig. 3.3.14. Data entered in the AppData function are passed to the main code for functions executed in calls throughout the code. Lines

86 through 91 define dimensions of variables used in the model. If these data are entered incorrectly, the code will fail in ways that are difficult to understand from MATLAB error messages.

The *Spatial Joint Data Table* (SJDT) is defined in lines 92 through 101 of Fig. 3.11.2. Table data for distance, spherical, cylindrical, revolute, translational, universal, strut, and revolute-spherical constraints and distance and rotational drivers are defined in lines 92 through 97, with explanation expanded and repeated here as follows:

```

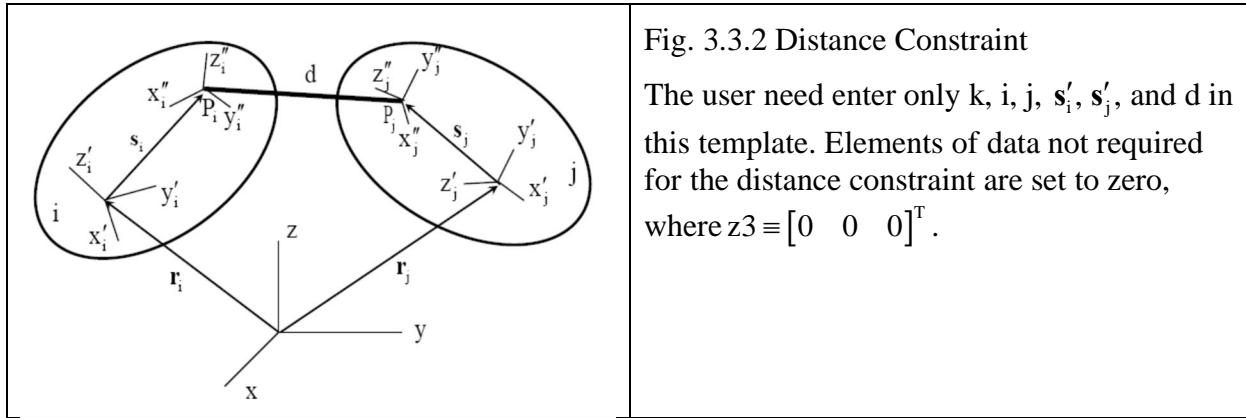
92 %SJDT(22,nh): Spatial Joint Data Table
93 %SJTd(:,k)=[T;i;j;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
94 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
95 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr, 10=RotDr); i&j=bodies conn.,i>0;
96 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=vectors in
97 %joints

```

Line 93 defines elements of data in each column of the table, with the function of each element defined in Lines 94 through 97. Time dependent drivers must occupy the bottom rows of the SJDT. Elements of data not required for a joint are entered as zeroes.

To be more specific, a column of the SJDT that defines joint k as a *distance constraint* between bodies i and j shown in Fig. 3.3.2, repeated here for clarity, is as follows:

$$\text{SJDT}(:,k) = [1; i; j; \mathbf{s}_i'; \mathbf{s}_j'; d; z3; z3; z3]$$



A column of the SJDT that defines joint k as a *spherical constraint* between bodies i and j shown in Fig. 3.3.2, repeated here for clarity, is as follows:

$$SJDT(:,k) = [2;i;j; \mathbf{s}'_i; \mathbf{s}'_j; 0; z3; z3; z3; z3]$$

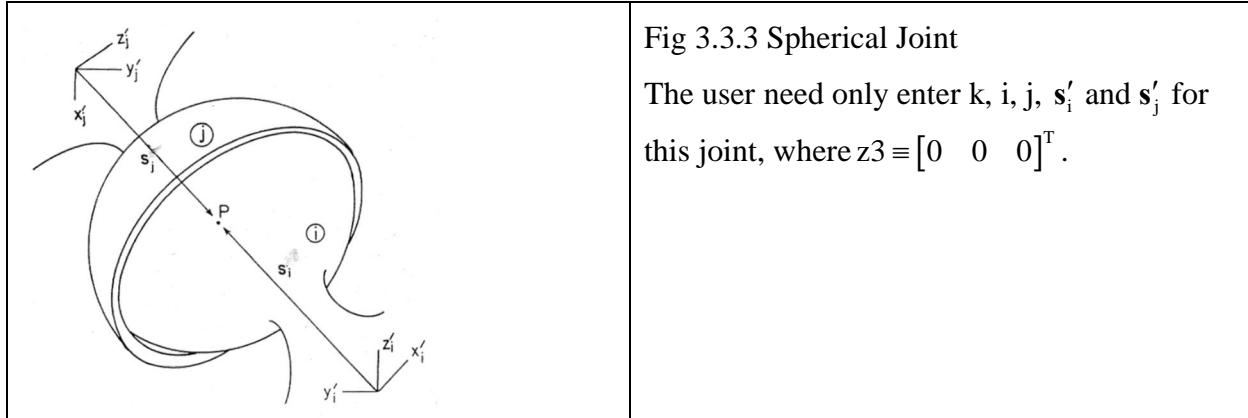


Fig 3.3.3 Spherical Joint

The user need only enter k, i, j,  $\mathbf{s}'_i$  and  $\mathbf{s}'_j$  for this joint, where  $z3 = [0 \ 0 \ 0]^T$ .

Columns of the SJDT that define joint k as a *cylindrical constraint, revolute constraint, or translational constraint*, respectively between bodies i and j shown in Fig. 3.3.4, repeated here for clarity, are as follows:

$$SJDT(:,k) = [3;i;j; \mathbf{s}'_i; \mathbf{s}'_j; 0; \mathbf{u}'^i; \mathbf{w}'^i; \mathbf{u}'^j; \mathbf{w}'^j]$$

$$SJDT(:,k) = [4;i;j; \mathbf{s}'_i; \mathbf{s}'_j; 0; \mathbf{u}'^i; \mathbf{w}'^i; \mathbf{u}'^j; \mathbf{w}'^j]$$

$$SJDT(:,k) = [5;i;j; \mathbf{s}'_i; \mathbf{s}'_j; 0; \mathbf{u}'^i; \mathbf{w}'^i; \mathbf{u}'^j; \mathbf{w}'^j]$$

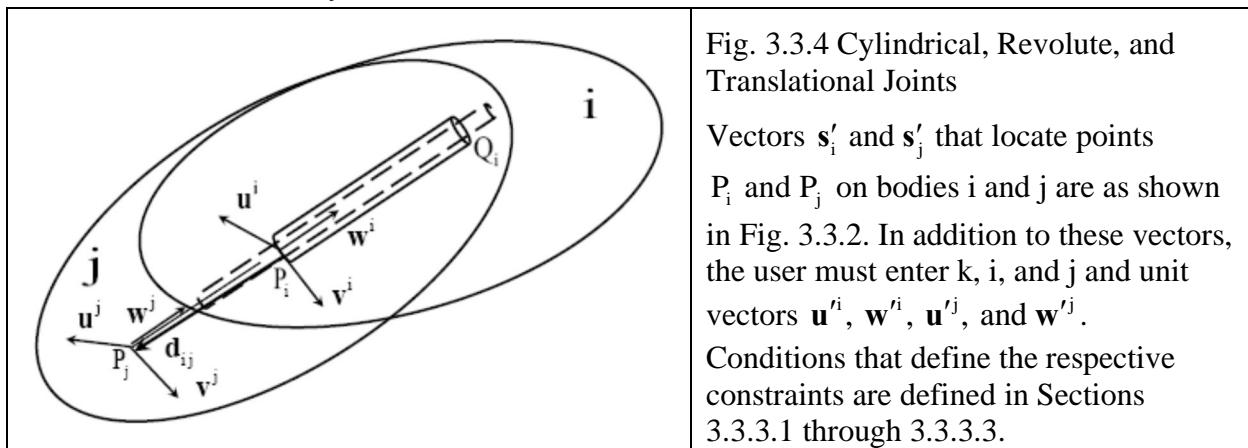


Fig. 3.3.4 Cylindrical, Revolute, and Translational Joints

Vectors  $\mathbf{s}'_i$  and  $\mathbf{s}'_j$  that locate points  $P_i$  and  $P_j$  on bodies i and j are as shown in Fig. 3.3.2. In addition to these vectors, the user must enter k, i, and j and unit vectors  $\mathbf{u}'^i$ ,  $\mathbf{w}'^i$ ,  $\mathbf{u}'^j$ , and  $\mathbf{w}'^j$ . Conditions that define the respective constraints are defined in Sections 3.3.3.1 through 3.3.3.3.

A column of the SJDT that defines joint k as a *universal joint* between bodies i and j shown in Fig. 3.3.5, repeated here for clarity, is as follows:

$$SJDT(:,k) = [6;i;j; \mathbf{s}'_i; \mathbf{s}'_j; 0;z3; \mathbf{w}'^i; z3; \mathbf{w}'^j]$$

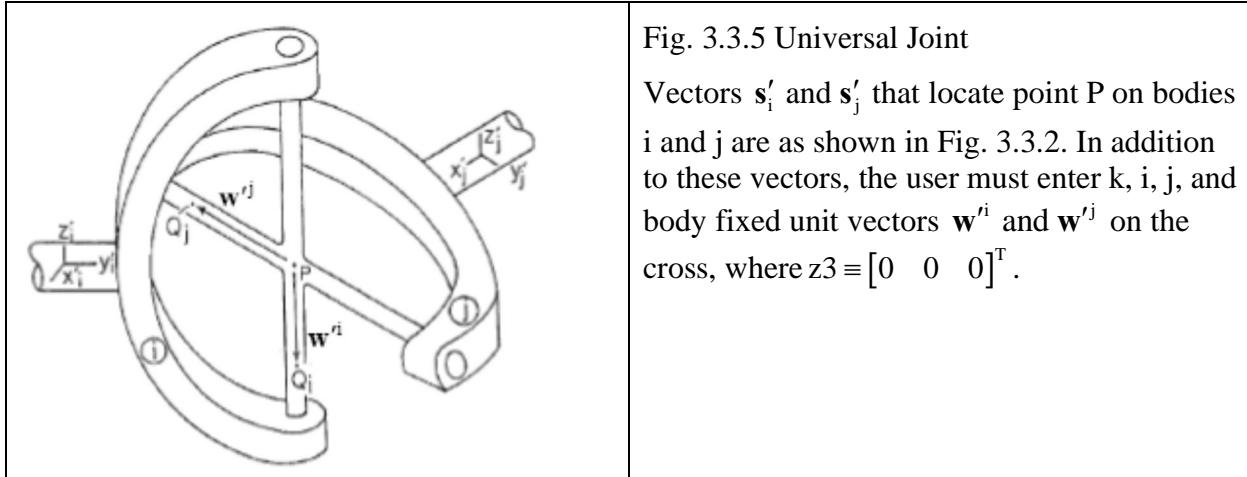


Fig. 3.3.5 Universal Joint

Vectors  $\mathbf{s}'_i$  and  $\mathbf{s}'_j$  that locate point P on bodies i and j are as shown in Fig. 3.3.2. In addition to these vectors, the user must enter k, i, j, and body fixed unit vectors  $\mathbf{w}'^i$  and  $\mathbf{w}'^j$  on the cross, where  $z3 \equiv [0 \ 0 \ 0]^T$ .

A column of the SJDT that defines joint k as a *strut joint* between bodies i and j shown in Fig. 3.3.8, repeated here for clarity, is as follows:

$$SJDT(:,k) = [7;i;j; \mathbf{s}'_i; \mathbf{s}'_j; 0;z3;z3; \mathbf{u}'^j; \mathbf{w}'^j]$$

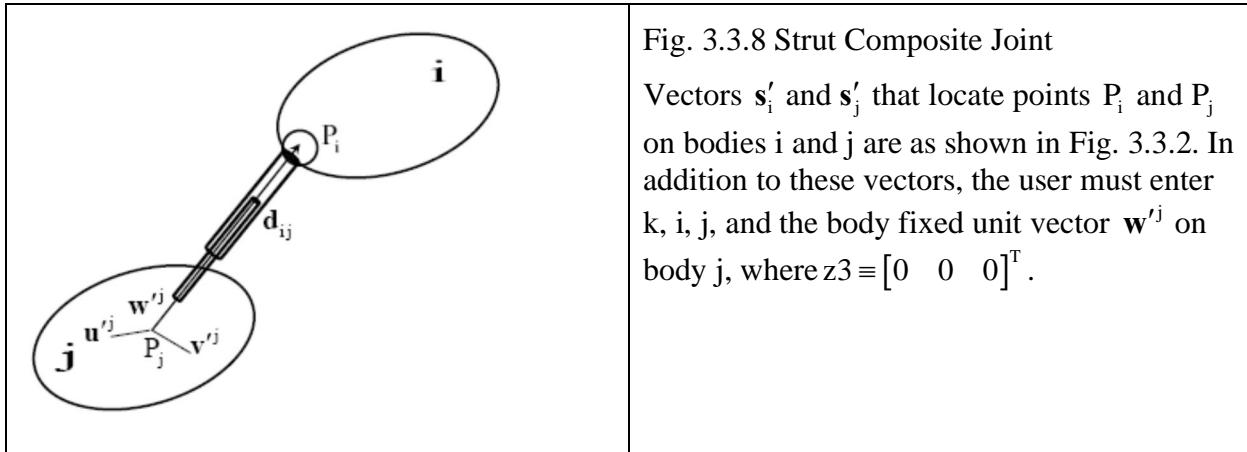


Fig. 3.3.8 Strut Composite Joint

Vectors  $\mathbf{s}'_i$  and  $\mathbf{s}'_j$  that locate points  $P_i$  and  $P_j$  on bodies i and j are as shown in Fig. 3.3.2. In addition to these vectors, the user must enter k, i, j, and the body fixed unit vector  $\mathbf{w}'^j$  on body j, where  $z3 \equiv [0 \ 0 \ 0]^T$ .

A column of the SJDT that defines joint k as a *revolute-spherical joint* between bodies i and j shown in Fig. 3.3.9, repeated here for clarity, is as follows:

$$SJDT(:,k)=[8;i;j;s'_i;s'_j;d;z3;z3;w'^j]$$

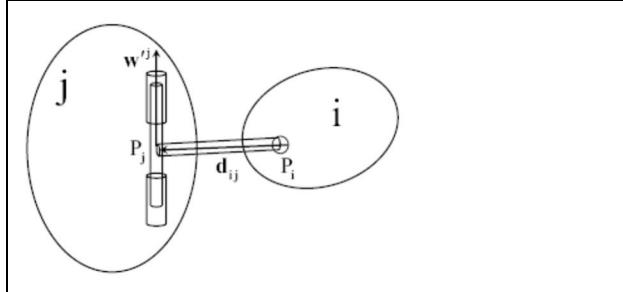


Fig. 3.3.9 Revolute-Spherical Joint

Vectors  $s'_i$  and  $s'_j$  that locate points  $P_i$  and  $P_j$  on bodies i and j are as shown in Fig. 3.3.2. In addition to these vectors, the user must enter k, i, j, d, and the body fixed unit vector  $w'^j$  on body j, where  $z3 = [0 \ 0 \ 0]^T$ .

A column of the SJDT that defines joint k as a *relative distance driver* between bodies i and j, as shown in Fig. 3.3.2, which imposes the constraint equation in Eq. (3.3.38),

$$\Phi^{distD} = (\mathbf{d}_{ij}^T \mathbf{d}_{ij} - d(t)^2)/2 = 0 \text{ where } \mathbf{d}_{ij} \text{ is as defined for the distance constraint, is as follows:}$$

$$SJDT(:,k)=[9;i;j;s'_i;s'_j;0;z3;z3], \text{ where } z3 = [0 \ 0 \ 0]^T.$$

The user need enter only k, i, j,  $s'_i$ , and  $s'_j$  in this template. The time dependent term  $-d(t)^2/2$  and its first and second time derivatives are entered in function P5.

A column of the SJDT that defines joint k as a *relative rotation driver* between bodies i and j shown in Fig. 3.3.10 that are connected by a revolute or cylindrical joint, repeated here for clarity, is as follows

$$SJDT(:,k)=[10;i;j;s'_i;s'_j;0;u'^i;w'^i;u'^j;w'^j]$$

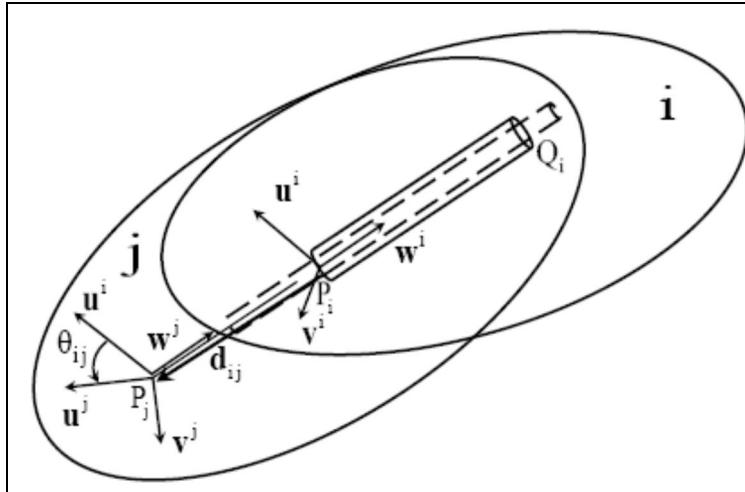


Fig. 3.3.10 Relative Rotation Drivers

Vectors  $s'_i$ ,  $s'_j$ ,  $u'^i$ ,  $w'^i$ ,  $u'^j$ , and  $w'^j$  are as defined in the host revolute or cylindrical joint shown in Fig. 3.3.10. The time dependent relative rotation  $\theta_{ij}(t)$  in Eq. (3.3.39) and its first and second time derivatives are specified in function P5.

It is **critical** that kinematic drivers be the last entered into the SJDT.

Initial generalized coordinate estimates are defined in lines 102 to 116, employing the three point method of Section 2.5.5, are used to initiate iterative solution for system configuration in the main code.

```

85 if app==3 %Spatial Slider-Crank, Bod 1 RotDr
86 nb=2; %Number of bodies
87 ngc=7*nb; %number of generalized coordinates
88 nh=3; %Number of holonomic constraints
89 nhc=11; %Number of holonomic constraint equations
90 nc=nhc+nb; %Number of constraint equations
91 nd=ngc-nc;
92 %SJDT(22,nh): Spatial Joint Data Table
93 %SJTd(:,k)=[T;i;j;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
94 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
95 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr, 10=RotDr); i&j=bodies conn.,i>0;
96 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=vectors in
97 %joints
98 SJDT(:,1)=[4;1;0;z3;0.1*uy+0.12*uz;0;uz;ux;uz;ux]; %Rev-Body1 to Ground
99 SJDT(:,2)=[5;2;0;z3;z3;0;uz;ux;uz;ux]; %Tran-Body2 to Ground
100 SJDT(:,3)=[1;1;2;0.08*uz;z3;0.24;z3;z3;z3]; %Dist-Body 1 to 2
101 SJDT(:,4)=[10;1;0;z3;0.1*uy+0.12*uz;0;uz;ux;uz;ux]; %RotDr-Body 1 to Ground
102 %Body Initial Configuration Data Estimate
103 %BPDDT(:,k)=[rO;rP;rQ];
104 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
105 %rQ=vector in x-y plane (as close to y axis as possible),
106 BPDDT(:,1)=[0.1*uy+0.12*uz;0.1*uy+0.12*uz+ux;1.1*uy+0.12*uz];
107 BPDDT(:,2)=[0.25*ux;1.25*ux;0.25*ux+uy];
108 %Initial generalized coordinate estimate
109 q0e=zeros(7*nb,1);
110 j=1;
111 while j<=nb
112 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
113 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
114 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
115 [q,p,A]=InitConfig(rO,rP,rQ);
116 q0e=Add(q0e,q,7*(j-1),0);
117 j=j+1;
118 end
119 end

```

*Figure 3.11.2 Spatial Joint Data Table (SJDT)*

The final component of user entered code is the *P5 Function* of Fig. 3.11.3 for the spatial slider-crank mechanism of Section 3.3.7.2 in Fig. 3.3.13. Using the *RotDrPart(k,SJDT)* function in line 223, data on bodies connected are obtained for evaluation of terms in the driving constraints needed, with time dependent  $\dot{\theta}_{ij}(t)$  defined in P5 for kinematic analysis. Data from the P5 function are used in the main code to define time dependence of constraints in position, velocity, and acceleration analysis. As in the planar kinematics Code 3.9, driving constraints follow kinematic constraints in the SJDT. In the spatial code, however, Euler parameter constraints follow driving constraints. This is illustrated in lines 263 to 265 of function P5 in Fig. 3.11.3, which computes right sides of position, velocity, and acceleration constraint equations; P, Pst, and Pstt. The following code is to be used as a template in entering time dependent RotD driver data for spatial applications

```

221 if app ==3 %Spatial Slider-Crank, RotDr Bod1-grnd
222 k=4;
223 [i,j,u1pr,v1pr,u2pr]=RotDrPart(k,SJDT);
225 omega=4;
226 theta=omega*tn;
227 thetad=omega;
228 thetadd=0;
230 [r1,p1]=qPart(q,i);
231 A1=ATran(p1);
233 if j==0
234 c=u1pr'*A1'*u2pr;
235 s=v1pr'*A1'*u2pr;
236 if abs(c)>=abs(s)
237 PD=-sin(theta);
238 PDd=-thetad*cos(theta);
239 PDdd=-thetadd*cos(theta)+(thetad^2)*sin(theta);
240 else
241 PD=-cos(theta);
242 PDd=thetad*sin(theta);
243 PDdd=thetadd*sin(theta)+(thetad^2)*cos(theta);
244 end
245 end
247 if j>=1
248 [r2,p2]=qPart(q,j);
249 A2=ATran(p2);
250 c=u1pr'*A1'*A2*u2pr;
251 s=v1pr'*A1'*A2*u2pr;
252 if abs(c)>=abs(s)
253 PD=-sin(theta);
254 PDd=-thetad*cos(theta);
255 PDdd=-thetadd*cos(theta)+(thetad^2)*sin(theta);
256 else
257 PD=-cos(theta);
258 PDd=thetad*sin(theta);
259 PDdd=thetadd*sin(theta)+(thetad^2)*cos(theta);
260 end
261 end
263 P=[zeros(nhc,1);PD;zeros(nb,1)];
264 Pst=[zeros(nhc,1);PDd;zeros(nb,1)];
265 Pstt=[zeros(nhc,1);PDdd;zeros(nb,1)];
266 end

```

Figure 3.11.3 P5 Function, Spatial Slider-Crank

### **3.11.2 Computational Components of Code**

*Computational flow* in the main program, which requires no input from the user, from line 25 through 74 is outlined in Fig. 3.11.4. Data to be reported for each application are entered by the user following line 76, with output data for the spatial slider-crank illustrated in lines 141 through 155.

```

25 %Kinematic Analysis
26 n=1; %Time step counter
27 t(1)=0;
28 tn=0;
29 while tn<tfinal
30 t(n)=tn;
31 %q-estimate
32 if n==1
33 q=q0e;
34 else
35 q=Q(:,n-1);
36 end
37 if n-1>1
38 q=q+h*Qd(:,n-1);
39 end
40 i=1; %Position iteration counter
41 err=qtol+1;
42 while err >qtol
43 Phi=PhiEval(tn,q,SJDT,par);
44 Phiq=PhiqEval(q,SJDT,par);
45 if i==1
46 CondPhiq(n)=cond(Phiq);
47 if CondPhiq(n)>maxCond %Check for ill conditioned constraint Jacobian
48 fprintf('Warning: Constraint Jacobian Ill Conditioned |n')
49 break
50 end
51 end
52 delq=-Phiq\Phi;
53 q=q+delq;
54 err=norm(Phi);
55 errpt(i)=err;
56 i=i+1;
57 if i>maxiter %Check for failure to converge in Newton-Raphson
58 fprintf('Warning: Newton-Raphson convergende failure')33 %Kinematic Analysis
59 break
60 end
61 end
62 iter(n)=i-1; %Report number of iterations
63 Q(:,n)=q;
64 %Evaluate qd
65 Phiq=PhiqEval(q,SJDT,par);
66 [P,Pst,Pstt]=P5Eval(tn,q,SJDT,par);
67 qd=-Phiq\Pst;
68 Qd(:,n)=qd;
69 % Evaluate qdd
70 %P2=P2Eval(q,qd,SJDT,par);
71 %Gam=P2*qd+Pstt;
72 Gam=GamEval(tn,q,qd,SJDT,par);
73 qdd=-Phiq\Gam;
74 Qdd(:,n)=qdd;
75 %Calculate output data of interest (Enter for each application)
76 if app==3 %Spatial Slider-Crank, RotDr Bod1-grnd
77 thetax(1)=0;

```

```

143 [r1,p1]=qPart(q,1);
144 [r1d,p1d]=qPart(qd,1);
145 [r1dd,p1dd]=qPart(qdd,1);
146 p1ddnorm(n)=norm(p1dd);
147 omega1x(n)=-2*ux'*EEval(p1)*p1d;
148 if n>1
149 thetax(n)=thetax(n-1)+h*omega1x(n);
150 end
151 omega1xd(n)=-2*ux'*EEval(p1)*p1dd;
152 x2(n)=q(8);
153 x2d(n)=qd(8);
154 x2dd(n)=qdd(8);
155 end

```

Figure 3.11.4 Main Code Computational Flow

*Computing functions* that underlie the main code outlined in Fig. 3.11.4 are identified in Fig. 3.11.5. Computing subroutines include the *Add Function* that enables adding nonzero submatrices to sparse matrices, below and to the right of the address of the 1-1 term in the submatrix added to the underlying matrix that was initialized to zeroes. Function atil evaluates the vector product matrix of Eq. (2.1.22), ATran evaluates the orientation transformation matrix of Eq. 2.5.22, EEval and GEval evaluate matrices **E** and **G** of Eqs. (2.6.1) and (2.6.2), and GamEval evaluates the term  $\gamma$  of Eq. (3.1.17). Finally, BTran and KEval evaluate matrices **B** and **K** of Eqs. (2.6.25) and (2.6.37).

#### Computing Functions

- Add
- atil
- ATran
- EEval
- GEval
- GamEval
- BTran
- KEval

#### Vector Partition Functions

- parPart
- qPart
- pNormPart

#### Constraint and Driver Partition Functions

- CylPart
- DistPart
- RevPart
- TranPart
- StrutPart
- RevSphPart
- UnivPart
- RotDrPart

#### Constraint and Derivative Evaluation Functions

- bbPhiDot1
- bbPhiqDot1

```

bbP2Dot1
bbxxx
.
.
.
PhiEval
PhiqEval
P2Eval
P5Eval

```

Figure 3.11.5 Computing Functions

Vector partition functions parPart, qPart, and xPart support partitioning of vectors involved into components that are used in kinematic and kinetic computations. Similarly, the constraint and driver partition functions listed provide access to constraint and driver data of the Spatial Joint Data Table (SJDT) that is defined in the AppData function to implement computation, as was the case for the planar code with its data table PJDT in Fig. 3.9.5.

Constraint and derivative evaluation functions listed evaluate kinematic constraint expressions and derivatives defined in Section 3.3 that are needed for kinematic analysis. The numerous bbxxx functions evaluate building block constraint functions and derivatives.

Internal details of the Functions listed in Fig. 3.11.4 are not presented, since each is documented internally and the user need not modify these functions.

Figure 3.11.6 presents the dot1 constraint function. This building block function is in turn used in functions PhiEval, PhiqEval, P2Eval, and P5Eval that evaluate vectors and matrices of system constraints and their derivatives. As in Fig. 3.9.6 for the planar case, care is taken to account for the fact that ground is designated  $j = 0$  and its generalized coordinates are constant, yielding no derivative contribution, but to include geometric quantities that define the constraint of body  $i$  with ground.

```

1 function Phi=bbPhidot1(i,j,a1pr,a2pr,q,par)
2 [nb,ngc,nh,nhc,nd,qtol,app]=parPart(par);
3 [r1,p1]=qPart(q,i);
4 A1=ATran(p1);
5 if j==0
6 Phi=a1pr'*A1'*a2pr;
7 end
8 if j>=1
9 [r2,p2]=qPart(q,j);
10 A2=ATran(p2);
11 Phi=a1pr'*A1'*A2*a2pr;
12 end
13 end

```

Figure 3.11.6 Building Block Constraint Evaluation Function, bbPhidot1

### 3.11.3 Code Output

In addition to output defined for each application in the AppData function, the code reports the following arrays that shed light on the application and performance of the code:

CondPhiq: the condition number of the Jacobian  $\Phi_q$  at each time step

`iter`; the number of Newton-Raphson iterations required at each time step

`Q`; the array of values of  $\mathbf{q}$  at each time step

`Qd`; the array of values of  $\dot{\mathbf{q}}$  at each time step

`Qdd`; the array of values of  $\ddot{\mathbf{q}}$  at each time step

Much as outlined in Section 3.9 for planar systems, Code 3.11 of Appendix 3.A is a tool for kinematic analysis of spatial systems. A general-purpose formulation and computer implementation are especially important for spatial systems, due to the analytical complexity associated with orientation of bodies and the resulting kinematic equations.

### 3.12 Spatial System Kinematic Analysis Using Code 3.11

Kinematic analyses of four spatial systems are carried out with MATLAB Code 3.11 to exercise various constraint and driver formulations and study system kinematic behavior with varying design and driver parameters. AppData sets are provided and driver definition using the P5 Function is illustrated.

#### 3.12.1 Two-Bar Mechanism with Two Rotation Drivers

The *two-bar mechanism* of Fig. 3.12.1 is comprised of two bars of length 1 m each. Body 1 is pivoted in ground about the global z axis and rotates in the x-y plane. Body 2 is pivoted in body 1 about the  $x'_1$  axis, so it rotates in a vertical plane that is orthogonal to the  $x'_1$  axis. As shown on the left of Fig. 3.12.2,  $\mathbf{w}'^0 = \mathbf{w}'^1 = \mathbf{u}_z$ ,  $\mathbf{u}'^0 = \mathbf{u}_x$ , and  $\mathbf{u}'^1 = \mathbf{u}_x$  define the revolute joint at the origin of the x-y-z frame. On the right of Fig. 3.12.2,  $\mathbf{w}'^1 = \mathbf{u}_x$ ,  $\mathbf{w}'^2 = \mathbf{u}_z$ ,  $\mathbf{u}'^1 = -\mathbf{u}_z$ , and  $\mathbf{u}'^2 = \mathbf{u}_x$  define the revolute joint at the intersection of the bodies. Data that define kinematics of the system are entered into the *AppData Function* of Fig.3.12.3. The initial configuration is with body 1 aligned with the x axis and body 2 aligned with the negative z axis.

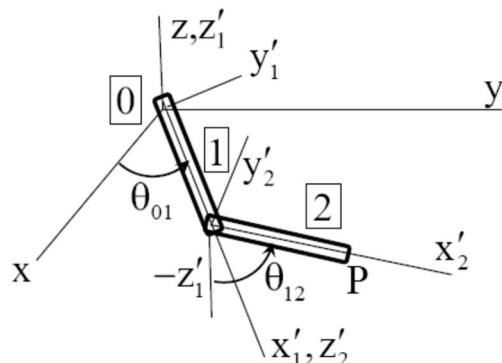


Figure 3.12.1 Two Bar Mechanism with Two Rotation Drivers

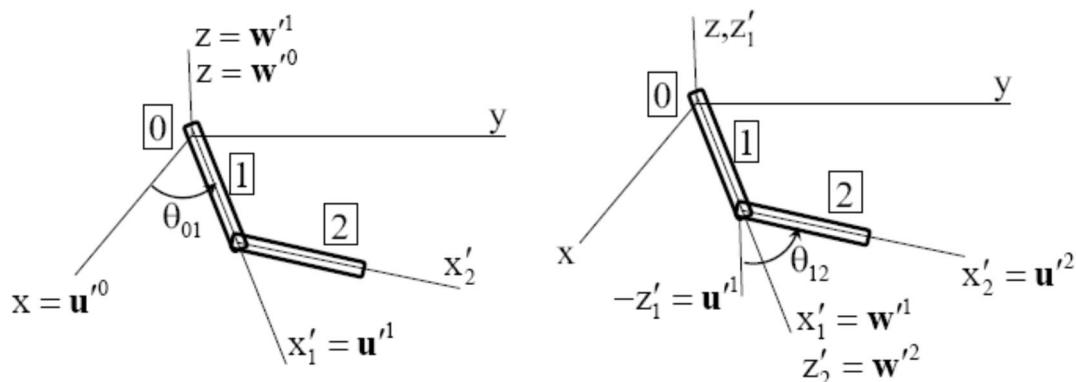


Figure 3.12.2 Data for Two Bar Mechanism Drivers

```

12 if app==1 %2 Bar with 2 rotational drivers
13 nb=2; %Number of bodies
14 ngc=7*nb; %number of generalized coordinates
15 nh=2; %Number of holonomic constraints
16 nhc=10; %Number of holonomic constraint equations

```

```

17 nc=nhc+nb; %Number of constraint equations
18 nd=ngc-nc;
19 %SJDT(22,nh): Spatial Joint Data Table
20 %SJDT(:,k)=[T;i;j;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
21 %k=joint No., T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
22 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr, 10=RotDr); i&j=bodies conn.,i>0;
23 %si&jpr=vectors to Pi&j; d=dist; uipr, wipr, ujpr, wjpr=joint vectors
24 SJDT(:,1)=[4;1;0;z3;z3;0;ux;uz;ux;uz]; %Rev-Body1 to Ground
25 SJDT(:,2)=[4;1;2;ux;z3;0;-uz;ux;ux;uz]; %Rev-Body1 to body 2
26 SJDT(:,3)=[10;1;0;z3;z3;0;ux;uz;ux;uz]; %RotDr-Body 1 to ground
27 SJDT(:,4)=[10;1;2;ux;z3;0;-uz;ux;ux;uz]; %RotDr-Body 1 to 2
28 %Body Initial Configuration Data Estimate
29 %BPDDT(:,k)=[rO;rP;rQ];
30 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
31 %rQ=vector in x-y plane (as close to y axis as possible),
32 BPDDT(:,1)=[z3;ux;uy];
33 BPDDT(:,2)=[ux;ux-uz;ux+uy];
34 %Initial generalized coordinate estimate
35 q0e=zeros(7*nb,1);
36 j=1;
37 while j<=nb
38 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
39 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
40 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
41 [q,p,A]=InitConfig(rO,rP,rQ);
42 q0e=Add(q0e,q,7*(j-1),0);
43 j=j+1;
44 end
45 end

```

Figure 3.12.3 AppData Function for Two-Bar Mechanism with 2 Rotational Drivers

*Rotational drivers* on body 1 relative to ground and on body 2 relative to body 1 shown in Fig. 3.12.1 are

$$\begin{aligned}\theta_{01} &= -\theta_{10} = \omega \times \mathbf{t}_n \\ \theta_{12} &= \omega \times \mathbf{t}_n\end{aligned}\quad (3.12.1)$$

They are implemented in lines 11 through 50 and 53 through 89 of the P5 function of Fig. 3.12.4. Note that the minus sign in lines 14 through 16 is required, since the rotation driver is on body  $j = 0$  (ground) relative to body  $i = 1$  in Fig. 3.3.10 and this is the negative of the angle  $\theta_1$  shown in Fig. 3.12.1. The remainder of the P5 function of Fig. 3.12.4 need not be modified by the user.

```

9 if app ==1 %2 Bar with 2 rotational drivers
10 %Rot driver body1 to ground
11 k=3;
12 [i,j,u1pr,v1pr,u2pr]=RotDrPart(k,SJDT);
13 omega=10;
14 theta10=-omega*t; %Angle from bod 1, to ground
15 theta10d=-omega;
16 theta10dd=0;
17 [r1,p1]=qPart(q,i);

```

```

18 A1=ATran(p1);
19 if j==0
20 c=u1pr'*A1'*u2pr;
21 s=v1pr'*A1'*u2pr;
22 if abs(c)>=abs(s)
23 PD=-sin(theta10);
24 PDd=-theta10d*cos(theta10);
25 PDdd=-theta10dd*cos(theta10)+(theta10d^2)*sin(theta10);
26 else
27 PD=-cos(theta10);
28 PDd=theta10d*sin(theta10);
29 PDdd=theta10dd*sin(theta10)+(theta10d^2)*cos(theta10);
30 end
31 end
32 if j>=1
33 [r2,p2]=qPart(q,j);
34 A2=ATran(p2);
35 c=u1pr'*A1'*A2*u2pr;
36 s=v1pr'*A1'*A2*u2pr;
37 if abs(c)>=abs(s)
38 PD=-sin(theta10);
39 PDd=-theta10d*cos(theta10);
40 PDdd=-theta10dd*cos(theta10)+(theta10d^2)*sin(theta10);
41 else
42 PD=-cos(theta10);
43 PDd=theta10d*sin(theta10);
44 PDdd=theta10dd*sin(theta10)+(theta10d^2)*cos(theta10);
45 end
46 end
47 PD1=PD;
48 PDd1=PDd;
49 PDdd1=PDdd;
50 %Rot driver body1 to body 2
51 k=4;
52 [i,j,u1pr,v1pr,u2pr]=RotDrPart(k,SJDT);
53 omega=10;
54 theta12=omega*tn; %Angle from bod 1, to bod 2
55 theta12d=omega;
56 theta12dd=0;
57 [r1,p1]=qPart(q,i);
58 A1=ATran(p1);
59 if j==0
60 c=u1pr'*A1'*u2pr;
61 s=v1pr'*A1'*u2pr;
62 if abs(c)>=abs(s)
63 PD=-sin(theta12);
64 PDd=-theta12d*cos(theta12);
65 PDdd=-theta12dd*cos(theta12)+(theta12d^2)*sin(theta12);

```

```

66 else
67 PD=-cos(theta12);
68 PDd=theta12d*sin(theta12);
69 PDdd=theta12dd*sin(theta12)+(theta12d^2)*cos(theta12);
70 end
71 end
72 if j>=1
73 [r2,p2]=qPart(q,j);
74 A2=ATran(p2);
75 c=u1pr'*A1'*A2*u2pr;
76 s=v1pr'*A1'*A2*u2pr;
77 if abs(c)>=abs(s)
78 PD=-sin(theta12);
79 PDd=-theta12d*cos(theta12);
80 PDdd=-theta12dd*cos(theta12)+(theta12d^2)*sin(theta12);
81 else
82 PD=-cos(theta12);
83 PDd=theta12d*sin(theta12);
84 PDdd=theta12dd*sin(theta12)+(theta12d^2)*cos(theta12);
85 end
86 end
87 PD2=PD;
88 PDd2=PDd;
89 PDdd2=PDdd;
90 P=[zeros(nhc,1);PD1;PD2;zeros(nb,1)];
91 Pst=[zeros(nhc,1);PDd1;PDd2;zeros(nb,1)];
92 Pstt=[zeros(nhc,1);PDdd1;PDdd2;zeros(nb,1)];
93 end

```

Figure 3.12.4 P5 Time Dependent Terms, 2 Bar with 2 Rotational Drivers

Results of kinematic analysis for the x and y position, velocity, and acceleration of the origin of the body 2 reference frame, over the period  $0 \leq t \leq 3$ , carried out with Code 3.11 are presented in Fig. 3.12.5.

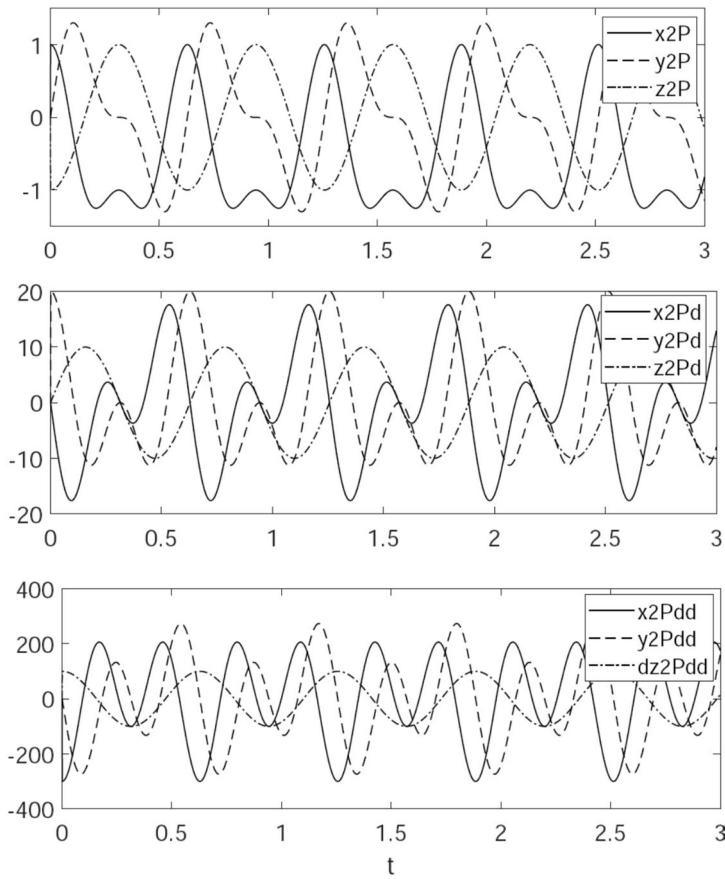


Figure 3.12.5 Position, Velocity, and Acceleration of Point P

### 3.12.2 Four-Bar Mechanism with Internal Rotation Driver

The two-bar mechanism of Fig. 3.12.1 is modified with the addition of a distance constraint of length  $\sqrt{18}$  m between the outboard end of body 2 and the point (0,4) on the y axis, to form the *closed loop four bar mechanism* that includes ground shown in Fig. 3.12.6. Data that define the model are presented in the *AppData Function* of Fig. 3.12.7. Since the *four-bar mechanism* has only one degree of freedom, a single driver is required, in this case the second relative rotation driver between bodies 1 and 2 of Eq. (3.12.1) that is implemented in the P5 Function of Fig. 3.12.8.

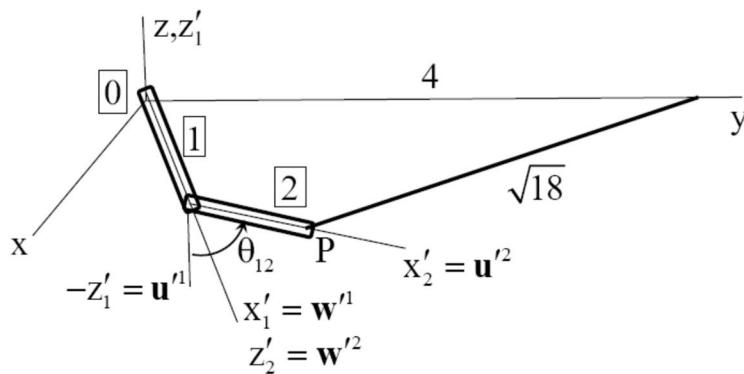


Figure 3.12.6 Four Bar Mechanism with Internal Rotational Driver

```

48 if app==2 %4Bar with internal rotational driver
49 nb=2; %Number of bodies
50 ngc=7*nb; %number of generalized coordinates
51 nh=3; %Number of holonomic constraints
52 nhc=11; %Number of holonomic constraint equations
53 nc=nhc+nb; %Number of constraint equations
54 nd=ngc-nc;
55 %SJDT(22,nh): Spatial Joint Data Table
56 %SJDT(:,k)=[T;i;j;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
57 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
58 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr, 10=RotDr); i&j=bodies conn.,i>0;
59 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=joint vectors
60 SJDT(:,1)=[4;1;0;z3;z3;0;ux;uz;ux;uz]; %Rev-Body1 to Ground
61 SJDT(:,2)=[4;1;2;ux;z3;0;-uz;ux;ux;uz]; %Rev-Body1 to body 2
62 SJDT(:,3)=[1;2;0;ux;4*uy;sqrt(18);z3;z3;z3;z3]; %Dist-Body 2 to Ground
63 SJDT(:,4)=[10;1;2;ux;z3;0;-uz;ux;ux;uz]; %RotDr-Body 1 to 2
64 %Body Initial Configuration Data Estimate
65 %BPDDT(:k)=[rO;rP;rQ];
66 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
67 %rQ=vector in x-y plane (as close to y axis as possible),
68 BPDDT(:,1)=[z3;ux;uy];
69 BPDDT(:,2)=[ux;ux-uz;ux+uy];
70 %Initial generalized coordinate estimate
71 q0e=zeros(7*nb,1);
72 j=1;
73 while j<=nb
74 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
75 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
76 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
77 [q,p,A]=InitConfig(rO,rP,rQ);
78 q0e=Add(q0e,q,7*(j-1),0);
79 j=j+1;
80 end
81 end

```

Figure 3.12.7 AppData Function for Four Bar Mechanism with Internal Rotational Driver

```

97 if app ==2 %4 Bar with rotational driver
98 %Rot driver body1 to body 2
99 k=4;
100 [i,j,u1pr,v1pr,u2pr]=RotDrPart(k,SJDT);
101 omega=10;
102 theta12=omega*tn; %Angle from bod 1, to bod 2
103 theta12d=omega;
104 theta12dd=0;
105 [r1,p1]=qPart(q,i);
106 A1=ATran(p1);
107 if j==0
108 c=u1pr'*A1'*u2pr;
109 s=v1pr'*A1'*u2pr;
110 if abs(c)>=abs(s)
111 PD=-sin(theta12);

```

```

112 PDd=-theta12d*cos(theta12);
113 PDdd=-theta12dd*cos(theta12)+(theta12d^2)*sin(theta12);
114 else
115 PD=-cos(theta12);
116 PDd=theta12d*sin(theta12);
117 PDdd=theta12dd*sin(theta12)+(theta12d^2)*cos(theta12);
118 end
119 end
120 if j>=1
121 [r2,p2]=qPart(q,j);
122 A2=ATran(p2);
123 c=u1pr'*A1'*A2*u2pr;
124 s=v1pr'*A1'*A2*u2pr;
125 if abs(c)>=abs(s)
126 PD=-sin(theta12);
127 PDd=-theta12d*cos(theta12);
128 PDdd=-theta12dd*cos(theta12)+(theta12d^2)*sin(theta12);
129 else
130 PD=-cos(theta12);
131 PDd=theta12d*sin(theta12);
132 PDdd=theta12dd*sin(theta12)+(theta12d^2)*cos(theta12);
133 end
134 end
135 PD2=PD;
136 PDd2=PDd;
137 PDdd2=PDdd;
138 P=[zeros(nhc,1);PD2;zeros(nb,1)];
139 Pst=[zeros(nhc,1);PDd2;zeros(nb,1)];
140 Pstt=[zeros(nhc,1);PDdd2;zeros(nb,1)];
141 end

```

Figure 3.12.8 P5 Function

Results of kinematic analysis for position, velocity, and acceleration of the origin of the body 2 reference frame, over the period  $0 \leq t \leq 3$ , carried out with Code 3.11 are presented in Fig. 3.12.9. The distinctly different characteristics of the responses shown in Figs. 3.12.5 and 3.12.9 are due to the addition of the distance constraint in Fig. 3.12.6. This is characteristic of the distinctive difference in performance of open loop and *closed loop mechanisms*. While kinematic analysis of closed loop mechanisms has traditionally been considered to be much more complicated than for *open loop mechanisms*, the computational approach presented here shows that this need not be the case.

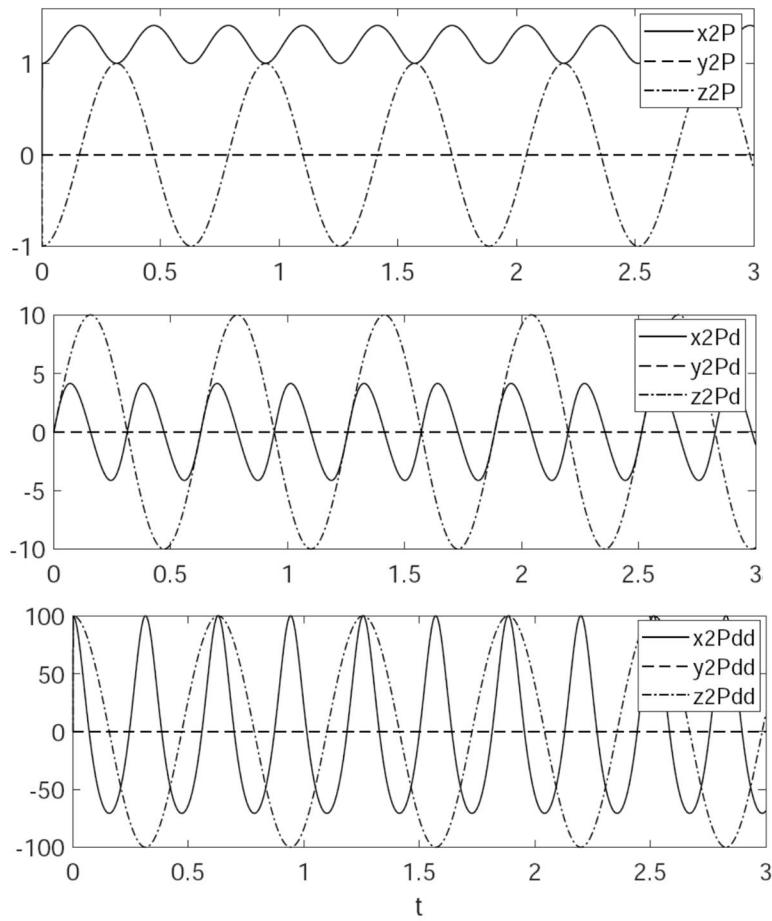


Figure 3.12.9 Position, Velocity, and Acceleration of Point P

### 3.12.3 Spatial Slider-Crank

The two-body spatial slider-crank model of Section 3.3.7.3, shown here in Fig. 3.12.10, is analyzed using Code 3.11 with data presented in Eq. (3.3.62) and incorporated in the *AppData Function* of Fig. 3.12.11 and the P5 Function of Fig. 3.12.12. The radius of the crank is 0.08 m and the rotational driver imposed on the crank is as in Eq. (3.3.62).

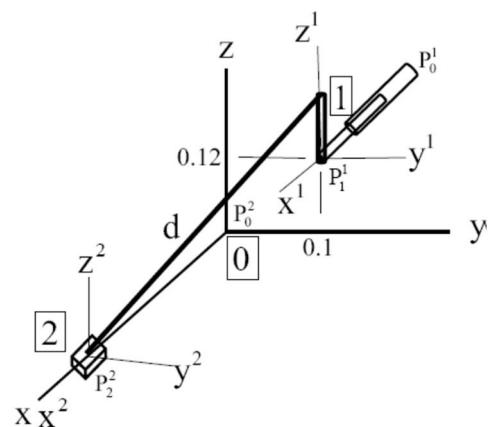


Figure 3.12.10 Spatial Slider-Crank

```

85 if app==3 %Spatial Slider-Crank, Bod 1 RotDr
86 nb=2; %Number of bodies
87 ngc=7*nb; %number of generalized coordinates
88 nh=3; %Number of holonomic constraints
89 nhc=11; %Number of holonomic constraint equations
90 nc=nhc+nb; %Number of constraint equations
91 nd=ngc-nc;
92 %SJDT(22,nh): Spatial Joint Data Table
93 %SJTd(:,k)=[T;ij;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
94 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
95 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr, 10=RotDr); i&j=bodies conn.,i>0;
96 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=vectors in
97 %joints
98 SJDT(:,1)=[4;1;0;z3;0.1*uy+0.12*uz;0;uz;ux;uz;ux]; %Rev-Body1 to Ground
99 SJDT(:,2)=[5;2;0;z3;z3;0;uz;ux;uz;ux]; %Tran-Body2 to Ground
100 SJDT(:,3)=[1;1;2;0.08*uz;z3;0.24;z3;z3;z3]; %Dist-Body 1 to 2
101 SJDT(:,4)=[10;1;0;z3;0.1*uy+0.12*uz;0;uz;ux;uz;ux]; %RotDr-Body 1 to Ground
102 %Body Initial Configuration Data Estimate
103 %BPDDT(:,k)=[rO;rP;rQ];
104 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
105 %rQ=vector in x-y plane (as close to y axis as possible),
106 BPDDT(:,1)=[0.1*uy+0.12*uz;0.1*uy+0.12*uz+ux;1.1*uy+0.12*uz];
107 BPDDT(:,2)=[0.25*ux;1.25*ux;0.25*ux+uy];
108 %Initial generalized coordinate estimate
109 q0e=zeros(7*nb,1);
110 j=1;
111 while j<=nb
112 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
113 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
114 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
115 [q,p,A]=InitConfig(rO,rP,rQ);
116 q0e=Add(q0e,q,7*(j-1),0);
117 j=j+1;
118 end
119 end

```

Figure 3.12.11 AppData Function for Spatial Slider-Crank

```

145 if app ==3 %Spatial Slider-Crank, RotDr Bod1-grnd
146 k=4;
147 [i,j,u1pr,v1pr,u2pr]=RotDrPart(k,SJDT);
148 omega=4;
149 theta=-omega*tn;
150 thetad=-omega;
151 thetadd=0;
152 [r1,p1]=qPart(q,i);
153 A1=ATran(p1);
154 if j==0
155 c=u1pr'*A1'*u2pr;
156 s=v1pr'*A1'*u2pr;
157 if abs(c)>=abs(s)
158 PD=-sin(theta);

```

```

159 PDd=-thetad*cos(theta);
160 PDdd=-thetadd*cos(theta)+(thetad^2)*sin(theta);
161 else
162 PD=-cos(theta);
163 PDd=thetad*sin(theta);
164 PDdd=thetadd*sin(theta)+(thetad^2)*cos(theta);
165 end
166 end
167 if j>=1
168 [r2,p2]=qPart(q,j);
169 A2=ATran(p2);
170 c=u1pr'*A1'*A2*u2pr;
171 s=v1pr'*A1'*A2*u2pr;
172 if abs(c)>=abs(s)
173 PD=-sin(theta);
174 PDd=-thetad*cos(theta);
175 PDdd=-thetadd*cos(theta)+(thetad^2)*sin(theta);
176 else
177 PD=-cos(theta);
178 PDd=thetad*sin(theta);
179 PDdd=thetadd*sin(theta)+(thetad^2)*cos(theta);
180 end
181 end
182 P=[zeros(nhc,1);PD;zeros(nb,1)];
183 Pst=[zeros(nhc,1);PDd;zeros(nb,1)];
184 Pstt=[zeros(nhc,1);PDdd;zeros(nb,1)];
185 end

```

Figure 3.12.12 P5 Function for Spatial Slider-Crank

To study the effect of varying crank angular velocity, the connecting rod length is selected as  $d = 0.3$  m. Plots of position, velocity, and acceleration ( $z_2$ ,  $z_{2d}$ , and  $z_{2dd}$ ) of the slider for crank angular velocities  $\omega = 1, 2$ , and  $4$  rad/sec are presented in Fig. 3.12.13. As expected, velocities vary linearly with input angular velocity and accelerations vary with the square of velocities.

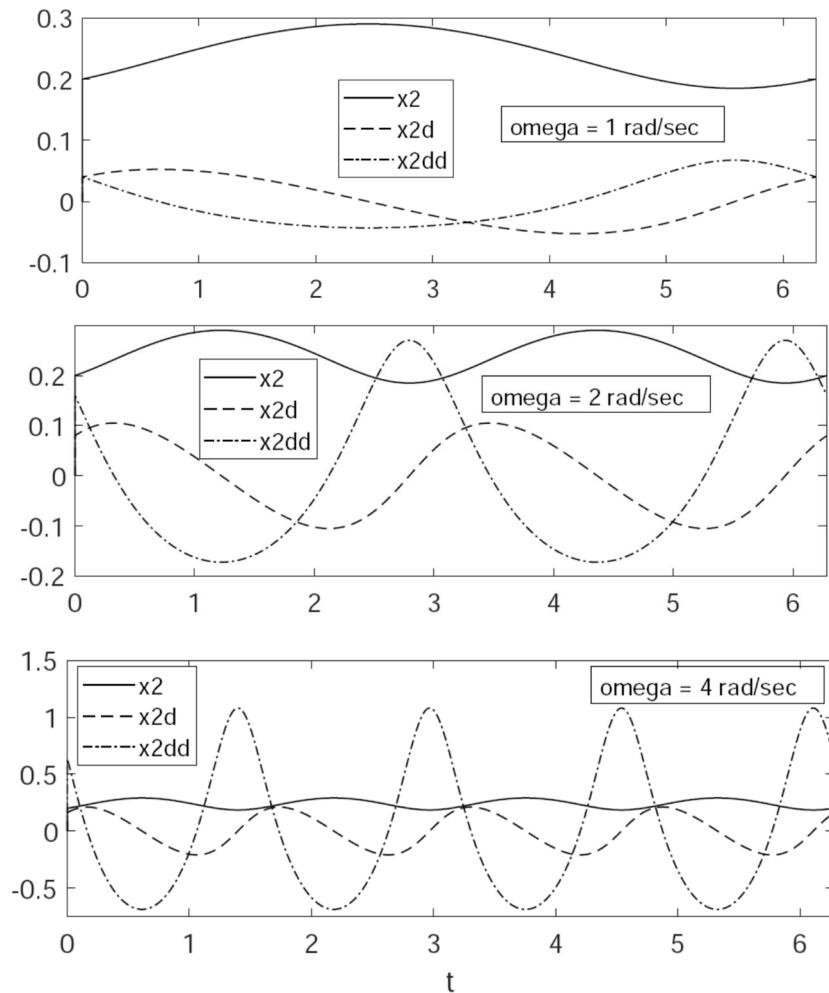


Figure 3.12.13 Slider Position, Velocity, and Acceleration,  $d = 0.3\text{m}$

To study the influence of variable connecting rod length, the crank is driven at  $\omega = 4 \text{ rad/sec}$  with a connecting rod length of 0.24 m. Plots of position, velocity, and acceleration of the slider are presented in Fig. 3.12.14. Since a connecting rod length of 0.23 m leads to a *singular configuration*, it is not surprising that accelerations become extreme with a connecting rod length of 0.24 m.

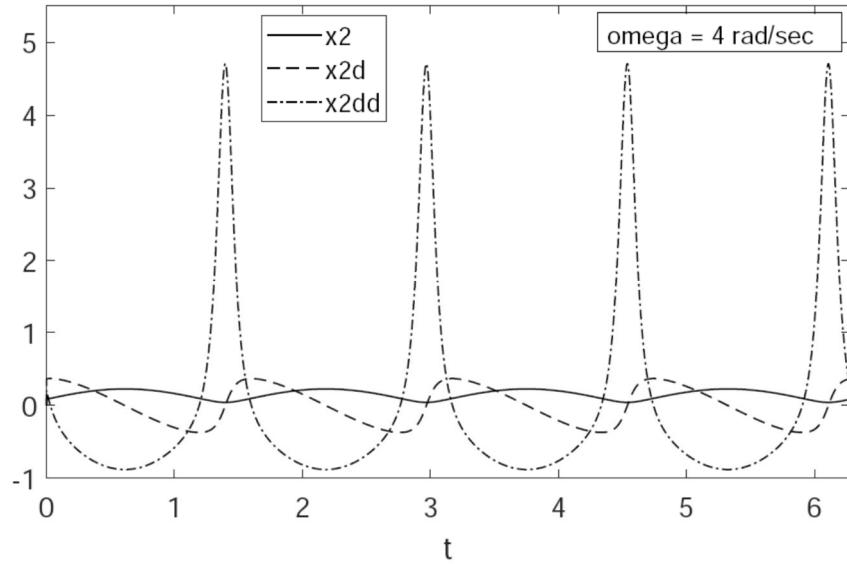


Figure 3.12.14 Slider Position, Velocity, and Acceleration,  $\omega = 4$ ,  $d = 0.24\text{m}$

### 3.12.4 Fly-Ball Governor

The *fly-ball governor* of Fig. 3.12.15, defined in Section 3.3.7.4, is comprised of four moving bodies and ground. Body 1 is the rotor in a revolute joint with ground. Body 2 is the collar that translates on and rotates with the rotor. Bodies 3 and 4 contain heavy balls at their ends and rotate in revolute joints at the top of the rotor. Two distance constraints connect bodies 3 and 4 with the collar, to control fuel feed. The revolute joints with bodies 3 and 4 on body 1 are offset 0.02 m from its centerline, as are the distance constraint attachment points on body 2. With three revolute and one translational joint, two distance constraints and four Euler parameter normalization constraints, there are a total of 26 constraint equations in 28 generalized coordinates. If the constraint Jacobian has full rank, the system has two degrees of freedom.

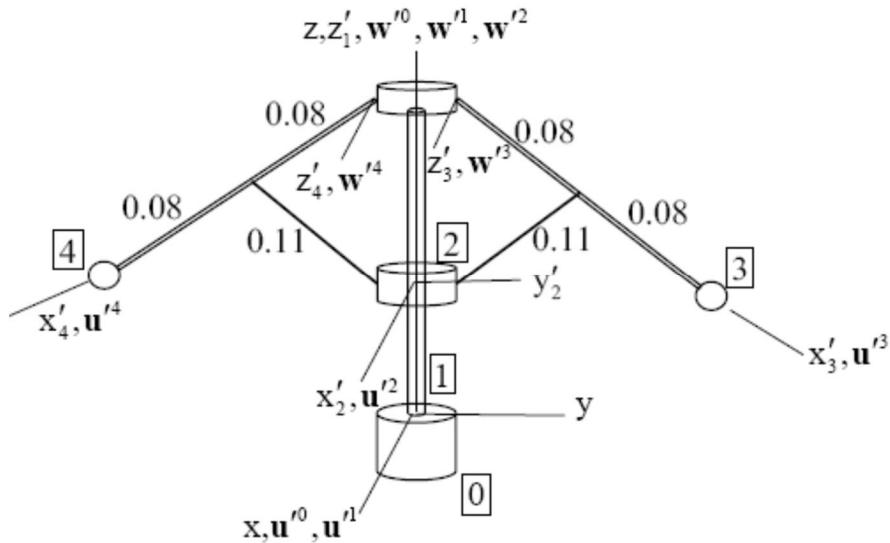


Figure 3.12.15 Fly-Ball Governor

Drivers that specify rotation of the shaft relative to ground and the distance of the collar from the base of the shaft are defined in Eq. (3.3.63). These data are implemented in AppData

and P5 Functions of Figs. 3.12.16 and 3.12.17, respectively, for simulation in Code 3.11. Numerical results providing the elevation of the collar and ball 3 are given in Fig. 3.12.18.

```

122 if app==4 %Fly-Ball Governor
123 nb=4; %Number of bodies
124 ngc=7*nb; %number of generalized coordinates
125 nh=6; %Number of holonomic constraints
126 nhc=22; %Number of holonomic constraint equations
127 nc=nhc+nb; %Number of constraint equations
128 nd=ngc-nc;
129 %SJDT(22,nh): Spatial Joint Data Table
130 %SJDT(:,k)=[T;i;j;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
131 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
132 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr, 10=RotDr); i&j=bodies conn.,i>0;
133 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=joint vectors
134 SJDT(:,1)=[4;1;0;z3;z3;0;ux;uz;ux;uz]; %Rev-Body1 to Ground
135 SJDT(:,2)=[5;1;2;z3;z3;0;ux;uz;ux;uz]; %Tran-Body1 to body 2
136 SJDT(:,3)=[4;1;3;0.2*uz+0.02*uy;z3;0;uz;ux;ux;uz]; %Rev-Body1 to body 3
137 SJDT(:,4)=[4;1;4;0.2*uz-0.02*uy;z3;0;uz;ux;ux;uz]; %Rev-Body1 to body 4
138 SJDT(:,5)=[1;2;3;0.02*uy;0.08*ux;0.11;z3;z3;z3]; %Dist-Body 2 to body3
139 SJDT(:,6)=[1;2;4;-0.02*uy;0.08*ux;0.11;z3;z3;z3]; %Dist-Body 2 to body4
140 SJDT(:,7)=[10;1;0;z3;z3;0;ux;uz;ux;uz]; %RotDr-Body 1 to ground
141 SJDT(:,8)=[9;1;2;z3;z3;0;z3;z3;z3]; %DistDr-Body 1 to 2
142 %Body Initial Configuration Data Estimate
143 %BPDDT(:,k)=[rO;rP;rQ];
144 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
145 %rQ=vector in x-y plane (as close to y axis as possible),
146 BPDDT(:,1)=[z3;ux;uy];
147 BPDDT(:,2)=[0.1*uz;0.1*uz+ux;0.1*uz+uy];
148 BPDDT(:,3)=[0.2*uz+0.02*uy;0.2*uz+0.02*uy-uz+uy;0.2*uz+0.02*uy+uz+uy];
149 BPDDT(:,4)=[0.2*uz-0.02*uy;0.2*uz-0.02*uy-uz-uy;0.2*uz-0.02*uy+uz-uy];
150 %Initial generalized coordinate estimate
151 q0e=zeros(7*nb,1);
152 j=1;
153 while j<=nb
154 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
155 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
156 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
157 [q,p,A]=InitConfig(rO,rP,rQ);
158 q0e=Add(q0e,q,7*(j-1),0);
159 j=j+1;
160 end
161 end

```

Figure 3.12.16 AppData Function for Fly-Ball Governor

```

189 if app ==4 %Fly-Ball Governor
190 %Rot driver ground to body 1
191 k=7;
192 [i,j,u1pr,v1pr,u2pr]=RotDrPart(k,SJDT);
193 omega=10;
194 theta10=omega*t; %Angle from bod 1, to ground
195 theta10d=-omega;

```

```

196 theta10dd=0;
197 [r1,p1]=qPart(q,i);
198 A1=ATran(p1);
199 if j==0
200 c=u1pr'*A1'*u2pr;
201 s=v1pr'*A1'*u2pr;
202 if abs(c)>=abs(s)
203 PD=-sin(theta10);
204 PDd=-theta10d*cos(theta10);
205 PDdd=-theta10dd*cos(theta10)+(theta10d^2)*sin(theta10);
206 else
207 PD=-cos(theta10);
208 PDd=theta10d*sin(theta10);
209 PDdd=theta10dd*sin(theta10)+(theta10d^2)*cos(theta10);
210 end
211 end
212 if j>=1
213 [r2,p2]=qPart(q,j);
214 A2=ATran(p2);
215 c=u1pr'*A1'*A2*u2pr;
216 s=v1pr'*A1'*A2*u2pr;
217 if abs(c)>=abs(s)
218 PD=-sin(theta10);
219 PDd=-theta10d*cos(theta10);
220 PDdd=-theta10dd*cos(theta10)+(theta10d^2)*sin(theta10);
221 else
222 PD=-cos(theta10);
223 PDd=theta10d*sin(theta10);
224 PDdd=theta10dd*sin(theta10)+(theta10d^2)*cos(theta10);
225 end
226 end
227 PD1=PD;
228 PDd1=PDd;
229 PDdd1=PDdd;
230 %Dist driver body1 to body 2
231 k=8;
232 [i,j,s1pr,s2pr]=DistDrPart(k,SJDT);
233 d=0.1+0.02*sin(omega*tn);
234 PD2=-0.5*d^2;
235 PDd2=-d*0.02*omega*cos(omega*tn);
236 PDdd2=d*0.02*omega^2*sin(omega*tn)-(0.02*omega*cos(omega*tn))^2;
237 P=[zeros(nhc,1);PD1;PD2;zeros(nb,1)];
238 Pst=[zeros(nhc,1);PDd1;PDd2;zeros(nb,1)];
239 Pstt=[zeros(nhc,1);PDdd1;PDdd2;zeros(nb,1)];
240 end

```

### 3.12.17 P5 Function for Fly-Ball Governor

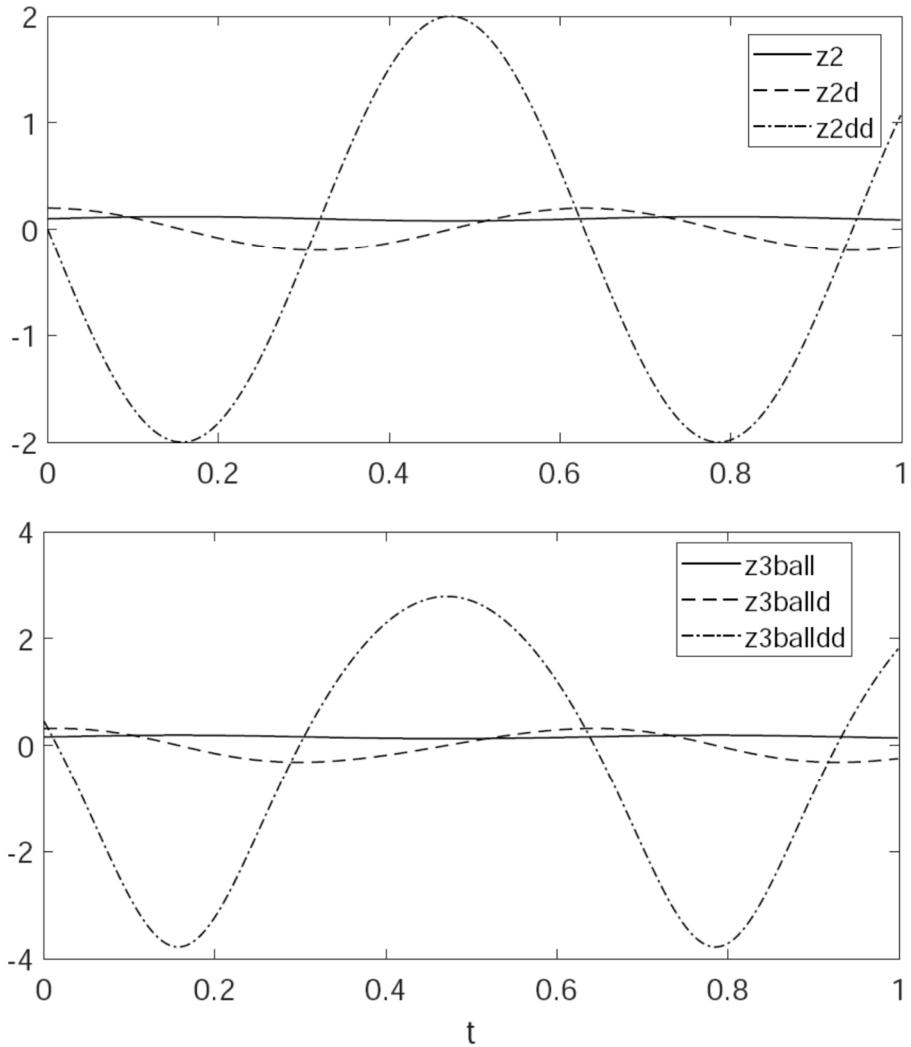


Figure 3.12.18 Elevation of Collar and Ball 3

For spatial system models with time dependent drivers that create the same number of constraint equations as generalized coordinates, kinematic analysis is carried out using iterative methods to solve nonlinear constraint equations for kinematically admissible configurations on a time grid that is specified for analysis. Linear velocity and acceleration equations are solved on the same time grid for dynamic response. Criteria for singular behavior are monitored, including the condition number of the Jacobian, rapid changes in acceleration, and loss of existence in position analysis.

Numerical analyses using Code 3.11 of Appendix 3.A are carried out with minimal investment of person time, in contrast with the extreme effort that would be associated with manual derivation and coding. Results confirm the criteria for onset of singular behavior and quadratic dependence of system acceleration on velocities.

## **Appendix 3.A Kinematics Code**

MATLAB computer code for planar and spatial kinematic analysis is provided in software files that accompany the text. Instruction on two substantial codes is provided in Sections 3.9 and 3.11. Codes contained in accompanying files are as follows:

***Code 3.6.1 Particle On Unit Sphere***

***Code 3.9 Planar Kinematic Analysis***

***Code 3.11 Spatial Kinematic Analysis***

### Appendix 3.B Key Formulas, Chapter 3

$$\Phi^k(q_i, q_j) = 0 \quad \Phi_{q_{ij}}^k(q_i, q_j) = \begin{bmatrix} \Phi_{q_i}^k(q_i, q_j) & \Phi_{q_j}^k(q_i, q_j) \end{bmatrix} \quad (3.1.7)$$

$$\Phi_{q_i}^k(q_i, q_j)\dot{q}_i + \Phi_{q_j}^k(q_i, q_j)\dot{q}_j = 0 \quad (3.1.8)$$

$$\Phi_{q_i}^k(q_i, q_j)\ddot{q}_i + \Phi_{q_j}^k(q_i, q_j)\ddot{q}_j = -\gamma^k(q_{ij}, \dot{q}_{ij}) \quad (3.1.9)$$

$$\gamma^k(q_{ij}, \dot{q}_{ij}) = \left( \Phi_{q_i}^k(q_i, \hat{q}_j) \hat{\dot{q}}_i \right)_{q_i} \dot{q}_i + 2 \left( \Phi_{q_i}^k(\hat{q}_i, q_j) \hat{\dot{q}}_i \right)_{q_j} \dot{q}_j + \left( \Phi_{q_j}^k(\hat{q}_i, q_j) \hat{\dot{q}}_j \right)_{q_j} \dot{q}_j \quad (3.1.17)$$

$$P2^k(q_{ij}, \dot{q}_{ij}) \equiv \left[ \left( \Phi_{q_i}^k \dot{q}_i \right)_{q_i} + \left( \Phi_{q_j}^k \dot{q}_j \right)_{q_i} - \left( \Phi_{q_i}^k \dot{q}_i \right)_{q_j} + \left( \Phi_{q_j}^k \dot{q}_j \right)_{q_i} \right] \quad (3.1.19)$$

$$\Phi^{\text{dist}} \equiv (d_{ij}^T d_{ij} - d^2) / 2 = 0 \quad d_{ij} \equiv r_j + A_j s'_j - r_i - A_i s'_i \quad (3.2.2)$$

$$\Phi^{\text{rev}} \equiv d_{ij} = r_j + A_j s'_j - r_i - A_i s'_i = 0 \quad \Phi^{\text{tan}} \equiv - \begin{bmatrix} v_i'^T P A_i^T d_{ij} \\ v_i'^T P A_i^T A_j v'_j \end{bmatrix} = 0 \quad (3.2.7) \quad (3.2.13)$$

$$\Phi^{\text{dist}}(P_i, P_j, d) \equiv (d_{ij}^T d_{ij} - d^2) / 2 = 0 \quad d_{ij} = r_j + A_j s'_j - r_i - A_i s'_i \quad (3.3.1) \quad (3.3.2)$$

$$\Phi^{\text{sph}}(P_i, P_j) \equiv d_{ij} = r_j + A_j s'_j - r_i - A_i s'_i = 0 \quad (3.3.8)$$

$$\Phi^{\text{dot1}}(a_i, a_j) = a_i'^T A_i^T A_j a_j = 0 \quad \Phi^{\text{dot2}}(a_j, d_{ij}) = a_j'^T A_j^T d_{ij} = 0 \quad (3.3.13) \quad (3.3.18)$$

$$\Phi(q, t) = [\Phi_1(q, t) \quad \dots \quad \Phi_{n_{\text{hc}}}(q, t)]^T = 0 \quad q = [q_1^T \quad q_2^T \quad \dots \quad q_{n_b}^T]^T \in R^{n_{\text{gc}}} \quad (3.4.19)$$

$$E(q, t)\dot{q} = e(q, t) \quad \Phi_q(q, t)\dot{q} = -\Phi_t(q, t) \equiv v_h(q, t) \quad (3.4.20) \quad (3.4.21)$$

$$C(q, t)\dot{q} \equiv \begin{bmatrix} \Phi_q(q, t) \\ E(q, t) \end{bmatrix} \dot{q} \equiv \begin{bmatrix} v_h(q, t) \\ e(q, t) \end{bmatrix} \equiv v(q, t) \quad (3.4.22)$$

$$C(q, t)\ddot{q} \equiv \begin{bmatrix} -\gamma_h(q, \dot{q}, t) \\ -\gamma_{nh}(q, \dot{q}, t) \end{bmatrix} \equiv -\gamma(q, \dot{q}, t) \quad (3.4.25)$$

$$C = \{q : \Phi(q) = 0\} \subset R^{ngc} \quad (3.5.2)$$

$$\tilde{C} = \left\{ q \in C : \text{rank } \Phi_q(q) = nhc \right\} \subset R^{ngc} \quad (3.5.3)$$

$$U = \Phi_q^T(q^0) \quad \Phi_q^0 V = U^T V = 0, V^T V = I \quad (3.5.20) \quad (3.5.21)$$

$$q = q^0 + Vv - Uh(v) \quad \Phi(q^0 + Vv - Uh(v)) = 0 \quad u = h(v) \quad (3.5.27) \quad (3.5.30)$$

$$u^{i+1} = u^i + B\Phi(q^0 + Vv - Uh(v)), \quad i = 1, 2, \dots \text{ until } \|\Phi(q^0 + Vv - Uh(v))\| \leq \text{utol} \quad (3.5.40)$$

$$B(q) \equiv (\Phi_q(q)U)^{-1} \quad (3.5.29)$$

$$B^{i+1} = 2B^i - B^i \Phi_q U B^i, \quad i = 1, 2, \dots, \text{ until } \|\Phi_q U B^{i+1} - I\| \leq \text{Btol} \quad (3.5.39)$$

$$\Phi_q(q^i, t_i) \Delta q^i = -\Phi(q^i, t_i); \quad q^{i+} = q^i + \Delta q^i \quad (3.8.13)$$

$$\Phi_q(q^i, t_i) \dot{q}^i = -\Phi_t(q^i, t_i) \equiv v(q^i, t_i) \quad (3.8.16)$$

$$\Phi_q(q^i, t_i) \ddot{q}^i = -\left( \Phi_q(q^i, t_i) \dot{q}^i \right)_q \dot{q}^i - 2\Phi_{qq}(q^i, t_i) \dot{q}^i - \Phi_{tt}(q^i, t_i) \equiv -\gamma(q^i, \dot{q}^i, t_i) \quad (3.8.17)$$

## CHAPTER 4

### Equations of Motion for Bodies and Multibody Systems

#### 4.0 Introduction

*Variational equations of motion* that are based on *Newton's laws of motion* for particles, *d'Alembert's principle*, and kinematic relations defined in Chapter 3 are derived for systems of particles in Section 4.1 and for planar and spatial rigid bodies in Sections 4.2 and 4.3. Inertia matrices are defined for planar and spatial bodies and their properties are analyzed in Section 4.4.

External and internal forces and torques that act on and between bodies are incorporated into the variational formulation in Section 4.5. Variational equations of motion for individual bodies are summed over all bodies in a system in Section 4.6, to obtain multibody system variational equations of motion. The concept of *virtual work* and *d'Alembert's principle*, in the form öconstraint reaction forces do no work as a result of kinematically admissible virtual displacementsö, are applied to obtain system *variational equations of motion*. Broadly applicable conditions are defined under which the multibody system mass matrix is positive definite on the space of *kinematically admissible virtual velocities*. This property of the system mass matrix is of critical importance in *computational dynamics*. A formulation that enables initial conditions to be defined for constrained equations of motion is presented. Examples are presented in Section 4.7, showing that variational equations of motion for holonomic systems with independent generalized coordinates can be reduced to second order *ordinary differential equations (ODE)* of motion.

*Runge-Kutta numerical integration methods* for first order ODE, with established stability and accuracy properties, are presented in Section 4.8. *First order ODE* methods are transformed to a form that is suitable for application to *second order ODE of dynamics*. MATLAB B Code 4.8 that implements the second order solution methods is presented in Appendix 4.A and user instruction is provided in Section 4.9.1. Numerical results for planar and spatial examples studied in Section 4.7.1 are obtained using MATLAB Codes 4.9.1 through 4.9.3 presented in Appendix 4.A, to illustrate the use of *numerical integration methods* in mechanical system dynamics.

*Lagrange multipliers* are introduced in Section 4.10 to evaluate *constraint reaction forces* in multibody systems. This leads to equations of motion in Lagrange multipliers that appear algebraically as unknowns in equations of motion that include algebraic kinematic constraint equations. The resulting system has become known as *differential-algebraic equations (DAE)* of motion.

## 4.1 Equations of Motion for Particles

The foundation on which the theory of mechanical system dynamics is based is three *laws of motion* postulated by Newton (1687) that govern the motion of particles and *d'Alembert's principle* that states that forces of constraint do no work (d'Alembert, 1743). These landmark contributions represent the fundamental physics that underpin a broadly applicable formulation for dynamics of mechanical systems. Definition of position, velocity, and acceleration of a point in space presupposes the existence of some *frame of reference*. For the purposes of dynamics, it is important that the frame of reference be an *inertial reference frame*, defined as a frame in which Newton's laws of motion are valid. An inertial reference frame may be thought of as a frame fixed in some ideal space relative to which motion of stars and planets can be described. In most engineering applications, a reference frame that is fixed to the surface of the earth is adequate. This is technically not an inertial reference frame, due to motion of the earth relative to the sun and other stars. Accuracy associated with an *earth-fixed reference frame* is, however, acceptable for many engineering purposes.

### 4.1.1 Newton's Laws

In an inertial reference frame, the following are *Newton's laws of motion*:

**First Law:** A particle continues in its state of rest or constant velocity if the resultant of forces acting on the particle is zero.

**Second Law:** The product of the mass of a particle and its acceleration in an inertial reference frame is equal to the resultant force acting on the particle.

**Third Law:** To every force of action on a particle there is an equal and opposite force of reaction on another particle; i.e., the mutual forces of interaction between two particles are equal in magnitude and opposite in direction.

The *first law* may be viewed as a special case of the *second law*, in which a zero resultant force implies zero acceleration, hence constant velocity. The first law, however, plays an interesting role in *static friction*, or *stiction*.

The *resultant force*  $\mathbf{F}(t)$  in the second law is the vector sum of all forces that act on a particle. The acceleration of the particle is the second time derivative of the position vector  $\mathbf{r} = \mathbf{r}(t)$  that locates the particle relative to the origin of an inertial reference frame and the *mass* of the particle is denoted  $m$ . the *second law* may thus be written in vector form as

$$m\ddot{\mathbf{r}} = \mathbf{F}(t) \quad (4.1.1)$$

In most applications, the *forces of action and reaction* between pairs of particles are not only equal in magnitude and opposite in direction, as guaranteed by the *third law*, but they are *collinear*; i.e., they act along the line between the particles. This is called the *strong form of the third law* (Goldstein, 1980) and is true for *gravitational forces*, rectilinear spring and damper forces, and *forces of contact*. Forces of interaction due to charged particles in an electric field need not, however, be collinear.

The relation between *mass*  $m$  and *weight*  $w$  of a particle is established by the *gravitational acceleration*  $\mathbf{g}$ , which has magnitude  $g$  and is directed toward the center of the

earth. The weight  $w$  of the particle is a scalar quantity that is related to its mass and the gravitational acceleration by

$$w = mg \quad (4.1.2)$$

Units employed in defining positions of particles, forces acting on particles, and masses of particles must be consistently defined in order that Eqs. (4.1.1) and (4.1.2) may be employed. *Standard International units (SI)* of *meters*, *newtons*, and *kilograms* are used in the text. In all systems of units, the *unit of time* is seconds.

Note that if  $\mathbf{F}(t)$  and  $m$  are known in Eq. (4.1.1), and no constraints act on the particle, Newton's equations of motion are *second-order ODE*. As is well known from the theory of ordinary differential equations (Teschl, 2012), *initial conditions* must be specified on  $\mathbf{r}$  and  $\dot{\mathbf{r}}$  at an *initial time*  $t^0$  to yield a unique solution of Eq. (4.1.1) for  $\mathbf{r}(t)$ , namely

$$\begin{aligned} \mathbf{r}(t^0) &= \mathbf{r}^0 \\ \dot{\mathbf{r}}(t^0) &= \dot{\mathbf{r}}^0 \end{aligned} \quad (4.1.3)$$

Equations (4.1.1) and (4.1.3) comprise an *initial-value problem* that determines the motion of an unconstrained particle.

#### 4.1.2 Virtual Displacements

Defining the *virtual displacement* of a particle as a perturbation  $\delta\mathbf{r}$ , its scalar product with both sides of Eq. (4.1.1) yields

$$\mathbf{r}^T (m\ddot{\mathbf{r}} - \mathbf{F}) = 0 \quad (4.1.4)$$

This equation holding for arbitrary  $\delta\mathbf{r}$  is equivalent to Eq. (4.1.1). It is called the *variational equation of motion* for the particle.

Lagrange (1788) published a treatise on *analytical dynamics* that introduced concepts of *virtual displacement* and *virtual work* associated with particles that move in space under the action of applied forces and forces due to constraints that act between them. For a system of  $n_p$  particles that are located by vectors  $\mathbf{r}_i$ ,  $i = 1, \dots, n_p$ , relative to an inertial reference frame, a vector of *generalized coordinates* of dimension  $3n_p$  is defined as

$$\mathbf{q} = \left[ \mathbf{r}_1^T \quad \cdots \quad \mathbf{r}_{n_p}^T \right]^T \quad (4.1.5)$$

that locates all of the particles in the inertial reference frame. As shown in Section 3.1, particles may be subjected to *kinematic constraints* that specify interactions between pairs of particles, such as massless rods that constrain the distance between particles to be constant, or *actuators* that specify distances between particles as functions of time. Such constraints may be written in equation form as *holonomic constraints*,

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (4.1.6)$$

where  $\Phi(\mathbf{q}, t) = [\phi_1(\mathbf{q}, t) \quad \cdots \quad \phi_{n_h}(\mathbf{q}, t)]^T$ . While most constraints, such as those encountered in Chapter 3, do not depend explicitly on time, time dependent constraints are admitted here to account for time dependent actuators between pairs of particles. Using the *vector calculus*

notation of Eq. (2.2.35), Eq. (4.1.6) may be differentiated with respect to time to obtain the *holonomic velocity constraint equation*

$$\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} = -\Phi_t(\mathbf{q}, t) \equiv v_h(\mathbf{q}, t) \quad (4.1.7)$$

that must hold throughout the time interval of interest. In order for the constraints to be independent, the *constraint Jacobian*  $\Phi_q(\mathbf{q}, t)$  must have *full rank* for all  $\mathbf{q}$  and  $t$ . Written in *differential form*, Eq. (4.1.7) is

$$\Phi_q(\mathbf{q}, t)d\mathbf{q} = v_h(\mathbf{q}, t)dt \quad (4.1.8)$$

A *differential*  $d\mathbf{q}$  that satisfies Eq. (4.1.8) is said to be an *admissible displacement* at  $(\mathbf{q}, t)$ , with *time increment*  $dt$ . With the goal of obtaining a variational formulation of the equations of dynamics for a system of particles that is based on Newton's equations of motion, the use of admissible displacements is problematic. Newton's equations of motion hold at each instant in time, they do not involve a concept of time increment. To resolve this dilemma, Lagrange is credited (Pars, 1965) with creating the concept of a *virtual displacement*, denoted  $\delta\mathbf{q}$ , that is *kinematically admissible* at an instant in time if it satisfies Eq. (4.1.8) with  $dt = 0$ ,

$$\Phi_q(\mathbf{q}, t)\delta\mathbf{q} = \mathbf{0} \quad (4.1.9)$$

so  $\delta\mathbf{q}$  is tangent to the *constraint manifold* defined in Section 3.5, for constraints of Eq. (4.1.6), with  $t$  fixed. The admissible displacement  $d\mathbf{q}$  of Eq. (4.1.8) is generally not tangent to the constraint manifold. It should be noted that if the holonomic constraint of Eq. (4.1.6) does not depend explicitly on time, then there is no distinction between admissible displacements and virtual displacements.

**Example 4.1.1.** Before proceeding to a general statement of variational equations of motion, it is instructive to illustrate the foregoing definitions and properties with a simple example. A particle of mass  $m$  slides without friction on a straight wire, shown as the dashed line in Fig. 4.1.1, that moves in the plane according to the equation  $y = x + t$ . Motion of the particle must therefore satisfy the *time dependent constraint*

$$(x, y, t) = y - x - t = 0 \quad (4.1.10)$$

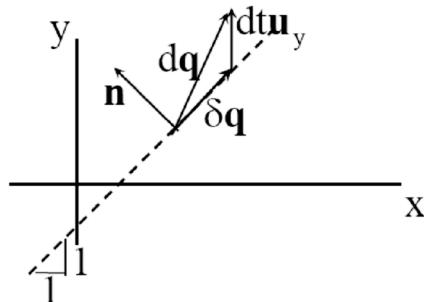


Figure 4.1.1. Particle on Moving Wire

With  $\mathbf{q} = [x \ y]^T$ , a *virtual displacement*  $\delta\mathbf{q}$  that is consistent with the constraint satisfies  ${}_{\mathbf{q}}\delta\mathbf{q} = \mathbf{n}^T\delta\mathbf{q} = 0$ , where  $\mathbf{n} = [-1 \ 1]^T$ , so

$$\delta \mathbf{q} = \alpha [1 \ 1]^T \quad (4.1.11)$$

where  $\alpha$  is arbitrary. An *admissible displacement* that is consistent with the constraint of Eq. (4.1.10) satisfies

$$_q d\mathbf{q} - dt = [-1 \ 1]d\mathbf{q} - dt = 0 \quad (4.1.12)$$

Substitution of  $d\mathbf{q} = \alpha [1 \ 1]^T + dt [0 \ 1]^T$  into Eq. (4.1.12) yields

$[-1 \ 1] (\alpha [1 \ 1]^T + dt [0 \ 1]^T) - dt = dt - dt = 0$ . Thus, with the unit vector  $\mathbf{u}_y = [0 \ 1]^T$ ,

$$d\mathbf{q} = \delta \mathbf{q} + dt \mathbf{u}_y \quad (4.1.13)$$

satisfies Eq. (4.1.12) for arbitrary  $\alpha$ ; i.e., it is an admissible displacement. As shown in Fig. 4.1.1, the kinematically admissible virtual displacement  $\delta \mathbf{q}$  is along the wire; i.e., tangent to the curve defined by Eq. (4.1.10). In contrast, the admissible displacement  $d\mathbf{q}$  is not tangent to the curve.

From the geometry of Fig. 4.1.1 and the fact that there is no friction force, the constraint reaction force  $\mathbf{F}^r = \gamma [-1 \ 1]^T$  must be orthogonal to the wire on which the particle slides, as shown in Fig. 4.1.2. Work done by the reaction force in the kinematically admissible virtual displacement  $\delta \mathbf{q}$  of Eq. (4.1.11) is

$$\delta \mathbf{q}^T \mathbf{F}^r = \alpha [1 \ 1]^T \gamma [-1 \ 1]^T = 0 \quad (4.1.14)$$

In contrast, work done by the reaction force in the admissible displacement  $d\mathbf{q}$  of Eq. (4.1.13) is

$$d\mathbf{q}^T \mathbf{F}^r = (\delta \mathbf{q}^T + dt [0 \ 1]^T) \gamma [-1 \ 1]^T = \gamma dt \neq 0 \quad (4.1.15)$$

This inconsistency is the reason that virtual displacements and not admissible displacements form the foundation for *variational methods in mechanics*.

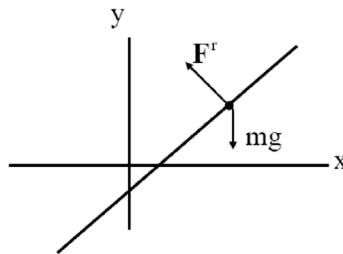


Figure 4.1.2. Forces Acting on Particle

### 4.1.3 *d'Alembert's Principle*

Example 4.1.1 is an illustration of the basis for *d'Alembert's principle* (Lanczos, 1949; Pars, 1965), which may be stated as follows:

***d'Alembert's Principle:*** Constraint reaction forces do no work under the action of kinematically admissible virtual displacements.

Example 4.1.1 shows that d'Alembert's Principle says nothing about work done by constraint forces under the action of admissible displacements. Lanczos (1949) argues that while d'Alembert's principle appears to be related to Newton's third law, as it involves constraint forces of *action and reaction*, it cannot be deduced from the third law. It is thus accepted as a fundamental tenant of the *variational approach to mechanics*. It is important to note that even if *friction* acts in constraints and is influenced by normal contact forces of constraint, those constraint forces do no work. The *force of friction* must be accounted for as an applied force whose magnitude is influenced by the workless normal *forces of constraint*, as is done in Chapter 8. d'Alembert's principle, together with Newton's equations of motion, is adopted in this text as a fundamental tenant of the variational approach to dynamics.

#### 4.1.4 Variational Equations of Motion for Systems of Particles

Newton's equation of motion for particle i, in a collection of np particles, is

$$m_i \ddot{r}_i = F_i^A + F_i^C \quad (4.1.16)$$

where  $F_i^A$  and  $F_i^C$  are *applied force* and *constraint reaction force* that act on particle i. For an arbitrary virtual displacement  $\delta r_i$ , Eq. (4.1.16) is equivalent to

$$\delta r_i^T (m_i \ddot{r}_i - F_i^A - F_i^C) = 0 \quad (4.1.17)$$

Using the generalized coordinates of Eq. (4.1.5) and defining

$$\begin{aligned} \mathbf{M} &= \text{diag}(m_1 \mathbf{I}_3 \ \cdots \ m_{np} \mathbf{I}_3) \\ \mathbf{F}^A &= [F_1^{AT} \ \cdots \ F_{np}^{AT}]^T \\ \mathbf{F}^C &= [F_1^{CT} \ \cdots \ F_{np}^{CT}]^T \end{aligned} \quad (4.1.18)$$

the sum of Eqs. (4.1.17) over all particles is

$$\sum_{i=1}^{np} \delta r_i^T (m_i \ddot{r}_i - F_i^A - F_i^C) = \delta q^T (\mathbf{M} \ddot{q} - \mathbf{F}^A - \mathbf{F}^C) = 0 \quad (4.1.19)$$

which must hold for arbitrary  $\delta q$ .

Under the condition that constraint reaction forces do no work under the action of a kinematically admissible virtual displacement; i.e.,

$$\delta q^T \mathbf{F}^C = 0 \quad (4.1.20)$$

for all  $\delta q$  that satisfy Eq. (4.1.9), Eq. (4.1.19) reduces to

$$\delta q^T (\mathbf{M} \ddot{q} - \mathbf{F}^A) = 0 \quad (4.1.21)$$

which must hold for all  $\delta q$  that satisfy Eq. (4.1.9). This condition is called the *first form of the fundamental equation of motion* by Pars (1965) and *d'Alembert's equation* by others. In any case, it is a *variational equation of motion* that is equivalent to Newton's equations of motion, provided Eq. (4.1.20) holds for all  $\delta q$  that satisfy Eq. (4.1.9).

---

**Example 4.1.2:** Newton's equations of motion for the particle of Example 4.1.1 that slides without friction on a wire, including the force of constraint, is

$$m\ddot{\mathbf{q}} + mg\mathbf{u}_y - (\gamma/\sqrt{2})[-1 \ 1]^T = \mathbf{0} \quad (4.1.22)$$

where  $\mathbf{q} = [x \ y]^T$  and  $\gamma$  is the magnitude of the *constraint reaction force*. The variational equation of motion of Eq. (4.1.21), for kinematically admissible virtual displacements of Eq. (4.1.11); i.e.,  $\delta\mathbf{q} = \alpha[1 \ 1]^T$  for arbitrary  $\alpha$ , is

$$\begin{aligned} \delta\mathbf{q}^T \left[ m\ddot{\mathbf{q}} + mg\mathbf{u}_y - (\gamma/\sqrt{2})[-1, 1]^T \right] &= \alpha[1 \ 1] \left[ m\ddot{\mathbf{q}} + mg\mathbf{u}_y - (\gamma/\sqrt{2})[-1 \ 1]^T \right] \\ &= \alpha[m(\ddot{x} + \ddot{y}) + mg] = 0 \end{aligned}$$

Since this must hold for arbitrary  $\alpha$ ,

$$(\ddot{x} + \ddot{y}) + g = 0 \quad (4.1.23)$$

Taking two derivatives of the constraint equation of Eq. (4.1.10),  $\ddot{y} - \ddot{x} = 0$ . Substituting this into Eq. (4.1.23),

$$\ddot{x} = -g/2 \quad (4.1.24)$$

For initial conditions at  $t^0 = 0$ ,  $\mathbf{q}^0 = \mathbf{0}$  and  $\dot{\mathbf{q}}^0 = [-1 \ 0]^T$  that are consistent with Eq. (4.1.10) and its time derivative  $\dot{y} - \dot{x} - 1 = 0$ , the solution of Eq. (4.1.24) is

$$\begin{aligned} \dot{x} &= -1 - gt/2 \\ x &= -t - gt^2/4 \end{aligned} \quad (4.1.25)$$

From the constraint equation of Eq. (4.1.10) and its derivatives,

$$\begin{aligned} \ddot{y} &= -g/2 \\ \dot{y} &= -gt/2 \\ y &= -gt^2/4 \end{aligned} \quad (4.1.26)$$

Equations (4.1.24), (4.1.25), and (4.1.26) comprise the solution of the variational equation of motion.

Substituting the solution of the variational equations into Newton's equations of motion of Eq. (4.1.22),

$$\begin{bmatrix} -mg/2 + \gamma/\sqrt{2} \\ -mg/2 + mg - \gamma/\sqrt{2} \end{bmatrix} = \mathbf{0} \quad (4.1.27)$$

which is satisfied with  $\gamma = mg/\sqrt{2}$ . This verifies that the solution of the variational equation of motion satisfies Newton's equations of motion.

---

Models of rigid bodies are next created, using collections of particles that are subjected to distance constraints.

**Example 4.1.3:** The pair of particles  $m_1 \neq 0 \neq m_2$  shown in the x-y plane in Fig. 4.1.3 is connected by a massless bar of unit length to form a body, sometimes called a *dumbbell*. With unit vector  $\mathbf{u}'_x$  from particle one to particle two along the  $x'$ -axis, vectors that locate the particles in the inertial x-y plane and give their accelerations and virtual displacements are

$$\begin{aligned}\mathbf{r}_1 &= \mathbf{r} \\ \mathbf{r}_2 &= \mathbf{r} + \mathbf{A}(\phi)\mathbf{u}'_x \\ \ddot{\mathbf{r}}_1 &= \ddot{\mathbf{r}} \\ \ddot{\mathbf{r}}_2 &= \ddot{\mathbf{r}} + \ddot{\phi}\mathbf{P}\mathbf{A}(\phi)\mathbf{u}'_x - \dot{\phi}^2\mathbf{A}(\phi)\mathbf{u}'_x \\ \delta\mathbf{r}_1 &= \delta\mathbf{r} \\ \delta\mathbf{r}_2 &= \delta\mathbf{r} + \delta\phi\mathbf{P}\mathbf{A}(\phi)\mathbf{u}'_x\end{aligned}\tag{4.1.28}$$

where Eqs. (2.3.40) and (2.3.43) have been used.

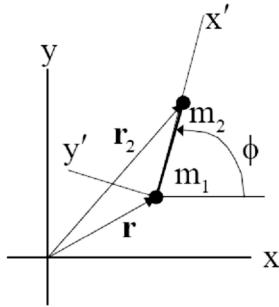


Figure 4.1.3 Two Particle Body in x-y Plane

Expanding the sum in Eq. (4.1.21) for this system, the variational equation of motion is

$$\delta\mathbf{r}^T \left( m_1 \ddot{\mathbf{r}} - \mathbf{F}_1^A \right) + \left( \delta\mathbf{r}^T + \delta\phi \mathbf{u}'_x^T \mathbf{A}^T(\phi) \mathbf{P}^T \right) \left[ m_2 \left( \ddot{\mathbf{r}} + \ddot{\phi} \mathbf{P} \mathbf{A}(\phi) \mathbf{u}'_x - \dot{\phi}^2 \mathbf{A}(\phi) \mathbf{u}'_x \right) - \mathbf{F}_2^A \right] = 0 \tag{4.1.29}$$

Define

$$\begin{aligned}\mathbf{m} &= m_1 + m_2 \\ \mathbf{s}'^c &= \frac{m_2}{m} \mathbf{u}'_x \\ \mathbf{J}' &= m_2 \mathbf{u}'_x^T \mathbf{u}'_x = m_2 \\ \mathbf{F}^A &= \mathbf{F}_1^A + \mathbf{F}_2^A \\ \mathbf{n}'^A &= \mathbf{u}'_x^T \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{F}_2^A\end{aligned}\tag{4.1.30}$$

where  $\mathbf{s}'^c$  locates the center of mass of the dumbbell and  $\mathbf{J}'$  is its *moment of inertia* relative to the origin of the  $x'$ - $y'$  frame. Expanding Eq. (4.1.29), using Eqs. (4.1.28) and (4.1.30),

$$\delta \mathbf{r}^T \left( m \ddot{\mathbf{r}} + m \left( \dot{\phi} \mathbf{P} \mathbf{A}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi) \right) \mathbf{s}'^c - \mathbf{F}^A \right) + \delta \phi \left( m \mathbf{s}'^{cT} \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} + J' \ddot{\phi} - n'^A \right) = 0 \quad (4.1.31)$$

Since arbitrary virtual displacements and rotations of the  $x'-y'$  frame leave the unit distance constraint between the masses unchanged, if no additional constraints act, Eq. (4.1.31) must hold for arbitrary  $\delta \mathbf{r}$  and  $\delta \phi$ . Their coefficients in Eq. (4.1.31) must therefore be zero; i.e.,

$$\begin{aligned} m \ddot{\mathbf{r}} + m \left( \dot{\phi} \mathbf{P} \mathbf{A}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi) \right) \mathbf{s}'^c &= \mathbf{F}^A \\ m \mathbf{s}'^{cT} \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} + J' \ddot{\phi} &= n'^A \end{aligned} \quad (4.1.32)$$

which are *second order ODE* for motion of the body defined by the pair of constrained particles. These are a special case of *planar rigid body equations of motion* derived in Section 4.2.

---

**Example 4.1.4:** A special case of a *spatial rigid body* is the three particle model shown in Fig. 4.1.4, where  $m_1 \neq 0$  is located at the origin of the  $x'-y'-z'$  frame and  $m_2 \neq 0 \neq m_3$  are located on the  $x'$  and  $y'$  axes, one unit from the origin at the origin of the  $x'-y'-z'$  frame. Using Eqs. (2.4.22), (2.4.57), and (2.4.67), with unit vectors  $\mathbf{u}'_x$  and  $\mathbf{u}'_y$  along the  $x'$  and  $y'$  axes, positions, virtual displacements, and accelerations of the three particles are

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{r} \\ \mathbf{r}_2 &= \mathbf{r} + \mathbf{A}(\mathbf{p}) \mathbf{u}'_x \\ \mathbf{r}_3 &= \mathbf{r} + \mathbf{A}(\mathbf{p}) \mathbf{u}'_y \\ \delta \mathbf{r}_1 &= \delta \mathbf{r} \\ \delta \mathbf{r}_2 &= \delta \mathbf{r} - \mathbf{A}(\mathbf{p}) \tilde{\mathbf{u}}'_x \delta \boldsymbol{\pi}' \\ \delta \mathbf{r}_3 &= \delta \mathbf{r} - \mathbf{A}(\mathbf{p}) \tilde{\mathbf{u}}'_y \delta \boldsymbol{\pi}' \\ \ddot{\mathbf{r}}_1 &= \ddot{\mathbf{r}} \\ \ddot{\mathbf{r}}_2 &= \ddot{\mathbf{r}} - \mathbf{A}(\mathbf{p}) \tilde{\mathbf{u}}'_x \dot{\boldsymbol{\omega}}' + \mathbf{A}(\mathbf{p}) \tilde{\boldsymbol{\omega}}' \tilde{\boldsymbol{\omega}}' \mathbf{u}'_x \\ \ddot{\mathbf{r}}_3 &= \ddot{\mathbf{r}} - \mathbf{A}(\mathbf{p}) \tilde{\mathbf{u}}'_y \dot{\boldsymbol{\omega}}' + \mathbf{A}(\mathbf{p}) \tilde{\boldsymbol{\omega}}' \tilde{\boldsymbol{\omega}}' \mathbf{u}'_y \end{aligned} \quad (4.1.33)$$

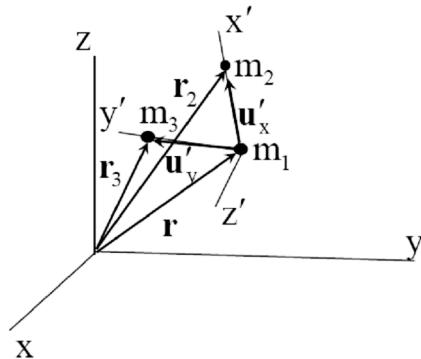


Figure 4.1.4 Three-Particle Body in Space

Writing the sum of terms in Eq. (4.1.21) over the three particles, using Eqs (4.1.33), expanding products, and collecting terms yields the variational equation of motion,

$$\begin{aligned} \delta\mathbf{r}^T & \left[ (m_1 + m_2 + m_3)\ddot{\mathbf{r}} - \mathbf{A}(\mathbf{p})(m_2\tilde{\mathbf{u}}'_x + m_3\tilde{\mathbf{u}}'_y)\dot{\boldsymbol{\omega}}' + \mathbf{A}(\mathbf{p})\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'(m_2\mathbf{u}'_x + m_3\mathbf{u}'_y) - (\mathbf{F}_1^A + \mathbf{F}_2^A + \mathbf{F}_3^A) \right] \\ & + \delta\boldsymbol{\pi}'^T \left[ (m_2\tilde{\mathbf{u}}'_x + m_3\tilde{\mathbf{u}}'_y)\mathbf{A}^T(\mathbf{p})\ddot{\mathbf{r}} - (m_2\tilde{\mathbf{u}}'_x\tilde{\mathbf{u}}'_x + m_3\tilde{\mathbf{u}}'_y\tilde{\mathbf{u}}'_y)\dot{\boldsymbol{\omega}}' + m_2\tilde{\mathbf{u}}'_x\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{u}'_x + m_3\tilde{\mathbf{u}}'_y\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{u}'_y \right] = 0 \\ & - \tilde{\mathbf{u}}'_x\mathbf{A}^T(\mathbf{p})\mathbf{F}_2^A - \tilde{\mathbf{u}}'_y\mathbf{A}^T(\mathbf{p})\mathbf{F}_3^A \end{aligned} \quad (4.1.34)$$

Using Eqs. (2.1.26), (2.1.27), and (2.1.29),

$$\tilde{\mathbf{u}}'_x\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{u}'_x = -\tilde{\mathbf{u}}'_x\tilde{\boldsymbol{\omega}}'\tilde{\mathbf{u}}'_x\boldsymbol{\omega}' = -\tilde{\boldsymbol{\omega}}'\tilde{\mathbf{u}}'_x\tilde{\mathbf{u}}'_x\boldsymbol{\omega}' - \boldsymbol{\omega}'\mathbf{u}'_x^T\tilde{\mathbf{u}}'_x\boldsymbol{\omega}' - \mathbf{u}'_x\boldsymbol{\omega}'^T\tilde{\mathbf{u}}'_x\boldsymbol{\omega}' = -\tilde{\boldsymbol{\omega}}'\tilde{\mathbf{u}}'_x\tilde{\mathbf{u}}'_x\boldsymbol{\omega}' \quad (4.1.35)$$

Similarly, for the corresponding term involving  $\mathbf{u}'_y$ ,  $\tilde{\mathbf{u}}'_y\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{u}'_y = -\tilde{\boldsymbol{\omega}}'\tilde{\mathbf{u}}'_y\tilde{\mathbf{u}}'_y\boldsymbol{\omega}'$ . Using these results and the definitions

$$\begin{aligned} \mathbf{m} & \equiv \mathbf{m}_1 + \mathbf{m}_2 + \mathbf{m}_3 \\ \mathbf{J}' & \equiv - (m_2\tilde{\mathbf{u}}'_x\tilde{\mathbf{u}}'_x + m_3\tilde{\mathbf{u}}'_y\tilde{\mathbf{u}}'_y) \\ \mathbf{s}'^c & \equiv (1/m)(m_2\mathbf{u}'_x + m_3\mathbf{u}'_y) \\ \mathbf{F}^A & \equiv \mathbf{F}_1^A + \mathbf{F}_2^A + \mathbf{F}_3^A \\ \mathbf{n}'^A & \equiv \tilde{\mathbf{u}}'_x\mathbf{A}^T\mathbf{F}_2^A + \tilde{\mathbf{u}}'_y\mathbf{A}^T\mathbf{F}_3^A \end{aligned} \quad (4.1.36)$$

Eq. (4.1.34) reduces to

$$\begin{aligned} \delta\mathbf{r}^T & \left( m\ddot{\mathbf{r}} - m\mathbf{A}(\mathbf{p})\tilde{\mathbf{s}}'^c\dot{\boldsymbol{\omega}}' + m\mathbf{A}(\mathbf{p})\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{s}'^c - \mathbf{F}^A \right) \\ & + \delta\boldsymbol{\pi}'^T \left( m\tilde{\mathbf{s}}'^c\mathbf{A}^T(\mathbf{p})\ddot{\mathbf{r}} + \mathbf{J}'\dot{\boldsymbol{\omega}}' + \tilde{\boldsymbol{\omega}}'\mathbf{J}'\boldsymbol{\omega}' - \mathbf{n}'^A \right) = 0 \end{aligned} \quad (4.1.37)$$

Equation (4.1.37) must hold for arbitrary  $\delta\mathbf{r}$  and  $\delta\boldsymbol{\pi}'$ , since virtual displacement and rotation of the  $x'$ - $y'$ - $z'$  frame do not change the distance between particles and no additional constraints act on the system. Thus, the coefficients of  $\delta\mathbf{r}$  and  $\delta\boldsymbol{\pi}'$  in Eq. (4.1.34) must be zero; i.e.,

$$\begin{aligned} m\ddot{\mathbf{r}} - m\mathbf{A}(\mathbf{p})\tilde{\mathbf{s}}'^c\dot{\boldsymbol{\omega}}' + m\mathbf{A}(\mathbf{p})\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\mathbf{s}'^c &= \mathbf{F}^A \\ m\tilde{\mathbf{s}}'^c\mathbf{A}^T(\mathbf{p})\ddot{\mathbf{r}} + \mathbf{J}'\dot{\boldsymbol{\omega}}' + \tilde{\boldsymbol{\omega}}'\mathbf{J}'\boldsymbol{\omega}' - \mathbf{n}'^A &= 0 \end{aligned} \quad (4.1.38)$$

which are *second order ODE of motion* of the three-particle model of a rigid body in space. Expansion of the second of Eqs. (4.1.36) with  $m_2 \neq 0 \neq m_3$  yields the inertia matrix

$$\mathbf{J}' = \begin{bmatrix} m_3 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_2 + m_3 \end{bmatrix}$$

which is symmetric and positive definite. As will be seen in Section 4.3, Eqs. (4.1.38) are the well-known *Newton-Euler equations of motion* of a rigid body in space, written in terms of generalized coordinates of a body-fixed *noncentroidal reference frame*.

Newton's three laws that govern the motion of particles, published in 1687, form the foundation for dynamics of multiparticle systems. In 1788, Lagrange published work contributing the concept of virtual displacement and associated virtual work that is the basis for the modern theory of variational mechanics used in this text. This theory is encapsulated in the concise statement of d'Alembert's Principle: "Constraint reaction forces do no work under the action of kinematically admissible virtual displacements," published by d'Alembert in 1743.

Using d'Alembert's principle and Newton's laws, two and three particle models of rigid planar and spatial bodies, respectively, are formulated using kinematic relationships presented in Chapter 3. They yield equations of motion that are identical in form to more general rigid body models derived in Sections 4.2 and 4.3.

## 4.2 Equations of Motion for Planar Rigid Bodies

The equations of motion of a planar rigid body are derived, beginning with Newton's laws for a particle and a rigid body model that is comprised of a constrained system of particles. The variational approach introduced in Section 4.1 for particles provides a self-contained derivation of equations of motion for rigid bodies.

Consider the planar rigid body shown in Fig. 4.2.1, with a body-fixed  $x'$ - $y'$  reference frame that is located and oriented relative to the inertial  $x$ - $y$  reference frame by the vector  $\mathbf{r}$  and angle of rotation  $\phi$ . A *differential mass*  $dm(P)$  at point P is located in the body by the vector  $\mathbf{s}^P = \mathbf{A}\mathbf{s}'^P$ , where  $\mathbf{s}'^P$  is its fixed location in the body. Forces that act on this differential mass include *external force*  $\mathbf{F}(P)$  per unit of mass  $dm(P)$  at point P and *internal force*  $\mathbf{f}(P,R)$  per units of mass  $dm(P)$  and  $dm(R)$  located at points P and R, as shown in Fig. 4.2.1.

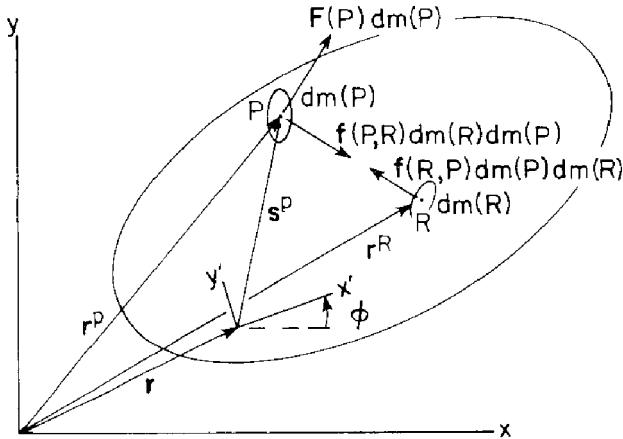


Figure 4.2.1 Forces Acting on a Planar Rigid Body

As a model of a rigid body, let a distance constraint (a massless bar) act between each pair of differential masses in the body, thought of as particles. With this model, internal forces  $\mathbf{f}(P, R) dm(R)$  and  $\mathbf{f}(R, P) dm(P)$  of interaction on  $dm(P)$  and  $dm(R)$  act along the massless bar between points P and R, due to the *distance constraint* between particles  $dm(P)$  and  $dm(R)$ , so they are equal in magnitude, opposite in direction, and *collinear*. While  $\mathbf{f}(P, R) dm(R) dm(P) = -\mathbf{f}(R, P) dm(P) dm(R)$  follows from Newton's third law, in general Newton said nothing about collinearity in his law of action and reaction. Collinearity of  $\mathbf{f}(P, R)$  and  $\mathbf{f}(R, P)$  in the model used here for a rigid body has been called the *strong form of Newton's second law* (Goldstein, 1980) and is adequate to represent *internal gravitational attraction*. The model used here requires that, if forces of interaction occur in a body due to electric or magnetic fields effects that violate the collinearity assumption, they must be accounted for as external forces.

*Newton's equation of motion* for differential mass  $dm(P)$  is

$$\ddot{\mathbf{r}}^P dm(P) = \mathbf{F}(P) dm(P) + \left[ \int_m \mathbf{f}(P, R) dm(R) \right] dm(P) \quad (4.2.1)$$

where integration of the internal forces that act on  $dm(P)$  is taken over the total mass  $m$  of the body. Equation (4.2.1) is difficult to use, since it explicitly involves internal forces that act within the body. Furthermore, it is written for every differential element of mass, yielding an infinite number of equations of motion. A unifying concept that resolves this dilemma and yields broadly applicable tools for mechanical system dynamics is the variational approach that is presented in Section 4.1.

Let  $\delta\mathbf{r}^P$  denote a *virtual displacement* of point P; i.e., an *infinitesimal variation* in the location of point P that is kinematically admissible relative to the distance constraints between particles, with time held fixed, as introduced in Section 4.1. Multiplying both sides of Eq. (4.2.1) on the left by  $\delta\mathbf{r}^{PT}$  and integrating over the total mass  $m$  of the body yields

$$\int_m \delta\mathbf{r}^{PT} \ddot{\mathbf{r}}^P dm(P) = \int_m \delta\mathbf{r}^{PT} \mathbf{F}(P) dm(P) + \int_m \int_m \delta\mathbf{r}^{PT} \mathbf{f}(P, R) dm(R) dm(P) \quad (4.2.2)$$

which must hold for arbitrary  $\delta\mathbf{r}^P$ . Manipulation of the double integral appearing on the right of Eq. (4.2.2) yields

$$\begin{aligned} \int_m \int_m \delta\mathbf{r}^{PT} \mathbf{f}(P, R) dm(R) dm(P) &= \frac{1}{2} \int_m \int_m [\delta\mathbf{r}^{PT} \mathbf{f}(P, R) + \delta\mathbf{r}^{PT} \mathbf{f}(R, P)] dm(R) dm(P) \\ &= \frac{1}{2} \int_m \int_m \delta\mathbf{r}^{PT} \mathbf{f}(P, R) dm(R) dm(P) + \frac{1}{2} \int_m \int_m \delta\mathbf{r}^{RT} \mathbf{f}(R, P) dm(P) dm(R) \end{aligned} \quad (4.2.3)$$

where, since P and R are variables of integration, they may be renamed without affecting the value of the integral. Furthermore, the order of carrying out the integration can be reversed without affecting the value of the integral and  $\mathbf{f}(P, R) dm(R) dm(P) = -\mathbf{f}(R, P) dm(P) dm(R)$  can be used to obtain

$$\int_m \int_m \delta\mathbf{r}^{PT} \mathbf{f}(P, R) dm(R) dm(P) = \frac{1}{2} \int_m \int_m \delta\mathbf{r}^{PT} \mathbf{f}(P, R) dm(R) dm(P) - \frac{1}{2} \int_m \int_m \delta\mathbf{r}^{RT} \mathbf{f}(P, R) dm(R) dm(P) \quad (4.2.4)$$

Combining the integrals on the right and using the fact from differential calculus that  $\delta\mathbf{r}^P - \delta\mathbf{r}^R = \delta(\mathbf{r}^P - \mathbf{r}^R)$ ,

$$\int_m \int_m \mathbf{r}^{PT} \mathbf{f}(P, R) dm(R) dm(P) = \frac{1}{2} \int_m \int_m (\mathbf{r}^P - \mathbf{r}^R)^T \mathbf{f}(P, R) dm(R) dm(P) \quad (4.2.5)$$

Recall that the definition of a rigid body requires that the distance between points P and R is constant; i.e.,

$$(\mathbf{r}^P - \mathbf{r}^R)^T (\mathbf{r}^P - \mathbf{r}^R) = c \quad (4.2.6)$$

Taking the differential of both sides of Eq. (4.2.6), using the rules of differential calculus, and multiplying both sides by 1/2 yields

$$\delta(\mathbf{r}^P - \mathbf{r}^R)^T (\mathbf{r}^P - \mathbf{r}^R) = 0 \quad (4.2.7)$$

Since  $\mathbf{f}(P,R)$  acts along the vector  $(\mathbf{r}^P - \mathbf{r}^R)$  between points P and Q,  $\delta(\mathbf{r}^P - \mathbf{r}^R)^T \mathbf{f}(P,R) = 0$ . Thus, the integrand on the right of Eq. (4.2.5) is zero and Eq. (4.2.2) simplifies to

$$\int_m \delta \mathbf{r}^{PT} \dot{\mathbf{r}}^P dm(P) = \int_m \delta \mathbf{r}^{PT} \mathbf{F}(P) dm(P) \quad (4.2.8)$$

which must hold for all  $\delta \mathbf{r}^P$  that are consistent with *rigid-body motion*.

It is important to note that while Eq. (4.2.2) holds for arbitrary  $\delta \mathbf{r}^P$ , Eq. (4.2.8) does not. The price for eliminating internal forces in the second term on the right of Eq. (4.2.2) is that  $\delta \mathbf{r}^P$  must be consistent with the definition of rigid-body motion. To take full advantage of Eq. (4.2.8), the virtual displacement  $\delta \mathbf{r}^P$  of point P must be written in terms of variations in generalized coordinates of the body reference frame. Using Eqs. (2.3.43) and (2.3.47),

$$\ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} + \dot{\phi} \mathbf{PA}(\phi) \mathbf{s}'^P - \dot{\phi}^2 \mathbf{A}(\phi) \mathbf{s}'^P \quad (4.2.9)$$

$$\delta \mathbf{r}^P = \delta \mathbf{r} + \delta \phi \mathbf{PA}(\phi) \mathbf{s}'^P \quad (4.2.10)$$

Equations (4.2.9) and (4.2.10) may be substituted in Eq. (4.2.8) to obtain the *variational equation of planar motion*,

$$\begin{aligned} & \delta \mathbf{r}^T \ddot{\mathbf{r}} \int_m dm(P) + \delta \mathbf{r}^T (\ddot{\phi} \mathbf{PA}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi)) \int_m \mathbf{s}'^P dm(P) + \delta \phi \int_m \mathbf{s}'^P T dm(P) \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} \\ & + \delta \phi \ddot{\phi} \int_m \mathbf{s}'^P T \mathbf{s}'^P dm(P) - \delta \phi \dot{\phi}^2 \int_m \mathbf{s}'^P T \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{A}(\phi) \mathbf{s}'^P dm(P) \\ & = \delta \mathbf{r}^T \int_m \mathbf{F}(P) dm(P) + \delta \phi \int_m \mathbf{s}'^P T \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{F}(P) dm(P) \end{aligned} \quad (4.2.11)$$

where terms that do not depend on location within the body are taken outside integrals. While Eq. (4.2.8) holds only for  $\delta \mathbf{r}^P$  that are consistent with the definition of rigid-body motion, Eq. (4.2.11) holds for any  $\delta \mathbf{r}$  and  $\delta \phi$  that are consistent with external constraints that act on the body. This is true since  $\delta \mathbf{r}^P$  given by Eq. (4.2.10) is consistent with Eq. (4.2.7), which defines rigid-body motion.

The form of the equations of motion of a planar rigid body in Eq. (4.2.11) can be substantially simplified. The first integral on the left is the total mass of the body; i.e.,

$$m = \int_m dm(P) \quad (4.2.12)$$

By definition of *centroid*, the vector from the origin of the x'-y' frame to the centroid is

$$\mathbf{s}'^c \equiv \frac{1}{m} \int_m \mathbf{s}'^P dm(P) \quad (4.2.13)$$

and the *polar moment of inertia* of the body relative to the origin of the x'-y' frame is

$$J' \equiv \int_m \mathbf{s}'^P T \mathbf{s}'^P dm(P) \quad (4.2.14)$$

Using  $\mathbf{P}^T = -\mathbf{P}$  and Eq. (2.3.33),  $\mathbf{s}'^{PT} \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{A}(\phi) \mathbf{s}'^P = (\mathbf{P} \mathbf{s}^P)^T \mathbf{s}^P = 0$ , since  $\mathbf{P} \mathbf{s}^P$  is orthogonal to  $\mathbf{s}^P$ . Thus, the last term on the left of Eq. (4.2.11) is  $\int_m \mathbf{s}'^{PT} \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{A}(\phi) \mathbf{s}'^P dm(P) = 0$ . The first integral on the right of Eq. (4.2.11) is the total *external force* that acts on the body,

$$\mathbf{F} \equiv \int_m \mathbf{F}(P) dm(P) \quad (4.2.15)$$

and the second integral is the total *external torque* that acts about the origin of the  $x'-y'$  frame, due to the distributed applied force; i.e.,

$$\begin{aligned} \mathbf{n}' &\equiv \int_m \left[ -s_y^P F_{x'}(P) + s_{x'}^P F_y(P) \right] dm(P) \\ &= \int_m \left( \mathbf{P} \mathbf{s}'^P \right)^T \mathbf{F}'(P) dm(P) = \int_m \mathbf{s}'^{PT} \mathbf{P}^T \mathbf{A}^T(\phi) \mathbf{F}(P) dm(P) \\ &= \int_m \mathbf{s}'^{PT} \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{F}(P) dm(P) \end{aligned} \quad (4.2.16)$$

where  $\mathbf{A}^T(\phi) \mathbf{P}^T = \mathbf{P}^T \mathbf{A}^T(\phi)$  from Eq. (2.3.33) has been used.

If  $\mathbf{F}(P) = g\mathbf{a}$  is the *gravitational force* in the direction defined by unit vector  $\mathbf{a}$ , then Eqs. (4.2.15) and (4.2.16) yield

$$\begin{aligned} \mathbf{F}_g &= \int_m g \mathbf{a} dm(P) = g \mathbf{a} \int_m dm(P) = mg \mathbf{a} \\ \mathbf{n}'_g &= \int_m \mathbf{s}'^{PT} \mathbf{A}^T(\phi) \mathbf{P}^T g \mathbf{a} dm(P) = \int_m \mathbf{s}'^{PT} dm(P) \mathbf{A}^T(\phi) \mathbf{P}^T g \mathbf{a} = mgs'^c \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{a} \end{aligned} \quad (4.3.17)$$

Substituting the foregoing results into Eq. (4.2.11) yields the *noncentroidal planar variational equation of motion*,

$$\delta \mathbf{r}^T \left[ m \ddot{\mathbf{r}} + m \left( \dot{\phi} \mathbf{PA}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi) \right) \mathbf{s}'^c - \mathbf{F} \right] + \delta \phi \left[ ms'^c \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} + \ddot{\phi} \mathbf{J}' - \mathbf{n}' \right] = 0 \quad (4.2.18)$$

In terms of body generalized coordinates  $\mathbf{q} = \begin{bmatrix} \mathbf{r}^T & \phi \end{bmatrix}^T$ , this is

$$\delta \mathbf{q}^T [\mathbf{M} \ddot{\mathbf{q}} - \mathbf{S} - \mathbf{Q}] = 0 \quad (4.2.19)$$

where

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} m\mathbf{I} & m\mathbf{PA}(\phi)\mathbf{s}'^c \\ ms'^c \mathbf{A}^T(\phi) \mathbf{P}^T & \mathbf{J}' \end{bmatrix} \\ \mathbf{S} &= \begin{bmatrix} \dot{\phi}^2 m \mathbf{A}(\phi) \mathbf{s}'^c \\ 0 \end{bmatrix} \\ \mathbf{Q} &= \begin{bmatrix} \mathbf{F} \\ \mathbf{n}' \end{bmatrix} \end{aligned} \quad (4.2.20)$$

If the origin of the  $x'$ - $y'$  frame is selected to be at the centroid of the body; i.e.,  $\mathbf{s}'^c = \mathbf{0}$ , Eq. (4.2.18) reduces to the *centroidal planar variational equation of motion*,

$$\delta\mathbf{r}^T [m\ddot{\mathbf{r}} - \mathbf{F}] + \delta\phi [\ddot{\phi}\mathbf{J}' - \mathbf{n}'] = 0 \quad (4.2.21)$$

In matrix form, this is the same as Eqs. (4.2.19) and (4.2.20), but with  $\mathbf{M} = m\mathbf{I}$  diagonal and  $\mathbf{S} = \mathbf{0}$ , since  $\mathbf{s}'^c = \mathbf{0}$ .

If no external constraints act on the body,  $\delta\mathbf{r}$  and  $\delta\phi$  are arbitrary, so their coefficients in Eqs. (4.2.18) and (4.2.21) must be zero. This yields the *planar body differential equations of motion*

$$\begin{aligned} m\ddot{\mathbf{r}} + m(\ddot{\phi}\mathbf{PA}(\phi) - \dot{\phi}^2\mathbf{A}(\phi))\mathbf{s}'^c &= \mathbf{F} \\ m\mathbf{s}'^{cT}\mathbf{A}^T(\phi)\mathbf{P}^T\ddot{\mathbf{r}} + \ddot{\phi}\mathbf{J}' &= \mathbf{n}' \end{aligned} \quad (4.2.22)$$

for a *noncentroidal reference frame*, and

$$\begin{aligned} m\ddot{\mathbf{r}} &= \mathbf{F} \\ J'\ddot{\phi} &= n' \end{aligned} \quad (4.2.23)$$

for a *centroidal reference frame*. Note that Eq. (4.2.22) is exactly the same as Eq. (4.1.35) in Example 4.1.3 for motion of a rigid body that is made up of two particles in a plane that are connected by a single distance constraint.

It is important to remember that Eqs. (4.2.22) and (4.2.23) are valid only if the force vector  $\mathbf{F}$  and the torque  $\mathbf{n}'$  that act on the body represent all forces and torques that act on the body. A manipulation identical to that carried out in obtaining Eq. (4.2.16) shows that the counterclockwise torque about the origin of the  $x'$ - $y'$  frame due to a force  $\mathbf{F}^P$  applied at point P in the  $x'$ - $y'$  frame, located by the vector  $\mathbf{s}'^P$  is given by

$$\mathbf{n}' = \mathbf{s}'^{PT}\mathbf{A}^T(\phi)\mathbf{P}^T\mathbf{F}^P = \mathbf{s}'^{PT}\mathbf{P}^T\mathbf{A}^T(\phi)\mathbf{F}^P = \mathbf{s}'^{PT}\mathbf{P}^T\mathbf{F}'^P \quad (4.2.24)$$

Since it is clear that the centroidal formulation of the equations of motion is simpler than the noncentroidal formulation, one might ask; why would the noncentroidal formulation ever be used? One reason is that analysis of the effects of *design variation* requires use of a noncentroidal formulation. If  $\mathbf{b} = [b_1 \ \dots \ b_k]^T$  is a vector of *design variables*, the location of the centroid will almost always be a function of design; i.e.,  $\mathbf{s}'^c = \mathbf{s}'^c(\mathbf{b})$ . If a centroidal reference frame is selected for a nominal design  $\mathbf{b}^0$ ; i.e.,  $\mathbf{s}'^c(\mathbf{b}^0) = \mathbf{0}$ , in general,  $\frac{\partial \mathbf{s}'^c}{\partial \mathbf{b}}(\mathbf{b}^0) \neq \mathbf{0}$  and design *sensitivity analysis* will be in error if the centroidal form of the equations of motion is used.

Variational and differential equations of motion for a planar rigid body are derived for a body that is modeled as a constrained system of particles, using only d'Alembert's principle and Newton's laws. This takes advantage of the fact that the strong form of Newton's third law is valid for the model used; i.e., constraint forces that act on particles that are connected by a massless rod are equal in magnitude, opposite in direction, and collinear.

## Key Formulas

$$\delta \mathbf{r}^T [ m \ddot{\mathbf{r}} + m( \ddot{\phi} \mathbf{P} \mathbf{A}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi) ) \mathbf{s}'^c - \mathbf{F} ] + \delta \phi [ m s'^{cT} \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} + \ddot{\phi} \mathbf{J}' - \mathbf{n}' ] = 0 \quad (4.2.18)$$

$$\delta \mathbf{r}^T [ m \ddot{\mathbf{r}} - \mathbf{F} ] + \delta \phi [ \ddot{\phi} \mathbf{J}' - \mathbf{n}' ] = 0 \quad (4.2.21)$$

$$m \ddot{\mathbf{r}} + m( \ddot{\phi} \mathbf{P} \mathbf{A}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi) ) \mathbf{s}'^c = \mathbf{F} \quad m s'^{cT} \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} + \ddot{\phi} \mathbf{J}' = \mathbf{n}' \quad (4.2.22)$$

$$m \ddot{\mathbf{r}} = \mathbf{F} \quad \ddot{\phi} \mathbf{J}' = \mathbf{n}' \quad (4.2.23)$$

### 4.3 Equations of Motion for Spatial Rigid Bodies

Consider the *spatial rigid body* shown in Fig. 4.3.1 that is located in space by a vector  $\mathbf{r}$  and Euler parameter generalized coordinates that define location and orientation of the  $x'$ - $y'$ - $z'$  body-fixed reference frame, relative to an  $x$ - $y$ - $z$  *inertial reference frame*. The spatial formulation of equations of motion presented here is almost identical to that presented in Example 4.1.4 for constrained particles in space and in Section 4.2 for planar bodies, prior to the point at which *angular velocity* is introduced.

#### 4.3.1 Angular Velocity Form of Equations of Motion

A differential mass  $dm(P)$  at a typical point  $P$  in the rigid body is located by vector  $\mathbf{s}^P = \mathbf{A}(\mathbf{p})\mathbf{s}'^P$ , where  $\mathbf{s}'^P$  is fixed in the body. As a model of a rigid body, let a *distance constraint* act between pairs of differential masses, as in Fig. 4.3.1.

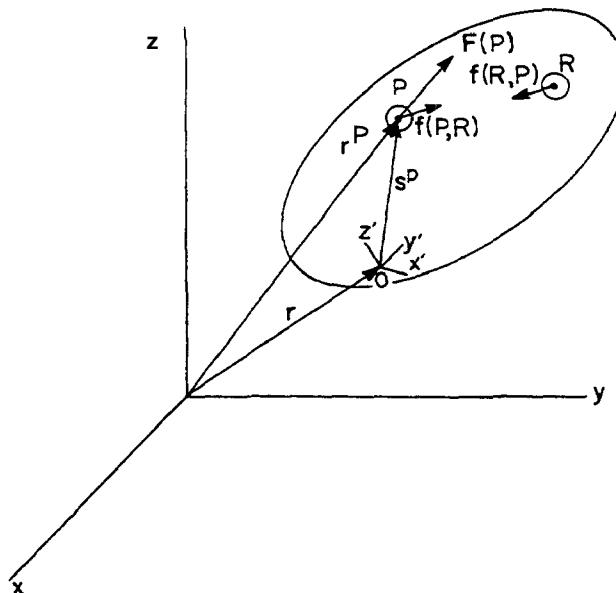


Figure 4.3.1 Forces Acting on a Rigid Body in Space

Forces that act on a differential mass at point  $P$  include the *external force*  $\mathbf{F}(P)$  per unit of mass at point  $P$  and the *internal force*  $\mathbf{f}(P, R)$  per units of masses  $dm(P)$  and  $dm(R)$  at points  $P$  and  $R$ , as shown in Fig. 4.3.1. Internal forces in this model of a rigid body are due only to gravitational interaction and distance constraints. By *Newton's third law*, reaction forces acting on  $dm(P)$  and  $dm(R)$  are equal in magnitude and opposite in direction; i.e.,

$$\mathbf{f}(P, R) dm(P) dm(R) = -\mathbf{f}(R, P) dm(P) dm(R) \quad (4.3.1)$$

In general, as noted in Sections 4.1 and 4.2, Newton's third law does not require reaction forces to be collinear. For gravitational interaction and constraint forces due to a distance constraint, however, they are collinear.

Newton's equation of motion for differential mass  $dm(P)$  is

$$\ddot{\mathbf{r}}^P dm(P) = \mathbf{F}(P) dm(P) + \left( \int_m \mathbf{f}(P, R) dm(R) \right) dm(P) \quad (4.3.2)$$

where integration of the internal force  $\mathbf{f}(P, R)$  with respect to differential mass at point  $R$  is taken over the entire body. Let  $\mathbf{r}^P$  denote an arbitrary *infinitesimal variation*, or *virtual displacement*, of point  $P$ . Multiplying both sides of Eq. (4.3.2) on the left by  $\mathbf{r}^{P^T}$  and integrating over the total mass of the body yields

$$\int_m \mathbf{r}^{P^T} \ddot{\mathbf{r}}^P dm(P) = \int_m \mathbf{r}^{P^T} \mathbf{F}(P) dm(P) + \iint_m \mathbf{r}^{P^T} \mathbf{f}(P, R) dm(R) dm(P) \quad (4.3.3)$$

where  $\mathbf{r}^{P^T}$  may be taken inside the integral on the right of Eq. (4.3.2), since it is constant relative to integration over mass distributed at points  $R$ .

An argument identical to that presented in Section 4.2 shows that the double integral on the right of Eq. (4.3.3) is zero. Thus, Eq. (4.3.3) reduces to

$$\int_m \mathbf{r}^{P^T} \ddot{\mathbf{r}}^P dm(P) = \int_m \mathbf{r}^{P^T} \mathbf{F}(P) dm(P) \quad (4.3.4)$$

Using the acceleration relation of Eq. (2.4.53) and the variational relation of Eq. (2.4.63), with the Euler parameter form of the orientation transformation matrix, the left side of Eq. (4.3.4) may be expanded to obtain

$$\begin{aligned} \int_m \mathbf{r}^{P^T} \ddot{\mathbf{r}}^P dm(P) &= \int_m \left( \mathbf{r} - \mathbf{A}(\mathbf{p}) \tilde{\mathbf{s}}'^P \boldsymbol{\pi}' \right)^T \left( \ddot{\mathbf{r}} + \mathbf{A}(\mathbf{p}) \tilde{\boldsymbol{\omega}}' \mathbf{s}'^P + \mathbf{A}(\mathbf{p}) \tilde{\boldsymbol{\omega}}' \tilde{\boldsymbol{\omega}}' \mathbf{s}'^P \right) dm(P) \\ &= \mathbf{r}^T \left( \ddot{\mathbf{r}} \int_m dm(P) + \left( \mathbf{A}(\mathbf{p}) \tilde{\boldsymbol{\omega}}' + \mathbf{A}(\mathbf{p}) \tilde{\boldsymbol{\omega}}' \tilde{\boldsymbol{\omega}}' \right) \int_m \mathbf{s}'^P dm(P) \right) \\ &\quad + \boldsymbol{\pi}'^T \left( \int_m \tilde{\mathbf{s}}'^P dm(P) \mathbf{A}^T(\mathbf{p}) \ddot{\mathbf{r}} + \int_m \tilde{\mathbf{s}}'^P \tilde{\boldsymbol{\omega}}' \mathbf{s}'^P dm(P) + \int_m \tilde{\mathbf{s}}'^P \tilde{\boldsymbol{\omega}}' \tilde{\boldsymbol{\omega}}' \mathbf{s}'^P dm(P) \right) \end{aligned} \quad (4.3.5)$$

where use has been made of the fact that  $(\tilde{\mathbf{s}}'^P)^T = -\tilde{\mathbf{s}}'^P$  and quantities that do not depend on the location of point  $P$  and that appear as left or right factors in an integrand have been moved outside the integral. Using vector identities  $\tilde{\boldsymbol{\omega}}' \mathbf{s}'^P = -\tilde{\mathbf{s}}'^P \dot{\boldsymbol{\omega}}'$  and  $\tilde{\mathbf{s}}'^P \tilde{\boldsymbol{\omega}}' = \tilde{\boldsymbol{\omega}}' \tilde{\mathbf{s}}'^P + \boldsymbol{\omega}' \mathbf{s}'^{P^T} - \mathbf{s}'^P \boldsymbol{\omega}'^T$  of Eqs. (2.1.27) and (2.1.33), the last two terms on the right of Eq. (4.3.5) may be written as

$$\begin{aligned} \boldsymbol{\pi}'^T \left( \int_m \tilde{\mathbf{s}}'^P \tilde{\boldsymbol{\omega}}' \mathbf{s}'^P dm(P) + \int_m \tilde{\mathbf{s}}'^P \tilde{\boldsymbol{\omega}}' \tilde{\boldsymbol{\omega}}' \mathbf{s}'^P dm(P) \right) \\ = -\boldsymbol{\pi}'^T \left( \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \dot{\boldsymbol{\omega}}' + \tilde{\boldsymbol{\omega}}' \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \boldsymbol{\omega}' \right) \end{aligned}$$

where  $\mathbf{s}'^{P^T} \tilde{\mathbf{s}}'^P = \mathbf{0} = \boldsymbol{\omega}'^T \tilde{\boldsymbol{\omega}}'$ , from Eq. (2.1.27), have been used. Substituting this result into Eq. (4.3.5) yields

$$\begin{aligned} \int_m \mathbf{r}^{PT} \ddot{\mathbf{r}}^P dm(P) &= \mathbf{r}^T \left( \ddot{\mathbf{r}} \int_m dm(P) + (\mathbf{A}(\mathbf{p}) \tilde{\omega}' + \mathbf{A}(\mathbf{p}) \tilde{\omega}' \tilde{\omega}') \int_m \mathbf{s}'^P dm(P) \right) \\ &\quad + \pi'^T \left( \int_m \tilde{\mathbf{s}}'^P dm(P) \mathbf{A}^T(\mathbf{p}) \ddot{\mathbf{r}} - \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \dot{\omega}' - \tilde{\omega}' \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \omega' \right) \end{aligned} \quad (4.3.6)$$

Using Eq. (2.4.63), the right side of Eq. (4.3.4) may be written as

$$\begin{aligned} \int_m \mathbf{r}^{PT} \mathbf{F}(P) dm(P) &= \int_m \left( \mathbf{r}^T + \pi'^T \tilde{\mathbf{s}}'^P \mathbf{A}^T(\mathbf{p}) \right) \mathbf{F}(P) dm(P) \\ &= \mathbf{r}^T \int_m \mathbf{F}(P) dm(P) + \pi'^T \int_m \tilde{\mathbf{s}}'^P \mathbf{A}^T(\mathbf{p}) \mathbf{F}(P) dm(P) \\ &= \mathbf{r}^T \int_m \mathbf{F}(P) dm(P) + \pi'^T \int_m \tilde{\mathbf{s}}'^P \mathbf{F}'(P) dm(P) \end{aligned} \quad (4.3.7)$$

Substituting from Eqs. (4.3.6) and (4.3.7) into Eq. (4.3.4) and collecting terms,

$$\begin{aligned} &\mathbf{r}^T \left( \ddot{\mathbf{r}} \int_m dm(P) + \mathbf{A}(\mathbf{p}) (\tilde{\omega}' + \tilde{\omega}' \tilde{\omega}') \int_m \mathbf{s}'^P dm(P) - \int_m \mathbf{F}(P) dm(P) \right) \\ &+ \pi'^T \left( \int_m \tilde{\mathbf{s}}'^P dm(P) \mathbf{A}^T(\mathbf{p}) \ddot{\mathbf{r}} - \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \dot{\omega}' - \tilde{\omega}' \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \omega' - \int_m \tilde{\mathbf{s}}'^P \mathbf{F}'(P) dm(P) \right) = 0 \end{aligned} \quad (4.3.8)$$

Terms in Eq. (4.3.8) have the following physical and mathematical significance:

$$m \equiv \int_m dm(P) \quad (4.3.9)$$

is the total mass of the body,

$$\mathbf{s}'^c \equiv \frac{1}{m} \int_m \mathbf{s}'^P dm(P) \quad (4.3.10)$$

is the vector from the origin of the body-fixed  $x'$ - $y'$ - $z'$  frame to the *centroid*, or *center of mass*, of the body,

$$\begin{aligned} \mathbf{J}' &= - \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \\ &= \int m \begin{bmatrix} (y'^P)^2 + (z'^P)^2 & -x'^P y'^P & -x'^P z'^P \\ -x'^P y'^P & (x'^P)^2 + (z'^P)^2 & -y'^P z'^P \\ -x'^P z'^P & -y'^P z'^P & (x'^P)^2 + (y'^P)^2 \end{bmatrix} dm(P) \end{aligned} \quad (4.3.11)$$

is the *inertia tensor*, or *inertia matrix*, of the body relative to the  $x'$ - $y'$ - $z'$  frame,

$$\mathbf{F} \equiv \int_m \mathbf{F}(P) dm(P) \quad (4.3.12)$$

is the *resultant force* that acts at the origin of the body-fixed  $x'$ - $y'$ - $z'$  frame, represented in the  $x$ - $y$ - $z$  frame, and

$$\mathbf{n}' \equiv \int_m \tilde{\mathbf{s}}'^P \mathbf{F}'(P) dm(P) \quad (4.3.13)$$

is the *resultant torque* that acts on the body about the origin of the  $x'$ - $y'$ - $z'$  frame, represented in that frame.

If  $\mathbf{F}(P) = g\mathbf{a}$  is the gravitational force in the direction defined by unit vector  $\mathbf{a}$ , then Eqs. (4.3.12) and (4.3.13) are

$$\begin{aligned} \mathbf{F}_g &= \int_m g\mathbf{a} dm(P) = g\mathbf{a} \int_m dm(P) = m g \mathbf{a} \\ \mathbf{n}'_g &= \int_m \tilde{\mathbf{s}}'^P \mathbf{A}^T(\mathbf{p}) g\mathbf{a} dm(P) = \int_m \tilde{\mathbf{s}}'^P dm(P) \mathbf{A}^T(\mathbf{p}) g\mathbf{a} = m g \tilde{\mathbf{s}}'^c \mathbf{A}^T(\mathbf{p}) \mathbf{a} \end{aligned} \quad (4.3.14)$$

In terms of the foregoing quantities, and noting that  $\int_m \tilde{\mathbf{s}}'^P dm(P) = \widetilde{\int_m \tilde{\mathbf{s}}'^P dm(P)} = m \tilde{\mathbf{s}}'^c$ , Eq.

(4.3.8) may be written as

$$\begin{aligned} &\mathbf{r}^T (m\ddot{\mathbf{r}} - m\mathbf{A}(\mathbf{p})\tilde{\mathbf{s}}'^c \dot{\omega}' + m\mathbf{A}(\mathbf{p})\tilde{\omega}' \tilde{\omega}' \mathbf{s}'^c - \mathbf{F}) \\ &+ \pi'^T (m\tilde{\mathbf{s}}'^c \mathbf{A}^T(\mathbf{p})\ddot{\mathbf{r}} + \mathbf{J}'\dot{\omega}' + \tilde{\omega}' \mathbf{J}'\omega' - \mathbf{n}') = 0 \end{aligned} \quad (4.3.15)$$

This is the *noncentroidal variational equation of motion* of the body.

If no external constraints act on the body, Eq. (4.3.15) must hold for arbitrary virtual displacement  $\mathbf{r}$  and virtual rotation  $\pi'$ . Thus, the coefficients of these vectors in Eq. (4.3.15) must be zero, yielding the classical *noncentroidal Newton-Euler equations of motion* (Pars, 1965),

$$\begin{aligned} m\ddot{\mathbf{r}} - m\mathbf{A}(\mathbf{p})\tilde{\mathbf{s}}'^c \dot{\omega}' + m\mathbf{A}(\mathbf{p})\tilde{\omega}' \tilde{\omega}' \mathbf{s}'^c &= \mathbf{F} \\ m\tilde{\mathbf{s}}'^c \mathbf{A}^T(\mathbf{p})\ddot{\mathbf{r}} + \mathbf{J}'\dot{\omega}' + \tilde{\omega}' \mathbf{J}'\omega' &= \mathbf{n}' \end{aligned} \quad (4.3.16)$$

It is interesting to note that Eq. (4.3.16) is exactly the same as Eq. (4.1.41) in Example 4.1.4, in which equations of motion of a three-particle model of a rigid body were derived.

It is common in applications to select a body-fixed  $x'$ - $y'$ - $z'$  reference frame whose origin is at the *center of mass*, or *centroid*. In this case,  $\mathbf{s}'^c = \mathbf{0}$  and the variational equations of motion of Eq. (4.3.15) simplify to

$$\mathbf{r}^T (m\ddot{\mathbf{r}} - \mathbf{F}) + \pi'^T (\mathbf{J}'\dot{\omega}' + \tilde{\omega}' \mathbf{J}'\omega' - \mathbf{n}') = 0 \quad (4.3.17)$$

If no external constraints act on the body, this yields the *centroidal Newton-Euler equations of motion*

$$\begin{aligned} m\ddot{\mathbf{r}} &= \mathbf{F} \\ \mathbf{J}'\dot{\omega}' + \tilde{\omega}' \mathbf{J}'\omega' &= \mathbf{n}' \end{aligned} \quad (4.3.18)$$

While the simplified form of the centroidal equations of motion is aesthetically appealing and reduces computational complexity by eliminating three terms, it does not account for the effect of a noncentroidal body-fixed  $x'$ - $y'$ - $z'$  reference frame. This is critical when dealing with *design sensitivity analysis*, as explained at the end of Section 4.2. While the simplified forms of centroidal equations of motion yield some computational benefit, in order to use them, modeling of bodies must be done in a centroidal reference frame, often a complicated process. With available computing power, the penalty in using noncentroidal equations of motion is, in fact, minimal.

#### 4.3.2 Euler Parameter Form of Equations of Motion

Identities of Eqs. (2.6.77), (2.6.68), (2.6.63), (2.6.15), and (2.4.43) yield

$$\begin{aligned}\dot{\omega}' &= 2G\ddot{p} \\ \delta\pi' &= 2G\delta p \\ \omega' &= 2G\dot{p} = -2\dot{G}p \\ \tilde{\omega}' &= A^T \dot{A} = -\dot{A}^T A\end{aligned}\tag{4.3.19}$$

where  $G \equiv G(p)$ ,  $\dot{G} \equiv G(\dot{p})$ , and  $A \equiv A(p)$ . These relations may be used to transform the equations of motion of Section 4.3.1 to Euler parameter form.

Substituting the identities of Eq. (4.3.19) into the variational equation of motion of Eq. (4.3.15) yields

$$\begin{aligned}\delta r^T &\left[ m\ddot{r} - 2mA\tilde{s}'^c G\ddot{p} - m\dot{A}A^T As'^c - F \right] \\ &+ 2\delta p^T G^T \left[ m\tilde{s}'^c A^T \ddot{r} + 2J'G\ddot{p} - 2A^T \dot{A}J'\dot{G}p - n' \right] = 0\end{aligned}\tag{4.3.20}$$

which must hold for all  $r$  and  $p$  for which  $p^T p = 0$ . This requirement is imposed so that the transformations of Eqs. (4.3.19) are valid. Using Eq. (2.6.54) and Euler parameter identities of Eqs. (2.6.9) and (2.6.10) that hold provided the Euler parameter normalization constraint is satisfied,

$$\begin{aligned}4G^T A^T \dot{A}J'\dot{G}p &= 4G^T GE^T (2E\dot{G}^T) J'\dot{G}p \\ &= 8(I - pp^T)(I - pp^T) \dot{G}^T J'\dot{G}p \\ &= 8(I - 2pp^T + p(p^T p)p^T) \dot{G}^T J'\dot{G}p \\ &= 8\dot{G}^T J'\dot{G}p - 8pp^T \dot{G}^T J'\dot{G}p\end{aligned}$$

and

$$\begin{aligned}\dot{A}\dot{A}^T As'^c &= 2E\dot{G}^T (2E\dot{G}^T)^T EG^T s'^c \\ &= 4E\dot{G}^T \dot{G}(I - pp^T) G^T s'^c \\ &= 4E\dot{G}^T \dot{G}G^T s'^c\end{aligned}$$

Substituting these relations into Eq. (4.3.20),

$$\begin{aligned} \delta \mathbf{r}^T & \left[ m\ddot{\mathbf{r}} - 2m\mathbf{A}\tilde{\mathbf{s}}'^c\mathbf{G}\ddot{\mathbf{p}} - m4\mathbf{E}\dot{\mathbf{G}}^T\dot{\mathbf{G}}\mathbf{G}^T\mathbf{s}'^c - \mathbf{F} \right] \\ & + \delta \mathbf{p}^T \left[ 2m\mathbf{G}^T\tilde{\mathbf{s}}'^c\mathbf{A}^T\ddot{\mathbf{r}} + 4\mathbf{G}^T\mathbf{J}'\mathbf{G}\ddot{\mathbf{p}} - 8\dot{\mathbf{G}}^T\mathbf{J}'\dot{\mathbf{G}}\mathbf{p} + 8\mathbf{p}\mathbf{p}^T\dot{\mathbf{G}}^T\mathbf{J}'\dot{\mathbf{G}}\mathbf{p} - 2\mathbf{G}^T\mathbf{n}' \right] = 0 \end{aligned} \quad (4.3.21)$$

Since  $\mathbf{p}^T\mathbf{p} = 0$ , this reduces to

$$\begin{aligned} \delta \mathbf{r}^T & \left[ m\ddot{\mathbf{r}} - 2m\mathbf{A}\tilde{\mathbf{s}}'^c\mathbf{G}\ddot{\mathbf{p}} - 4m\mathbf{E}\dot{\mathbf{G}}^T\dot{\mathbf{G}}\mathbf{G}^T\mathbf{s}'^c - \mathbf{F} \right] \\ & + \delta \mathbf{p}^T \left[ 2m\mathbf{G}^T\tilde{\mathbf{s}}'^c\mathbf{A}^T\ddot{\mathbf{r}} + 4\mathbf{G}^T\mathbf{J}'\mathbf{G}\ddot{\mathbf{p}} - 8\dot{\mathbf{G}}^T\mathbf{J}'\dot{\mathbf{G}}\mathbf{p} - 2\mathbf{G}^T\mathbf{n}' \right] = 0 \end{aligned} \quad (4.3.22)$$

where  $\mathbf{F}$  is the *resultant force* acting at the origin of the body-fixed x'-y'-z' frame, represented in the x-y-z frame, and  $\mathbf{n}'$  is the *resultant torque* acting on the body about the origin of the x'-y'-z' frame, represented in that frame. Since gravitational force  $mga$  in direction  $\mathbf{a}$ , represented in the x-y-z frame, acts at the centroid, the gravitational torque  $mg\tilde{\mathbf{s}}'^c\mathbf{A}^T\mathbf{a}$  must be included in  $\mathbf{n}'$ .

For a centroidal body fixed reference frame,  $\mathbf{s}'^c = \mathbf{0}$  and Eq. (4.3.22) reduces to

$$\delta \mathbf{r}^T \left[ m\ddot{\mathbf{r}} - \mathbf{F} \right] + \delta \mathbf{p}^T \left[ 4\mathbf{G}^T\mathbf{J}'\mathbf{G}\ddot{\mathbf{p}} - 8\dot{\mathbf{G}}^T\mathbf{J}'\dot{\mathbf{G}}\mathbf{p} - 2\mathbf{G}^T\mathbf{n}' \right] = 0 \quad (4.3.23)$$

In terms of *Cartesian generalized coordinates*  $\mathbf{q} = [\mathbf{r}^T \mathbf{p}^T]^T$ , the *noncentroidal variational equation of motion* of Eq. (4.3.22) may be written as

$$\delta \mathbf{q}^T \{ \mathbf{M}\ddot{\mathbf{q}} - \mathbf{S} - \mathbf{Q} \} = 0 \quad (4.3.24)$$

which must hold for all  $\mathbf{q} = [\mathbf{r}^T \mathbf{p}^T]^T$  that satisfy  $\mathbf{p}^T \mathbf{p} = 0$ , where

$$\begin{aligned} \mathbf{M} & \equiv \begin{bmatrix} m\mathbf{I} & -2m\mathbf{A}\tilde{\mathbf{s}}'^c\mathbf{G} \\ 2m\mathbf{G}^T\tilde{\mathbf{s}}'^c\mathbf{A}^T & 4\mathbf{G}^T\mathbf{J}'\mathbf{G} \end{bmatrix} \\ \mathbf{S} & \equiv \begin{bmatrix} 4m\mathbf{E}\dot{\mathbf{G}}^T\dot{\mathbf{G}}\mathbf{G}^T\mathbf{s}'^c \\ 8\dot{\mathbf{G}}^T\mathbf{J}'\dot{\mathbf{G}}\mathbf{p} \end{bmatrix} \\ \mathbf{Q} & \equiv \begin{bmatrix} \mathbf{F} \\ 2\mathbf{G}^T\mathbf{n}' \end{bmatrix} \end{aligned} \quad (4.3.25)$$

With a centroidal body fixed reference frame, Eq. (4.3.24) reduces to

$$\delta \mathbf{q}^T \{ \mathbf{M}\ddot{\mathbf{q}} - \mathbf{S} - \mathbf{Q} \} = 0 \quad (4.3.26)$$

which must hold for all  $\mathbf{q} = [\mathbf{r}^T \mathbf{p}^T]^T$  that satisfy  $\mathbf{p}^T \mathbf{p} = 0$ , where

$$\begin{aligned} \mathbf{M} & = \begin{bmatrix} m\mathbf{I} & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}^T\mathbf{J}'\mathbf{G} \end{bmatrix} \\ \mathbf{S} & = \begin{bmatrix} \mathbf{0} \\ 8\dot{\mathbf{G}}^T\mathbf{J}'\dot{\mathbf{G}}\mathbf{p} \end{bmatrix} \end{aligned} \quad (4.3.27)$$

In either case, Eq. (4.3.24) or Eq. (4.3.26) may be written in the form

$$\delta \mathbf{q}^T [\mathbf{M}\ddot{\mathbf{q}} - \mathbf{S} - \mathbf{Q}] = 0 \quad (4.3.28)$$

which must hold for all  $\mathbf{q} = [ \mathbf{r}^T \quad \mathbf{p}^T ]^T$  that satisfy  $\mathbf{p}^T \mathbf{p} = 0$ .

It is important to recall that the *generalized force*  $\mathbf{Q}$  that acts on the body includes both externally applied forces and torques and constraint forces and torques, if any external constraints act on the body. Further, since  $\mathbf{q}$  in Eqs. (4.3.24), (4.3.26), and (4.3.28) is not arbitrary, it is not possible to obtain differential equations of motion, as was possible in Eqs. (4.3.16) and (4.3.18) in angular velocity generalized coordinates. Since Eq. (4.3.28) holds for all  $\mathbf{q} = [ \mathbf{r}^T \quad \mathbf{p}^T ]^T$  that satisfy  $\mathbf{p}^T \mathbf{p} = \mathbf{p}^T \mathbf{p} = 0$ , the *Lagrange Multiplier Theorem* of Section 2.2.5 implies the existence of a scalar  $\lambda$  such that

$$\delta \mathbf{q}^T [\mathbf{M}\ddot{\mathbf{q}} + \mathbf{p}^T \mathbf{p} - \mathbf{S} - \mathbf{Q}] = 0$$

for all  $\mathbf{q}$ , so

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{p}^T \mathbf{p} = \mathbf{S} + \mathbf{Q} \quad (4.3.29)$$

This is a special case of more general *differential-algebraic equations* (DAE) that will be encountered in Section 4.10. It is called a *DAE* because the *Lagrange multiplier* appears algebraically and the algebraic constraint  $\mathbf{p}^T \mathbf{p} = 1$  must be satisfied. Equation (4.3.29) is thus not an ODE.

Equations of motion for a spatial rigid body are derived for a model that is based on a constrained system of particles, using only d'Alembert's principle and Newton's laws, with the strong form of Newton's third law, leading to both variational and differential equations of motion in angular velocities. As shown in Section 2.6.5, however, angular velocity is not the derivative of anything, so equations of motion in angular velocity are not second order ODE.

Transforming the variational equation of motion to Euler parameter form yields a variational equation that cannot be reduced to an ODE, because  $\delta \mathbf{p}$  must satisfy  $\mathbf{p}^T \delta \mathbf{p} = 0$ . While the equations of motion can then be written as a DAE, it is not an ODE.

## Key Formulas

$$\begin{aligned} & \mathbf{r}^T (\mathbf{m}\ddot{\mathbf{r}} - \mathbf{m}\mathbf{A}(\mathbf{p})\tilde{\mathbf{s}}'^c \dot{\omega}' + \mathbf{m}\mathbf{A}(\mathbf{p})\tilde{\omega}' \tilde{\omega}' \mathbf{s}'^c - \mathbf{F}) \\ & + \pi'^T (\mathbf{m}\tilde{\mathbf{s}}'^c \mathbf{A}^T(\mathbf{p})\ddot{\mathbf{r}} + \mathbf{J}'\dot{\omega}' + \tilde{\omega}' \mathbf{J}'\omega' - \mathbf{n}') = 0 \end{aligned} \quad (4.3.15)$$

$$\mathbf{r}^T (\mathbf{m}\ddot{\mathbf{r}} - \mathbf{F}) + \pi'^T (\mathbf{J}'\dot{\omega}' + \tilde{\omega}' \mathbf{J}'\omega' - \mathbf{n}') = 0 \quad (4.3.17)$$

$$\begin{aligned} & \delta \mathbf{r}^T [\mathbf{m}\ddot{\mathbf{r}} - 2\mathbf{m}\mathbf{A}\tilde{\mathbf{s}}'^c \mathbf{G}\ddot{\mathbf{p}} - 4\mathbf{m}\mathbf{E}\mathbf{G}^T \dot{\mathbf{G}} \mathbf{G}^T \mathbf{s}'^c - \mathbf{F}] \\ & + \delta \mathbf{p}^T [2\mathbf{m}\mathbf{G}^T \tilde{\mathbf{s}}'^c \mathbf{A}^T \ddot{\mathbf{r}} + 4\mathbf{G}^T \mathbf{J}' \mathbf{G}\ddot{\mathbf{p}} - 8\dot{\mathbf{G}}^T \mathbf{J}' \dot{\mathbf{G}} \mathbf{p} - 2\mathbf{G}^T \mathbf{n}'] = 0 \end{aligned} \quad (4.3.22)$$

$$\delta \mathbf{r}^T [\mathbf{m}\ddot{\mathbf{r}} - \mathbf{F}] + \delta \mathbf{p}^T [4\mathbf{G}^T \mathbf{J}' \mathbf{G}\ddot{\mathbf{p}} - 8\dot{\mathbf{G}}^T \mathbf{J}' \dot{\mathbf{G}} \mathbf{p} - 2\mathbf{G}^T \mathbf{n}'] = 0 \quad (4.3.23)$$

## 4.4 Centroid and Moments and Products of Inertia

Inertia properties of a spatial rigid body are derived, including transformation of the *inertia matrix* from an arbitrary body fixed reference frame to a *centroidal frame*. Relations that enable location of the centroid and evaluation of the inertia matrix for *composite bodies* are derived. Transformation of the *polar moment of inertia* from an arbitrary body fixed frame to a centroidal frame for planar bodies is obtained as a special case.

### 4.4.1 Location of Centroid

To find the centroid of a spatial body, let a *noncentroidal body-fixed reference frame*  $x''-y''-z''$  with origin  $O''$  be given. Let the centroid be designated as point  $O'$  and let the origin of the  $x'-y'-z'$  frame be  $O'$ , as shown in Fig. 4.4.1.

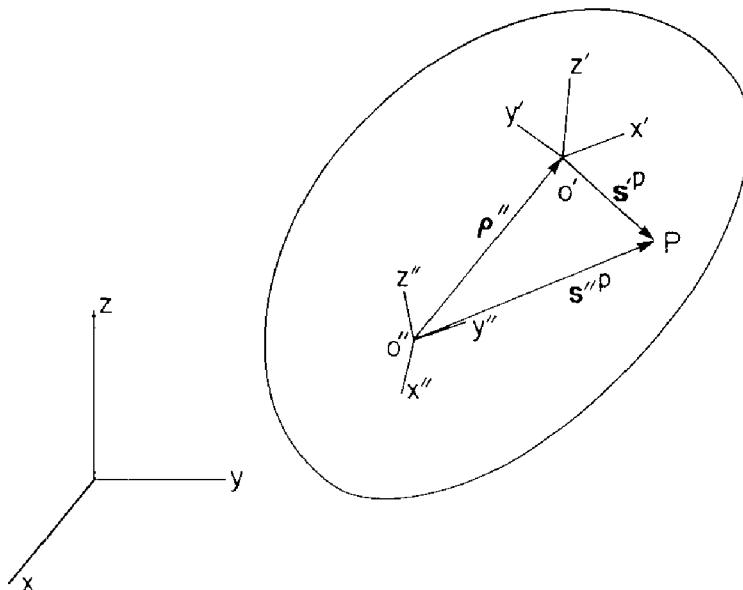


Figure 4.4.1 Location of Centroid

The vector  $\mathbf{p}''$  in the  $x''-y''-z''$  frame that locates the *centroid*  $O'$  may be determined by writing the vector to point  $P$  in the  $x'-y'-z'$  frame as

$$\mathbf{s}'^P = \mathbf{C}(\mathbf{s}''^P - \mathbf{p}'') \quad (4.4.1)$$

where  $\mathbf{s}'^P = \mathbf{C}(\mathbf{s}''^P - \mathbf{p}'')$  emanates from the origin of the  $x'-y'-z'$  frame and  $\mathbf{C}$  is the constant orthogonal transformation matrix from the  $x''-y''-z''$  frame to the  $x'-y'-z'$  frame; i.e.,  $\mathbf{s}' = \mathbf{Cs}''$ . Note that  $\mathbf{s}'^P \neq \mathbf{Cs}''^P$ , since the vectors are not defined in reference frames with a common origin. Using Eq. (4.3.10),

$$\begin{aligned}
\mathbf{0} &= \int_m s'^p dm(P) = \int_m C(s''^p - \rho'') dm(P) \\
&= C \int_m s''^p dm(P) - C \rho'' \int_m dm(P) = C \int_m s''^p dm(P) - m C \rho''
\end{aligned}$$

Multiplying both sides of this equation on the left by  $C^T = C^{-1}$ ,

$$\rho'' = \frac{1}{m} \int_m s''^p dm(P) \quad (4.4.2)$$

Consider the body shown in Fig. 4.4.2, for which the location of points and mass distribution are symmetric about a plane with normal  $\mathbf{n}$ . Select a body  $x''-y''-z''$  frame so that its origin is on the *plane of symmetry*. Thus, for points  $P^+$  and  $P^-$  and associated differential masses that are symmetrically placed with respect to the plane of symmetry,

$$\begin{aligned}
\mathbf{n}^T s''^{P+} &= -\mathbf{n}^T s''^{P-} \\
dm(P^-) &= dm(P^+)
\end{aligned} \quad (4.4.3)$$

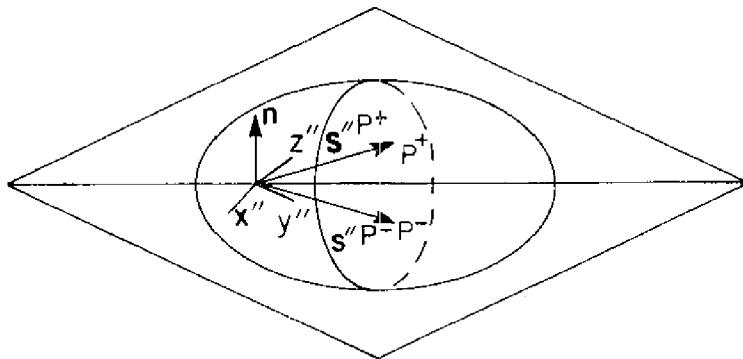


Figure 4.4.2 Body with Plane of Symmetry

Taking the scalar product of both sides of Eq. (4.4.2) with  $\mathbf{n}$  and using Eq. (4.4.3) to change variables of integration,

$$\begin{aligned}
\mathbf{n}^T \rho'' &= \frac{1}{m} \int_m \mathbf{n}^T s''^p dm(P) \\
&= \frac{1}{m} \int_{m^+} \mathbf{n}^T s''^{P+} dm(P^+) + \frac{1}{m} \int_{m^-} \mathbf{n}^T s''^{P-} dm(P^-) \\
&= \frac{1}{m} \int_{m^+} \mathbf{n}^T s''^{P+} dm(P^+) + \frac{1}{m} \int_{m^+} \mathbf{n}^T (-s''^{P+}) dm(P^+) = 0
\end{aligned} \quad (4.4.4)$$

Equation (4.4.4) shows that if a body has a plane of symmetry, then the centroid lies on that plane. If a body has an *axis of symmetry*; i.e., every plane that contains the axis of symmetry is a plane of symmetry, then the centroid lies on the axis of symmetry. This is the case for *bodies of revolution* that may be formed on a lathe. Note that if a body is geometrically symmetric, but is asymmetrically nonhomogeneous, these helpful geometric results are no longer valid.

#### 4.4.2 Transformation of Inertia Matrices

The *inertia matrix* of Eq. (4.3.11) relative to the  $x''-y''-z''$  frame in Fig. 4.4.1 is

$$\mathbf{J}'' = - \int_m \tilde{\mathbf{s}}''^P \tilde{\mathbf{s}}''^P dm(P) \quad (4.4.5)$$

The transpose of the inertia matrix is

$$\mathbf{J}''^T = - \int_m \left( \tilde{\mathbf{s}}''^P \tilde{\mathbf{s}}''^P \right)^T dm(P) = - \int_m \left( \tilde{\mathbf{s}}''^{P T} \tilde{\mathbf{s}}''^{P T} \right) dm(P) = - \int_m \left( (-\tilde{\mathbf{s}}''^P)(-\tilde{\mathbf{s}}''^P) \right) dm(P) = - \int_m \left( \tilde{\mathbf{s}}''^P \tilde{\mathbf{s}}''^P \right) dm(P)$$

so it is symmetric. If there are three elements of nonzero mass in the body that are noncoincident and not on a straight line, as shown in Example 4.1.4,  $\mathbf{J}''$  is positive definite. A more general criteria for positive definiteness is presented in what follows.

To obtain a relationship between  $\mathbf{J}''$  and the inertia matrix  $\mathbf{J}'$  with respect to the  $x'-y'-z'$  *centroidal reference frame*, with  $\mathbf{s}''^P = \tilde{\mathbf{p}}'' + \mathbf{C}^T \mathbf{s}'^P$ , first let  $\mathbf{a}' = \mathbf{s}'^P$  and  $\mathbf{a}'' = \mathbf{s}''^P - \tilde{\mathbf{p}}''$ , where  $\mathbf{a}' = \mathbf{C} \mathbf{a}''$ . Applying Eq. (2.4.27),

$$\tilde{\mathbf{s}}'^P = \tilde{\mathbf{a}}' = \mathbf{C} \tilde{\mathbf{a}}'' \mathbf{C}^T = \mathbf{C} (\tilde{\mathbf{s}}''^P - \tilde{\mathbf{p}}'') \mathbf{C}^T = \mathbf{C} \tilde{\mathbf{s}}''^P \mathbf{C}^T - \mathbf{C} \tilde{\mathbf{p}}'' \mathbf{C}^T \quad (4.4.6)$$

Thus,

$$\tilde{\mathbf{s}}''^P = \tilde{\mathbf{p}}'' + \mathbf{C}^T \tilde{\mathbf{s}}'^P \mathbf{C} \quad (4.4.7)$$

Substituting this into Eq. (4.4.5),

$$\begin{aligned} \mathbf{J}'' &= - \int_m \left( \tilde{\mathbf{p}}'' + \mathbf{C}^T \tilde{\mathbf{s}}'^P \mathbf{C} \right) \left( \tilde{\mathbf{p}}'' + \mathbf{C}^T \tilde{\mathbf{s}}'^P \mathbf{C} \right) dm(P) \\ &= - m \tilde{\mathbf{p}}'' \tilde{\mathbf{p}}'' - \tilde{\mathbf{p}}'' \mathbf{C}^T \int_m \tilde{\mathbf{s}}'^P dm(P) \mathbf{C} - \mathbf{C}^T \int_m \tilde{\mathbf{s}}'^P dm(P) \mathbf{C} \tilde{\mathbf{p}}'' - \mathbf{C}^T \int_m \tilde{\mathbf{s}}'^P \tilde{\mathbf{s}}'^P dm(P) \mathbf{C} \end{aligned} \quad (4.4.8)$$

Since the  $x'-y'-z'$  frame is centroidal,  $\int_m \tilde{\mathbf{s}}'^P dm(P) = \mathbf{0}$  and Eq. (4.4.8) reduces to

$$\mathbf{J}'' = \mathbf{C}^T \mathbf{J}' \mathbf{C} - m \tilde{\mathbf{p}}'' \tilde{\mathbf{p}}'' \quad (4.4.9)$$

Using Eq. (2.1.30), Eq. (4.4.9) may be written in the more conventional form,

$$\mathbf{J}'' = \mathbf{C}^T \mathbf{J}' \mathbf{C} + m \left( \mathbf{p}''^T \mathbf{p}'' \mathbf{I} - \mathbf{p}'' \mathbf{p}''^T \right) \quad (4.4.10)$$

In the special case  $\mathbf{C} = \mathbf{I}$ ; i.e., the  $x'-y'-z'$  and  $x''-y''-z''$  frames are parallel, moments of inertia on the diagonal of Eq. (4.4.10) are related by

$$\begin{aligned} J_{x''x''} &= J_{x'x'} + m \left( \frac{2}{y''} + \frac{2}{z''} \right) \\ J_{y''y''} &= J_{y'y'} + m \left( \frac{2}{x''} + \frac{2}{z''} \right) \\ J_{z''z''} &= J_{z'z'} + m \left( \frac{2}{x''} + \frac{2}{y''} \right) \end{aligned} \quad (4.4.11)$$

Thus, moments of inertia with respect to a noncentroidal  $x''-y''-z''$  frame are those with respect to the parallel centroidal  $x'-y'-z'$  frame plus the mass of the body times the square of the distances

between the respective parallel axes. This is the so-called *parallel axis theorem*. For parallel axes, *products of inertia* are obtained from off-diagonal terms in Eq. (4.4.10) as

$$\begin{aligned} J_{x''y''} &= J_{x'y'} - m \int_{x''}^{x'} \int_{y''}^{y'} dm(P) \\ J_{x''z''} &= J_{x'z'} - m \int_{x''}^{x'} \int_{z''}^{z'} dm(P) \\ J_{y''z''} &= J_{y'z'} - m \int_{y''}^{y'} \int_{z''}^{z'} dm(P) \end{aligned} \quad (4.4.12)$$

Consider a special case in which one of the planes of an  $x''-y''-z''$  frame is a plane of symmetry; e.g., the  $x''-y''$  plane. Then, using the symmetry argument employed in deriving Eq. (4.4.4),

$$J_{z''x''} = J_{z''y''} = 0 \quad (4.4.13)$$

Similarly, if the  $x''-z''$  plane is a plane of symmetry,

$$J_{x''y''} = J_{z''y''} = 0 \quad (4.4.14)$$

and if the  $y''-z''$  plane is a plane of symmetry,

$$J_{x''y''} = J_{x''z''} = 0 \quad (4.4.15)$$

If any two coordinate planes of the  $x''-y''-z''$  frame are planes of symmetry; i.e., have both geometric and mass distribution symmetry, then from Eqs. (4.4.13) to (4.4.15), all products of inertia are zero. This is a very helpful property, since bodies in mechanical systems often have two planes of symmetry. A common case of pairs of planes of symmetry is a homogeneous body of revolution about one of the axes of a reference frame.

#### 4.4.3 Principal Axes

It is possible to find a *centroidal reference frame* with respect to which the products of inertia are all zero. Let the  $x''-y''-z''$  and  $x'-y'-z'$  frames have their origins at the centroid of a body. Then,  $\rho'' = \mathbf{0}$ . Presume  $\mathbf{J}''$  is known, in general as a full symmetric matrix. Multiplying Eq. (4.4.9) on the left by the orthogonal matrix  $\mathbf{C}$  such that  $\mathbf{a}' = \mathbf{Ca}''$  and on the right by  $\mathbf{C}^T$ ,

$$\mathbf{J}' = \mathbf{CJ}''\mathbf{C}^T \quad (4.4.16)$$

The task is now to find an orthogonal transformation matrix  $\mathbf{C}$  from the  $x''-y''-z''$  frame to an  $x'-y'-z'$  frame for which  $\mathbf{J}'$  is diagonal.

From the definition of the symmetric matrix  $\mathbf{J}''$  in Eq. (4.4.5), for any vector  $\mathbf{a}$ ,

$$\begin{aligned} \mathbf{a}^T \mathbf{J}'' \mathbf{a} &= - \int_m \mathbf{a}^T \tilde{\mathbf{s}}'' \tilde{\mathbf{s}}''^T \mathbf{a} dm(P) = \int_m \mathbf{a}^T \tilde{\mathbf{s}}''^T \tilde{\mathbf{s}}'' \mathbf{a} dm(P) \\ &= \int_m (\tilde{\mathbf{s}}'' \mathbf{a})^T (\tilde{\mathbf{s}}'' \mathbf{a}) dm(P) \geq 0 \end{aligned} \quad (4.4.17)$$

Thus,  $\mathbf{J}''$  is *positive semidefinite*. In fact, providing that mass density is nowhere zero in the body,  $\mathbf{a}^T \mathbf{J}'' \mathbf{a} = 0$  implies that  $\tilde{\mathbf{s}}'' \mathbf{a} = \mathbf{0} = -\tilde{\mathbf{a}} \mathbf{s}''$ , for all  $\mathbf{s}''$  in the body. If there are three linearly independent vectors  $\mathbf{s}_i''$ ,  $i = 1, 2, 3$ , to points in the body, then

$$\tilde{\mathbf{a}}[\mathbf{s}_1'' \quad \mathbf{s}_2'' \quad \mathbf{s}_3''] = \mathbf{0}$$

and since  $[\mathbf{s}_1'' \quad \mathbf{s}_2'' \quad \mathbf{s}_3'']$  is nonsingular,  $\tilde{\mathbf{a}} = \mathbf{0}$  and  $\mathbf{a} = \mathbf{0}$ . Thus,  $\mathbf{J}''$  is *positive definite*. This is the case for any homogeneous body whose volume is not zero.

The  $3 \times 3$  positive semidefinite matrix  $\mathbf{J}''$  has three *orthonormal eigenvectors* (Kreyszig, 2011), denoted  $\mathbf{f}''$ ,  $\mathbf{g}''$ , and  $\mathbf{h}''$  in the  $x''$ - $y''$ - $z''$  frame, with corresponding nonnegative *eigenvalues*,  $\zeta_i$ ,  $i = 1, 2, 3$ , ordered such that  $\zeta_1 \geq \zeta_2 \geq \zeta_3$ , where

$$\begin{aligned}\mathbf{J}''\mathbf{f}'' &= \zeta_1\mathbf{f}'' \\ \mathbf{J}''\mathbf{g}'' &= \zeta_2\mathbf{g}'' \\ \mathbf{J}''\mathbf{h}'' &= \zeta_3\mathbf{h}''\end{aligned}\tag{4.4.18}$$

Since  $-\mathbf{g}''$  and  $-\mathbf{h}''$  are also eigenvectors, their signs can be arranged so that  $\mathbf{f}''$ ,  $\mathbf{g}''$ , and  $\mathbf{h}''$  are unit coordinate vectors for a right-hand  $x'$ - $y'$ - $z'$  Cartesian frame. The transformation matrix  $\mathbf{C}^T$  from the  $x'$ - $y'$ - $z'$  frame to the  $x''$ - $y''$ - $z''$  frame is, by Eq. (2.4.16),

$$\mathbf{C}^T = [\mathbf{f}'' \quad \mathbf{g}'' \quad \mathbf{h}'']\tag{4.4.19}$$

Substituting  $\mathbf{C}^T$  from Eq. (4.4.19) into Eq. (4.4.16),

$$\begin{aligned}\mathbf{J}' &= \mathbf{CJ}''\mathbf{C}^T = [\mathbf{f}'' \quad \mathbf{g}'' \quad \mathbf{h}'']^T \mathbf{J}'' [\mathbf{f}'' \quad \mathbf{g}'' \quad \mathbf{h}''] \\ &= [\mathbf{f}'' \quad \mathbf{g}'' \quad \mathbf{h}'']^T [\mathbf{J}''\mathbf{f}'' \quad \mathbf{J}''\mathbf{g}'' \quad \mathbf{J}''\mathbf{h}''] \\ &= [\mathbf{f}'' \quad \mathbf{g}'' \quad \mathbf{h}'']^T [\zeta_1\mathbf{f}'' \quad \zeta_2\mathbf{g}'' \quad \zeta_3\mathbf{h}''] \\ &= [\mathbf{f}'' \quad \mathbf{g}'' \quad \mathbf{h}'']^T [\mathbf{f}'' \quad \mathbf{g}'' \quad \mathbf{h}''] \begin{bmatrix} \zeta_1 & 0 & 0 \\ 0 & \zeta_2 & 0 \\ 0 & 0 & \zeta_3 \end{bmatrix} \\ &= \begin{bmatrix} \zeta_1 & 0 & 0 \\ 0 & \zeta_2 & 0 \\ 0 & 0 & \zeta_3 \end{bmatrix}\end{aligned}\tag{4.4.20}$$

Thus, in the  $x'$ - $y'$ - $z'$  frame,  $J_{x'x'} = \zeta_1$ ,  $J_{y'y'} = \zeta_2$ ,  $J_{z'z'} = \zeta_3$ , and  $J_{x'y'} = J_{x'z'} = J_{y'z'} = 0$ . The axes of such a centroidal frame are called *principal axes* and the associated moments of inertia are called *principal moments of inertia*. By construction, the *principal products of inertia* are zero.

#### 4.4.4 Transformation of Planar Polar Moment of Inertia

As a special case, consider the body shown in Fig. 4.4.1 to be planar. The location of the centroid relative to the  $x''$ - $y''$  frame is  $\mathbf{p}'' = (1/m) \int s''^P dm(P)$ , but the *polar moment of inertia* is the simpler form of Eq. (4.2.14),  $J'' = \int s''^{P_T} s''^P dm(P)$ . To relate polar moments of inertia in the

noncentroidal  $x''$ - $y''$  and centroidal  $x'$ - $y'$  frames, substitute  $\mathbf{s}''^P = \mathbf{p}'' + \mathbf{s}'^P$  from Fig. 4.4.1 into the integral for  $\mathbf{p}''$ ,

$$\begin{aligned} J'' &= \int_m \left( \mathbf{p}'' + \mathbf{s}'^P \right)^T \left( \mathbf{p}'' + \mathbf{s}'^P \right) dm(P) \\ &= \mathbf{p}''^T \mathbf{p}'' \int_m dm(P) + 2\mathbf{p}'' \int_m \mathbf{s}'^P dm(P) + \int_m \mathbf{s}'^{P T} \mathbf{s}'^P dm(P) \\ &= J' + m|\mathbf{p}''|^2 \end{aligned} \quad (4.4.21)$$

This is a much simpler transformation than in Eq. (4.4.10).

#### 4.4.5 Inertia Properties of Complex Bodies

*Components of machines* are often made up of subcomponents that have standard shapes; e.g., circles, disks, cylinders, spheres, and rectangular solids. A typical example of such a component is shown in Fig. 4.4.3, in which each subcomponent and void are of some standard shape that is, typical of those resulting from standard manufacturing processes. The objective now is to develop expressions for the inertia properties of the composite body, using easily calculated properties of individual subcomponents.

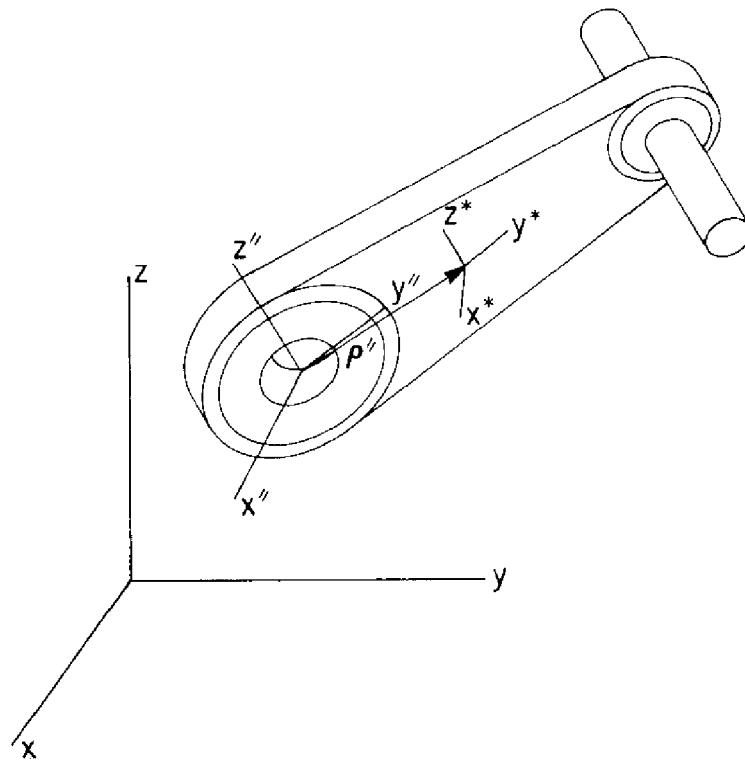


Figure 4.4.3 Composite Body Made up of Subcomponents

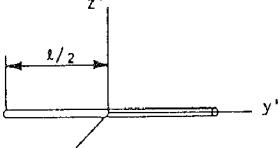
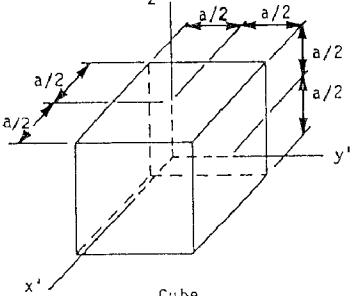
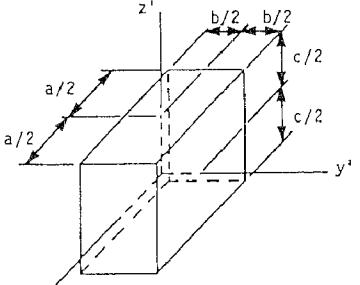
Let an  $x''$ - $y''$ - $z''$  frame be fixed in the *composite body* at a convenient location and orientation; e.g., as shown in Fig. 4.4.3. In this frame, the centroid of the composite body may be

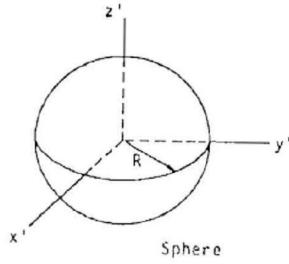
obtained using the definition of Eq. (4.4.2), employing the property that an integral over the entire mass may be written as the sum of integrals over subcomponents  $m_i$ , to obtain

$$\rho'' = \frac{1}{m} \sum_{i=1}^k \int s''^p dm(P) = \frac{1}{m} \sum_{i=1}^k m_i \rho''_i \quad (4.4.22)$$

where  $m = \sum_{i=1}^k m_i$ . To use this result, the centroid  $\rho''_i$  is first located in the  $x''-y''-z''$  frame, using mass and centroid location information from Table 4.4.1 or from direct numerical calculation. Equation (4.4.22) is then used to locate the centroid of the composite body in the  $x''-y''-z''$  frame.

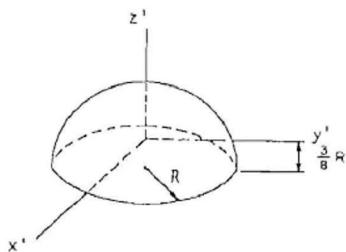
Table 4.4.1 Inertia Properties of Some Homogeneous Bodies

Body	Mass and Moment of Inertia ( $\gamma$ = mass density) ( $A$ = cross-sectional area)
 Thin Rod	$m = \gamma \ell A$ $J_{x'x'} = J_{z'z'} = \frac{m}{12} \ell^2$ $J_{y'y'} = 0$
 Cube	$m = \gamma a^3$ $J_{x'x'} = J_{y'y'} = J_{z'z'} = \frac{1}{6} m a^2$
 Rectangular Prism	$m = \gamma abc$ $J_{x'x'} = \frac{1}{12} m(b^2 + c^2)$ $J_{y'y'} = \frac{1}{12} m(a^2 + c^2)$ $J_{z'z'} = \frac{1}{12} m(a^2 + b^2)$



$$m = \frac{4}{3}\pi\gamma R^3$$

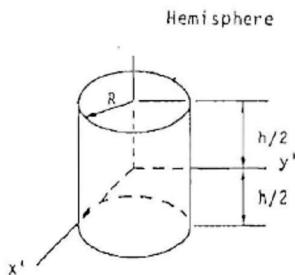
$$J_{x'x'} = J_{y'y'} = J_{z'z'} = \frac{2}{5}mR^2$$



$$m = \frac{2}{3}\pi\gamma R^3$$

$$J_{x'x'} = J_{y'y'} = \frac{83}{320}mR^2$$

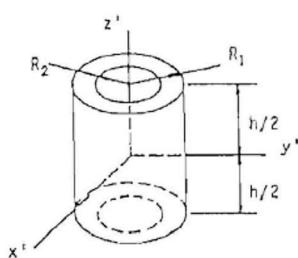
$$J_{z'z'} = \frac{2}{5}mR^2$$



$$m = \pi\gamma R^2 h$$

$$J_{x'x'} = J_{y'y'} = \frac{1}{2}m(3R^2 + h^2)$$

$$J_{z'z'} = \frac{1}{2}mR^2$$



$$m = \pi\gamma h(R_1^2 - R_2^2)$$

$$J_{x'x'} = J_{y'y'} = \frac{1}{2}m(3R_1^2 + 3R_2^2 + h^2)$$

$$J_{z'z'} = \frac{1}{2}m(R_1^2 + R_2^2)$$

Hollow Right Circular Cylinder

The *inertia matrix*  $\mathbf{J}^*$  with respect to the  $x^*-y^*-z^*$  *centroidal reference frame* of the composite body; e.g., as shown in Fig. 4.4.3, is now to be calculated. The notation  $x^*-y^*-z^*$  is selected here to avoid confusion with the centroidal  $x'_i-y'_i-z'_i$  frame of subcomponent  $i$ . Equation (4.4.20) may be written and its integral evaluated as a sum of integrals over subcomponents that make up the composite body, to obtain

$$\mathbf{J}^* = - \int_m \tilde{\mathbf{s}}^{*P} \tilde{\mathbf{s}}^{*P} dm(P) = \sum_{i=1}^k \left( - \int_{m_i} \tilde{\mathbf{s}}^{*P} \tilde{\mathbf{s}}^{*P} dm(P) \right) = \sum_{i=1}^k \mathbf{J}_i^* \quad (4.4.23)$$

To use Eq. (4.4.23), the inertia matrix  $\mathbf{J}_i^*$  of subcomponent  $i$  in the  $x^*-y^*-z^*$  frame must be calculated. To do this, a convenient  $x'_i-y'_i-z'_i$  centroidal frame is defined for subcomponent  $i$  in the

composite body. Denote by  $\mathbf{J}'_i$  the inertia matrix of subcomponent  $i$  with respect to its selected centroidal  $x'-y'-z'$  frame and by  $\mathbf{p}'_i$  the vector that locates the centroid of the subcomponent in the  $x^*-y^*-z^*$  frame. Since the  $x^*-y^*-z^*$  frame is not centroidal for the individual subcomponents, the inertia transformation of Eq. (4.4.10) is used to calculate the inertia matrix of subcomponent  $i$  with respect to the composite body centroidal  $x^*-y^*-z^*$  frame. Denoting by  $\mathbf{C}'_i$  the transformation matrix from the  $x^*-y^*-z^*$  frame to the  $x'_i-y'_i-z'_i$  frame, Eq. (4.4.10) yields

$$\mathbf{J}_i^* = \mathbf{C}'_i^T \mathbf{J}'_i \mathbf{C}'_i + m_i (\mathbf{p}'_i^T \mathbf{p}'_i \mathbf{I} - \mathbf{p}'_i \mathbf{p}'^T) \quad (4.4.24)$$

This result may be substituted into Eq. (4.4.23), to obtain the desired inertia matrix for the composite body. If there are voids in a composite body, Eq. (4.4.23) may be used by defining  $-\mathbf{J}_i^*$  as the inertia matrix of the void, where  $\mathbf{J}_i^*$  is the inertia matrix of a body that would occupy the void, with the same material density as the body from which it is removed. Inertia properties of a few homogeneous bodies, for use in such calculations, are provided in Table 4.4.1.

**Example 4.4.1:** The body of Fig. 4.4.4 is formed by a rectangular solid with a hole of radius  $R = c/4$  about the  $x$  axis. Since all three coordinate planes are planes of symmetry, the centroid is at the origin of the  $x$ - $y$ - $z$  frame and the products of inertia with respect to this frame are zero.

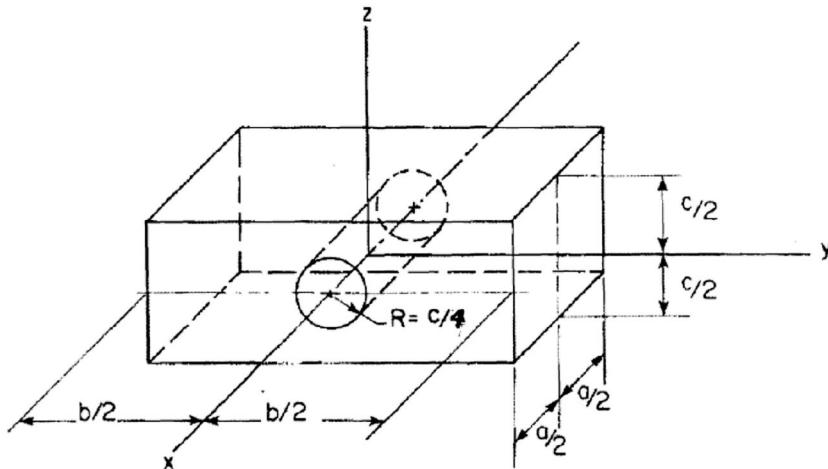


Figure 4.4.4 Rectangular Solid with Hole

From Eq. (4.4.24), with  $\mathbf{J}_1$  as the inertia matrix of the rectangular solid without a hole and  $\mathbf{J}_2$  as the inertia matrix of a cylinder that would occupy the hole,  $\mathbf{J} = \mathbf{J}_1 - \mathbf{J}_2$ . Using Table 4.4.1,

$$J_{xx} = \frac{abc}{12} (b^2 + c^2) - \frac{c^4 a}{512}$$

$$J_{yy} = \frac{abc}{12} (a^2 + c^2) - \frac{c^2 a}{192} \left( \frac{3c^2}{16} + a^2 \right)$$

$$J_{zz} = \frac{abc}{12} (a^2 + b^2) - \frac{c^2 a}{192} \left( \frac{3c^2}{16} + a^2 \right)$$

The relatively elementary composite body of Example 4.4.1 shows that the transformation relations for inertial properties and the theory of calculating centroids and inertia properties of complex bodies can be used systematically. The reader may, however, despair over the extent of such computations for *complex composite bodies*, such as automobile bodies, aircraft landing gear assemblies, vending machine components, and robot arms. Indeed, manual application of the methods used in this section would lead to extensive algebraic manipulations that are time consuming and prone to error when carried out analytically by the engineer.

The availability of large-scale computer codes for system dynamic simulation creates new opportunities for complex applications, but requires detailed inertia data that have not normally been calculated in engineering practice. This dilemma may be alleviated using *computer-aided design* and *computer-aided engineering* systems that contain extensive modeling software that describes the geometry of complex mechanical assemblies. The methods developed and illustrated in this section form the foundation for computer implementation with *geometric modelers* to determine the locations of centroids and moments and products of inertia with respect to reference frames that are specified by the engineer. The technical challenge in implementing such methods is computer generation of the geometric definition of complex bodies, which is the domain of *solid modeling* methods that are available in the form of practical computational tools in modern computer-aided design and computer-aided engineering systems.

Once geometric and material property information is defined and entered into a computer database, implementation of the analytical methods presented in this section for calculating the locations of centroids and moments and products of inertia is a relatively simple matter. The viewpoint taken for the remainder of this text is that such modern computer-aided engineering tools are available to the engineer for computation of the inertial properties of machine components.

The integral representation of the inertia matrix and location of centroid enable derivation of formulas for transformation of these properties to body-fixed coordinate systems that have attractive properties; e.g., a diagonal inertia matrix. Symmetry of bodies with respect of planes and axes likewise yields simplified expressions. Finally, transformation formulas can be used with composite bodies that are comprised of subsets with common shapes to evaluate inertia properties, as is done in modern computer-aided engineering systems.

## 4.5 Internal Generalized Forces

Externally applied forces are accounted for in Eqs. (4.2.15) and (4.2.16) for planar systems and Eqs. (4.3.12) and (4.3.13) for spatial systems. *Internal forces* due to *force elements* that act between points fixed in pairs of bodies and exert forces of action and reaction along the line between the attachment points are treated in this section. Such elements include passive devices such as *springs* and *dampers* that act on pairs of bodies. Perhaps more important in modern systems are *actuators* associated with control systems that exert forces and torques between bodies using electric and hydraulic actuators. Internal forces associated with a *translational spring-damper-actuator* (TSDA) that acts between points on bodies and a *rotational spring-damper-actuator* (RSDA) that acts about common rotational axes in the bodies are treated in this section, for both planar and spatial systems.

### 4.5.1 Internal Forces in Spatial Systems

Consider the *spatial TSDA* shown in Fig. 4.5.1 that connects points  $P_i$  and  $P_j$  on bodies  $i$  and  $j$ , respectively. The magnitude of the *internal force* that acts in the *TSDA*, with tension (drawing  $P_i$  and  $P_j$  together) taken as positive, is

$$f = K(\ell - \ell_0) + C\dot{\ell} + F(\ell, \dot{\ell}) \quad (4.5.1)$$

where  $K$  is a *spring constant*,  $\ell_0$  is the *free length* of the spring,  $C$  is a *damping coefficient*, and  $F(\ell, \dot{\ell})$  is a general *actuator force*.

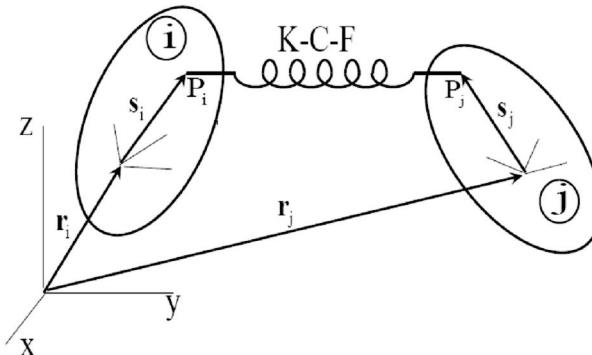


Figure 4.5.1 Translational Spring-Damper-Actuator (TSDA)

A *unilateral TSDA* is defined as one for which  $K = C = F = \mathbf{0}$  if  $\ell - \ell_0 > 0$ . Logic is used in computer implementation so that values of  $K$ ,  $C$ , and  $F$  specified are implemented if  $\ell - \ell_0 < 0$  and zeros are implemented if  $\ell - \ell_0 > 0$ . To avoid numerical problems with discontinuous damping forces; in computerese, when  $\ell - \ell_0 < 0$ ,  $C = 100C(\ell_0 - \ell)/\ell_0$  if  $100(\ell_0 - \ell)/\ell_0 < 1$  and  $C = 0$  if  $100(\ell_0 - \ell)/\ell_0 > 1$ . This element is valuable when *bump stops* or *impacts* are to be modeled.

Since an increase in  $\ell$  acts in the opposite sense as the force  $f$ , the *virtual work* done on the system is negative if both  $f$  and  $\ell$  are positive, so

$$W = -f\ell \quad (4.5.2)$$

To see that the minus sign is required, with the augmentation of Fig. 4.5.2, note that if  $\ell > 0$ ,  $f > 0$ , and as the attachment points on the bodies move apart, work done on the bodies is negative and  $W = -f \ell$ .

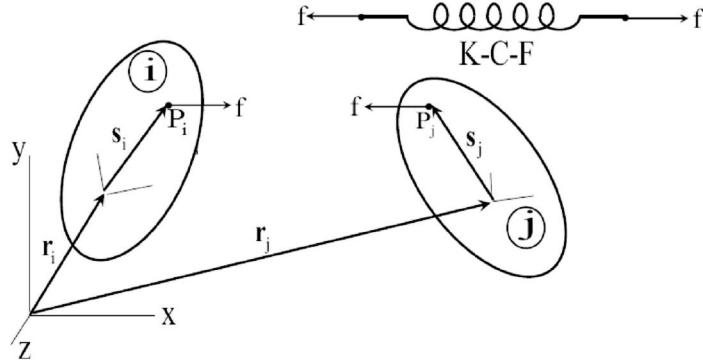


Figure 4.5.2 Translational Spring-Damper-Actuator (TSDA)

The goal is to expand the right side of Eq. (4.5.2) in terms of variations in generalized coordinates, to obtain

$$\delta W = \delta \mathbf{q}_i^T \mathbf{Q}_i + \delta \mathbf{q}_j^T \mathbf{Q}_j \quad (4.5.3)$$

where  $\mathbf{Q}_i$  and  $\mathbf{Q}_j$  are vectors of *generalized force* that act on bodies i and j. This is a generally applicable approach to determining generalized forces on bodies, using the *virtual work* of applied forces.

The vector from  $P_i$  to  $P_j$  is

$$\mathbf{d}_{ij} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i \quad (4.5.4)$$

Thus, the length  $\ell$  of the spring-damper-actuator is given by

$$\ell^2 = \mathbf{d}_{ij}^T \mathbf{d}_{ij} \geq 0 \quad (4.5.5)$$

It is assumed that the TSDA is designed so that  $\ell > 0$ . Differentiating Eq. (4.5.5) with respect to time yields

$$2\ell \dot{\ell} = 2\mathbf{d}_{ij}^T \dot{\mathbf{d}}_{ij} \quad (4.5.6)$$

so, using Eq. (2.6.26),

$$\dot{\ell} = (1/\ell) \mathbf{d}_{ij}^T \dot{\mathbf{d}}_{ij} = (1/\ell) \mathbf{d}_{ij}^T (\dot{\mathbf{r}}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) \dot{\mathbf{p}}_j - \dot{\mathbf{r}}_i - \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \dot{\mathbf{p}}_i) \quad (4.5.7)$$

where it is important that the design requires that  $\ell > 0$ . Similarly, taking the variation of Eq. (4.5.5), using Eq. (2.4.67),

$$\ell = (1/\ell) \mathbf{d}_{ij}^T (\mathbf{r}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{p}_j - \mathbf{r}_i - \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{p}_i) \quad (4.5.8)$$

Substituting this result into Eq. (4.5.2) yields

$$\begin{aligned}
W &= -\left(\frac{f/\ell}{\ell}\right) \mathbf{d}_{ij}^T \left( \mathbf{r}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{p}_j - \mathbf{r}_i - \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{p}_i \right) \\
&= \begin{bmatrix} \mathbf{r}_j^T & \mathbf{p}_j^T \end{bmatrix} \left( -\left(\frac{f/\ell}{\ell}\right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{d}_{ij} \end{bmatrix} \right) + \begin{bmatrix} \mathbf{r}_i^T & \mathbf{p}_i^T \end{bmatrix} \left( \left(\frac{f/\ell}{\ell}\right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{d}_{ij} \end{bmatrix} \right)
\end{aligned} \quad (4.5.9)$$

The Euler parameter form of *spatial TSDA generalized forces* on bodies i and j are thus

$$\begin{aligned}
\mathbf{Q}_i &= \left(\frac{f/\ell}{\ell}\right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{d}_{ij} \end{bmatrix} \\
\mathbf{Q}_j &= -\left(\frac{f/\ell}{\ell}\right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{d}_{ij} \end{bmatrix}
\end{aligned} \quad (4.5.10)$$

Another useful *internal force element* in spatial systems is the *spatial rotational spring-damper-actuator* (RSDA) that acts about the common axis of relative rotation in a revolute or cylindrical joint between bodies i and j is defined in Section 3.3, as shown schematically in Fig. 4.5.3. The *spatial RSDA torque*

$$\tau = K(\theta_{ij} - \theta_{0ij}) + C\dot{\theta}_{ij} + T(t) \quad (4.5.11)$$

acts between axes fixed in bodies i and j, about the common  $\mathbf{w}^i$  and  $\mathbf{w}^j$  axes, with a positive torque drawing the axes together. The angle  $\theta_{0ij}$  in the RSDA corresponds to the free length  $\ell_0$  of a TSDA. The virtual work due to the torque is thus

$$\delta W = -\tau \delta \theta_{ij} \quad (4.5.12)$$

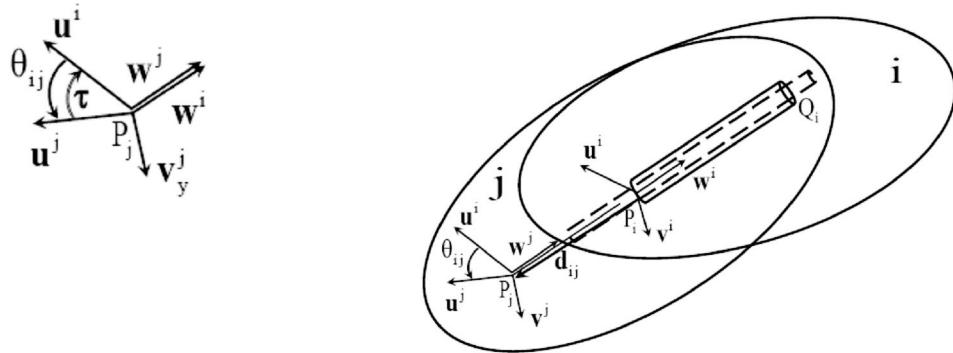


Figure 4.5.3 Spatial Rotational Spring-Damper-Actuator (RSDA)

The angle  $\theta_{ij}(\mathbf{p}_i, \mathbf{p}_j)$  is given by Eq. (2.4.37),

$$\theta_{ij} = \begin{cases} -\pi - \text{Arcsin}(s), & \text{if } s \leq 0 \text{ and } c < 0 \\ \text{Arcsin}(s), & \text{if } c \geq 0 \\ \pi - \text{Arcsin}(s), & \text{if } s \geq 0 \text{ and } c < 0 \end{cases} \quad (4.5.13)$$

where  $c \equiv \cos \theta_{ij} = \mathbf{u}^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}^j$  and  $s \equiv \sin \theta_{ij} = \mathbf{v}^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}^j$ , and may be numerically evaluated using Code 2.4 in Section 2.B.2 of Appendix 2.B. In terms of angular velocities of bodies i and j and the vector  $\mathbf{w}^i$ ,

$$\begin{aligned}\dot{\omega}_{ij} &= \mathbf{w}^{iT} (\omega_j - \omega_i) = \mathbf{w}'^{iT} \mathbf{A}_i^T (\mathbf{A}_j \omega'_j - \mathbf{A}_i \omega'_i) = \mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \omega'_j - \omega'_i) \\ &= 2\mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\mathbf{p}_j) \dot{\mathbf{p}}_j - \mathbf{G}(\mathbf{p}_i) \dot{\mathbf{p}}_i)\end{aligned}\quad (4.5.14)$$

Similarly,

$$\dot{\omega}_{ij} = 2\mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\mathbf{p}_j) \dot{\mathbf{p}}_j - \mathbf{G}(\mathbf{p}_i) \dot{\mathbf{p}}_i) \quad (4.5.15)$$

The torque of Eq. (4.5.11) that acts between the bodies is thus

$$\tau = K(\theta_{ij}(\mathbf{p}_i, \mathbf{p}_j) - \theta_{0ij}) + C2\mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\mathbf{p}_j) \dot{\mathbf{p}}_j - \mathbf{G}(\mathbf{p}_i) \dot{\mathbf{p}}_i) + T(t) \quad (4.5.16)$$

and the virtual work of Eq. (4.5.12) may be written as

$$\begin{aligned}W &= -2\tau \mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\mathbf{p}_j) \dot{\mathbf{p}}_j - \mathbf{G}(\mathbf{p}_i) \dot{\mathbf{p}}_i) \\ &= 2\tau \mathbf{p}_i^T \mathbf{G}^T(\mathbf{p}_i) \mathbf{w}'^i - 2\tau \mathbf{p}_j^T \mathbf{G}^T(\mathbf{p}_j) \mathbf{A}_j^T \mathbf{A}_i \mathbf{w}'^i\end{aligned}\quad (4.5.17)$$

Generalized forces that act on bodies i and j, as a result of the RSDA, are thus

$$\begin{aligned}\mathbf{Q}_i &= \begin{bmatrix} \mathbf{0} \\ 2\tau \mathbf{G}^T(\mathbf{p}_i) \mathbf{w}'^i \end{bmatrix} \\ \mathbf{Q}_j &= \begin{bmatrix} \mathbf{0} \\ -2\tau \mathbf{G}^T(\mathbf{p}_j) \mathbf{A}_j^T \mathbf{A}_i \mathbf{w}'^i \end{bmatrix}\end{aligned}\quad (4.5.18)$$

The effect of TSDA and RSDA generalized forces on the equations of motion of a system is accounted for by inserting the expressions of Eqs. (4.5.10) and (4.5.18) into the *variational equations of motion*.

#### 4.5.2 Internal Forces in Planar Systems

The derivation of *TSDA* generalized forces for planar systems is the same as that for spatial systems, through Eq. (4.5.7). Taking the derivative and variation of  $\ell$  in Eq. (4.5.5) yields

$$\begin{aligned}\dot{\ell} &= (1/\ell) \mathbf{d}_{ij}^T (\dot{\mathbf{r}}_j + \dot{\phi}_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j - \dot{\mathbf{r}}_i - \dot{\phi}_i \mathbf{P} \mathbf{A}_i \mathbf{s}'_i) \\ \ell &= (1/\ell) \mathbf{d}_{ij}^T (\mathbf{r}_j + \phi_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \phi_i \mathbf{P} \mathbf{A}_i \mathbf{s}'_i)\end{aligned}\quad (4.5.19)$$

Substituting  $\ell$  into Eq. (4.5.2) yields

$$\begin{aligned}W &= -(f/\ell) \mathbf{d}_{ij}^T (\mathbf{r}_j + \phi_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \phi_i \mathbf{P} \mathbf{A}_i \mathbf{s}'_i) \\ &= \begin{bmatrix} \mathbf{r}_j^T & \phi_j \end{bmatrix} \left( -\left(f/\ell\right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \end{bmatrix} \right) + \begin{bmatrix} \mathbf{r}_i^T & \phi_i \end{bmatrix} \left( \left(f/\ell\right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_i \mathbf{s}'_j \end{bmatrix} \right)\end{aligned}\quad (4.5.20)$$

Thus, *planar TSDA generalized forces* that act on bodies i and j are

$$\begin{aligned}\mathbf{Q}_i &= \left( f/\ell \right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \end{bmatrix} \\ \mathbf{Q}_j &= -\left( f/\ell \right) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \end{bmatrix}\end{aligned}\quad (4.5.21)$$

Another useful *internal force element* in planar systems is the *rotational spring-damper-actuator* (RSDA) that acts about the axis of relative rotation in a revolute joint between bodies i and j at common points  $P_i$  and  $P_j$  shown schematically in Fig. 4.5.2. The *planar RSDA torque*

$$\tau = K(\phi_j - \phi_i - \phi_0) + C(\dot{\phi}_j - \dot{\phi}_i) + T \quad (4.5.22)$$

acts between axes fixed in bodies i and j, parallel to the  $x'_i$  and  $x'_j$  axes, with a positive torque drawing the axes together. The angle  $\phi_0$  in the RSDA corresponds to the free length  $\ell_0$  of a TSDA. The virtual work due to the torque is thus

$$\delta W = -\tau(\delta\phi_j - \delta\phi_i) = -\tau\delta\phi_j + \tau\delta\phi_i \quad (4.5.23)$$

The *planar RSDA generalized forces* acting on bodies i and j are thus

$$\begin{aligned}\mathbf{Q}_i &= \begin{bmatrix} \mathbf{0} \\ \tau \end{bmatrix} \\ \mathbf{Q}_j &= -\begin{bmatrix} \mathbf{0} \\ \tau \end{bmatrix}\end{aligned}\quad (4.5.24)$$

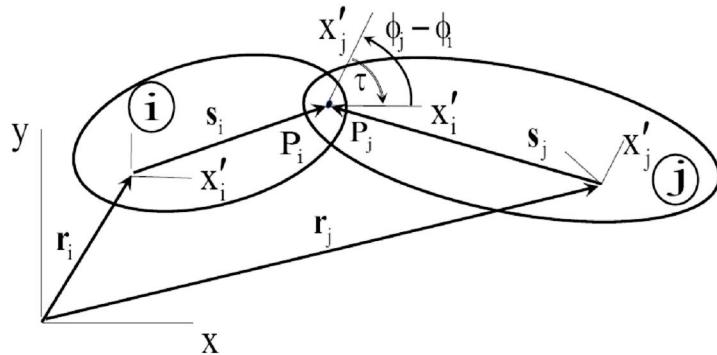


Figure 4.5.2 Planar Rotational Spring-Damper-Actuator (RSDA)

Generalized forces due to internal force elements such as springs, dampers, and actuators that act between pairs of bodies are accounted for using virtual displacements and rotations to calculate the associated virtual work.

## 4.6 Variational Equations of Motion for Multibody Systems

Mechanical systems that are made up of multiple particles, planar bodies, and spatial bodies are considered. Holonomic kinematic constraints are accounted for. The *variational equation of motion* for such systems is formed by summing variational equations for each of its particles and bodies. Constraints that act between particles and bodies in the system are accounted for by requiring that virtual displacements satisfy the linearized form of constraint equations, with time fixed.

### 4.6.1 System Constraints and Virtual Displacements

For systems that are comprised of particles, planar bodies, and spatial bodies, *holonomic constraints* between bodies  $i$  and  $j$  are of the form

$$\Phi^k(\mathbf{q}_i, \mathbf{q}_j) = 0 \quad (4.6.1)$$

$k = 1 \dots nh$ , where Euler parameter normalization conditions for spatial bodies are included in Eq. (4.6.1). The associated *kinematic velocity equations* are obtained by taking the time derivative of Eq. (4.6.1),

$$\Phi_{\dot{\mathbf{q}}_i}^k + \Phi_{\dot{\mathbf{q}}_j}^k = 0 \quad (4.6.2)$$

and virtual displacement equations are obtained by taking the differential of Eq. (4.6.1), with time held fixed,

$$\Phi_{\mathbf{q}_i}^k \mathbf{q}_i + \Phi_{\mathbf{q}_j}^k \mathbf{q}_j = 0 \quad (4.6.3)$$

With system generalized coordinates  $\mathbf{q} = [\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \dots \quad \mathbf{q}_{nb}^T]^T$ , *system holonomic constraints* are

$$\Phi(\mathbf{q}) = \begin{bmatrix} \Phi^1(\mathbf{q}) \\ \vdots \\ \Phi^{nhc}(\mathbf{q}) \end{bmatrix} = 0 \quad (4.6.4)$$

*system velocity constraints* are

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = 0 \quad (4.6.5)$$

and *system virtual displacement constraints* are

$$\Phi_q(\mathbf{q})\delta\mathbf{q} = 0 \quad (4.6.6)$$

### 4.6.2 Variational Equations of Motion

For a particle with generalized coordinate  $\mathbf{q}_i = \mathbf{r}_i$  or a planar body with centroidal generalized coordinate  $\mathbf{q}_i = [\mathbf{r}_i^T \quad \phi_i]^T$ , its *variational equation of motion*, from Eqs. (4.2.20), is

$$\delta\mathbf{q}_i^T (\mathbf{M}_i \ddot{\mathbf{q}}_i - \mathbf{Q}_i^A) = 0 \quad (4.6.7)$$

where for a particle,

$$\begin{aligned}\mathbf{M}_i &= m_i \mathbf{I}_3 \\ \mathbf{Q}_i^A &= \mathbf{F}_i\end{aligned}\quad (4.6.8)$$

and for a *centroidal body reference frame*,

$$\begin{aligned}\mathbf{M}_i &= \begin{bmatrix} m_i \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & J'_i \end{bmatrix} \\ \mathbf{Q}_i^A &= \begin{bmatrix} \mathbf{F}_i^A \\ \mathbf{n}_i'^A \end{bmatrix}\end{aligned}\quad (4.6.9)$$

If *noncentroidal body reference frames* are used for planar bodies, kinetic terms of Eqs. (4.2.19) must be used.

For a spatial body with generalized coordinates  $\mathbf{q}_i = [\mathbf{r}_i^T \quad \mathbf{p}_i^T]^T$ , the *variational equation of motion* of Eq. (4.3.23) is

$$\delta \mathbf{q}_i^T (\mathbf{M}_i \ddot{\mathbf{q}}_i - \mathbf{S}_i - \mathbf{Q}_i^A) = \mathbf{0} \quad (4.6.10)$$

where, for a *centroidal body reference frame*,

$$\begin{aligned}\mathbf{M}_i &= \begin{bmatrix} m_i \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}^T(\mathbf{p}_i) \mathbf{J}'_i \mathbf{G}(\mathbf{p}_i) \end{bmatrix} \\ \mathbf{S}_i &= \begin{bmatrix} \mathbf{0} \\ 8\mathbf{G}^T(\dot{\mathbf{p}}_i) \mathbf{J}'_i \mathbf{G}(\mathbf{p}_i) \mathbf{p}_i \end{bmatrix} \\ \mathbf{Q}_i^A &\equiv \begin{bmatrix} \mathbf{F}_i^A \\ 2\mathbf{G}^T(\mathbf{p}_i) \mathbf{n}_i'^A \end{bmatrix}\end{aligned}\quad (4.6.11)$$

If the body reference frame is noncentroidal, kinetic terms of Eqs. (4.3.26) must be used.

With  $\mathbf{S}_i = \mathbf{0}$  for particles and summing the variational equations of Eqs. (4.6.7) and (4.6.10) over all bodies, the *system variational equation of motion* is

$$\sum_{i=1}^{nb} \delta \mathbf{q}_i^T (\mathbf{M}_i \ddot{\mathbf{q}}_i - \mathbf{S}_i - \mathbf{Q}_i^A) = \mathbf{0} \quad (4.6.12)$$

which must hold for all  $\delta \mathbf{q}_i$  that satisfy Eqs. (4.6.6). Forces of constraint in the system do not appear in Eq. (4.6.12), because they do no work under the action of *kinematically admissible virtual displacements*.

Defining

$$\begin{aligned}
\mathbf{q} &= \left[ \mathbf{q}_1^T \cdots \mathbf{q}_{nb}^T \right]^T \\
\delta \mathbf{q} &= \left[ \delta \mathbf{q}_1^T \cdots \delta \mathbf{q}_{nb}^T \right]^T \\
\mathbf{M} &= \text{diag}(\mathbf{M}_1 \cdots \mathbf{M}_{nb}) \\
\mathbf{Q}^A &= \left[ \mathbf{Q}_{1A}^T \cdots \mathbf{Q}_{nbA}^T \right]^T \\
\mathbf{S} &= \left[ \mathbf{S}_1^T \cdots \mathbf{S}_{nb}^T \right]^T
\end{aligned} \tag{4.6.13}$$

Eq. (4.6.12) may be written as

$$\mathbf{q}^T (\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0 \tag{4.6.14}$$

The *system variational equation of motion* is thus that Eq. (4.6.14) holds for all  $\delta \mathbf{q}$  that satisfy Eq. (4.6.6); i.e., for all *kinematically admissible virtual displacements*.

#### 4.6.3 Positive Definiteness of System Mass Matrix

For the single spatial body shown in Fig. 4.3.1, the velocity of a differential element of mass at point P is

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \tilde{\omega} \mathbf{s}^P = \dot{\mathbf{r}} + \mathbf{A} \tilde{\omega}' \mathbf{s}'^P = \dot{\mathbf{r}} - \mathbf{A} \tilde{\mathbf{s}}'^P \omega' \tag{4.6.15}$$

With this relation, the *kinetic energy* of body i, suppressing index i, is

$$\begin{aligned}
KEB^i &= \frac{1}{2} \int_m (\dot{\mathbf{r}} - \mathbf{A} \tilde{\mathbf{s}}'^P \omega')^T (\dot{\mathbf{r}} - \mathbf{A} \tilde{\mathbf{s}}'^P \omega') dm(P) \\
&= \frac{m}{2} \dot{\mathbf{r}}^T \dot{\mathbf{r}} - \frac{1}{2} \int_m \omega'^T \tilde{\mathbf{s}}'^P \mathbf{A}^T \mathbf{A} \tilde{\mathbf{s}}'^P \omega' dm(P) - \int_m \dot{\mathbf{r}}^T \mathbf{A} \tilde{\mathbf{s}}'^P \omega' dm(P) \\
&= \frac{m}{2} \dot{\mathbf{r}}^T \dot{\mathbf{r}} + \frac{1}{2} \omega'^T \mathbf{J}' \omega' - m \dot{\mathbf{r}}^T \mathbf{A} \tilde{\mathbf{s}}'^c \omega' \\
&= \frac{m}{2} \dot{\mathbf{r}}^T \dot{\mathbf{r}} + 2 \dot{\mathbf{p}}^T \mathbf{G}^T \mathbf{J}' \mathbf{G} \dot{\mathbf{p}} - 2m \dot{\mathbf{r}}^T \mathbf{A} \tilde{\mathbf{s}}'^c \mathbf{G} \dot{\mathbf{p}} = \frac{1}{2} \dot{\mathbf{q}}^{iT} \mathbf{M}_i \dot{\mathbf{q}}^i
\end{aligned} \tag{4.6.16}$$

where Eq. (2.6.63) is used to write the equation in terms of Euler parameter time derivatives and the definition of the *noncentroidal mass matrix* is given in Eq. (4.3.26). Since the integrand on the right of Eq. (4.6.16) is nonnegative at all points in the body,  $KEB^i \geq 0$  for all  $\dot{\mathbf{r}}$  and  $\omega$ , hence all  $\dot{\mathbf{q}}^i$ . Thus, the body mass matrix  $\mathbf{M}_i$  is *positive semidefinite*.

To see that  $\mathbf{M}_i$  for a spatial body in the Euler parameter form of the equations of motion is not *positive definite*, let  $\dot{\bar{\mathbf{p}}} = \bar{\mathbf{p}} \neq \mathbf{0}$  and  $\dot{\bar{\mathbf{r}}} = \mathbf{0}$ ; i.e.,  $\dot{\bar{\mathbf{q}}} = [\mathbf{0} \quad \bar{\mathbf{p}}^T]^T \neq \mathbf{0}$ . The identity of Eq. (2.6.6), which holds even if  $\bar{\mathbf{p}}$  does not satisfy the normalization condition, shows that  $\mathbf{G} \dot{\bar{\mathbf{p}}} \equiv \mathbf{G}(\bar{\mathbf{p}}) \dot{\bar{\mathbf{p}}} = \mathbf{G}(\bar{\mathbf{p}}) \bar{\mathbf{p}} = \mathbf{0}$ . For this  $\dot{\bar{\mathbf{q}}}$ , the last equality of Eq. (4.6.16) shows that  $\dot{\bar{\mathbf{q}}}^T \mathbf{M}_i \dot{\bar{\mathbf{q}}} = 0$ . Thus,  $\mathbf{M}_i$  is not positive definite. In fact, it is singular. Note, however, that the generalized

velocity  $\dot{\bar{\mathbf{p}}} = \bar{\mathbf{p}}$  chosen satisfies  $\bar{\mathbf{p}}^T \dot{\bar{\mathbf{p}}} = 1$  if  $\bar{\mathbf{p}}$  satisfies the Euler parameter normalization constraint. Thus, the velocity  $\dot{\bar{\mathbf{p}}}$  is not kinematically admissible.

Summing Eq. (4.6.16) over all bodies in the system, using the notation of Eq. (4.6.13), yields the *system kinetic energy*

$$KES = \frac{1}{2} \sum_i \dot{\mathbf{q}}_i^T \mathbf{M}_i \dot{\mathbf{q}}_i = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \quad (4.6.17)$$

Since each body mass matrix  $\mathbf{M}_i$  is positive semidefinite, the system mass matrix  $\mathbf{M}$  is positive semidefinite.

*Kinematically admissible virtual velocities* are those that satisfy the holonomic constraint velocity equations with time held fixed; i.e.,

$$\Phi_q \delta \dot{\mathbf{q}} = 0$$

where, for a proper design and model,  $\Phi_q(\mathbf{q})$  has *full rank*.

The *kinetic energy* of virtual velocities is  $(1/2)\delta \dot{\mathbf{q}}^T \mathbf{M} \delta \dot{\mathbf{q}}$ . Physical reasoning leads to the conclusion that the only kinematically admissible virtual velocity for which system kinetic energy is zero is the zero velocity; i.e.,  $\delta \dot{\mathbf{q}} = \mathbf{0}$ . Otherwise, there exists a motion with nonzero virtual velocity that is realized without having done any work on the system to achieve that motion from a configuration at rest. If the reader can find such a system in the real world, fame and fortune will surely follow. No such real system exists. Thus, real *system kinetic energy* is *positive definite* on the space of kinematically admissible virtual velocities; i.e., the *null space* of the *constraint Jacobian*. As shown above, this does not say that matrix  $\mathbf{M}$  is positive definite.

The foregoing result does not say that the system mass matrix of a physically bad model is positive definite on the null space of the model constraint Jacobian. If a model is created using one or more bodies of finite dimension but zero mass and inertia matrix, there is no guarantee that the resulting mass matrix is positive definite on said null space. Similarly, if an Euler parameter normalization condition is not included in the vector of holonomic constraints, the definiteness result is invalidated. Simply stated, the foregoing mathematical result depends on *kinematic modeling* and *kinetic modeling* that represent the actual mechanics of the system considered.

The results of this section are valid for planar systems and systems of particles, as may be verified by even simpler manipulations, using the equations of Section 4.6.1.

#### 4.6.4 Initial Conditions

In addition to the equations of motion of Eqs. (4.6.14) and (4.6.6), *initial conditions* must be specified for  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at an *initial time*  $t^0$ , consistent with constraints of Eqs. (4.6.4), (4.6.5), and (4.6.6). The number of initial position and orientation conditions should be equal to the number of *degrees of freedom*; i.e., the number of generalized coordinates ngc minus the number nhc of independent holonomic constraint equations of Eq. (4.6.1), written in the form

$$\Phi(\mathbf{q}^0) = \begin{bmatrix} \Phi^1(\mathbf{q}^0) \\ \vdots \\ \Phi^{n_{hc}}(\mathbf{q}^0) \end{bmatrix} = 0 \quad (4.6.18)$$

A general form of n<sub>gc</sub> – n<sub>hc</sub> specified *initial configuration conditions* is

$$\Phi^I(\mathbf{q}^0, t^0) = \mathbf{0} \quad (4.6.19)$$

These conditions must be independent among themselves, as well as with the kinematic constraints of Eqs. (4.6.18) at  $t_0$ . The combined conditions,

$$\begin{bmatrix} \Phi(\mathbf{q}^0) \\ \Phi^I(\mathbf{q}^0, t^0) \end{bmatrix} = \mathbf{0} \quad (4.6.20)$$

must uniquely determine  $\mathbf{q}^0 = \mathbf{q}(t^0)$ , which requires the Jacobian of Eq. (4.6.20) with respect to  $\mathbf{q}^0$  is nonsingular. Equations (4.6.20) may be solved using iterative numerical methods for  $\mathbf{q}^0$ .

Similarly, initial velocity conditions may be specified at  $t^0$  as

$$\mathbf{B}^I(\mathbf{q}^0)\dot{\mathbf{q}}^0 = \mathbf{v}^I \quad (4.6.21)$$

where the  $(n_{gc} - n_{hc} - n_d) \times n_{gc}$  matrix  $\mathbf{B}^I(\mathbf{q}^0)$  must be of full rank and independent of the *initial velocity constraint* of Eq. (4.6.5) at  $t^0$ .

The system of n<sub>gc</sub> linear equations of Eqs. (4.6.21) and (4.6.5) may thus be solved for *initial velocity conditions*  $\dot{\mathbf{q}}^0$ .

If initial conditions for spatial bodies are specified in terms of angular velocity at  $t^0$ ; i.e.,

$$\mathbf{B}^I\dot{\mathbf{r}}^0 + \mathbf{C}^I\dot{\mathbf{p}}^0 + \mathbf{D}^I\omega'^0 = \mathbf{v}^I$$

they may be transformed to the form of Eq. (4.6.21) by using the relation  $\omega' = 2\mathbf{G}(\mathbf{p}^0)\dot{\mathbf{p}}$  of Eq. (2.6.63); i.e.,

$$\mathbf{B}^I\dot{\mathbf{r}}^0 + \mathbf{C}^I\dot{\mathbf{p}}^0 + \mathbf{D}^I\omega'^0 = \mathbf{B}^I\dot{\mathbf{r}}^0 + \mathbf{C}^I\dot{\mathbf{p}}^0 + 2\mathbf{D}^I\mathbf{G}(\mathbf{p}^0)\dot{\mathbf{p}}^0 = \mathbf{v}^I$$

Initial conditions for planar systems may be defined in the same way.

Variational equations of motion for a multibody system are obtained by summing variational equations for each body in the system. The result is a single variational equation of motion that must hold for all generalized coordinate variations (virtual displacements) that are consistent with holonomic constraints that act on the system. Due to d'Alembert's principle, this variational equation need not include constraint reaction forces.

Under the physically meaningful condition that kinetic energy is positive for any nonzero kinematically admissible virtual velocity, the system mass matrix is positive definite on the space of kinematically admissible virtual velocities. This is a critically important property in implementing any numerical solution method to simulate the dynamics of multibody systems.

## Key Formulas

$$\Phi^k(\mathbf{q}_i, \mathbf{q}_j, t) = 0 \quad (4.6.1)$$

$$\mathbf{q} = \left[ \mathbf{q}_1^T \cdots \mathbf{q}_{nb}^T \right]^T \quad \delta \mathbf{q} = \left[ \delta \mathbf{q}_1^T \cdots \delta \mathbf{q}_{nb}^T \right]^T \quad \mathbf{M} = \text{diag}(\mathbf{M}_1 \cdots \mathbf{M}_{nb}) \quad (4.6.13)$$

$$\mathbf{Q}^A = \left[ \mathbf{Q}_1^{AT} \cdots \mathbf{Q}_{nb}^{AT} \right]^T \quad \mathbf{S} = \left[ \mathbf{S}_1^T \cdots \mathbf{S}_{nb}^T \right]^T$$

$$\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0 \quad \Phi_q(\mathbf{q})\delta \mathbf{q} = \mathbf{0} \quad (4.6.14) \quad (4.6.6)$$

## 4.7 ODE of Motion in Independent Generalized Coordinates

For holonomic multibody systems whose kinematics can be defined using *independent generalized coordinates*, the variational equations of motion derived in Section 4.6 can be reduced to *ordinary differential equations* (ODE). To illustrate the process, examples that involve particles, planar bodies, and spatial bodies are analyzed in Section 4.7.1. A general formulation is presented in Section 4.7.2, and results on existence, uniqueness, and differentiability of solutions of ODE are summarized in Section 4.7.3.

### 4.7.1 Examples of Systems with Independent Generalized Coordinates

**Example 4.7.1** The *planar double pendulum* model of Fig. 4.7.1 is comprised of two particles with nonzero mass that are connected by massless bars of unit length. Bar one is pivoted at the origin of the x-y frame and bar two is pivoted relative to bar one at mass  $m_1$ . The angles of rotation  $v_1$  and  $v_2$  serve as generalized coordinates that are subject to no constraint; i.e., they are *independent generalized coordinates*. Gravitational force acts in the negative y-direction.

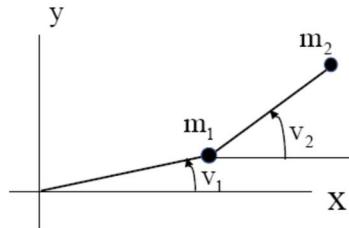


Figure 4.7.1 Planar Double Pendulum

Vectors that locate the masses in the inertial x-y frame are

$$\begin{aligned}\mathbf{r}_1 &= [\cos(v_1) \quad \sin(v_1)]^T \\ \mathbf{r}_2 &= [\cos(v_1) \quad \sin(v_1)]^T + [\cos(v_2) \quad \sin(v_2)]^T\end{aligned}\tag{4.7.1}$$

Their first derivatives and variations yield velocities and virtual displacements of the masses as

$$\begin{aligned}\dot{\mathbf{r}}_1 &= [-\sin(v_1) \quad \cos(v_1)]^T \dot{v}_1 \\ \dot{\mathbf{r}}_2 &= [-\sin(v_1) \quad \cos(v_1)]^T \dot{v}_1 + [-\sin(v_2) \quad \cos(v_2)]^T \dot{v}_2 \\ \delta\mathbf{r}_1 &= [-\sin(v_1) \quad \cos(v_1)]^T \delta v_1 \\ \delta\mathbf{r}_2 &= [-\sin(v_1) \quad \cos(v_1)]^T \delta v_1 + [-\sin(v_2) \quad \cos(v_2)]^T \delta v_2\end{aligned}\tag{4.7.2}$$

Finally, the derivative of velocities yields accelerations of the masses,

$$\begin{aligned}\ddot{\mathbf{r}}_1 &= [-\sin(v_1) \quad \cos(v_1)]^T \ddot{v}_1 - [\cos(v_1) \quad \sin(v_1)]^T (\dot{v}_1)^2 \\ \ddot{\mathbf{r}}_2 &= [-\sin(v_1) \quad \cos(v_1)]^T \ddot{v}_1 - [\cos(v_1) \quad \sin(v_1)]^T (\dot{v}_1)^2 \\ &\quad + [-\sin(v_2) \quad \cos(v_2)]^T \ddot{v}_2 - [\cos(v_2) \quad \sin(v_2)]^T (\dot{v}_2)^2\end{aligned}\tag{4.7.3}$$

The variational equation of motion of the system of two particles, from Eq. (4.1.25), is

$$\delta \mathbf{r}_1^T (m_1 \ddot{\mathbf{r}}_1 + m_1 g \mathbf{u}_y) + \delta \mathbf{r}_2^T (m_2 \ddot{\mathbf{r}}_2 + m_2 g \mathbf{u}_y) = 0 \quad (4.7.4)$$

where  $\mathbf{u}_y$  is the unit vector along the positive y axis and  $g$  is gravitational acceleration.

Substituting from Eqs. (4.7.2) and (4.7.3) and collecting terms, Eq. (4.7.4) reduces to

$$\begin{aligned} \delta v_1 & \left( (m_1 + m_2) \ddot{v}_1 + m_2 (\sin(v_1) \sin(v_2) + \cos(v_1) \cos(v_2)) \ddot{v}_2 \right. \\ & \left. + m_2 (\sin(v_1) \cos(v_2) - \cos(v_1) \sin(v_2)) (\dot{v}_2)^2 + (m_1 + m_2) g \cos(v_1) \right) = 0 \\ \delta v_2 & \left( m_2 (\sin(v_1) \sin(v_2) + \cos(v_1) \cos(v_2)) \ddot{v}_1 + m_2 \ddot{v}_2 \right. \\ & \left. + m_2 (\cos(v_1) \sin(v_2) - \sin(v_1) \cos(v_2)) (\dot{v}_1)^2 + m_2 g \cos(v_2) \right) = 0 \end{aligned} \quad (4.7.5)$$

Since  $v_1$  and  $v_2$  are *independent generalized coordinates*; i.e., subject to no constraints,  $\delta v_1$  and  $\delta v_2$  are arbitrary. Thus, their coefficients in Eq. (4.7.5) must be zero. Using the trigonometric identities  $\sin(v_1) \sin(v_2) + \cos(v_1) \cos(v_2) = \cos(v_2 - v_1)$  and  $\cos(v_1) \sin(v_2) - \sin(v_1) \cos(v_2) = \sin(v_2 - v_1)$  in Eq. (4.7.5) yields the second order system of ODE

$$\begin{aligned} (m_1 + m_2) \ddot{v}_1 + m_2 \cos(v_2 - v_1) \ddot{v}_2 - m_2 \sin(v_2 - v_1) (\dot{v}_2)^2 &= -(m_1 + m_2) g \cos(v_1) \\ m_2 \cos(v_2 - v_1) \ddot{v}_1 + m_2 \ddot{v}_2 + m_2 \sin(v_2 - v_1) (\dot{v}_1)^2 &= -m_2 g \cos(v_2) \end{aligned} \quad (4.7.6)$$

In matrix form, Eqs. (4.7.6) are

$$\begin{bmatrix} (m_1 + m_2) & m_2 \cos(v_2 - v_1) \\ m_2 \cos(v_2 - v_1) & m_2 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} -(m_1 + m_2) g \cos(v_1) \\ -m_2 g \cos(v_2) \end{bmatrix} - \begin{bmatrix} -m_2 \sin(v_2 - v_1) (\dot{v}_2)^2 \\ m_2 \sin(v_2 - v_1) (\dot{v}_1)^2 \end{bmatrix} \quad (4.7.7)$$

The determinant of the symmetric acceleration coefficient matrix of Eq. (4.7.7) is

$$(m_1 + m_2)m_2 - (m_2)^2 (\cos(v_2 - v_1))^2 = m_1 m_2 + (m_2)^2 (1 - (\cos(v_2 - v_{11}))^2) > 0$$

so the matrix is nonsingular. In fact, since each of the *principal subdeterminants* is positive, it is *positive definite* (Strang, 1980).

With initial conditions; e.g., with both masses at rest on the x-axis at time  $t^0 = 0$ ,

$$\begin{aligned} v_1(0) &= v_2(0) = 0 \\ \dot{v}_1(0) &= \dot{v}_2(0) = 0 \end{aligned} \quad (4.7.8)$$

the *initial-value problem* of Eqs. (4.7.7) and (4.7.8) has a unique solution (Teschl, 2012).

---

**Example 4.7.2** The *planar slider-crank* mechanism model of Fig. 4.7.2 is comprised of a crank that is modeled as a planar rigid body of radius  $R$  and polar moment of inertia  $J_1$ , relative to its centroid at the origin of its  $x'-y'$  body reference frame, and a slider along the  $x$ -axis that is modeled as a point mass  $m_2$ . The crank rotates about the origin of the inertial  $x-y$  frame and is connected to the slider by a massless connecting rod of length 2 units.

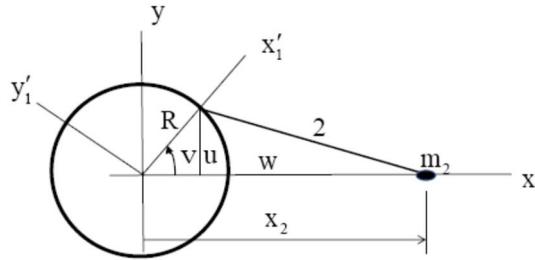


Figure 4.7.2 Planar Slider-Crank

Provided  $R < 2$ , the angle of rotation  $v$  of the crank is an *independent generalized coordinate* that uniquely determines the geometry of the mechanism. With  $u = R\sin(v)$  and  $u^2 + w^2 = (R\sin(v))^2 + w^2 = 4$ ,  $w = \sqrt{4 - (R\sin(v))^2}$  and

$$x_2 = R\cos(v) + w = R\cos(v) + \sqrt{4 - (R\sin(v))^2} \quad (4.7.9)$$

Differentiation yields the velocity, virtual displacement, and acceleration of the slider,

$$\begin{aligned} \dot{x}_2 &= \left( -R\sin(v) - \frac{R^2\sin(v)\cos(v)}{\sqrt{4 - (R\sin(v))^2}} \right) \dot{v} \\ \delta x_2 &= \left( -R\sin(v) - \frac{R^2\sin(v)\cos(v)}{\sqrt{4 - (R\sin(v))^2}} \right) \delta v \\ \ddot{x}_2 &= \left( -R\sin(v) - \frac{R^2\sin(v)\cos(v)}{\sqrt{4 - (R\sin(v))^2}} \right) \ddot{v} \\ &\quad + \left( -R\cos(v) - \frac{R^2(\cos(v)^2 - \sin(v)^2)}{\sqrt{4 - (R\sin(v))^2}} - \frac{R^4(\sin(v)\cos(v))^2}{(4 - (R\sin(v))^2)^{3/2}} \right) \dot{v}^2 \end{aligned} \quad (4.7.10)$$

For the crank, the position, velocity, virtual displacement, and acceleration are zero and the angular acceleration and virtual rotation are  $\ddot{v}$  and  $\delta v$ , respectively.

Summing the variational equations of motion of the planar crank from Eq. (4.2.20) and of the point mass slider from Eq. (4.1.21), the *system variational equation of motion* is

$$\delta v J_1 \ddot{v} + \delta x_2 m_2 \ddot{x}_2 = 0 \quad (4.7.11)$$

Substituting from Eqs. (4.7.10) and collecting terms, Eq. (4.7.11) reduces to

$$\delta v(M(v)\ddot{v} - g(v,\dot{v})) = 0 \quad (4.7.12)$$

where

$$\begin{aligned} M(v) &\equiv J_1 + m_2 \left( R \sin(v) + \frac{R^2 \sin(v) \cos(v)}{\sqrt{4 - (R \sin(v))^2}} \right)^2 > 0 \\ g(v, \dot{v}) &\equiv -m_2 \left( R \sin(v) + \frac{R^2 \sin(v) \cos(v)}{\sqrt{4 - (R \sin(v))^2}} \right) \\ &\quad \times \left( R \cos(v) + \frac{R^2 (\cos(v)^2 - \sin(v)^2)}{\sqrt{4 - (R \sin(v))^2}} - \frac{R^4 (\sin(v) \cos(v))^2}{(4 - (R \sin(v))^2)^{3/2}} \right) \dot{v}^2 \end{aligned} \quad (4.7.13)$$

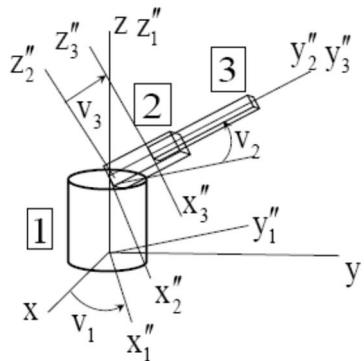
Since  $v$  is independent,  $\delta v$  is arbitrary and its coefficient in Eq. (4.7.12) must be zero. This yields the *second order ODE*

$$M(v)\ddot{v} = g(v,\dot{v}) \quad (4.7.14)$$

Since the coefficient of  $\ddot{v}$  is never zero, the *initial-value problem* of Eq. (4.7.14) and initial conditions such as  $v(0) = 0$  and  $\dot{v}(0) = \omega_0$  has a unique solution, provided  $R < 2$  (Teschl, 2012). Since a closed form solution is out of the question, numerical solution methods are called for.

---

**Example 4.7.3** A three degree of freedom *robotic manipulator*, modeled as a special case of that shown in Fig. 1.2.4, is shown in Fig. 4.7.3. The base, body 1, rotates by angle  $v_1$  about the global  $z$  axis. Body 2 rotates by angle  $v_2$  about the body fixed  $x''_2$  axis that is parallel to the  $x''_1$  axis in body 1. Finally, body 3 translates by distance  $v_3$  along the common  $y''_2$  and  $y''_3$  axes. The height of body 1 is 2m, with the centroid at the midpoint. Bodies 2 and 3 are 2m and 4m in length, respectively, with centroids at their midpoints.



4.7.3 Three Degree of Freedom Robotic Manipulator

For body 1,

$$\begin{aligned}\mathbf{r}_1 &= \mathbf{u}_z \\ \dot{\mathbf{r}}_1 &= \ddot{\mathbf{r}}_1 = \mathbf{0}\end{aligned}\quad (4.7.15)$$

and its angular velocity in the global frame is  $\boldsymbol{\omega}_1 = \dot{\mathbf{v}}_1 \mathbf{u}_z$ . In the body 1 reference frame,

$$\boldsymbol{\omega}'_1 = \mathbf{A}_1^T \boldsymbol{\omega}_1 = \dot{\mathbf{v}}_1 \mathbf{u}_z = [\mathbf{u}_z \quad \mathbf{0} \quad \mathbf{0}] \dot{\mathbf{v}} \equiv \mathbf{W}_1 \dot{\mathbf{v}} \quad (4.7.16)$$

where, with  $c_1 \equiv \cos(v_1)$  and  $s_1 \equiv \sin(v_1)$ ,

$$\mathbf{A}_1 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7.17)$$

Taking differential and derivative yields virtual rotation and angular acceleration

$$\begin{aligned}\delta \boldsymbol{\pi}'_1 &= \mathbf{W}_1 \delta \mathbf{v} \\ \dot{\boldsymbol{\omega}}'_1 &= \mathbf{W}_1 \ddot{\mathbf{v}}\end{aligned}\quad (4.7.18)$$

For body 2, with  $c_2 \equiv \cos(v_2)$  and  $s_2 \equiv \sin(v_2)$ , and the transformation matrix from the body 2 frame to the body 1 frame,

$$\mathbf{A}_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_2 & -s_2 \\ 0 & s_2 & c_2 \end{bmatrix} \quad (4.7.19)$$

$$\mathbf{r}_2 = 2\mathbf{r}_1 + \mathbf{A}_1 \mathbf{A}_{12} \mathbf{u}_y \quad (4.7.20)$$

Taking time derivatives and variation yields velocity, acceleration, and virtual displacement,

$$\begin{aligned}\dot{\mathbf{r}}_2 &= \dot{\mathbf{v}}_1 \mathbf{A}'_1 \mathbf{A}_{12} \mathbf{u}_{y''} + \dot{\mathbf{v}}_2 \mathbf{A}_1 \mathbf{A}'_{12} \mathbf{u}_{y''} = [\mathbf{A}'_1 \mathbf{A}_{12} \mathbf{u}_{y''} \quad \mathbf{A}_1 \mathbf{A}'_{12} \mathbf{u}_{y''} \quad \mathbf{0}] \dot{\mathbf{v}} \equiv \mathbf{V}_2 \dot{\mathbf{v}} \\ \delta \mathbf{r}_2 &= \mathbf{V}_2 \delta \mathbf{v} \\ \ddot{\mathbf{r}}_2 &= \mathbf{V}_2 \ddot{\mathbf{v}} + \dot{\mathbf{v}}_1^2 \mathbf{A}''_1 \mathbf{A}_{12} \mathbf{u}_{y''} + 2\dot{\mathbf{v}}_1 \dot{\mathbf{v}}_2 \mathbf{A}'_1 \mathbf{A}'_{12} \mathbf{u}_{y''} + \dot{\mathbf{v}}_2^2 \mathbf{A}_1 \mathbf{A}''_{12} \mathbf{u}_{y''} \equiv \mathbf{V}_2 \ddot{\mathbf{v}} + \mathbf{U}_2\end{aligned}\quad (4.7.21)$$

where a prime denotes derivative of entries in the associated matrices with respect to their argument  $v_1$  or  $v_2$  and a double prime denotes two derivatives.

The angular velocity of body 2, represented in the body 1 reference frame, is  $(\boldsymbol{\omega}_2)'_1 = \dot{\mathbf{v}}_2 \mathbf{A}_{12} \mathbf{u}_{x''_1}$ . Transforming to the global frame, the angular velocity of body 2 is

$$\boldsymbol{\omega}_2 = \boldsymbol{\omega}_1 + \mathbf{A}_1 (\boldsymbol{\omega}_2)'_1 = \dot{\mathbf{v}}_1 \mathbf{u}_z + \dot{\mathbf{v}}_2 \mathbf{A}_1 \mathbf{A}_{12} \mathbf{u}_{x''_1}$$

Transforming this vector to the body 2 reference frame,

$$\boldsymbol{\omega}'_2 = \mathbf{A}_{12}^T \mathbf{A}_1^T \boldsymbol{\omega}_2 = \dot{\mathbf{v}}_1 \mathbf{A}_{12}^T \mathbf{A}_1^T \mathbf{u}_z + \dot{\mathbf{v}}_2 \mathbf{u}_{x'_1} = [\mathbf{A}_{12}^T \mathbf{A}_1^T \mathbf{u}_z \quad \mathbf{u}_{x'_1} \quad \mathbf{0}] \dot{\mathbf{v}} \equiv \mathbf{W}_2 \dot{\mathbf{v}} \quad (4.7.22)$$

taking the variation and time derivative yield the virtual rotation and angular acceleration

$$\begin{aligned}\delta\boldsymbol{\pi}'_2 &= \mathbf{W}_2 \delta\mathbf{v} \\ \dot{\boldsymbol{\omega}}'_2 &= \mathbf{W}_2 \ddot{\mathbf{v}} + \dot{v}_1^2 \mathbf{A}_{12}^T \mathbf{A}'_{12} \mathbf{u}_z + \dot{v}_1 \dot{v}_2 \mathbf{A}'_{12} \mathbf{A}_1^T \mathbf{u}_z \equiv \mathbf{W}_2 \ddot{\mathbf{v}} + \mathbf{X}_2\end{aligned}\quad (4.7.23)$$

Body 3 is constrained to have the same orientation as body 2, so

$$\begin{aligned}\boldsymbol{\omega}'_3 &= \boldsymbol{\omega}'_2 = \mathbf{W}_2 \dot{\mathbf{v}} \\ \delta\boldsymbol{\pi}'_3 &= \delta\boldsymbol{\pi}'_2 = \mathbf{W}_2 \delta\mathbf{v} \\ \dot{\boldsymbol{\omega}}'_3 &= \dot{\boldsymbol{\omega}}'_2 = \mathbf{W}_2 \ddot{\mathbf{v}} + \mathbf{X}_2\end{aligned}\quad (4.7.24)$$

The position of the centroid of body 3 is  $\mathbf{r}_3 = 2\mathbf{r}_1 + (\mathbf{v}_3 + 2)\mathbf{A}_1 \mathbf{A}_{12} \mathbf{u}_{y_2''}$ . Taking derivatives and variation yield

$$\begin{aligned}\dot{\mathbf{r}}_3 &= \dot{v}_1(\mathbf{v}_3 + 2)\mathbf{A}'_1 \mathbf{A}_{12} \mathbf{u}_{y_2''} + \dot{v}_2(\mathbf{v}_3 + 2)\mathbf{A}_1 \mathbf{A}'_{12} \mathbf{u}_{y_2''} + \dot{v}_3 \mathbf{A}_1 \mathbf{A}_{12} \mathbf{u}_{y_2''} \\ &= \left[ (\mathbf{v}_3 + 2)\mathbf{A}'_1 \mathbf{A}_{12} \mathbf{u}_{y_2''} \quad (\mathbf{v}_3 + 2)\mathbf{A}_1 \mathbf{A}'_{12} \mathbf{u}_{y_2''} \quad \mathbf{A}_1 \mathbf{A}_{12} \mathbf{u}_{y_2''} \right] \dot{\mathbf{v}} \equiv \mathbf{V}_3 \dot{\mathbf{v}} \\ \delta\mathbf{r}_3 &= \mathbf{V}_3 \delta\mathbf{v} \\ \ddot{\mathbf{r}}_3 &= \mathbf{V}_3 \ddot{\mathbf{v}} + \left( \begin{array}{c} \dot{v}_1^2(\mathbf{v}_3 + 2)\mathbf{A}''_1 \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_2(\mathbf{v}_3 + 2)\mathbf{A}'_1 \mathbf{A}'_{12} \mathbf{u}_{y_2''} \\ + \dot{v}_2^2(\mathbf{v}_3 + 2)\mathbf{A}_1 \mathbf{A}''_{12} \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_3 \mathbf{A}'_1 \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_2 \dot{v}_3 \mathbf{A}_1 \mathbf{A}'_{12} \mathbf{u}_{y_2''} \end{array} \right) \\ &\equiv \mathbf{V}_3 \ddot{\mathbf{v}} + \mathbf{U}_3\end{aligned}\quad (4.7.25)$$

Using inertia matrices of the bodies,

$$\begin{aligned}\mathbf{J}'_1 &= J_1 \mathbf{I}_3 \\ \mathbf{J}'_2 &= \mathbf{J}'_3 = \text{diag}(20 \quad 5 \quad 20)\end{aligned}\quad (4.7.26)$$

and summing variational equations of motion of Eq. (4.3.16), the system variational equation of motion is

$$\begin{aligned}\delta\boldsymbol{\pi}'_1^T (\mathbf{J}'_1 \dot{\boldsymbol{\omega}}'_1 + \tilde{\boldsymbol{\omega}}'_1 \mathbf{J}'_1 \boldsymbol{\omega}'_1) + \delta\mathbf{r}_2^T (\mathbf{m}_2 \ddot{\mathbf{r}}_2 + \mathbf{m}_2 \mathbf{g} \mathbf{u}_z) \\ + 2\delta\boldsymbol{\pi}'_2^T (\mathbf{J}'_2 \dot{\boldsymbol{\omega}}'_2 + \tilde{\boldsymbol{\omega}}'_2 \mathbf{J}'_2 \boldsymbol{\omega}'_2) + \delta\mathbf{r}_3^T (\mathbf{m}_3 \ddot{\mathbf{r}}_3 + \mathbf{m}_3 \mathbf{g} \mathbf{u}_z) = 0\end{aligned}\quad (4.7.27)$$

Substitution from Eqs. (4.7.16) through (4.7.26), the variational equation of motion in the variable  $\mathbf{v}$  is

$$\mathbf{v}^T (\mathbf{M}(\mathbf{v}) \ddot{\mathbf{v}} - \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}})) = 0 \quad (4.7.28)$$

where

$$\begin{aligned}\mathbf{M}(\mathbf{v}) &= \mathbf{W}_1^T \mathbf{J}'_1 \mathbf{W}_1 + \mathbf{m}_2 \mathbf{V}_2^T \mathbf{V}_2 + 2\mathbf{W}_2^T \mathbf{J}'_2 \mathbf{W}_2 + \mathbf{m}_3 \mathbf{V}_3^T \mathbf{V}_3 \\ \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}}) &= - \left( \begin{array}{c} \mathbf{W}_1^T (\widetilde{\mathbf{W}_1} \dot{\mathbf{v}}) \mathbf{J}'_1 \mathbf{W}_1 \dot{\mathbf{v}} + \mathbf{m}_2 \mathbf{V}_2^T (\mathbf{U}_2 + \mathbf{g} \mathbf{u}_z) + 2\mathbf{W}_2^T \mathbf{J}'_2 \mathbf{X}_2 \\ + 2\mathbf{W}_2^T (\widetilde{\mathbf{W}_2} \dot{\mathbf{v}}) \mathbf{J}'_2 \mathbf{W}_2 \dot{\mathbf{v}} + \mathbf{m}_3 \mathbf{V}_3^T (\mathbf{U}_3 + \mathbf{g} \mathbf{u}_z) \end{array} \right)\end{aligned}\quad (4.7.29)$$

Since  $\mathbf{v}$  is arbitrary, its coefficient in Eq. (4.7.28) must be zero, yielding the ODE

$$\mathbf{M}(\mathbf{v}) \ddot{\mathbf{v}} = \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}}) \quad (4.7.30)$$

The *system kinetic energy* is

$$\begin{aligned} \mathbf{KE} &= (1/2)\dot{\mathbf{v}}^T \left( m_1 \mathbf{V}_1^T \mathbf{V}_1 + \mathbf{W}_1^T \mathbf{J}'_1 \mathbf{W}_1 + m_2 \mathbf{V}_2^T \mathbf{V}_2 + 2 \mathbf{W}_2^T \mathbf{J}'_2 \mathbf{W}_2 + m_3 \mathbf{V}_3^T \mathbf{V}_3 \right) \dot{\mathbf{v}} \\ &= (1/2)\dot{\mathbf{v}}^T \mathbf{M}(\mathbf{v}) \dot{\mathbf{v}} \geq 0 \end{aligned} \quad (4.7.31)$$

so  $\mathbf{M}(\mathbf{v})$  is positive semidefinite. In fact, since kinetic energy must be positive for any nonzero velocity  $\dot{\mathbf{v}}$ ,  $\mathbf{M}(\mathbf{v})$  must be positive definite, hence nonsingular. Thus, the initial-value problem of Eq. (4.7.30) and initial conditions such as  $\mathbf{v}(0) = \dot{\mathbf{v}}(0) = \mathbf{0}$  has a unique solution (Teschl, 2012).

Oppressive calculations and manipulations would be required to obtain an explicit analytical form of equations of motion, for even this modest multibody system. Needless to say, a closed form solution is not to be had, so numerical methods are required to obtain the solution. For computation, terms in Eq. (4.7.29) can be computer generated and numerical methods of solution may be applied. This strongly suggests that systematic computer methods are required to construct equations of motion for realistic multibody systems, as well as to obtain their solution.

#### 4.7.2 General Form of ODE of Multibody Dynamics

Consider a multibody system with generalized coordinates  $\mathbf{q} \in \mathbb{R}^{ngc}$  that satisfy nhc *independent holonomic constraints*

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (4.7.32)$$

i.e.,  $\Phi_q(\mathbf{q})$  has full row rank. Time derivatives of Eq. (4.7.32) yield the velocity and acceleration constraints

$$\begin{aligned} \Phi_q(\mathbf{q})\dot{\mathbf{q}} &= \mathbf{0} \\ \Phi_q(\mathbf{q})\ddot{\mathbf{q}} + (\Phi_q(\mathbf{q})\ddot{\mathbf{q}})_q &\equiv \Phi_q(\mathbf{q})\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0} \end{aligned} \quad (4.7.33)$$

Define *independent generalized coordinates*  $\mathbf{v} \in \mathbb{R}^m$ ,  $m = ngc - nhc$ , that determine *dependent generalized coordinates* through the transformation

$$\mathbf{q} = \psi(\mathbf{v}) \quad (4.7.34)$$

and its differential and derivatives

$$\begin{aligned} \delta\mathbf{q} &= \psi_v \delta\mathbf{v} \\ \dot{\mathbf{q}} &= \psi_v \dot{\mathbf{v}} \\ \ddot{\mathbf{q}} &= \psi_v \ddot{\mathbf{v}} + (\psi_v \hat{\mathbf{v}})_v \dot{\mathbf{v}} \end{aligned} \quad (4.7.35)$$

If  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  identically satisfy the holonomic constraints; i.e.,

$$\begin{aligned} \Phi(\psi(\mathbf{v})) &= \mathbf{0} \\ \Phi_q(\psi(\mathbf{v}))\psi_v(\mathbf{v})\dot{\mathbf{v}} &= \mathbf{0} \\ \Phi_q(\psi(\mathbf{v}))\psi_v(\mathbf{v})\ddot{\mathbf{v}} + (\Phi_q(\mathbf{q})\hat{\psi}_v(\mathbf{v})\hat{\mathbf{v}})_q \psi_v(\mathbf{v})\dot{\mathbf{v}} + \Phi_q(\psi(\mathbf{v}))\psi(\phi_v(\mathbf{v})\hat{\mathbf{v}})_v \dot{\mathbf{v}} &= \mathbf{0} \end{aligned} \quad (4.7.36)$$

for arbitrary  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ , then  $\psi(\mathbf{v})$  is said to be a *parameterization of the regular constraint manifold* of Section 3.5,

$$\tilde{\mathbf{C}} = \left\{ \mathbf{q} : \Phi(\mathbf{q}) = \mathbf{0}, \text{rank} \Phi_q(\mathbf{q}) = \text{nhc} \right\} \quad (4.7.37)$$

In the language of mechanics (Pars, 1965),  $\mathbf{v}$  and its derivatives are called *Lagrangian generalized coordinates*.

Substituting from Eqs. (4.7.35) into the *variational equation of motion* of the multibody system, Eq. (4.6.16),

$$\delta \mathbf{v}^T \begin{bmatrix} \psi_v^T(\mathbf{v}) \mathbf{M}(\psi(\mathbf{v})) \psi_v(\mathbf{v}) \ddot{\mathbf{v}} + \psi_v^T(\mathbf{v}) \mathbf{M}(\psi(\mathbf{v})) (\psi_v(\mathbf{v}) \hat{\mathbf{v}})_v \dot{\mathbf{v}} \\ -\psi_v^T(\mathbf{v}) \mathbf{S}(\psi(\mathbf{v}), \psi_v(\mathbf{v}) \dot{\mathbf{v}}) - \psi_v^T(\mathbf{v}) \mathbf{Q}^A(\psi(\mathbf{v}), \psi_v(\mathbf{v}) \dot{\mathbf{v}}) \end{bmatrix} = 0 \quad (4.7.38)$$

which must hold for arbitrary  $\delta \mathbf{v}$ . This yields the second order ODE

$$\begin{aligned} & \psi_v^T(\mathbf{v}) \mathbf{M}(\psi(\mathbf{v})) \psi_v(\mathbf{v}) \ddot{\mathbf{v}} + \psi_v^T(\mathbf{v}) \mathbf{M}(\psi(\mathbf{v})) (\psi_v(\mathbf{v}) \hat{\mathbf{v}})_v \dot{\mathbf{v}} \\ & - \psi_v^T(\mathbf{v}) \mathbf{S}(\psi(\mathbf{v}), \psi_v(\mathbf{v}) \dot{\mathbf{v}}) - \psi_v^T(\mathbf{v}) \mathbf{Q}^A(\psi(\mathbf{v}), \psi_v(\mathbf{v}) \dot{\mathbf{v}}) = 0 \end{aligned} \quad (4.7.39)$$

Since the columns of  $\psi_v(\mathbf{v})$  are *kinematically admissible virtual velocities*, the coefficient matrix  $\psi_v^T(\mathbf{v}) \mathbf{M}(\psi(\mathbf{v})) \psi_v(\mathbf{v})$  is positive definite, hence nonsingular. The *initial-value problem* of Eq. (4.7.39) and initial conditions such as  $\mathbf{v}(0) = \dot{\mathbf{v}}(0) = \mathbf{0}$  thus has a unique solution (Teschl, 2012).

The conditions of Eq. (4.7.36) on Lagrangian generalized coordinates are very severe. For example, as shown in Section 3.6, *Lagrangian generalized coordinates* do not exist for the unit sphere in  $\mathbb{R}^3$ . If such coordinates can be found; e.g., as in examples 4.7.1 through 4.7.3, or using the tangent space parameterization of Section 3.5.2, *ODE of dynamics* can be obtained. The benefit of using Lagrangian generalized coordinate thus becomes evident. The generalized coordinates  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  of Eqs. (4.7.34) and (4.7.35) identically satisfy the constraints of Eqs. (4.7.32) and (4.7.33) and satisfy the variational equation of motion, Eq. (4.6.16). They are thus the unique solution of the equations of mechanical system dynamics.

**Example 4.7.4:** For the planar double pendulum of Section 4.7.1,

$$\mathbf{q} = \begin{bmatrix} \mathbf{r}_1^T & \mathbf{r}_2^T \end{bmatrix}^T = \begin{bmatrix} \cos(v_1) & \sin(v_1) & \cos(v_1) + \cos(v_2) & \sin(v_1) + \sin(v_2) \end{bmatrix}^T \equiv \psi(\mathbf{v})$$

Terms required in the ODE of Eq. (4.7.39) are  $\mathbf{S} = \mathbf{0}$  and

$$\psi_v(\mathbf{v}) = \begin{bmatrix} -\sin(v_1) & 0 \\ \cos(v_1) & 0 \\ -\sin(v_1) & -\sin(v_2) \\ \cos(v_1) & \cos(v_2) \end{bmatrix}$$

$$(\psi_v(\mathbf{v}) \hat{\mathbf{v}})_v \dot{\mathbf{v}} = \begin{bmatrix} -\cos(v_1) \dot{v}_1^2 & -\sin(v_1) \dot{v}_1^2 & -\cos(v_1) \dot{v}_1^2 - \cos(v_2) \dot{v}_2^2 & -\sin(v_1) \dot{v}_1^2 - \sin(v_2) \dot{v}_2^2 \end{bmatrix}^T$$

$$\mathbf{M} = \text{diag}(m_1 \ m_1 \ m_2 \ m_2)$$

$$\mathbf{Q}^A = [0 \ -m_1 g \ 0 \ -m_2 g]^T$$

Substituting these relations into Eq. (4.7.39) and collecting terms yields

$$\begin{bmatrix} (m_1 + m_2) & m_2 \cos(v_2 - v_1) \\ m_2 \cos(v_2 - v_1) & m_2 \end{bmatrix} \begin{bmatrix} \ddot{v}_1 \\ \ddot{v}_2 \end{bmatrix} + \begin{bmatrix} -m_2 \sin(v_2 - v_1) (\dot{v}_2)^2 \\ m_2 \sin(v_2 - v_1) (\dot{v}_1)^2 \end{bmatrix} = \begin{bmatrix} -(m_1 + m_2) g \cos(v_1) \\ -m_2 g \cos(v_2) \end{bmatrix} \quad (4.7.40)$$

which is identical to Eq. (4.7.7)

The fact that agreement is achieved in Example 4.7.4 with the results of Section 4.7.1 is not the point here. The takeaway is that matrix manipulations using just the terms leading to Eqs. (4.7.40) can be carried out on the computer to form the ODE of motion for numerical integration, without the ad-hoc manipulations of example 4.7.1. This observation is the basis for a general ODE formulation and numerical methods for solution of the equations of multibody dynamics in Chapter 5, using the tangent space parameterization formulation of Section 3.5.2.

It is not possible in all applications to find Lagrangian generalized coordinates that hold over the full range of motion of the system. It is often possible, however, to define a family of parameterizations  $\mathbf{q} = \psi^i(\mathbf{v}^i)$  such that  $\Phi(\psi^i(\mathbf{v}^i)) = \mathbf{0}$  for  $\mathbf{v}^i \in V_i$ , where  $V_i$  are open sets such that the union of  $U_i = \psi^i(V_i) \subset \tilde{C}$  covers the constraint manifold of Eq. (4.7.37); i.e.,  $\tilde{C} = \cup_i \psi^i(V_i)$  and  $(\psi^i(\mathbf{v}^i), V_i)$  are called *charts* that comprise an *atlas* that covers the constraint manifold  $\tilde{C}$  (Schlichtkrull, 2015; Dundas, 2018). The differential equation of Eq. (4.7.39) is formed and solved until criteria dictate transition to a neighboring chart on which a new differential equation is formed and solved. This yields a continuation of the solution that is shown schematically in Fig. 3.6.2. This approach is adopted in Chapter 5 to create a broadly applicable ODE formulation for multibody dynamics.

### 4.7.3 Theory of ODE

A first order initial-value problem in unknown  $\mathbf{y} \in \mathbb{R}^n$  is

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{f}(\mathbf{y}, \mathbf{b}, t) \\ \mathbf{y}(t^0) &= \mathbf{y}^0 \end{aligned} \quad (4.7.41)$$

where  $\mathbf{f}(\mathbf{y}, \mathbf{b}, t) \in \mathbb{R}^n$  is assumed to be a *continuously differentiable function* of  $\mathbf{y}, \mathbf{b} \in \mathbb{R}^k$ , and  $t$  on the set  $D = \{\mathbf{y}, \mathbf{b}, t : 0 < \|\mathbf{y} - \mathbf{y}^0\| \leq a, 0 < \|\mathbf{b} - \mathbf{b}^0\| \leq c, 0 \leq t - t^0 \leq d\}$ . The vector  $\mathbf{b}$  contains *problem data* such as *design variables* that define the system being modeled. While the assumption of continuous differentiability of  $\mathbf{f}(\mathbf{y}, \mathbf{b}, t)$  can be relaxed slightly to *Lipschitz continuity* (Teschl, 2012), the associated level of mathematical complexity is bypassed here. Results that are proved in texts on the theory of ODE; e.g., (Teschl, 2012), include the following:

**ODE Existence Theorem:** With the foregoing hypotheses, there exist  $\alpha > 0$ ,  $\beta > 0$ , and a unique solution  $\mathbf{y}(t, \mathbf{b})$  of Eq. (4.7.41) on  $0 < t - t^0 \leq \alpha$  and  $0 < \|\mathbf{b} - \mathbf{b}^0\| \leq \beta$  that is continuously differentiable with respect to  $\mathbf{b}$  and  $t$ .

While the *ODE existence theorem* does not specify the magnitudes of nonzero parameters  $\alpha$  and  $\beta$ , continuation concepts such as illustrated in Fig. 3.6.2 show that the solution may be extended until the hypotheses of the theorem are violated.

A *second order initial-value problem* in unknown  $\mathbf{v} \in \mathbb{R}^n$  is

$$\begin{aligned}\ddot{\mathbf{v}} &= \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}, t) \\ \mathbf{v}(t^0) &= \mathbf{v}^0 \\ \dot{\mathbf{v}}(t^0) &= \dot{\mathbf{v}}^0\end{aligned}\tag{4.7.42}$$

with the same hypotheses as in the first order problem of Eq. (4.7.41). Defining

$\mathbf{u}_1 = \mathbf{v}$ ,  $\mathbf{u}_2 = \dot{\mathbf{v}} = \dot{\mathbf{u}}_1$ , and  $\mathbf{u} = [\mathbf{u}_1^T \quad \mathbf{u}_2^T]^T$ , Eq. (4.7.42) may be written as the first order initial-value problem

$$\begin{aligned}\dot{\mathbf{u}} &= \mathbf{F}(\mathbf{u}, \mathbf{b}, t) \equiv [\mathbf{u}_2^T \quad \mathbf{g}^T(\mathbf{u}_1, \mathbf{u}_2, \mathbf{b}, t)]^T \\ \mathbf{u}(t^0) &= \mathbf{u}^0 \equiv [\mathbf{u}_1^{0T} \quad \mathbf{u}_2^{0T}]^T\end{aligned}\tag{4.7.43}$$

Applying the ODE Existence Theorem to Eq. (4.7.43), the equivalent problem of Eq. (4.7.42) has a unique solution  $\mathbf{v}(t, \mathbf{b})$  that is twice continuously differentiable with respect to  $t$  and continuously differentiable with respect to  $\mathbf{b}$ .

The first and second order initial-value problems of Eqs. (4.7.41) and (4.7.42), with the stated hypotheses, are *well posed*, in that they have unique solutions that depend continuously on problem data.

While the ODE existence theorem guarantees there is a unique solution of the initial-value problem in a neighborhood of the initial values, it does not say anything about the nature of the solution. For example, the system may possess highly nonlinear components such as stiff vehicle suspension rebound stops that become active almost instantaneously in an extreme maneuver, exciting high frequency and *highly damped response* over a short period of time. Such behavior is characterized by low frequency smooth behavior most of the time and extreme highly damped high frequency response that must be reacted to and accounted for by solution methods over a short period of time and practically ignored after the amplitude of high frequency response has dissipated. Behavior of this sort occurs in well-designed mechanical systems that encounter moderately violent events, react in a controlled way, and return to a more normal mode of operation. Systems with behavior such as this are called *stiff*, or are said to exhibit *stiff behavior*. While there is no rigorous definition of a *stiff system*, the above noted modes of dynamic response let one recognize stiff behavior when it is encountered. A more practical criterion for a stiff system is that explicit numerical integration methods fail when assigned the task of simulating motion of a stiff system. This criterion is addressed in Section 4.8.

Variational equations of motion of three simple multibody systems, two planar and one spatial, are written in terms of Lagrangian generalized coordinates, yielding ODE of motion. The analytical manipulations required in this process are shown to be intricate and ad-hoc.

An approach that generalizes the procedure used in the examples is outlined and applied to one of the examples. It is shown that analytical implementation of the approach is not likely to be successful for general multibody systems, but the process can be continued on the constraint manifold until singular behavior is encountered. If functions that define first and second order ODE initial-value problems are smooth, the problems are well posed; i.e., unique solutions exist that depend continuously on time and problem data.

## 4.8 Runge-Kutta Methods for Numerical Solution of ODE

A vast literature has evolved during the past century on numerical methods for solution of *first order ordinary differential equation* (ODE) *initial-value problems*,

$$\begin{aligned}\dot{\mathbf{y}} &= \mathbf{f}(\mathbf{y}, t) \\ \mathbf{y}(t^0) &= \mathbf{y}^0\end{aligned}\tag{4.8.1}$$

where  $\mathbf{y} = [y_1 \dots y_m]^T$  and the function  $\mathbf{f}$  is continuously differentiable and bounded.

Numerical algorithms are generally based on approximating a solution on a *discrete time grid*  $t_i$ ,  $i=1, 2, \dots$  until a desired final time  $t_f$  is reached. The two most common methods are (1) *Runge-Kutta* (RK) methods that are based on *Taylor series approximation* of the solution at time  $t_n$  and leading to an approximate solution at time step  $t_{n+1}$  and (2) *backward differentiation* (BDF) methods that are based on interpolating solution data at  $t_n$  and prior time steps and projecting to an approximate solution at  $t_{n+1}$ . For a number of reasons associated with *integration on the constraint manifold*, *Runge-Kutta methods* are employed in this text. Attempting to cite even the most significant contributions to this field is a hopeless task. The most comprehensive documentation of developments to date are two volumes by Hairer, et.al., (1993; 1996) that provide the foundation for methods presented in this section.

The focus of this section is on *Runge-Kutta numerical methods* that are well suited for integration of the equations of mechanical system dynamics, with no attempt at deriving such methods. References to the literature for properties of algorithms and their computer implementation are considered adequate for the purposes of this text. Two distinctly different approaches to numerical integration are *explicit numerical integration methods* that use data at the current time step to predict the solution at the next step and *implicit numerical integration methods* that include the approximate solution at the next time step in the numerical integration equations, thus requiring an *iterative solution of nonlinear equations*. The former approach is attractive for its simplicity and accuracy for many applications. The latter approach is more complex, but is applicable for solution of mechanical systems with *stiff behavior* that defeats explicit methods.

Section 4.8.1 summarizes explicit and implicit *Runge-Kutta methods for solution of first order ODE*. Section 4.8.2 reduces second order differential equations to a pair of first order equations and applies Runge-Kutta methods to the resulting first order equations. The result is transformed back to the second order setting, yielding a family of methods for *numerical solution of second order ODE*. MATLAB computer codes that implement these numerical integration methods are contained in files that accompany this text, use of which is presented in Appendix 4.A.

### 4.8.1 Runge-Kutta Methods for First Order ODE

Runge-Kutta numerical integration methods have been well developed for the solution of the first order ODE of Eq. (4.8.1), where  $\mathbf{y}$  and  $\mathbf{f}$  are  $m$ -vectors. Unless otherwise indicated, results presented in this section are based on (Hairer, et. al., 1993; 1996). Runge-Kutta methods are *one step methods* that use no data prior to  $t_n$  to approximate the solution at time step  $t_{n+1}$ ,

using formulas based on *Taylor series approximation* at  $t_n$ . A broad range of *RK integrators* for this problem can be written in the form of *stage equations*

$$\mathbf{K}_i = \mathbf{f} \left( t_n + c_i h, \mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{K}_j \right), \quad i=1, \dots, s \quad (4.8.2)$$

and the *step solution equation*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{K}_i \quad (4.8.3)$$

where  $t_n$  is the current time step;  $\mathbf{y}_n$  is the approximate solution at  $t_n$ ;  $a_{ij}$ ,  $b_i$ , and  $c_i = \sum_{j=1}^i a_{ij}$  are constants;  $\sum_{i=1}^s b_i = 1$ ;  $s$  is the number of *stages* in integrating from  $t_n$  to  $t_{n+1}$ ;  $\mathbf{K}_i$  are *stage variables*; and  $h$  is the *integration step-size*. It is important to note that stage variables are not solution variables  $\mathbf{y}$  or their derivatives.

A concise representation of data for a broad spectrum of RK methods is the *Butcher tableau* of Table 4.8.1. In all cases treated here,  $a_{ij} = 0$  for  $j > i$ . If  $a_{ii} = 0$  for all  $i = 1, \dots, s$ , then  $\mathbf{K}_i$  does not appear on the right side of Eq. (4.8.2) and  $\mathbf{K}_i$  is explicitly determined by Eq. (4.8.2) as a function of known data, so the method is called an *explicit method*. If  $a_{ii} \neq 0$  for any  $i$ , then  $\mathbf{K}_i$  appears on both sides of Eq. (4.8.2), constituting an *implicit method*. Since  $\mathbf{f}$  is virtually always a nonlinear function of its arguments, an iterative method for solving for the stage variables in implicit methods is required.

Table 4.8.1. Butcher Tableau

$c_i$	$a_{ij}$			
$c_1$	$a_{11}$	0	0	0
$c_2$	$a_{21}$	$a_{22}$	0	0
$\vdots$	$\vdots$		$\ddots$	
$c_s$	$a_{s1}$	$a_{s2}$	$\cdots$	$a_{ss}$
$b_i$	$b_1$	$b_2$	$\cdots$	$b_s$

It is intuitively clear that an explicit method will have difficulty responding to an event such as encounter of a stiff spring and associated damping, called a *stiff event*, until significant error has accumulated. A brief discussion of situations in mechanical system dynamics that yield stiffness is presented in Section 4.7.3. An implicit method, in contrast, must deal with the onset of an extreme event at the instant it occurs. *Variable step size* mechanisms associated with implicit methods enable immediate reduction of step size to accurately capture a stiff event and to increase the step size after the amplitude of response has reduced below an established error tolerance. This is a pseudo definition of *stiffness* as behavior that defeats explicit integration methods and requires implicit integration methods. These concepts will become clear as

integration methods are presented in this section and used in subsequent chapters to treat systems with stiff behavior.

#### 4.8.1.1 Explicit Runge-Kutta Methods for First Order ODE

One of the most popular RK methods is of order 4, called *Runge-Kutta4*. It is specified by the tableau of Table 4.8.2 (Hairer, Norsett, and Wanner, 1993, p. 138). Its local truncation error is of order  $h^5$ . Since all elements on the diagonal are zero, it is an *explicit method*. The *stage equations* of Eq. (4.8.2), with data from Table 4.8.2 are

$$\begin{aligned} \mathbf{K}_1 &= \mathbf{f}(t_n, \mathbf{y}_n) \\ \mathbf{K}_2 &= \mathbf{f}\left(t_n + h/2, \mathbf{y}_n + (h/2)\mathbf{K}_1\right) \\ \mathbf{K}_3 &= \mathbf{f}\left(t_n + h/2, \mathbf{y}_n + (h/2)\mathbf{K}_2\right) \\ \mathbf{K}_4 &= \mathbf{f}\left(t_n + h, \mathbf{y}_n + h\mathbf{K}_3\right) \end{aligned} \quad (4.8.4)$$

The approximate solution at  $t_{n+1}$  is given by Eq. (4.8.3), with data from Table 4.8.2 ,

$$\mathbf{y}_{n+1} = \mathbf{y}_n + (h/6)(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4) \quad (4.8.5)$$

Table 4.8.2. Runge-Kutta 4

$c_i$	$a_{ij}$			
0	0			
1/2	1/2	0		
1/2	0	1/2	0	
1	0	0	1	0
$b_i$	1/6	2/6	2/6	1/6

RK methods for first order ODE have been developed using *embedded methods* that have the same coefficients  $a_{ij}$ , but different  $b_i$  in Table 4.8.1. This leads to two approximate solutions of different orders of accuracy, whose difference serves as an *estimate of error* in the underlying method. The tableau of Table 4.8.1 for these methods is extended to the form of Table 4.8.3.

Table 4.8.3. Butcher Tableau for Embedded Method

$c_i$	$a_{ij}$			
$c_1$	$a_{11}$	0	0	0
$c_2$	$a_{21}$	$a_{22}$	0	0
$\vdots$	$\vdots$	$\ddots$		
$c_s$	$a_{s1}$	$a_{s2}$	$\cdots$	$a_{ss}$
$b_i$	$b_1$	$b_2$	$\cdots$	$b_s$
$d_i$	$d_1$	$d_2$	$\cdots$	$d_s$

Given stage variables  $\mathbf{K}_i$  that satisfy Eq. (4.8.2), two approximate solutions are obtained using Eq. (4.8.3),

$$\begin{aligned}\mathbf{y}_{n+1} &= \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{K}_i \\ \hat{\mathbf{y}}_{n+1} &= \mathbf{y}_n + h \sum_{i=1}^s d_i \mathbf{K}_i\end{aligned}\tag{4.8.6}$$

The difference between the solutions,  $\mathbf{y}_n - \hat{\mathbf{y}}_n$ , is an *estimate of solution error*. A *measure of integration error* at time  $t_n$  is suggested as (Hairer, Norsett, and Wanner, 1993)

$$\text{err} = \sqrt{\frac{1}{m} \sum_{i=1}^m \left( \frac{y_{ni} - \hat{y}_{ni}}{sc_i} \right)^2}\tag{4.8.7}$$

where a desired bound on error is

$$sc_i = Atol + |y_{ni}| Rtol\tag{4.8.8}$$

and Atol and Rtol are *absolute error tolerance* and *relative error tolerance* set by the analyst. Provided  $\text{err} < 1$ , satisfactory solution accuracy is declared. If  $p$  is the order of the underlying method, its error is proportional to  $h^{p+1}$  (Hairer, Norsett, and Wanner, 1993). Assuming the same constant of proportionality for the embedded method of order  $h^p$ , an *optimum step size* is

$$h_{\text{opt}} = h \times (1/\text{err})^{1/(p+1)}\tag{4.8.9}$$

If, during the numerical integration process, err of Eq. (4.8.7) approaches one, a new *step size* based on Eq. (4.8.9) is selected. Factors are introduced to assure that the new step size leads to stable results, limiting the rate of decrease or increase of  $h$ .

One of the best known *adaptive explicit RK methods* is the *Runge-Kutta-Fehlberg45* method, given by the tableau of Table 4.8.4 (Hairer, Norsett, and Wanner, 1993, p. 177). The method has error bounded by  $h^5$  and is able to take steps as large as possible, while controlling error using Eqs. (4.8.6) to (4.8.9).

Table 4.8.4. Runge-Kutta-Fehlberg 45

$c_i$	$a_{ij}$					
0	0					
$\frac{1}{4}$	$\frac{1}{4}$	0				
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$	0			
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$	0		
1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	0	
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	0
$b_i$	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$	0
$d_i$	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$

#### 4.8.1.2 Implicit Runge-Kutta Methods for First Order ODE

While it can be interpreted as a Runge-Kutta method (Ascher and Petzold, 1998), the *implicit trapezoidal integration formula* (Atkinson, 1989) is presented here in terms of generalized coordinates, rather than stage variables,

$$\mathbf{y}_n = \mathbf{y}_{n-1} + (h/2)(\dot{\mathbf{y}}_{n-1} + \dot{\mathbf{y}}_n) \quad (4.8.10)$$

The *trapezoidal formula* is an order two, *A-stable* method. The important thing to point out about it is that, although the formula is *stiffly accurate*, this does not result in *L-stability* (Hairer and Wanner, 1996), so it has difficulty in treating *stiff systems*.

Numerical integration can be implemented with a *singly diagonal implicit Runge-Kutta* (SDIRK) formula, in which all diagonal entries in the tableau have the same nonzero value  $a_{ii} = \alpha \neq 0$ . For *SDIRK formulas*, the associated Butcher tableau assumes the form shown in

Table 4.8.5. The  $i$ th equation of Eqs. (4.8.2) is  $\mathbf{K}_i = \mathbf{f}\left(t_n + c_i h, \mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{K}_j\right)$ , where  $\mathbf{K}_1 \cdots \mathbf{K}_{i-1}$

have been determined as solutions for  $i = 1 \cdots s$ . Since  $\mathbf{K}_i$  appears on both sides of this equation and  $\mathbf{f}(t, \mathbf{y})$  is generally nonlinear, an iterative solution method for solving the equation, written in residual form,

$$\mathbf{R}(\mathbf{K}_i) \equiv \mathbf{K}_i - \mathbf{f} \left( t_n + c_i h, \mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{K}_j \right) = \mathbf{0} \quad (4.8.11)$$

must be employed. With an estimate  $\mathbf{K}_i^j$  and Jacobian  $\mathbf{R}_{\mathbf{K}_i}(\mathbf{K}_i^j)$  of the residual, *Newton-Raphson iteration* is

$$\begin{aligned} \mathbf{R}_{\mathbf{K}_i}(\mathbf{K}_i^j) \Delta \mathbf{K}_i^j &= (\mathbf{I} - h \mathbf{f}_y(\mathbf{K}_i^j)) \Delta \mathbf{K}_i^j = -\mathbf{R}(\mathbf{K}_i^j) \\ \mathbf{K}_i^{j+1} &= \mathbf{K}_i^j + \Delta \mathbf{K}_i^j \end{aligned} \quad (4.8.12)$$

$j = 0, 1, \dots$ , until convergence criteria are met. A reasonable estimate for  $\mathbf{K}_j^0$  is the solution from the preceding stage and, for  $\mathbf{K}_1^0$ ,  $\dot{\mathbf{y}}_n$  from previous integration result. The approximate solution at  $t_{n+1}$  is then evaluated using Eq. (4.8.3). This class of methods has accuracy of order  $h^s$  and attractive *stability properties*, including the ability to successfully integrate stiff systems that defeat explicit methods.

Table 4.8.5. Butcher's Tableau for SDIRK Methods

$c_i$	$a_{ij}$			
$c_1$	$\alpha$			
$c_2$	$a_{21}$	$\alpha$	$\dots$	
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$c_s$	$a_{s1}$	$a_{s2}$	$\dots$	$\alpha$
$b_i$	$b_1$	$b_2$	$\dots$	$b_s$

The SDIRK formula selected for integrating potentially stiff differential equations should be L-stable (Hairer and Wanner, 1996) and of average order. The L-stability attribute ensures good stability properties and order-preservation, even for stiff problems. The formula commonly chosen is of order  $p = 4$ , with  $s = 5$  stages, hence the name *SDIRK54*. This order is high enough to ensure good efficiency for tolerances typically used in simulations of engineering application, namely  $10^{-2}$  to  $10^{-5}$ . The *stiffly-accurate*, L-stable, 5 stage, order 4 *SDIRK54* formula for first order ODE, with *imbedded error estimation*, is defined in Table 4.8.6 (Hairer and Wanner, 1996, p.100).

Table 4.8.6 SDIRK54

$c_i$	$a_{ij}$				
1/4	1/4				
3/4		1/2	1/4		
11/20		17/50	-1/25	1/4	
1/2		371/1360	-137/2720	15/544	1/4
1		25/24	-49/48	125/16	-85/12
$b_i$		25/24	-49/48	125/16	-85/12
$d_i$	59/48	-17/96	225/32	-85/12	0

#### 4.8.2 Runge-Kutta Methods for Second Order ODE

In order to adapt Runge-Kutta methods such as those presented in Section 4.8.1 for first order ODE for integration of *second order ODE*, the second order equation is first reduced to a pair of first order equations. The method of Eqs. (4.8.2) and (4.8.3) is then applied and the result is transformed back to the second order setting, as a *second order integration algorithm*.

For the *second order initial-value problem* in  $\mathbf{v} \in \mathbb{R}^m$  and  $t$ ,

$$\begin{aligned}\ddot{\mathbf{v}} &= \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \\ \mathbf{v}(t^0) &= \mathbf{v}^0 \\ \dot{\mathbf{v}}(t^0) &= \dot{\mathbf{v}}^0\end{aligned}\tag{4.8.13}$$

define the pair of variables

$$\begin{aligned}\mathbf{u} &= \mathbf{v} \\ \mathbf{w} &= \dot{\mathbf{v}} = \dot{\mathbf{u}}\end{aligned}\tag{4.8.14}$$

The second order equation of Eq. (4.8.13) is thus equivalent to the system of first order equations in  $\mathbf{y} \equiv [\mathbf{u}^T \quad \mathbf{w}^T]^T$ ,

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{w} \\ \mathbf{g}(t, \mathbf{u}, \mathbf{w}) \end{bmatrix} \equiv \mathbf{f}(t, \mathbf{u}, \mathbf{w})\tag{4.8.15}$$

Applying Eq. (4.8.2), with  $\mathbf{K}_j = [\mathbf{k}\mathbf{u}_j^T \quad \mathbf{k}\mathbf{w}_j^T]^T$  and

$$\mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{K}_j = \begin{bmatrix} \mathbf{u}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{u}_j \\ \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \end{bmatrix}\tag{4.8.16}$$

the stage equations for the ODE of Eq. (4.8.15) are

$$\begin{bmatrix} \mathbf{k}\mathbf{u}_i \\ \mathbf{k}\mathbf{w}_i \end{bmatrix} = \begin{bmatrix} \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \\ \mathbf{g} \left( t_n + c_i h, \mathbf{u}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{u}_j, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \right) \end{bmatrix} \quad (4.8.17)$$

In component form, this is

$$\mathbf{k}\mathbf{u}_i = \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \quad (4.8.18)$$

$$\mathbf{k}\mathbf{w}_i = \mathbf{g} \left( t_n + c_i h, \mathbf{u}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{u}_j, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \right) \quad (4.8.19)$$

Writing Eq. (4.8.18) in the form  $\mathbf{k}\mathbf{u}_j = \mathbf{w}_n + h \sum_{\ell=1}^j a_{j\ell} \mathbf{k}\mathbf{w}_\ell$ , the upper limit of summation can be changed to  $s$ , because if  $\ell > j$ , then  $a_{j\ell} = 0$ . Substituting this into the second argument on the right of Eq. (4.8.19),

$$\begin{aligned} \mathbf{k}\mathbf{w}_i &= \mathbf{g} \left( t_n + c_i h, \mathbf{u}_n + h \sum_{j=1}^i a_{ij} \left( \mathbf{w}_n + h \sum_{\ell=1}^s a_{j\ell} \mathbf{k}\mathbf{w}_\ell \right), \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \right) \\ &= \mathbf{g} \left( t_n + c_i h, \mathbf{u}_n + h \left( \sum_{j=1}^i a_{ij} \right) \mathbf{w}_n + h^2 \sum_{j=1}^s \sum_{\ell=1}^s a_{ij} a_{j\ell} \mathbf{k}\mathbf{w}_\ell, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \right) \end{aligned} \quad (4.8.20)$$

where the upper limit of summation over  $j$  is changed to  $s$ , since if  $j > i$ , then  $a_{ij} = 0$ . Define

$$A_{i\ell} = \sum_{j=1}^s a_{ij} a_{j\ell} \quad (4.8.21)$$

or in matrix form,  $\mathbf{A} = [A_{ij}] = \mathbf{a}\mathbf{a}$ , where  $\mathbf{a} = [a_{ij}]$ . The only nonzero terms in the sum of Eq. (4.8.21) are with  $j \leq i$  and  $\ell \leq j$ ; i.e., with  $\ell \leq i$ . Thus, if  $\ell > i$ , either  $a_{ij} = 0$  or  $a_{j\ell} = 0$ , so  $A_{i\ell} = 0$  and the matrix  $\mathbf{A}$  has zeros above the diagonal, just as  $\mathbf{a}$ .

These manipulations lead to the stage equation of Eq. (4.8.20) for the second order differential equation of Eq. (4.8.13), in terms of only stage variable  $\mathbf{k}\mathbf{w}_i$ ,

$$\mathbf{k}\mathbf{w}_i = \mathbf{g} \left( t_n + c_i h, \mathbf{u}_n + h c_i \mathbf{w}_n + h^2 \sum_{\ell=1}^i A_{i\ell} \mathbf{k}\mathbf{w}_\ell, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}\mathbf{w}_j \right) \quad (4.8.22)$$

where  $c_i = \left( \sum_{j=1}^i a_{ij} \right)$  and  $A_{ij} = 0$  if  $j > i$ , so the upper limit of summation is changed from  $s$  to  $i$ .

Using Eq. (4.8.3), the approximate solution of Eq. (4.8.15) is

$$\begin{bmatrix} \mathbf{u}_{n+1} \\ \mathbf{w}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_n \\ \mathbf{w}_n \end{bmatrix} + h \sum_{i=1}^s b_i \begin{bmatrix} \mathbf{k}\mathbf{u}_i \\ \mathbf{k}\mathbf{w}_i \end{bmatrix} \quad (4.8.23)$$

or in component form

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + h \sum_{j=1}^s b_j \mathbf{k}\mathbf{u}_j \\ \mathbf{w}_{n+1} &= \mathbf{w}_n + h \sum_{j=1}^s b_j \mathbf{k}\mathbf{w}_j \end{aligned} \quad (4.8.24)$$

Substituting from Eq. (4.8.18) into the first of Eqs. (4.8.24) yields

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + h \sum_{j=1}^s b_j \left( \mathbf{w}_n + h \sum_{\ell=1}^j a_{j\ell} \mathbf{k}\mathbf{w}_\ell \right) \\ &= \mathbf{u}_n + h \left( \sum_{j=1}^s b_j \right) \mathbf{w}_n + h^2 \sum_{j=1}^s \sum_{\ell=1}^s b_j a_{j\ell} \mathbf{k}\mathbf{w}_\ell \\ &= \mathbf{u}_n + h \mathbf{w}_n + h^2 \sum_{\ell=1}^s B_\ell \mathbf{k}\mathbf{w}_\ell \end{aligned} \quad (4.8.25)$$

where  $\sum_{j=1}^s b_j = 1$  has been used, the upper limit of summation on the right has been increased to  $s$ ,

for the usual reason, and  $B_\ell$  is defined as

$$B_\ell = \sum_{j=1}^s b_j a_{j\ell} \quad (4.8.26)$$

or in matrix form,  $\mathbf{B} = \mathbf{b}\mathbf{a}$ .

Returning to the variables  $v$  and  $\dot{v}$ , using Eqs. (4.8.14) to eliminate  $\mathbf{u}$  and  $\mathbf{w}$ , and defining  $\mathbf{k}_i = \mathbf{k}\mathbf{w}_i$ , yields the *second order stage equation* of Eq. (4.8.22) for *Runge-Kutta integration of second order ODE*,

$$\mathbf{k}_i = \mathbf{g} \left( t_n + c_i h, v_n + h c_i \dot{v}_n + h^2 \sum_{j=1}^i A_{ij} \mathbf{k}_j, \dot{v}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}_j \right) \quad (4.8.27)$$

and, from Eqs. (4.8.24) and (4.8.25), the approximate solution at  $t_{n+1}$  is

$$\begin{aligned} \mathbf{v}_{n+1} &= \mathbf{v}_n + h \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^s B_j \mathbf{k}_j \\ \dot{\mathbf{v}}_{n+1} &= \dot{\mathbf{v}}_n + h \sum_{j=1}^s b_j \mathbf{k}_j \end{aligned} \quad (4.8.28)$$

The tableau of Table 4.8.1 is extended to represent RK formulas for second order ODE as Table 4.8.7. It contains all data needed to implement Eqs. (4.8.27) and (4.8.28).

Table 4.8.7. Runge-Kutta Tableau for Second Order ODE

$c_i$	$a_{ij}$				$A_{ij}$				
$c_1$	$a_{11} \quad 0 \quad 0 \quad 0$				$A_{11} \quad 0 \quad 0 \quad 0$				
$c_2$	$a_{21} \quad a_{22} \quad 0 \quad 0$				$A_{21} \quad A_{22} \quad 0 \quad 0$				
$\vdots$	$\vdots \quad \ddots$				$\vdots \quad \ddots$				
$c_s$	$a_{s1} \quad a_{s2} \quad \cdots \quad a_{ss}$				$A_{s1} \quad A_{s2} \quad \cdots \quad A_{ss}$				
$b_i \rightarrow$	$b_1 \quad b_2 \quad \cdots \quad b_s$				$B_1 \quad B_2 \quad \cdots \quad B_s$				$\leftarrow B_i$

If any  $a_{ii} \neq 0$  in Table 4.8.7, the unknown  $\mathbf{k}_i$  appears on both sides of Eq. (4.8.27) and the method is implicit. Nonlinear discretized equations for the ODE of Eq. (4.8.27) must then be solved for  $\mathbf{k}_i$ ,  $i = 1, \dots, s$ . To focus on the solution variable  $\mathbf{k}_i$  in the  $i$ th equation, where  $\mathbf{k}_j$ ,  $j = 1, \dots, i-1$ , have been determined by the first  $i-1$  of Eqs. (4.8.27), define

$$\begin{aligned}\mathbf{v}'_i &= \mathbf{v}_n + hc_i \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^{i-1} A_{ij} \mathbf{k}_j \\ \dot{\mathbf{v}}'_i &= \dot{\mathbf{v}}_n + h \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j\end{aligned}\tag{4.8.29}$$

Equation (4.8.27) may thus be written as

$$\mathbf{k}_i = \mathbf{g}(t_n + c_i h, \mathbf{v}'_i + h^2 A_{ii} \mathbf{k}_i, \dot{\mathbf{v}}'_i + h a_{ii} \mathbf{k}_i)\tag{4.8.30}$$

This form of the equation shows clearly the dependence on  $\mathbf{k}_i$  and identifies terms that do not vary during iteration for  $\mathbf{k}_i$ . In particular, if  $a_{ii} = A_{ii} = 0$ ,  $i = 1, \dots, s$ , the right side of Eq. (4.8.30) does not depend on  $\mathbf{k}_i$  and the associated method is explicit.

#### 4.8.2.1 Explicit Runge-Kutta Methods for Second Order ODE

Several explicit RK methods are used in solving second order ODE. A few of the most common are presented here. For calculation with  $a_{ii} = A_{ii} = 0$ ,  $i = 1, \dots, s$ , in solving Eq. (4.8.13), Eq. (4.8.30) yields

$$\mathbf{k}_i = \mathbf{g}(t_n + c_i h, \mathbf{v}'_i, \dot{\mathbf{v}}'_i)\tag{4.8.31}$$

An explicit RK method for solving second order differential equations is the order 4 *Nystrom4* method (Hairer, Norsett, and Wanner, 1993, p.284) that is characterized by the tableau of Table 4.8.8. Note that  $A_{ij}$  in Table 4.8.8 do not satisfy Eq. (4.8.21).

Table 4.8.8. Nystrom4

$c_i$	$a_{ij}$				$A_{ij}$				
0	0				0				
1/2	1/2 0				1/8 0				
1/2	0 1/2 0				1/8 0 0				
1	0 0 1 0				0 0 1/2 0				
$b_i \rightarrow$	1/6 2/6 2/6 1/6				1/6 1/6 1/6 0				$\leftarrow B_i$

Stage equations of Eq. (4.8.31) for this method, in solving Eq. (4.8.13), are

$$\begin{aligned}
 \mathbf{k}_1 &= \mathbf{g}(t_n, \mathbf{v}_n, \dot{\mathbf{v}}_n) \\
 \mathbf{k}_2 &= \mathbf{g}\left(t_n + \frac{h}{2}, \mathbf{v}_n + \frac{h}{2}\dot{\mathbf{v}}_n + \frac{h^2}{8}\mathbf{k}_1, \dot{\mathbf{v}}_n + \frac{h}{2}\mathbf{k}_1\right) \\
 \mathbf{k}_3 &= \mathbf{g}\left(t_n + \frac{h}{2}, \mathbf{v}_n + \frac{h}{2}\dot{\mathbf{v}}_n + \frac{h^2}{8}\mathbf{k}_1, \dot{\mathbf{v}}_n + \frac{h}{2}\mathbf{k}_2\right) \\
 \mathbf{k}_4 &= \mathbf{g}\left(t_n + h, \mathbf{v}_n + h\dot{\mathbf{v}}_n + \frac{h^2}{2}\mathbf{k}_3, \dot{\mathbf{v}}_n + h\mathbf{k}_3\right)
 \end{aligned} \tag{4.8.32}$$

The approximate solution at  $t_{n+1}$ , from Eqs. (4.8.28), is

$$\begin{aligned}
 \mathbf{v}_{n+1} &= \mathbf{v}_n + h\dot{\mathbf{v}}_n + \frac{h^2}{6}(\mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3) \\
 \dot{\mathbf{v}}_{n+1} &= \dot{\mathbf{v}}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)
 \end{aligned} \tag{4.8.33}$$

The second order ODE tableau of Table 4.8.9 is based on the RK4 tableau of Table 4.8.2 and satisfies Eq. (4.8.21). The tableau of Table 4.8.8, however, has yielded equally accurate results on problems with known solutions.

Table 4.8.9. Second Order Runge-Kutta4

$c_i$	$a_{ij}$				$A_{ij}$				
0	0				0				
1/2	1/2 0				0 0				
1/2	0 1/2 0				1/4 0 0				
1	0 0 1 0				0 1/2 0 0				
$b_i \rightarrow$	1/6 2/6 2/6 1/6				1/6 1/6 1/6 0				$\leftarrow B_i$

To create a second order ODE integrator with *error control*, based on the Runge-Kutta-Fehlberg45 method of Section 4.8.1.1, parameters  $A_{ij}$  are defined by Eq. (4.8.21), second order solution evaluation parameters  $B_i$  are defined by Eq. (4.8.26), and embedded error estimation parameters  $D_i$  are defined as

$$D_i = \sum_{j=1}^s d_j a_{ji} \quad (4.8.34)$$

The resulting tableau is presented in Table 4.8.10, called *Runge-Kutta-Fehlberg-Nystrom45*, or *RKFN45*, to reflect the contributing components. Numerical values for  $A_{ij}$ ,  $B_i$ , and  $D_i$  are given to four places to illustrate their orders of magnitude. Numerical values computed from Eqs. (4.8.21), (4.8.26), and (4.8.34) that are accurate to computer precision are used in implementation of the algorithm.

Table 4.8.10. Runge-Kutta-Fehlberg-Nystrom45

$c_i$	$a_{ij}$						$A_{ij}$						
0	0						0						
$\frac{1}{4}$	$\frac{1}{4}$ 0						0 0						
$\frac{3}{8}$	$\frac{3}{32}$ $\frac{9}{32}$ 0						0.0703 0 0						
$\frac{12}{13}$	$\frac{1932}{2197}$ $-\frac{7200}{2197}$ $\frac{7296}{2197}$ 0						-0.5080 0.9340 0 0						
1	$\frac{439}{216}$ -8						$\frac{3680}{513}$ $-\frac{845}{4104}$ 0						
$\frac{1}{2}$	$-\frac{8}{27}$ 2						$-\frac{3544}{2565}$ $\frac{1859}{4104}$ $-\frac{11}{40}$ 0						
							0.2099 0.3269 -0.4684						$D_i$
							0.0566 0 0						
$b_i$	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$	0	0.1157	0	0.3431	0.0412	0	0	$B_i$
$\rightarrow$													$\leftarrow$
$d_i$	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$	0.1171	0	0.3393	0.0535	-0.01	0	$D_i$
$\rightarrow$													$\leftarrow$

For the second order equation of Eq. (4.8.13), with stage values  $k_i$  of Eq. (4.8.27), the first solution is evaluated in Eq. (4.8.28) and the second by its analog,

$$\begin{aligned} \ddot{\mathbf{v}}_{n+1} &= \mathbf{v}_n + h\dot{\mathbf{v}}_n + h^2 \sum_{j=1}^s D_j \mathbf{k}_j \\ \ddot{\mathbf{v}}_{n+1} &= \dot{\mathbf{v}}_n + h \sum_{j=1}^s d_j \mathbf{k}_j \end{aligned} \quad (4.8.35)$$

Since  $\mathbf{y} = [\mathbf{u}^T \quad \mathbf{w}^T]^T = [\mathbf{v}^T \quad \dot{\mathbf{v}}^T]^T$ ,

$$\begin{aligned} \text{scv}_i &= \text{Atol} + |v_{ni}| \text{Rtol} \\ \text{sc}\dot{v}_i &= \text{Atol} + |\dot{v}_{ni}| \text{Rtol} \end{aligned} \quad (4.8.36)$$

and Eq. (4.8.7) is

$$\text{err} = \sqrt{\frac{1}{2m} \left[ \left( \sum_{i=1}^m \left( \frac{v_i - \ddot{v}_i}{\text{scv}_i} \right)^2 + \sum_{i=1}^m \left( \frac{\dot{v}_i - \ddot{\dot{v}}_i}{\text{sc}\dot{v}_i} \right)^2 \right) \right]} \quad (4.8.37)$$

*Step size control* thus follows Eq. (4.8.9).

#### 4.8.2.2 Implicit Runge-Kutta Methods for Second Order ODE

Since one or more of the Runge-Kutta coefficients  $a_{ii}$  is nonzero in an implicit method, Eqs. (4.8.27) must be solved numerically, usually with an iterative Newton-Raphson method.

The *implicit trapezoidal* method can be viewed as a Runge-Kutta method (Ascher and Petzold, 1998). To take advantage of its form in terms of generalized coordinates, rather than stage variables, it is presented as an extension of Eq. (4.8.10). Writing Eq. (4.8.10) for both  $v$  and  $\dot{v}$ ,

$$\begin{aligned} v_n &= v_{n-1} + (h/2)(\dot{v}_{n-1} + \dot{v}_n) \\ \dot{v}_n &= \dot{v}_{n-1} + (h/2)(\ddot{v}_{n-1} + \ddot{v}_n) \end{aligned} \quad (4.8.38)$$

Substituting the second equation into the first yields a second order formula,

$$\begin{aligned} v_n &= v_{n-1} + (h/2)(\dot{v}_{n-1} + \dot{v}_n + (h/2)(\ddot{v}_{n-1} + \ddot{v}_n)) \\ &= v_{n-1} + h\dot{v}_{n-1} + (h^2/4)(\ddot{v}_{n-1} + \ddot{v}_n) \end{aligned} \quad (4.8.39)$$

With the second of Eqs. (4.8.38), this yields the *second order trapezoidal formula*

$$\begin{aligned} v_n &= v_{n-1} + h\dot{v}_{n-1} + (h^2/4)(\ddot{v}_{n-1} + \ddot{v}_n) \\ \dot{v}_n &= \dot{v}_{n-1} + (h/2)(\ddot{v}_{n-1} + \ddot{v}_n) \end{aligned} \quad (4.8.40)$$

The trapezoidal formula is an order two, A-stable method that is often used and is analyzed in detail by Atkinson (1989). The important thing to point out about it is that, although the formula is stiffly accurate, this does not result in L-stability, hence it is expected to have difficulties with stiff systems.

Numerical integration can be implemented with a *singly diagonal implicit Runge-Kutta* (SDIRK) formula, in which all diagonal entries have the same nonzero value,  $\alpha \neq 0$ . For SDIRK formulas, the Butcher tableau assumes the form of Table 4.8.5.

For *first order SDIRK formulas*,  $a_{ii} = \alpha \neq 0$ ,  $i = 1, \dots, s$ . The matrix  $\mathbf{a} = [a_{ij}]$  given in Table 4.8.5 is the coefficient matrix of the formula. For SDIRK methods  $\mathbf{a}$  is nonsingular. Note that the diagonal terms in the *second order tableau* of Eq. (4.8.21),  $\mathbf{A} = [A_{ij}] = \mathbf{aa}^\top$ , are

$A_{ii} = \sum_{j=1}^s a_{ij}a_{ji} = (a_{ii})^2 = \alpha^2$ , since if  $i \neq j$ , one of the terms in the product forming the sum is zero, leaving only the product with  $j = i$ . Thus, the second order method is also singly diagonal.

In residual form, Eq. (4.8.30) is

$$\mathbf{R} = \mathbf{k}_i - \mathbf{g}(t_n + c_i h, \mathbf{v}'_i + h^2 \alpha^2 \mathbf{k}_i, \dot{\mathbf{v}}'_i + h \alpha \mathbf{k}_i) = \mathbf{0} \quad (4.8.41)$$

The Jacobian in iterative solution for  $\mathbf{k}_i$  is thus

$$\mathbf{R}_{\mathbf{k}_i} \equiv \mathbf{J} = \mathbf{I} - h^2 \alpha^2 \mathbf{g}_v - h \alpha \mathbf{g}_v \quad (4.8.42)$$

whose form does not depend on  $i$ . Since the arguments of the function  $\mathbf{g}$  vary little over the stages,  $i = 1, \dots, s$ , and even over several time steps, the Jacobian of Eq. (4.8.42) can be held fixed over all stages, and possibly over several time steps, until excessive iterations are encountered in solving Eq. (4.8.41), using a method such as Newton-Raphson iteration. Once Eqs. (4.8.41) are solved for  $\mathbf{k}_i$ ,  $i = 1, \dots, s$ , the approximate solution at  $t_{n+1} = t_n + h$  is given by Eqs. (4.8.28).

As in the case of first order ODE, the SDIRK formula sought for integrating potentially *stiff differential equations* should be L-stable, and of average order. The 5 stage, order 4 *SDIRK54 formula* for first order ODE defined in Table 4.8.6 has this property. From Eqs. (4.8.21) and (4.8.26), the coefficients for the second order ODE in Tableau 4.8.7, to five place accuracy, are

$$\mathbf{A} = \begin{bmatrix} 0.0625 & 0 & 0 & 0 & 0 \\ 0.25 & 0.0625 & 0 & 0 & 0 \\ 0.15 & -0.02 & 0.0625 & 0 & 0 \\ 0.12058 & -0.02628 & 0.01378 & 0.0625 & 0 \\ 0.73437 & -0.46614 & 3.71093 & -3.54166 & 0.0625 \end{bmatrix}$$

$$\mathbf{B} = [0.73437 \quad -0.46614 \quad 3.71093 \quad -3.54166 \quad 0.0625] \quad (4.8.43)$$

$$\mathbf{D} = [0.67708, 0.03125, 1.5625, -1.77083, 0]$$

where  $\mathbf{D}$  contains coefficients for error estimation from Eq. (4.8.34). In applications, Eqs. (4.8.21), (4.8.26), and (4.8.34) are used, with precise values of Table 4.8.6, to evaluate  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{D}$  for use in implementing numerical integration.

### 4.8.3 Generalized Form of Second Order ODE

As seen in examples presented in Section 4.7, second order ODE arising in system dynamics are generally of the form

$$\mathbf{M}(\mathbf{v})\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (4.8.44)$$

where the coefficient matrix  $\mathbf{M}(\mathbf{v})$  is nonsingular. Multiplying by the inverse of the coefficient matrix yields the standard form

$$\ddot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{v})\mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (4.8.45)$$

Applying the Runge-Kutta discretization of Eq. (4.8.30) yields

$$\mathbf{k}_i = \mathbf{M}^{-1}(\mathbf{v}'_i + h^2 A_{ii}\mathbf{k}_i)\mathbf{g}(t_n + c_i h, \mathbf{v}'_i + h^2 A_{ii}\mathbf{k}_i, \dot{\mathbf{v}}'_i + h a_{ii}\mathbf{k}_i) \quad (4.8.46)$$

Multiplying by  $\mathbf{M}(\mathbf{v}'_i + h^2 A_{ii}\mathbf{k}_i)$  and writing the result in residual form yields

$$\bar{\mathbf{R}} \equiv \mathbf{M}(\mathbf{v}'_i + h^2 A_{ii}\mathbf{k}_i)\mathbf{k}_i - \mathbf{g}(t_n + c_i h, \mathbf{v}'_i + h^2 A_{ii}\mathbf{k}_i, \dot{\mathbf{v}}'_i + h a_{ii}\mathbf{k}_i) = \mathbf{0} \quad (4.8.47)$$

The Jacobian in iterative solution for  $\mathbf{k}_i$  is

$$\bar{\mathbf{R}}_{\mathbf{k}_i} \equiv \mathbf{M} + h^2 A_{ii} \left( (\mathbf{M}\mathbf{k}_i)'_v - \mathbf{g}_v \right) - h a_{ii} \mathbf{g}_v \quad (4.8.48)$$

where  $A_{ii} = \alpha^2$  and  $a_{ii} = \alpha$  in the case of SDIRK methods.

Similarly, for the trapezoidal formula of Eq. (4.8.38), the residual equation is

$$\begin{aligned} \bar{\mathbf{R}} \equiv & \mathbf{M} \left( \mathbf{v}_{n-1} + h \dot{\mathbf{v}}_{n-1} + (h^2 / 4) (\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \right) \ddot{\mathbf{v}}_n \\ & - \mathbf{g} \left( t_n, \mathbf{v}_{n-1} + h \dot{\mathbf{v}}_{n-1} + (h^2 / 4) (\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n), \dot{\mathbf{v}}_{n-1} + (h / 2) (\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \right) = \mathbf{0} \end{aligned} \quad (4.8.49)$$

and the Jacobian in iterative solution for  $\ddot{\mathbf{v}}_n$  is

$$\bar{\mathbf{R}}_{\ddot{\mathbf{v}}_n} = \mathbf{M} + (h^2 / 4) \left( (\mathbf{M}\ddot{\mathbf{v}}_n)'_v - \mathbf{g}_v \right) - (h / 2) \mathbf{g}_{\dot{\mathbf{v}}} \quad (4.8.50)$$

To implement both Eqs. (4.8.48) and (4.8.50), the term

$$\mathbf{M}2(\mathbf{v}, \boldsymbol{\mu}) \equiv (\mathbf{M}(\mathbf{v})\boldsymbol{\mu})'_v \quad (4.8.51)$$

is coded and used with  $\boldsymbol{\mu} = \mathbf{k}_i$  in Eq. (4.8.48) and  $\boldsymbol{\mu} = \ddot{\mathbf{v}}_n$  in Eq. (4.8.50)

#### 4.8.4 Second Order Trapezoidal Method

The *second order trapezoidal method* of Eq. (4.8.40)

$$\begin{aligned} \mathbf{v}_n &= \mathbf{v}_{n-1} + h \dot{\mathbf{v}}_{n-1} + (h^2 / 4) (\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \\ \dot{\mathbf{v}}_n &= \dot{\mathbf{v}}_{n-1} + (h / 2) (\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \end{aligned} \quad (4.8.52)$$

has properties that differ from most Runge-Kutta methods and deserves treatment in its own right. The method and well-known variants such as the *Newmark method* (Newmark, 1959) and the *HHT- method* (Hilber, Hughes, and Taylor, 1977; Chung and Hulbert, 1993) that were developed for use in structural dynamics are well suited for integration of the equations of multibody dynamics (Negrut, Rampalli, Otterson, and Sajdak, 2007; Negrut, Jay, and Khude, 2009). The trapezoidal method is treated in this section, and variants are treated in Chapter 7 for integrating the *differential-algebraic equation* formulation of the equations of multibody dynamics.

The *trapezoidal method* of Eq. (4.8.52) has attractive properties for use in multibody dynamic simulation. It is simple in form, with variables of integration that are generalized coordinates, as opposed to stage variables in Runge-Kutta methods that are not generalized

coordinates. As a result, it can be substituted directly into equations of motion to provide discretized equations at each time step.

Disadvantages of the method include the fact it is only *second order accurate*, as opposed to the higher order of accuracy of most Runge-Kutta methods. Perhaps more concerning is the fact that the method lacks the degree of stability to be effective in treating stiff behavior that may be encountered in multibody dynamics. Finally, there is no imbedded integration formula that can be used for error and step size control.

#### 4.8.5 Third Derivative for Variable Step Size Error Control

In dealing with the *generalized form of second order ODE* of Eq. (4.8.44),

$$\mathbf{M}(\mathbf{v})\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (4.8.53)$$

an approximation for the third derivative of the solution  $\ddot{\mathbf{v}}_n$  at time  $t_n$  may be used to create an error estimate for the solution  $\mathbf{v}_{n+1}$  and  $\dot{\mathbf{v}}_{n+1}$ . Using Taylor's formula (Corwin and Szczerba, 1982),

$$\begin{aligned}\bar{\mathbf{v}}_{n+1} &= \mathbf{v}_n + h\dot{\mathbf{v}}_n + (h^2 / 2)\ddot{\mathbf{v}}_n + (h^3 / 6)\dddot{\mathbf{v}}_n + O(h^4) \\ \bar{\dot{\mathbf{v}}}_{n+1} &= \dot{\mathbf{v}}_n + h\ddot{\mathbf{v}}_n + (h^2 / 2)\dddot{\mathbf{v}}_n + O(h^3)\end{aligned} \quad (4.8.54)$$

*Time step control* of Eqs. (4.8.37) and (4.8.9) may be used with these estimates; i.e.,

$$\begin{aligned}\text{err} &= \sqrt{\frac{1}{2nv} \left[ \left( \sum_{i=1}^{nv} \left( \frac{\mathbf{v}_i - \bar{\mathbf{v}}_i}{scv_i} \right)^2 + \sum_{i=1}^{nv} \left( \frac{\dot{\mathbf{v}}_i - \bar{\dot{\mathbf{v}}}_i}{sc\dot{v}_i} \right)^2 \right) \right]} \\ h_{\text{opt}} &= h \times (1/\text{err})^{1/(3)}\end{aligned} \quad (4.8.55)$$

where  $scv_i = Atol + |\mathbf{v}_{ni}|Rtol$  and  $sc\dot{v}_i = Atol + |\dot{\mathbf{v}}_{ni}|Rtol$ .

A simple approximation of  $\ddot{\mathbf{v}}_n$  is the difference formula

$$\ddot{\mathbf{v}}_n = (1/h)(\dot{\mathbf{v}}_n - \dot{\mathbf{v}}_{n-1}) \quad (4.8.56)$$

While not necessarily accurate, this is an inexpensive way to implement error and step size control. A better estimate may be had by differentiating Eq. (4.8.53) with respect to time to obtain

$$\begin{aligned}\mathbf{M}(\mathbf{v}_n)\ddot{\mathbf{v}}_n &= \left( -(\mathbf{M}(\mathbf{v}_n)\ddot{\mathbf{v}}_n)_{v_n} + \mathbf{g}(t_n, \mathbf{v}_n, \dot{\mathbf{v}}_n)_{v_n} \right) \dot{\mathbf{v}}_n \\ &\quad + \mathbf{g}(t_n, \dot{\mathbf{v}}_n, \ddot{\mathbf{v}}_n)_{\dot{v}_n} \ddot{\mathbf{v}}_n + \mathbf{g}(t_n, \ddot{\mathbf{v}}_n, \ddot{\mathbf{v}}_n)_{t_n}\end{aligned} \quad (4.8.57)$$

The third derivative  $\ddot{\mathbf{v}}_n$  may thus be determined using one equation solution. While this may appear to be a costly calculation, it may not be. Since the trapezoidal method is implicit, all terms in Eq. (4.8.57) must be evaluated for iterative solution and the cost of factoring  $\mathbf{M}(\mathbf{v}_n)$  is less than the cost of factoring the Jacobian and one evaluation of the residual in iterative solution of the discretized equations of motion. Further, the estimate

$$\ddot{\mathbf{v}}_{n+1}^e = \ddot{\mathbf{v}}_n + h\ddot{\mathbf{v}}_n \quad (4.8.58)$$

of the solution of the discretized equations may be accurate enough to avoid an occasional iteration and the improved step size control associated with an accurate  $\ddot{\mathbf{v}}_n$  should favorably impact compute time. It should be noted that to implement the third derivative computation, the term  $\mathbf{g}_t(t, \mathbf{v}, \dot{\mathbf{v}})$ , which is not required in any of the foregoing integrators, must be evaluated.

Explicit Runge-Kutta numerical integration methods such as the Nystrom4 method can take advantage of this error control approach. Since it requires four function evaluations in each time step, the cost of the calculation of Eq. (4.8.57) may be acceptable, provided large steps are achieved with error control.

Methods for numerical integration of ODE, primarily initial-value problems with first order ODE, have been well-known for over a century. Especially for one step Runge-Kutta methods, a deep theory of stability, error control, and accuracy has evolved. Numerical integration formulas for first order ODE, both explicit that involve no iterative solution and implicit that involve iterative solution, are summarized for six methods, two of which include error control.

A transformation of integration methods for first order ODE is presented that creates algorithms for integration of second order ODE. A MATLAB implementation of six second order methods is presented in Code 4.8 of Appendix 4.A. Components of Code 4.8 with which the user interacts are presented in Section 4.9.1. Use of Code 4.8 is illustrated for solution of multibody system examples in Section 4.9.2.

## Key Formulas

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t) \quad \mathbf{y}(t^0) = \mathbf{y}^0 \quad (4.8.1)$$

$$\mathbf{K}_i = \mathbf{f}\left(t_n + c_i h, \mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{K}_j\right), \quad i=1, \dots, s \quad \mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{K}_i \quad (4.8.2) \quad (4.8.3)$$

$$\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad \mathbf{v}(t^0) = \mathbf{v}^0 \quad \dot{\mathbf{v}}(t^0) = \dot{\mathbf{v}}^0 \quad (4.8.13)$$

$$\mathbf{v}_n = \mathbf{v}_{n-1} + h\dot{\mathbf{v}}_{n-1} + (h^2 / 4)(\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \quad \dot{\mathbf{v}}_n = \dot{\mathbf{v}}_{n-1} + (h / 2)(\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \quad (4.8.40)$$

$$\mathbf{k}_i = \mathbf{g}\left(t_n + c_i h, \mathbf{v}_n + h c_i \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^i A_{ij} \mathbf{k}_j, \dot{\mathbf{v}}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}_j\right) \quad (4.8.27)$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + h\dot{\mathbf{v}}_n + h^2 \sum_{j=1}^s B_j \mathbf{k}_j \quad \dot{\mathbf{v}}_{n+1} = \dot{\mathbf{v}}_n + h \sum_{j=1}^s b_j \mathbf{k}_j \quad (4.8.28)$$

$$A_{il} = \sum_{j=1}^s a_{ij} a_{jl} \quad B_\ell = \sum_{j=1}^s b_j a_{jl} \quad (4.8.21) \quad (4.8.26)$$

## 4.9 Numerical Solution for Systems Governed by ODE

A brief user-oriented outline of Code 4.8 of Appendix 4.A is presented in Section 4.A.1 of Appendix 4.A. Following a brief introduction in Section 4.9.1, three examples of solutions obtained using tailored forms of Code 4.8 are presented in Sections 4.9.2.

### 4.9.1 Code 4.8 for Second Order ODE Solution

MATLAB computer Code 4.8 of Appendix 4.A is comprised of six numerical integration functions for solving the *second order initial-value problem*

$$\mathbf{M}(\mathbf{v})\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (4.9.1)$$

$$\begin{aligned} \mathbf{v}(0) &= \mathbf{v}^0 \\ \dot{\mathbf{v}}(0) &= \dot{\mathbf{v}}^0 \end{aligned} \quad (4.9.2)$$

Explicit solvers implemented are *Nystrom4*, *RungeKutta4*, *Kutta38*, and *RKFN45* of Section 4.8.2.1. Implicit solvers implemented are *Trapezoidal* and *SDIRK54* of Section 4.8.2.2. Code 4.8 is intended to be a template of numerical solution methods that will be adapted for numerical solution of numerous forms of second order ODE of mechanical system dynamics, rather than a stand-alone code.

### 4.9.2 Numerical Examples of Systems Governed by ODE

To illustrate numerical integration methods presented in Section 4.8, using Code 4.8 of Appendix 4.A that is documented in Section 4.A.1 of Appendix 4.A, examples of Section 4.7 are studied. In each example, quantities that are required for implementation of numerical integration methods are derived and implemented in a version of Code 4.8, identified as Codes 4.9.1 through 4.9.3 in Appendix 4.A.

#### 4.9.2.1 Planar Double Pendulum

The equations of motion of Eq. (4.7.7) for the *planar double pendulum* with concentrated masses of Section 4.7.1 are of the form of the second order ODE of Eq. (4.9.1); i.e.,  $\mathbf{M}(\mathbf{v})\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}})$ , with

$$\begin{aligned} \mathbf{M}(\mathbf{v}) &= \begin{bmatrix} (m_1 + m_2) & m_2 \cos(v_2 - v_1) \\ m_2 \cos(v_2 - v_1) & m_2 \end{bmatrix} \\ \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) &= \begin{bmatrix} -(m_1 + m_2)g \cos(v_1) \\ -m_2 g \cos(v_2) \end{bmatrix} - \begin{bmatrix} -m_2 \sin(v_2 - v_1)(\dot{v}_2)^2 \\ m_2 \sin(v_2 - v_1)(\dot{v}_1)^2 \end{bmatrix} \end{aligned} \quad (4.9.3)$$

Data that must be entered in the user component of Code 4.8, to create Code 4.9.1 for this example, are presented in Fig. 4.9.1. To assist in writing code in terms of parameters  $m_1$  and  $m_2$ , the function *AdatPart* is adapted in Code 4.8 for this application. In the present example, the variables are simple enough that the parameter partitioning function would not be necessary, but it is used as an illustration in Fig 4.9.2.

```

1 %4.9.1 Planar Double Pendulum; Based on Code 4.8, Second Order ODE Solver
2 intol=10^-8; %Tolerance in solving discretized equations of motion
3 Atol=10^-8; %Absolute error tolerance for variable step methods
4
5 h=0.0001; %Step size
6 hmax=0.001; %Maximum Allowable Step size
7 hvar=1; %hvar=1, variable step;hvar=2, fixed step
8
9 tfinal=6; %Final time
10
11 %Explicit Integration Methods:
12 %Integ=1, Nystrom4; Integ=2, RungeKutta4;
13 %Integ=4, Kutta3/8; integ=4, RKFN
14 %Implicit Integration Methods:
15 %Integ=5, Trapezoidal; Integ=6, SDIRK54
16 Integ=6; %Numerical Integration Method Selected
17
18 nv=2; %Use variable dimension
19
20 %Fixed Parameter Data List-Complete Partitioning in function parPart if
21 %explicit use of variable names is desired in user supplied functions.
22 par=[nv;intol;Atol;hmax;hvar];
23 % Problem Data
24 m1=10;
25 m2=10;
26 K1=10000000;
27 C1=20000;
28 K2=10^10;
29
30 %Problem Parameter Data List-Partitioning in function AdatPart if
31 %use of variable names is desired in user supplied functions.
32 dat=[m1;m2;K1;C1;K2];
33
34 % Data Storage Arrays
35 V=zeros(nv,10);
36 Vd=zeros(nv,10);
37 Vdd=zeros(nv,10);
38
39 % Initial Conditions
40 v0=[0;0]; %User define initial position
41 vd0=[0;0]; %User define initial velocity
42 V(:,1)=v0;
43 Vd(:,1)=vd0;

```

Figure 4.9.1 Code 4.9.1 Simulation Parameters, Double Pendulum

If only explicit integration is to be used, expressions of Eq. (4.9.3) are entered in functions Agf and AM of Fig. 4.9.5. If implicit integration is to be used, terms in Eqs. (4.8.49), (4.8.51), and (4.8.52) are

$$\begin{aligned}
\mathbf{g}_v &= \begin{bmatrix} (m_1 + m_2)g\sin(v_1) - m_2\cos(v_2 - v_1)(\dot{v}_2)^2 & m_2\cos(v_2 - v_1)(\dot{v}_2)^2 \\ m_2\cos(v_2 - v_1)(\dot{v}_1)^2 & m_2g\sin(v_2) - m_2\cos(v_2 - v_1)(\dot{v}_1)^2 \end{bmatrix} \\
\mathbf{g}_v &= \begin{bmatrix} \mathbf{0} & 2m_2\sin(v_2 - v_1)\dot{v}_2 \\ -2m_2\sin(v_2 - v_1)\dot{v}_1 & \mathbf{0} \end{bmatrix} \\
\mathbf{M2}(v, \mu) &= \sin(v_2 - v_1)m_2 \begin{bmatrix} \mu_2 & -\mu_2 \\ \mu_1 & -\mu_1 \end{bmatrix}
\end{aligned} \tag{4.9.4}$$

These terms are entered in functions Agfsvdd and AM2 of Fig 4.92.

```

1 function [m1,m2]=AdatPart(dat)
2
3 %User define elements in array dat by name, as in BparPart
4
5 m1=dat(1);
6 m2=dat(2);

1 function g=Agf(t,v,vd,par,dat)
2
3 [nv,intol,Atol,hmax,hvar]=BparPart(par);
4 [m1,m2]=AdatPart(dat);
5 v1=v(1);
6 v2=v(2);
7 vd1=vd(1);
8 vd2=vd(2);
9 %Enter right side of ODE, Eq. (4.7.7)
10
11 g=[-(m1+m2)*9.8*cos(v1);-m2*9.8*cos(v2)];...
12 [-m2*sin(v2-v1)*(vd2^2);m2*sin(v2-v1)*(vd1^2)];;

1 function [gsv,gsvd]=Agfsvvd(t,v,vd,par,dat)
2
3 [nv,intol,Atol,hmax,hvar]=BparPart(par);
4 [m1,m2]=AdatPart(dat);
5 v1=v(1);
6 v2=v(2);
7 v1d=vd(1);
8 v2d=vd(2);
9
10 %Enter derivatives of g with respect to v and vd
11 s1=sin(v1);
12 s2=sin(v2);
13 c2m1=cos(v2-v1);
14 s2m1=sin(v2-v1);
15
16 gsv=[(m1+m2)*9.8*s1-m2*c2m1*v2d^2,m2*c2m1*v2d^2;...
17 m2*c2m1*v1d^2,m2*9.8*s2-m2*c2m1*v1d^2];
18 gsvd=[0,2*m2*s2m1*v2d;-2*m2*s2m1*v1d,0];

1 function M=AM(v,par,dat)

```

```

2
3 [nv,intol,Atol,hmax,hvar]=BparPart(par);
4 [m1,m2]=AdatPart(dat);
5 v1=v(1);
6 v2=v(2);
7
8 % Enter Mass Matrix M=M(v,par)
9
10 M=[m1+m2,m2*cos(v2-v1);m2*cos(v2-v1),m2];

1 function M2=AM2(v,mu,par,dat)
2
3 [nv,intol,Atol,hmax,hvar]=BparPart(par);
4 [m1,m2]=AdatPart(dat);
5 v1=v(1);
6 v2=v(2);
7 mu1=mu(1);
8 mu2=mu(2);
9
10 % Enter M2=(M(q,par)mu)sq
11
12 M2=sin(v2-v1)*m2*[mu2,-mu2;mu1,-mu1];

```

Figure 4.9.2 Code 4.9.1 User Input Functions, Double Pendulum

With  $m_1 = m_2 = 1$  kg, plots of  $v_1$  and  $v_2$  in Fig 4.9.3 obtained using a constant step size of  $h = 0.001$  sec with the Nystrom4 explicit integrator indicate a relatively smooth swinging motion. *Total energy* for this *conservative system* was constant to within a maximum variation of  $2e-9$ . Essentially identical results were obtained with all six integrators. Variable step size simulations using the RKNF45 and SDIRK54 integrators with a maximum allowed step size  $h_{max} = 0.01$  sec yielded comparable results, with step sizes selected at the upper limit.

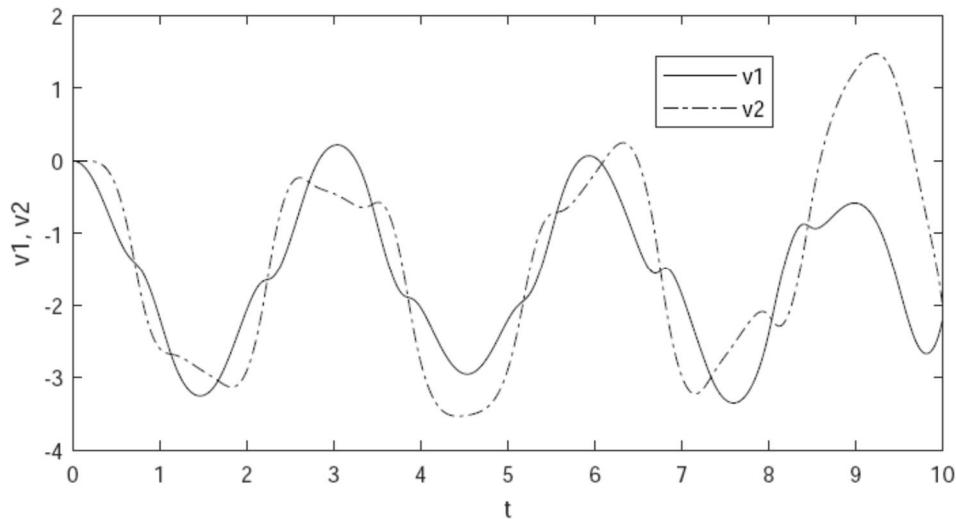


Figure 4.9.3  $v_i$  vs  $t$ , Double Pendulum,  $m_1 = m_2 = 1$  kg,  $h = 0.001$  sec

To test the strength of the numerical integration formulations, mass 1 is held at unity and mass 2 is set to 100 kg. Numerical results presented in Fig. 4.9.4 were obtained using SDIRK54

with a constant step size  $h = 0.0001$ . They show that moderately high frequency oscillation occurs. With constant step size SDIRK54, the variation in total energy was less than  $7e-5$ , whereas with the explicit Nystrom4 integrator the variation was  $4e-4$ . Allowing *variable step size*, with an upper bound  $h_{max} = 0.001$ , SDIRK54 chose step sizes ranging from just below 0.001 to less than 0.0001, with results essentially identical to those of Fig. 4.9.3. However, the variation in total energy grew to  $9e-3$ .

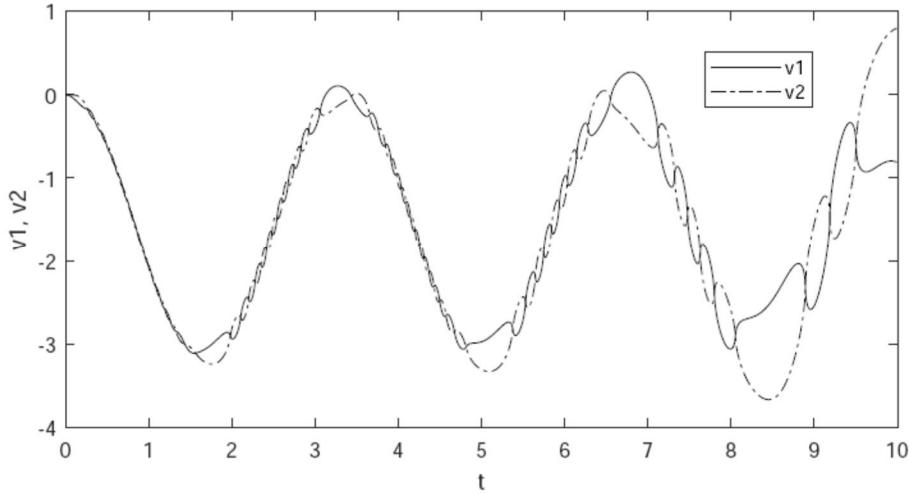


Figure 4.9.4  $v_i$  vs  $t$ , Double Pendulum,  $m_1 = 1$ ,  $m_2 = 100$ ,  $h = 0.001$  sec

#### 4.9.2.2 Planar Slider-Crank

The equations of motion of Eq. (4.7.14) for the *planar slider-crank* of Section 4.7.2, with connecting rod length 2 m and crank radius  $r$  m, are of the form of the ODE of Eq. (4.8.45); i.e.,

$$\mathbf{M}(\mathbf{v})\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (4.9.5)$$

with

$$\begin{aligned} \mathbf{M} &= J_1 + m_2 A^2 \\ \mathbf{g} &= -m_2 AB\dot{\mathbf{v}}^2 \end{aligned} \quad (4.9.6)$$

where

$$\begin{aligned} A &\equiv R\sin(v) + \frac{R^2\sin(v)\cos(v)}{\sqrt{4 - (R\sin(v))^2}} \\ A_v &= B \equiv R\cos(v) + \frac{R^2(\cos(v)^2 - \sin(v)^2)}{\sqrt{4 - (R\sin(v))^2}} + \frac{R^4(\sin(v)\cos(v))^2}{(4 - (R\sin(v))^2)^{3/2}} \end{aligned}$$

If only explicit numerical integration is to be used, expressions of Eq. (4.9.6) are entered in functions Agf and AM in Code 4.9.2 of Appendix 4.A.

If implicit integration is to be used, terms in Eqs. (4.8.49), (4.8.51), and (4.8.52) are

$$\begin{aligned}
\mathbf{g}_v &= -m_2 \mathbf{B}^2 \dot{v}^2 - m_2 \mathbf{A} \mathbf{C} \dot{v}^2 \\
\mathbf{g}_{\dot{v}} &= -2m_2 \mathbf{A} \mathbf{B} \dot{v} \\
M2(v, \mu) &= 2m_2 \mu \mathbf{A} \mathbf{B}
\end{aligned} \tag{4.9.7}$$

where

$$C \equiv -R \sin(v) - \frac{2R^2 \sin(v) \cos(v)}{\sqrt{4 - (R \sin(v))^2}} + \frac{3R^4 (\cos(v)^2 - \sin(v)^2) \sin(v) \cos(v)}{(4 - (R \sin(v))^2)^{3/2}} + \frac{3R^6 (\sin(v) \cos(v))^3}{2(4 - (R \sin(v))^2)^{5/2}}$$

These terms are entered in functions Agfsvdd and AM2 of Code 4.9.2. Rather than including user input files in the text, as was done in Section 4.9.2.1, the reader is referred to Code 4.9.2 in Appendix 4.A for details.

As an initial simulation, parameters are set as  $R=1.5$  m,  $J_1 = 20 \text{ kg} \cdot \text{m}^2$ ,  $m_2 = 20 \text{ kg}$ , and  $g = 9.8 \text{ m/sec}^2$ . Plots of crank angular velocity  $\dot{v} = v_d$  and  $x_2 = x_2$  and its derivatives are presented in Fig. 4.9.5, obtained using the explicit Nystrom4 integrator, with an initial crank angular velocity of 10 rad/sec and step size  $h = 0.001$  sec. Expressions for slider position  $x_2$  and its derivatives, as functions of  $v$  and its derivatives, are given by Eqs. (4.7.9) and (4.7.10). Comparable results were obtained using all integrators and kinetic energy varied only 0.01% over the simulation period for this conservative system.

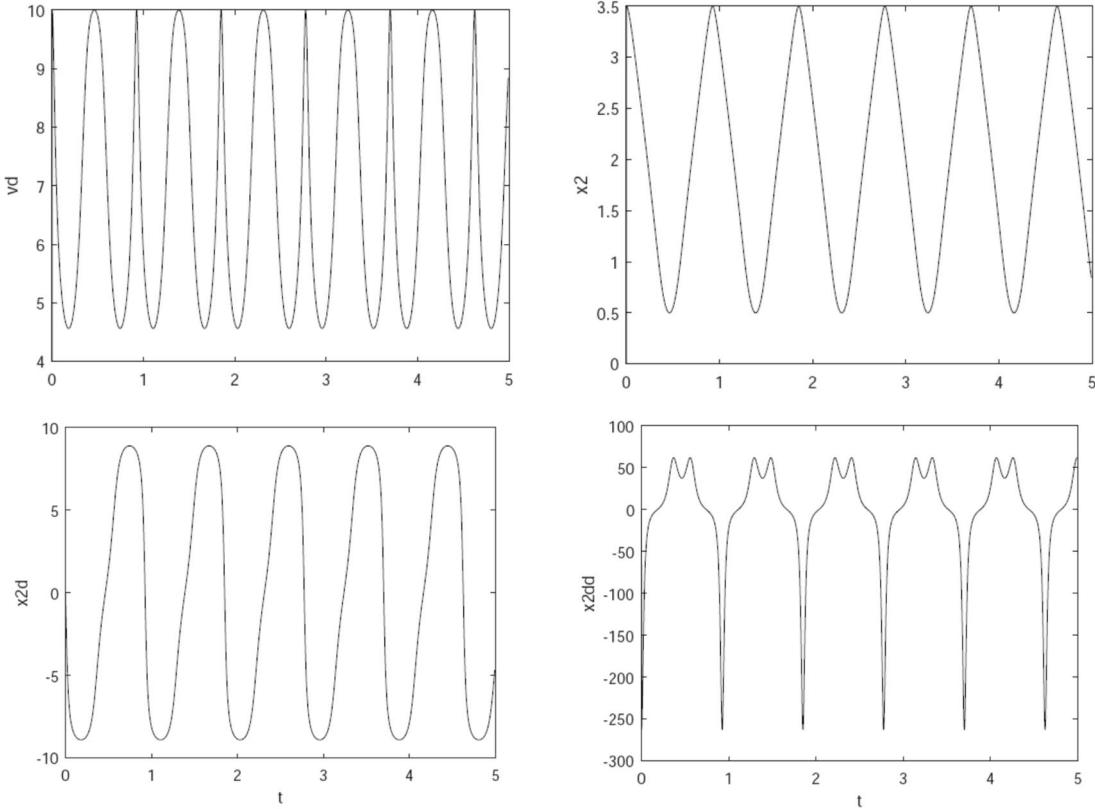


Figure 4.9.5  $\dot{v}$ ,  $x_2$ ,  $\dot{x}_2$ , and  $\ddot{x}_2$  for Slider Crank,  $R=1.5$ ,  $m_2 = 20$ ,  $\omega_0 = 10$ ,  $h = 0.001$

As shown in Example 3.7.1, the slider-crank has a singularity if  $R \geq 2$ . Setting  $R = 1.98$ , retaining the remaining data above, yields simulation results for crank angular velocity and  $\ddot{x}_2$ , using the SDIRK54 integrator with  $h = 0.001$ , shown in Fig. 4.9.6. As noted, substantially larger accelerations, hence larger associated bearing forces, occur.

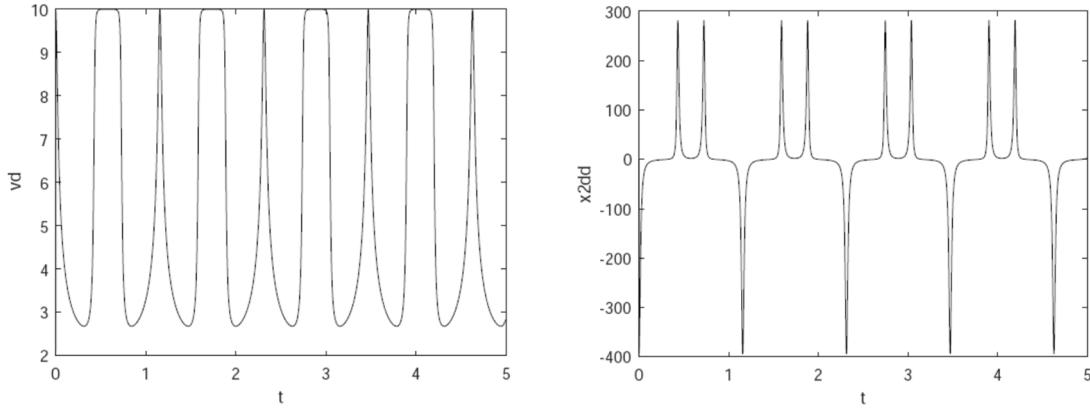


Figure 4.9.6  $\ddot{v}$  and  $\ddot{x}_2$  for Slider Crank,  $R=1.98$ ,  $m_2 = 20$ ,  $\omega_0 = 10$ ,  $h = 0.001$

The observation in Chapter 1 that nonlinearity in machine kinematics and dynamics is both a blessing and a curse is borne out with the slider crank. This simple mechanism makes possible the internal combustion engine that has been at the heart of automotive and manufacturing technology for well over a century. Its nonlinear kinematic behavior makes possible its monumental contributions. Its nonlinear dynamic behavior, including large accelerations and associated loads, makes dynamic analysis and *design of bearings* for long life a challenge. As with many aspects of engineering, one must overcome such challenges to take advantage of benefits offered by nonlinear mechanical systems

#### 4.9.2.3 Spatial Robotic Manipulator

The equations of motion of Eqs. (4.7.29) and (4.7.30) for the *spatial robotic manipulator* of Section 4.7.3 are of the form of the second order ODE of Eq. (4.8.45); i.e.,

$$\mathbf{M}(\mathbf{v})\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (4.9.8)$$

with  $\mathbf{v} \in \mathbb{R}^3$  and

$$\begin{aligned} \mathbf{M}(\mathbf{v}) &= \mathbf{W}_1^T \mathbf{J}'_1 \mathbf{W}_1 + m_2 \mathbf{V}_2^T \mathbf{V}_2 + 2 \mathbf{W}_2^T \mathbf{J}'_2 \mathbf{W}_2 + m_3 \mathbf{V}_3^T \mathbf{V}_3 \\ \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}}) &= - \left( \begin{array}{l} \mathbf{W}_1^T (\widetilde{\mathbf{W}_1 \dot{\mathbf{v}}}) \mathbf{J}'_1 \mathbf{W}_1 \dot{\mathbf{v}} + m_2 \mathbf{V}_2^T (\mathbf{U}_2 + g \mathbf{u}_z) + 2 \mathbf{W}_2^T \mathbf{J}'_2 \mathbf{X}_2 \\ + 2 \mathbf{W}_2^T (\widetilde{\mathbf{W}_2 \dot{\mathbf{v}}}) \mathbf{J}'_2 \mathbf{W}_2 \dot{\mathbf{v}} + m_3 \mathbf{V}_3^T (\mathbf{U}_3 + g \mathbf{u}_z) \end{array} \right) \end{aligned} \quad (4.9.9)$$

If only explicit integration is to be used, expressions of Eq. (4.9.9) are entered in functions Agf and AM in Code 4.8 of Appendix 4.A and one of the explicit integrators is invoked to obtain a numerical solution.

If implicit integration is to be used, terms in Eqs. (4.8.49), (4.8.51), and (4.8.52) must be calculated. Since  $\mathbf{U}_i(\mathbf{v}, \dot{\mathbf{v}})$  and  $\mathbf{X}_i(\mathbf{v}, \dot{\mathbf{v}})$  are column vectors, derivatives  $\mathbf{U}_{iv}$ ,  $\mathbf{U}_{iiv}$ ,  $\mathbf{X}_{iv}$ , and  $\mathbf{X}_{iiv}$  can

be evaluated. Since  $\mathbf{V}_i(\mathbf{v})$  and  $\mathbf{W}_i(\mathbf{v})$  are matrices, derivatives  $(\mathbf{V}_i \ddot{\mathbf{a}})_v$ ,  $(\mathbf{W}_i \ddot{\mathbf{p}})_v$ ,  $(\mathbf{V}_i^T \ddot{\mathbf{g}})_v$ , and  $(\mathbf{W}_i^T \ddot{\mathbf{d}})_v$  are to be evaluated.

To evaluate  $(\mathbf{M}(\mathbf{v}) \ddot{\mathbf{u}})_v$ , the fact that each of the four terms in the first of Eqs. (4.9.9) is symmetric leads to

$$\begin{aligned} (\mathbf{V}_i^T \mathbf{V}_i \ddot{\mathbf{u}})_v &= (\mathbf{V}_i^T \mathbf{V}_i \ddot{\mathbf{u}})_v + (\mathbf{V}_i^T \mathbf{V}_i \ddot{\mathbf{u}})_v = \mathbf{V}_i^T (\mathbf{V}_i \ddot{\mathbf{u}})_v + (\mathbf{V}_i^T \mathbf{V}_i \ddot{\mathbf{u}})_v = 2 \mathbf{V}_i^T (\mathbf{V}_i \ddot{\mathbf{u}})_v \\ (\mathbf{W}_i^T \mathbf{J}'_i \mathbf{W}_i \ddot{\mathbf{u}})_v &= (\mathbf{W}_i^T \mathbf{J}'_i \mathbf{W}_i \ddot{\mathbf{u}})_v + (\mathbf{W}_i^T \mathbf{J}'_i \mathbf{W}_i \ddot{\mathbf{u}})_v = \mathbf{W}_i^T \mathbf{J}'_i (\mathbf{W}_i \ddot{\mathbf{u}})_v + (\mathbf{W}_i^T \mathbf{J}'_i \mathbf{W}_i \ddot{\mathbf{u}})_v \\ &= 2 \mathbf{W}_i^T \mathbf{J}'_i (\mathbf{W}_i \ddot{\mathbf{u}})_v \end{aligned}$$

Since  $(\mathbf{W}_i \ddot{\mathbf{u}})_v = \mathbf{0}$ ,

$$\mathbf{M}2(\mathbf{v}, \ddot{\mathbf{u}}) = 2m_2 \mathbf{V}_2^T (\mathbf{V}_2 \ddot{\mathbf{u}})_v + 2 \mathbf{W}_2^T \mathbf{J}'_2 (\mathbf{W}_2 \ddot{\mathbf{u}})_v + 2m_3 \mathbf{V}_3^T (\mathbf{V}_3 \ddot{\mathbf{u}})_v \quad (4.9.10)$$

The terms  $((\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}})_v$  in  $\mathbf{g}(\mathbf{v}, \dot{\mathbf{v}})$  require a bit of derivation, as follows:

$$((\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}})_v = (\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i (\mathbf{W}_i \ddot{\mathbf{v}})_v + ((\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}})_v \quad (4.9.11)$$

Using the identity  $\tilde{\mathbf{a}}\mathbf{b} = -\tilde{\mathbf{b}}\mathbf{a}$  of Eq. (2.1.27),

$$((\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}})_v = -((\widetilde{\mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{W}_i \dot{\mathbf{v}})_v = -(\widetilde{\mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}}})(\mathbf{W}_i \dot{\mathbf{v}})_v$$

Thus, Eq. (4.9.11) becomes

$$\begin{aligned} ((\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}})_v &= (\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i (\mathbf{W}_i \ddot{\mathbf{v}})_v - ((\widetilde{\mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}}}) (\mathbf{W}_i \ddot{\mathbf{v}})_v \\ &= ((\widetilde{\mathbf{W}_i \dot{\mathbf{v}}}) \mathbf{J}'_i - ((\widetilde{\mathbf{J}'_i \mathbf{W}_i \dot{\mathbf{v}}})))(\mathbf{W}_i \ddot{\mathbf{v}})_v \end{aligned} \quad (4.9.12)$$

Since  $(\mathbf{W}_i^T \ddot{\mathbf{a}})_v = \mathbf{0}$ ,

$$\mathbf{g}_v = \left\{ \begin{array}{l} m_2 \left( \mathbf{V}_2^T (\mathbf{U}_2 + g \mathbf{u}_z) \right)_v + m_2 \mathbf{V}_2^T \mathbf{U}_{2v} + (\mathbf{W}_2^T \mathbf{J}'_2 \ddot{\mathbf{X}}_2)_v + \mathbf{W}_2^T \mathbf{J}'_2 \mathbf{X}_{2v} \\ + 2 \left( \mathbf{W}_2^T \left( (\widetilde{\mathbf{W}_2 \dot{\mathbf{v}}}) \mathbf{J}'_2 \mathbf{W}_2 \dot{\mathbf{v}} \right) \right)_v + 2 \mathbf{W}_2^T \left( ((\widetilde{\mathbf{W}_2 \dot{\mathbf{v}}}) \mathbf{J}'_2 - ((\widetilde{\mathbf{J}'_2 \mathbf{W}_2 \dot{\mathbf{v}}})) \right) (\mathbf{W}_2 \ddot{\mathbf{v}})_v \\ + m_3 \left( \mathbf{V}_3^T (\mathbf{U}_3 + g \mathbf{u}_z) \right)_v + m_3 \mathbf{V}_3^T \mathbf{U}_{3v} \end{array} \right\} \quad (4.9.13)$$

For dependence of  $\mathbf{g}(\mathbf{v}, \dot{\mathbf{v}})$  on  $\dot{\mathbf{v}}$ , using Eq. (2.1.26),

$$\begin{aligned}
((\widetilde{\mathbf{W}_1}\dot{\mathbf{v}})\mathbf{J}'\mathbf{W}_1\dot{\mathbf{v}})_{\dot{\mathbf{v}}} &= \left( (\widetilde{\mathbf{W}_1}\ddot{\mathbf{v}})\mathbf{J}'\mathbf{W}_1\dot{\mathbf{v}} \right)_{\dot{\mathbf{v}}} + \left( (\widetilde{\mathbf{W}_1}\dot{\mathbf{v}})\mathbf{J}'\mathbf{W}_1\ddot{\mathbf{v}} \right)_{\dot{\mathbf{v}}} \\
&= (\widetilde{\mathbf{W}_1}\dot{\mathbf{v}})\mathbf{J}'\mathbf{W}_1 - \left( (\mathbf{J}'\widetilde{\mathbf{W}_1}\ddot{\mathbf{v}})\mathbf{W}_1\dot{\mathbf{v}} \right)_{\dot{\mathbf{v}}} \\
&= ((\widetilde{\mathbf{W}_1}\dot{\mathbf{v}})\mathbf{J}' - (\mathbf{J}'\widetilde{\mathbf{W}_1}\ddot{\mathbf{v}}))\mathbf{W}_1
\end{aligned} \tag{4.9.14}$$

Thus

$$\mathbf{g}_{\dot{\mathbf{v}}} = - \begin{cases} \mathbf{W}_1^T \left( (\widetilde{\mathbf{W}_1}\dot{\mathbf{v}})\mathbf{J}' - (\mathbf{J}'\widetilde{\mathbf{W}_1}\ddot{\mathbf{v}}) \right) \mathbf{W}_1 + m_2 \mathbf{V}_2^T \mathbf{U}_{2\dot{\mathbf{v}}} + 2 \mathbf{W}_2^T \mathbf{J}'_2 \mathbf{X}_{2\dot{\mathbf{v}}} \\ + 2 \mathbf{W}_2^T \left( (\widetilde{\mathbf{W}_2}\dot{\mathbf{v}})\mathbf{J}'_2 - (\mathbf{J}'_2\widetilde{\mathbf{W}_2}\ddot{\mathbf{v}}) \right) \mathbf{W}_2 + m_3 \mathbf{V}_3^T \mathbf{U}_{3\dot{\mathbf{v}}} \end{cases} \tag{4.9.15}$$

For this example, from Eqs. (4.7.21) and (4.7.25),

$$\begin{aligned}
\mathbf{U}_{2\dot{\mathbf{v}}} &= [\mathbf{a} \quad \mathbf{b} \quad \mathbf{0}] \\
\mathbf{a} &= \dot{v}_1^2 \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_2 \mathbf{A}_1'' \mathbf{A}'_{12} \mathbf{u}_{y_2''} + \dot{v}_2^2 \mathbf{A}_1' \mathbf{A}_{12}'' \mathbf{u}_{y_2''} \\
\mathbf{b} &= \dot{v}_1^2 \mathbf{A}_1'' \mathbf{A}'_{12} \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_2 \mathbf{A}_1' \mathbf{A}_{12}'' \mathbf{u}_{y_2''} + \dot{v}_2^2 \mathbf{A}_1 \mathbf{A}_{12}''' \mathbf{u}_{y_2''}
\end{aligned}$$

where  $\mathbf{A}_1''' = -\mathbf{A}_1'$  and  $\mathbf{A}_{12}''' = -\mathbf{A}_{12}'$ .

$$\mathbf{U}_{2\dot{\mathbf{v}}} = [2\dot{v}_1 \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_2 \mathbf{A}_1' \mathbf{A}'_{12} \mathbf{u}_{y_2''} \quad 2\dot{v}_1 \mathbf{A}_1' \mathbf{A}_{12}' \mathbf{u}_{y_2''} + 2\dot{v}_2 \mathbf{A}_1 \mathbf{A}_{12}'' \mathbf{u}_{y_2''} \quad \mathbf{0}]$$

$$\begin{aligned}
\mathbf{U}_{3\dot{\mathbf{v}}} &= [\mathbf{c} \quad \mathbf{d} \quad \mathbf{e}] \\
\mathbf{c} &= \dot{v}_1^2 (v_3 + 2) \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_2 (v_3 + 2) \mathbf{A}_1'' \mathbf{A}'_{12} \mathbf{u}_{y_2''} \\
&\quad + \dot{v}_2^2 (v_3 + 2) \mathbf{A}_1' \mathbf{A}_{12}'' \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_3 \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_2 \dot{v}_3 \mathbf{A}_1' \mathbf{A}_{12}' \mathbf{u}_{y_2''} \\
\mathbf{d} &= \dot{v}_1^2 (v_3 + 2) \mathbf{A}_1'' \mathbf{A}'_{12} \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_2 (v_3 + 2) \mathbf{A}_1' \mathbf{A}_{12}'' \mathbf{u}_{y_2''} \\
&\quad + \dot{v}_2^2 (v_3 + 2) \mathbf{A}_1 \mathbf{A}_{12}''' \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_3 \mathbf{A}_1' \mathbf{A}_{12}' \mathbf{u}_{y_2''} + 2\dot{v}_2 \dot{v}_3 \mathbf{A}_1 \mathbf{A}_{12}'' \mathbf{u}_{y_2''} \\
\mathbf{e} &= \dot{v}_1^2 \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_1 \dot{v}_2 \mathbf{A}_1' \mathbf{A}'_{12} \mathbf{u}_{y_2''} + \dot{v}_2^2 \mathbf{A}_1 \mathbf{A}_{12}'' \mathbf{u}_{y_2''}
\end{aligned}$$

$$\begin{aligned}
\mathbf{U}_{3\dot{\mathbf{v}}} &= [\mathbf{f} \quad \mathbf{g} \quad \mathbf{h}] \\
\mathbf{f} &= 2\dot{v}_1 (v_3 + 2) \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_2 (v_3 + 2) \mathbf{A}_1' \mathbf{A}_{12}' \mathbf{u}_{y_2''} + 2\dot{v}_3 \mathbf{A}_1' \mathbf{A}_{12} \mathbf{u}_{y_2''} \\
\mathbf{g} &= 2\dot{v}_1 (v_3 + 2) \mathbf{A}_1' \mathbf{A}'_{12} \mathbf{u}_{y_2''} + 2\dot{v}_2 (v_3 + 2) \mathbf{A}_1 \mathbf{A}_{12}'' \mathbf{u}_{y_2''} + 2\dot{v}_3 \mathbf{A}_1 \mathbf{A}_{12}' \mathbf{u}_{y_2''} \\
\mathbf{h} &= 2\dot{v}_1 \mathbf{A}_1' \mathbf{A}_{12} \mathbf{u}_{y_2''} + 2\dot{v}_2 \mathbf{A}_1 \mathbf{A}_{12}' \mathbf{u}_{y_2''}
\end{aligned}$$

$$(\mathbf{V}_2 \ddot{\mathbf{u}})_v = [\alpha_1 \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + \alpha_2 \mathbf{A}_1' \mathbf{A}'_{12} \mathbf{u}_{y_2''} \quad \alpha_1 \mathbf{A}_1' \mathbf{A}_{12}' \mathbf{u}_{y_2''} + \alpha_2 \mathbf{A}_1 \mathbf{A}_{12}'' \mathbf{u}_{y_2''} \quad \mathbf{0}]$$

$$\begin{aligned}
(\mathbf{V}_3 \ddot{\mathbf{u}})_v &= [\mathbf{i} \quad \mathbf{j} \quad \mathbf{k}] \\
\mathbf{i} &= \alpha_1 (v_3 + 2) \mathbf{A}_1'' \mathbf{A}_{12} \mathbf{u}_{y_2''} + \alpha_2 (v_3 + 2) \mathbf{A}_1' \mathbf{A}'_{12} \mathbf{u}_{y_2''} + \alpha_3 \mathbf{A}_1' \mathbf{A}_{12} \mathbf{u}_{y_2''} \\
\mathbf{j} &= \alpha_1 (v_3 + 2) \mathbf{A}_1' \mathbf{A}'_{12} \mathbf{u}_{y_2''} + \alpha_2 (v_3 + 2) \mathbf{A}_1 \mathbf{A}_{12}'' \mathbf{u}_{y_2''} + \alpha_3 \mathbf{A}_1 \mathbf{A}_{12}' \mathbf{u}_{y_2''} \\
\mathbf{k} &= \alpha_1 \mathbf{A}_1' \mathbf{A}_{12} \mathbf{u}_{y_2''} + \alpha_2 \mathbf{A}_1 \mathbf{A}_{12}' \mathbf{u}_{y_2''}
\end{aligned}$$

$$\begin{aligned} \left(\mathbf{V}_2^T \ddot{\boldsymbol{\beta}}\right)_v &= \begin{bmatrix} \mathbf{u}_{y_2''}^T \mathbf{A}_{12}^T \mathbf{A}_1'^T \ddot{\boldsymbol{\beta}} \\ \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} \\ 0 \end{bmatrix}_v = \begin{bmatrix} \mathbf{u}_{y_2''}^T \mathbf{A}_{12}^T \mathbf{A}_1''^T \ddot{\boldsymbol{\beta}} & \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1'^T \ddot{\boldsymbol{\beta}} & 0 \\ \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1'^T \ddot{\boldsymbol{\beta}} & \mathbf{u}_{y_2''}^T \mathbf{A}_{12}''^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \left(\mathbf{V}_3^T \ddot{\boldsymbol{\beta}}\right)_v &= \begin{bmatrix} (v_3 + 2) \mathbf{u}_{y_2''}^T \mathbf{A}_{12}^T \mathbf{A}_1'^T \ddot{\boldsymbol{\beta}} \\ (v_3 + 2) \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} \\ \mathbf{u}_{y_2''}^T \mathbf{A}_{12}^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} \end{bmatrix}_v \\ &= \begin{bmatrix} (v_3 + 2) \mathbf{u}_{y_2''}^T \mathbf{A}_{12}^T \mathbf{A}_1''^T \ddot{\boldsymbol{\beta}} & (v_3 + 2) \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1'^T \ddot{\boldsymbol{\beta}} & \mathbf{u}_{y_2''}^T \mathbf{A}_{12}^T \mathbf{A}_1'^T \ddot{\boldsymbol{\beta}} \\ (v_3 + 2) \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1'^T \ddot{\boldsymbol{\beta}} & (v_3 + 2) \mathbf{u}_{y_2''}^T \mathbf{A}_{12}''^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} & \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} \\ \mathbf{u}_{y_2''}^T \mathbf{A}_{12}^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} & \mathbf{u}_{y_2''}^T \mathbf{A}_{12}'^T \mathbf{A}_1^T \ddot{\boldsymbol{\beta}} & 0 \end{bmatrix} \end{aligned}$$

From (4.7.16), (4.7.22), and (4.7.23),

$$\begin{aligned} \mathbf{X}_{2v} &= \begin{bmatrix} \dot{v}_1^2 \mathbf{A}_{12}^T \mathbf{A}_1''^T \mathbf{u}_z + \dot{v}_1 \dot{v}_2 \mathbf{A}_{12}'^T \mathbf{A}_1'^T \mathbf{u}_z & \dot{v}_1^2 \mathbf{A}_{12}'^T \mathbf{A}_1'^T \mathbf{u}_z + \dot{v}_1 \dot{v}_2 \mathbf{A}_{12}''^T \mathbf{A}_1^T \mathbf{u}_z & \mathbf{0} \end{bmatrix} \\ \mathbf{X}_{2\dot{v}} &= \begin{bmatrix} 2\dot{v}_1 \mathbf{A}_{12}^T \mathbf{A}_1'^T \mathbf{u}_z + \dot{v}_2 \mathbf{A}_{12}'^T \mathbf{A}_1^T \mathbf{u}_z & \dot{v}_2 \mathbf{A}_{12}'^T \mathbf{A}_1^T \mathbf{u}_z & \mathbf{0} \end{bmatrix} \\ (\mathbf{W}_2 \ddot{\boldsymbol{\gamma}})_v &= \begin{bmatrix} \gamma_1 \mathbf{A}_{12}^T \mathbf{A}_1'^T \mathbf{u}_z & \gamma_1 \mathbf{A}_{12}'^T \mathbf{A}_1^T \mathbf{u}_z & \mathbf{0} \end{bmatrix} \\ (\mathbf{W}_2^T \ddot{\boldsymbol{\delta}})_v &= \begin{bmatrix} \mathbf{u}_z^T \mathbf{A}_1 \mathbf{A}_{12} \ddot{\boldsymbol{\delta}} \\ \mathbf{u}_{x_1}^T \ddot{\boldsymbol{\delta}} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_z^T \mathbf{A}_1' \mathbf{A}_{12} \ddot{\boldsymbol{\delta}} & \mathbf{u}_z^T \mathbf{A}_1 \mathbf{A}_{12}' \ddot{\boldsymbol{\delta}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Due to the complexity of terms required in implicit numerical integration and the common matrices and trigonometric expressions involved, evaluation of  $\mathbf{M}(\mathbf{v})$  and  $\mathbf{g}(\mathbf{v}, \dot{\mathbf{v}})$  and their derivatives is carried out in a MATLAB function AMg in Code 4.9.3 of Appendix 4.A. Logic is imbedded to avoid computation of unnecessary derivatives when explicit numerical integration is being used and in implicit numerical integration when only equation residual evaluation is required. Code 4.9.3 supports the same four explicit integration methods and two implicit methods used in prior examples.

For numerical simulation, the inertia data of Eq. (4.7.26) are used and springs with coefficients  $K_1 = 100 \text{ Nm/rad}$ ,  $K_2 = 10,000 \text{ Nm/rad}$ , and  $K_3 = 10,000 \text{ N/m}$  are applied to the three degrees of freedom to retain variables near the zero initial condition,  $\mathbf{v}(0) = \mathbf{0}$ . An initial velocity  $\dot{\mathbf{v}}(0) = [1 \ 0 \ 0]^T$  is imposed and a 10 sec simulation is carried out. Results presented in Fig. 4.9.7 suggest that the rotational degree of freedom  $v_1$  is loosely coupled with the other two degrees of freedom. The variable  $v_1$  oscillates with a lower frequency than the other two degrees of freedom as a near sine wave with only higher frequency perturbation from the other two degrees of freedom. Total energy for this conservative system was constant to eight places in a simulation with constant step size  $h = 0.001 \text{ sec}$ , carried out using the Nystrom4 explicit integrator. Comparable results were obtained using the other five integrators. In simulations with

the explicit RKF45 and implicit SDIRK54 formulations with variable step size, the code quickly increased the step size to the upper limit of  $h = 0.01$  sec.

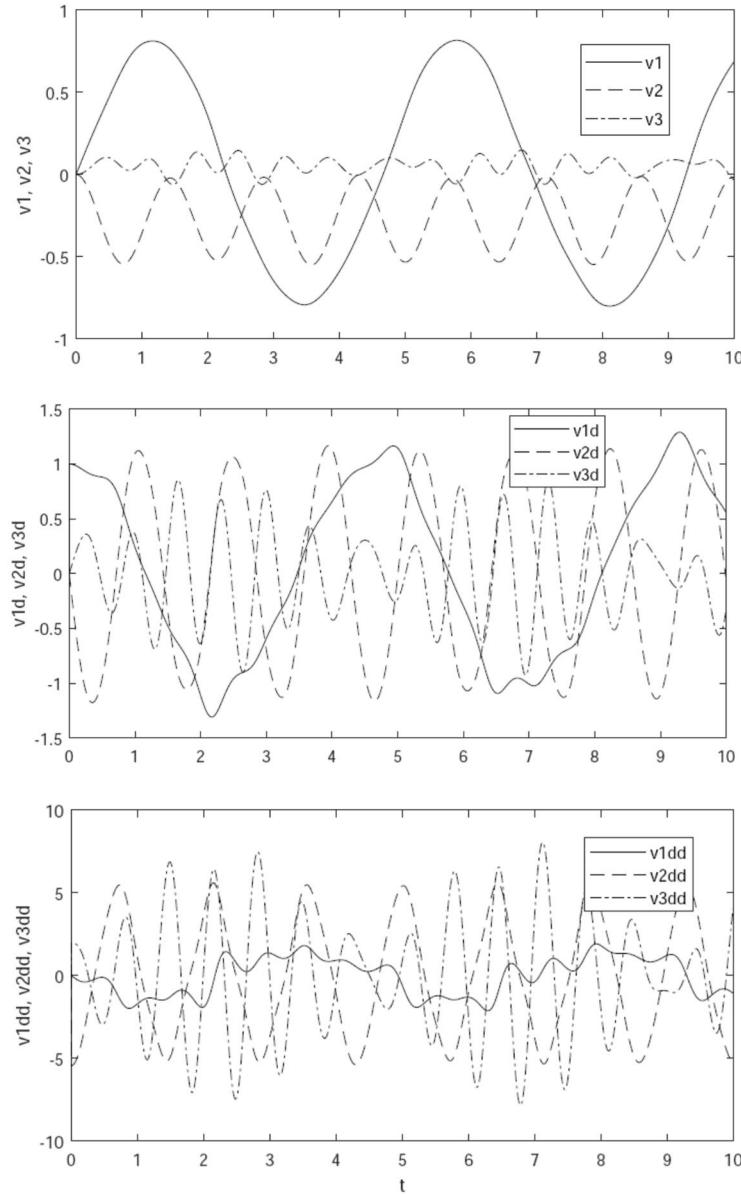


Figure 4.9.7 Manipulator Simulation

In use of the manipulator to control motion of an end-effector that is attached at the outboard end of body 3, torques and force that are determined by a controller,

$$\mathbf{Q}^A = [T_1 \quad T_2 \quad F_3]^T, \text{ would be applied to the system by addition to the right side of Eq. (4.9.8).}$$

Code 4.8 of Appendix 4.A provides six second order ODE numerical solvers for use in multibody dynamics. Integrators in Code 4.8 are used in numerous settings and applications in the remainder of the text.

Numerical integration methods presented in Section 4.8, implemented in Codes 4.9.1 through 4.9.3 of Appendix 4.A to treat the examples of Section 4.7, show excellent numerical performance. In each example, quantities required for implementation of numerical integration methods are implemented using Code 4.8 of Appendix 4.A.

While numerical results with examples presented are positive, it is clear that ad-hoc derivation and numerical solution of the ODE for each example is oppressive, and not conducive to treatment of general multibody systems. This experience motivates development of broadly applicable formulations and numerical solution methods that eliminate the need for intricate and error prone analytical manipulations to obtain and program the equations of mechanical system dynamics.

## 4.10 Constraint Forces and Differential-Algebraic Equations of Motion

Reaction forces that act on a pair of bodies that are connected by a constraint are derived, using d'Alembert's principle (Lanczos, 1962; Pars, 1965) and the *Lagrange multiplier theorem* of Section 2.2.2. With expressions for these forces, the variational equations of motion of multibody systems are written to include all generalized forces that act on the system, yielding *differential-algebraic equations* (DAE) of motion.

### 4.10.1. Generalized Constraint Reaction Forces

For a multibody system that is comprised of nb planar or spatial bodies, generalized coordinates that include position vectors and orientation variables,  $\mathbf{q}_i = \begin{bmatrix} \mathbf{r}_i^T & \phi_i \end{bmatrix}^T$  for a planar body and  $\mathbf{q}_i = \begin{bmatrix} \mathbf{r}_i^T & \mathbf{p}_i^T \end{bmatrix}^T$  for a spatial body, combined as system generalized coordinates  $\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \dots & \mathbf{q}_{nb}^T \end{bmatrix}^T$  for a planar or spatial system. Holonomic kinematic constraints on bodies i and j are

$$\Phi^k(\mathbf{q}_i, \mathbf{q}_j) = 0, \quad k = 1, \dots, nh \quad (4.10.1)$$

and Euler parameter normalization conditions for spatial bodies are

$$\Phi^{p_i} = (\mathbf{p}_i^T \mathbf{p}_i - 1)/2 = 0, \quad i = 1, \dots, nb \quad (4.10.2)$$

where nb is the number of bodies in the system. In vector form, Eqs. (4.10.1) and (4.10.2) are

$$\begin{aligned} \Phi^C(\mathbf{q}) &= \begin{bmatrix} \Phi^{1T}(\mathbf{q}) & \dots & \Phi^{nhT}(\mathbf{q}) \end{bmatrix}^T = \mathbf{0} \\ \Phi^P(\mathbf{p}) &= \begin{bmatrix} \Phi^{p_1}(\mathbf{p}_1) & \dots & \Phi^{p_{nb}}(\mathbf{p}_{nb}) \end{bmatrix}^T = \mathbf{0} \end{aligned} \quad (4.10.3)$$

Suppressing arguments of constraint functions, for notational convenience, the *virtual displacement equations* associated with Eq. (4.10.1) and the Euler parameter normalization conditions of Eq. (4.10.2) for bodies i and j are

$$\begin{aligned} \Phi_{\mathbf{q}_i}^k \mathbf{q}_i + \Phi_{\mathbf{q}_j}^k \mathbf{q}_j &= 0 \\ \mathbf{p}_i^T \delta \mathbf{p}_i &= 0 = \mathbf{p}_j^T \delta \mathbf{p}_j \end{aligned} \quad (4.10.4)$$

The virtual work of *generalized constraint reaction forces*  $\mathbf{Q}_i^{C_k}$  and  $\mathbf{Q}_j^{C_k}$  that act on bodies i and j due to constraint k is

$$W^{C_k} = \mathbf{Q}_i^{C_k T} \mathbf{q}_i + \mathbf{Q}_j^{C_k T} \mathbf{q}_j \quad (4.10.5)$$

Since *d'Alembert's principle* requires that the virtual work of constraint reaction forces is zero for all kinematically admissible virtual displacements,  $W^{C_k} = 0$  for all  $\mathbf{q}_i$  and  $\mathbf{q}_j$  that satisfy Eqs. (4.10.4). The *Lagrange multiplier theorem* of Section 2.2.2 implies existence of a vector  $\lambda^k$  with the same dimension as  $\Phi^k$  and scalars  $\frac{k}{p_i}$  and  $\frac{k}{p_j}$  such that

$$\mathbf{Q}_i^{C_k T} \mathbf{q}_i + \mathbf{Q}_j^{C_k T} \mathbf{q}_j + \lambda^{kT} \left( \Phi_{\mathbf{q}_i}^k \mathbf{q}_i + \Phi_{\mathbf{q}_j}^k \mathbf{q}_j \right) + \frac{k}{p_i} \mathbf{p}_i^T \delta \mathbf{p}_i + \frac{k}{p_j} \mathbf{p}_j^T \delta \mathbf{p}_j = 0 \quad (4.10.6)$$

for arbitrary  $\mathbf{q}_i$  and  $\mathbf{q}_j$ . This implies  $\mathbf{Q}_i^{C_k T} + \lambda^{kT} \Phi_{q_i}^k + \frac{k}{p_i} \begin{bmatrix} 0 & \mathbf{p}_i^T \end{bmatrix} = 0$  and  $\mathbf{Q}_j^{C_k T} + \lambda^{kT} \Phi_{q_j}^k + \frac{k}{p_j} \begin{bmatrix} 0 & \mathbf{p}_j^T \end{bmatrix} = 0$ . Thus,

$$\begin{aligned}\mathbf{Q}_i^{C_k} &= -\Phi_{q_i}^{kT} \lambda^k - \frac{k}{p_i} \begin{bmatrix} 0 & \mathbf{p}_i^T \end{bmatrix}^T \\ \mathbf{Q}_j^{C_k} &= -\Phi_{q_j}^{kT} \lambda^k - \frac{k}{p_j} \begin{bmatrix} 0 & \mathbf{p}_j^T \end{bmatrix}^T\end{aligned}\quad (4.10.7)$$

Deleting the Euler parameter terms and defining  $\mathbf{q}_i = [\mathbf{r}_i \ \phi_i]^T$ , Eq. (4.10.7) yields generalized constraint forces for planar systems.

#### 4.10.2 Differential-Algebraic Equations of Motion

Summing generalized constraint reaction forces on body  $i$  over all constraints that act on body  $i$ ,

$$\mathbf{Q}_i^C \equiv \sum_{k_i} \mathbf{Q}_{k_i}^{C_{k_i}} = -\sum_{k_i} \Phi_{q_i}^{kT} \lambda^{k_i} - \left( \sum_{k_i} \frac{k_i}{p_i} \right) \begin{bmatrix} 0 & \mathbf{p}_i^T \end{bmatrix}^T = -\sum_{k_i} \Phi_{q_i}^{kT} \lambda^{k_i} - \frac{\mathbf{p}_i}{p_i} \begin{bmatrix} 0 & \mathbf{p}_i^T \end{bmatrix}^T$$

where  $\frac{\mathbf{p}_i}{p_i} \equiv \left( \sum_{k_i} \frac{k_i}{p_i} \right)$ . The full vector of *constraint generalized forces* is thus

$$\mathbf{Q}^C = -\Phi_q^{CT}(\mathbf{q}) \lambda^C - \Phi_q^{pT}(\mathbf{q}) \lambda^p \quad (4.10.8)$$

where  $\mathbf{Q}^C = [\mathbf{Q}_1^{CT} \ \dots \ \mathbf{Q}_{nb}^{CT}]^T$ ,  $\lambda^C = [\lambda^{k_1 T} \ \dots \ \lambda^{k_{nb} T}]$ , and  $\lambda^p = [\mathbf{p}_1 \ \dots \ \mathbf{p}_{nb}]^T$ . The *variational equation of motion* for the multibody system of Eq. (4.6.15), including both *applied forces* and *constraint reaction forces*, is

$$\delta \mathbf{q}^T (\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_q^{CT}(\mathbf{q}) \lambda^C + \Phi_q^{pT}(\mathbf{q}) \lambda^p - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})) = 0 \quad (4.10.9)$$

where  $\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)$  is the vector of externally applied generalized forces and  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$  contains velocity coupling terms, sometimes called *Coriolis forces*. Combining constraints of Eq. (4.10.3),

$$\Phi(\mathbf{q}) = [\Phi^{CT}(\mathbf{q}) \ \Phi^{pT}(\mathbf{p})]^T = \mathbf{0} \quad (4.10.10)$$

And *associated Lagrange multipliers*,

$$\lambda = [\lambda^{CT} \ \lambda^{pT}]^T \quad (4.10.11)$$

Eq. (4.10.9) is

$$\delta \mathbf{q}^T (\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}) \lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})) = 0 \quad (4.10.12)$$

Since Eq. (4.10.12) includes all generalized forces that act on the system, all constraints are accounted for and  $\mathbf{q}$  is arbitrary. Its coefficient must therefore be zero, so

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}) \lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = 0 \quad (4.10.13)$$

These equations and the constraints of Eqs. (4.10.10) comprise a system of *differential-algebraic equations* (DAE) of motion for the multibody system. They are called “differential-algebraic” because (1) they involve derivatives of generalized coordinates and Lagrange multipliers  $\lambda$  that appear algebraically as unknowns and (2) they include the algebraic constraint equations of Eq. (4.10.10).

The Lagrange multiplier form of *constrained equations of motion* of Eqs. (4.10.13) and (4.10.10) have been known since the time of Lagrange (Lagrange, 1788), but were named *DAE* only late in the 20<sup>th</sup> century. These DAE of motion are obtained almost effortlessly, one might even say cheaply, in this section. As in many endeavors, the old saying “you get what you pay for” is operative. While an intense research effort has been focused on the DAE of dynamics during the past 1/2 century, their theoretical properties and numerical methods for their solution significantly lag that for ODE. These and related issues are addressed in Chapter 7.

It is important to be clear that the DAE of Eqs. (4.10.13) and (4.10.10) implicitly incorporates the velocity and acceleration equations that are implied by the configuration constraints of Eq. (4.10.10). The Full system DAE of dynamics, called the *Full DAE*, is thus the equation of motion of Eq. (4.10.13) and all three forms of constraint equations,

$$\begin{aligned}\Phi(\mathbf{q}) &= \mathbf{0} \\ \Phi_q(\mathbf{q})\dot{\mathbf{q}} &= \mathbf{0} \\ \Phi_q(\mathbf{q})\ddot{\mathbf{q}} &= -\left(\Phi_q(\mathbf{q})\ddot{\mathbf{q}}\right)_q \dot{\mathbf{q}} \equiv -\gamma(\mathbf{q}, \dot{\mathbf{q}})\end{aligned}\tag{4.10.14}$$

This definition is important, in contrast to formulations called index 3, index2, and index1 that enforce only the first, second, or third of Eqs (4.10.14), the remaining pair of constraints in each case being ignored. As might be expected, numerical performance of these formulations are shown in Chapter 7 to differ substantially.

While the development in this section has included Euler parameter normalization constraints and their associated Lagrange multipliers, suppressing them yields valid equations for planar systems

#### **4.10.3. Spatial Joint Constraint Reaction Force and Torque**

The virtual work of *constraint reaction force and torque* on body  $i$  due to joint  $k$ , acting at the origin of the  $x''_i$ - $y''_i$ - $z''_i$  joint definition frame shown in Fig. 4.10.1, is

$$\mathbf{W}_i^k = \mathbf{r}_i^{''PT} \mathbf{F}_i^{''k} + \boldsymbol{\pi}_i^{''T} \mathbf{T}_i^{''k}\tag{4.10.15}$$

where  $\boldsymbol{\pi}_i''$  is the virtual rotation of the  $x''_i$ - $y''_i$ - $z''_i$  frame, represented in that frame.

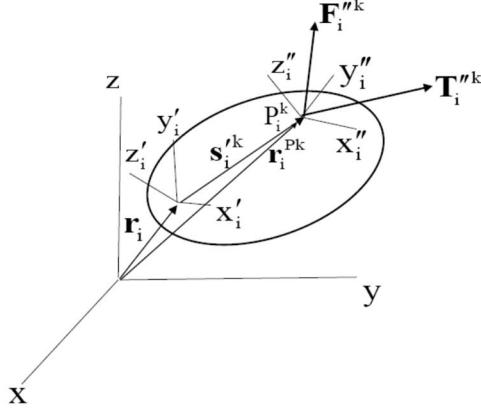


Figure 4.10.1. Constraint Reaction Force and Torque on Spatial Body i

The identity  $\mathbf{p}_i = (1/2)\mathbf{E}^T(\mathbf{p}_i) \boldsymbol{\pi}_i$  of Eq. (2.6.69) holds, since  $\mathbf{p}^T \mathbf{p} = 0$  is included in the definition of kinematically admissible virtual displacements of Eq. (4.10.4). The relation  $\mathbf{r}_i = \mathbf{A}(\mathbf{p}_i) \mathbf{C}_i^k \mathbf{r}_i''^P - \mathbf{A}(\mathbf{p}_i) \tilde{\mathbf{s}}_i^k \mathbf{C}_i^k \boldsymbol{\pi}_i''$  is from Eq. (2.4.67). Finally, from Eq. (2.6.4),  $\mathbf{A}_i = \mathbf{A}(\mathbf{p}_i) = \mathbf{E}(\mathbf{p}_i) \mathbf{G}^T(\mathbf{p}_i)$  and, from Eq. (2.6.9),  $\mathbf{E}^T(\mathbf{p}_i) \mathbf{E}(\mathbf{p}_i) = \mathbf{I} - \mathbf{p}\mathbf{p}^T$ , since the Euler parameter normalization condition on  $\mathbf{p}_i$  is included in the system constraints of Eq. (4.10.2). Using these identities, the virtual work of the generalized constraint force of Eq. (4.10.7) on body i is

$$\begin{aligned} W_i^k &= -\mathbf{q}_i^T \Phi_{\mathbf{q}_i}^{kT} \lambda^k - \sum_{\mathbf{p}_i} \mathbf{q}_i^T \begin{bmatrix} 0 & \mathbf{p}_i^T \end{bmatrix}^T = -\begin{bmatrix} \mathbf{r}_i^T & \mathbf{p}_i^T \end{bmatrix} \begin{bmatrix} \Phi_{\mathbf{r}_i}^{kT} \lambda^k \\ \Phi_{\mathbf{p}_i}^{kT} \lambda^k + \sum_{\mathbf{p}_i} \mathbf{p}_i^T \end{bmatrix} \\ &= -\left( \mathbf{r}_i^T \Phi_{\mathbf{r}_i}^{kT} \lambda^k + \mathbf{p}_i^T \Phi_{\mathbf{p}_i}^{kT} \lambda^k \right) = -\mathbf{r}_i^T \Phi_{\mathbf{r}_i}^{kT} \lambda^k - (1/2) \boldsymbol{\pi}_i^T \mathbf{E}(\mathbf{p}_i) \Phi_{\mathbf{p}_i}^{kT} \lambda^k \quad (4.10.16) \\ &= -\mathbf{r}_i''^{PT} \mathbf{C}_i^{kT} \mathbf{A}_i^T \Phi_{\mathbf{r}_i}^{kT} \lambda^k - \boldsymbol{\pi}_i''^T \left( (1/2) \mathbf{C}_i^{kT} \mathbf{G}(\mathbf{p}_i) \Phi_{\mathbf{p}_i}^{kT} \lambda^k - \mathbf{C}_i^T \tilde{\mathbf{s}}_i^k \mathbf{A}_i^T \Phi_{\mathbf{r}_i}^{kT} \lambda^k \right) \end{aligned}$$

where the identity  $\mathbf{p}_i^T \mathbf{p}_i = 0$  eliminates  $\lambda_{\mathbf{p}_i}^k$  from the equation.

Since  $\delta W_i^k$  of Eqs. (4.10.15) and (4.10.16) must be equal, for arbitrary  $\mathbf{r}_i''^P$  and  $\boldsymbol{\pi}_i''$ ,

$$\begin{bmatrix} \mathbf{F}_i''^k \\ \mathbf{T}_i''^k \end{bmatrix} = \begin{bmatrix} -\mathbf{C}_i^{kT} \mathbf{A}_i^T \Phi_{\mathbf{r}_i}^{kT} \lambda^k \\ -\mathbf{C}_i^{kT} \left( (1/2) \mathbf{G}(\mathbf{p}_i) \Phi_{\mathbf{p}_i}^{kT} - \tilde{\mathbf{s}}_i^k \mathbf{A}_i^T \Phi_{\mathbf{r}_i}^{kT} \right) \lambda^k \end{bmatrix} \quad (4.10.17)$$

These *joint constraint reaction forces and torques*, represented in the  $x_i''-y_i''-z_i''$  joint definition frame, are shown in Fig. 4.10.1. In the  $x_i'-y_i'-z_i'$  frame, Eq. (4.10.17) reduces to

$$\begin{bmatrix} \mathbf{F}_i'^k \\ \mathbf{T}_i'^k \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_i^T \Phi_{\mathbf{r}_i}^{kT} \lambda^k \\ -(1/2) \mathbf{G}(\mathbf{p}_i) \Phi_{\mathbf{p}_i}^{kT} - \tilde{\mathbf{s}}_i^k \mathbf{A}_i^T \Phi_{\mathbf{r}_i}^{kT} \end{bmatrix} \quad (4.10.18)$$

It is important to note that, whereas Lagrange multipliers associated with Euler parameter normalization conditions appear in the generalized constraint reaction forces of Eq. (4.10.7) and

in the equations of motion of Eq. (4.10.9), they do not appear in expressions for the joint constraint reaction force and torque of Eqs. (4.10.17) and (4.10.18).

#### 4.10.4 Planar Joint Constraint Reaction Force and Torque

The derivation of generalized *constraint reaction forces* of Section 4.10.3 is valid for *planar systems*, with Euler parameter constraints deleted. For constraint  $k$  with  $\mathbf{q} = \begin{bmatrix} \mathbf{r}_i^T & \phi_i \end{bmatrix}^T$ ,

$$\Phi^k(\mathbf{q}_i, \mathbf{q}_j) = 0 \quad (4.10.19)$$

there exist *Lagrange multipliers*  $\lambda^k$  of the same dimension as  $\Phi^k$  such that generalized constraint reaction forces are given by Eq. (4.10.7), with the Euler parameter term deleted; i.e.,

$$\begin{aligned} \mathbf{Q}_i^k &= -\Phi_{q_i}^{k T} \lambda^k \\ \mathbf{Q}_j^k &= -\Phi_{q_j}^{k T} \lambda^k \end{aligned} \quad (4.10.20)$$

*Constraint reaction force* and *constraint reaction torque* on body  $i$  are shown in Fig. 4.10.2, represented in the body fixed  $x'_i - y'_i$  frame at *joint definition point*  $P^k$ . The virtual work of constraint reaction force and torque on body  $i$  is

$$\delta W_i^k = \delta \mathbf{r}_i'^{P_k T} \mathbf{F}_i^k + \delta \phi_i \mathbf{T}_i^k \quad (4.10.21)$$

From Eq. (4.10.20), virtual work can also be written as

$$\delta W_i^k = \delta \mathbf{q}_i^T \mathbf{Q}_i^k = -\delta \mathbf{q}_i^T \Phi_{q_i}^{k T} \lambda^k = -\lambda^{k T} \Phi_{q_i}^k \delta \mathbf{q}_i = -\lambda^{k T} (\Phi_{r_i}^k \delta \mathbf{r}_i + \Phi_{\phi_i}^k \delta \phi_i) \quad (4.10.22)$$

The variation of the vector equation  $\mathbf{r}_i^{P_k} = \mathbf{r}_i + \mathbf{A}_i \mathbf{s}_i^{P_k}$  yields  $\delta \mathbf{r}_i^{P_k} = \delta \mathbf{r}_i + \delta \phi_i \mathbf{P} \mathbf{A}_i \mathbf{s}_i^{P_k}$ , which may be transformed to  $\delta \mathbf{r}_i = \delta \mathbf{r}_i^{P_k} - \delta \phi_i \mathbf{P} \mathbf{A}_i \mathbf{s}_i^{P_k} = \mathbf{A}_i \delta \mathbf{r}_i^{P_k} - \delta \phi_i \mathbf{P} \mathbf{A}_i \mathbf{s}_i^{P_k}$ . Substituting this result into Eq. (4.10.22) yields

$$\begin{aligned} \delta W_i^k &= -\lambda^{k T} (\Phi_{r_i}^k \mathbf{A}_i \delta \mathbf{r}_i^{P_k} - \delta \phi_i \Phi_{r_i}^k \mathbf{P} \mathbf{A}_i \mathbf{s}_i^{P_k} + \delta \phi_i \Phi_{\phi_i}^k) \\ &= -\delta \mathbf{r}_i'^{P_k T} \mathbf{A}_i^T \Phi_{r_i}^{k T} \lambda^k - \delta \phi_i (\mathbf{s}_i'^{P_k T} \mathbf{A}_i^T \mathbf{P} \Phi_{r_i}^{k T} + \Phi_{\phi_i}^{k T}) \lambda^k \end{aligned} \quad (4.10.23)$$

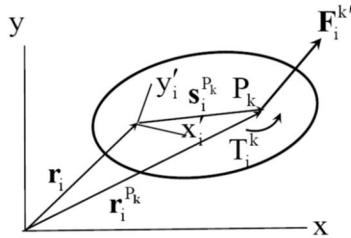


Figure 4.10.2 Constraint Reaction Force and Torque on Planar Body  $i$

Equating expressions for virtual work in Eqs. (4.10.21) and (4.10.23),

$$\delta \mathbf{r}_i'^{P_k T} \mathbf{F}_i^k + \delta \phi_i \mathbf{T}_i^k = -\delta \mathbf{r}_i'^{P_k T} \mathbf{A}_i^T \Phi_{r_i}^{k T} \lambda^k - \delta \phi_i (\mathbf{s}_i'^{P_k T} \mathbf{A}_i^T \mathbf{P} \Phi_{r_i}^{k T} + \Phi_{\phi_i}^{k T}) \lambda^k \quad (4.10.24)$$

Since Eq. (4.10.24) must hold for arbitrary  $\mathbf{r}_i'^{P_k}$  and  $\delta \phi_i$ ,

$$\begin{bmatrix} \mathbf{F}'_i \\ \mathbf{T}_i^k \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_i^T \Phi_{r_i}^{kT} \lambda^k \\ -(\mathbf{s}_i'^{P_k T} \mathbf{A}_i^T \mathbf{P} \Phi_{r_i}^{kT} + \Phi_{\phi_i}^{kT}) \lambda^k \end{bmatrix} \quad (4.10.25)$$

#### 4.10.5 Evaluation of Constraint Reaction Forces

Even if the DAE of Eq. (4.10.13) is not used to solve the equations of motion of multibody dynamics, it may be used with a solution that is obtained by other means to *evaluate constraint reaction forces* that are needed for design of joints and components of multibody systems. With a solution  $\mathbf{q}(t)$ , Eq. (4.10.13) may be written in the form

$$\Phi_q^T(\mathbf{q}) \lambda = \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} \quad (4.10.26)$$

Since the matrix  $\Phi_q(\mathbf{q})$  has full row rank,  $\Phi_q^T(\mathbf{q})$  has full column rank and the product  $\Phi_q(\mathbf{q})\Phi_q^T(\mathbf{q})$  is nonsingular. Multiplication on the left of Eq. (4.10.26) by  $\Phi_q(\mathbf{q})$  yields

$$\Phi_q(\mathbf{q})\Phi_q^T(\mathbf{q}) \lambda = \Phi_q(\mathbf{q})(\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}}) \quad (4.10.27)$$

Thus, Eq. (4.10.27) may be solved for all *Lagrange multipliers* and Eq. (4.10.18) or (4.10.25) may be used to evaluate constraint reaction force and torque that act on each body in the system.

Thus far in the text, the temptation to invoke the classical Lagrange multiplier form of multibody equations of motion has been resisted, for two reasons. First, it is not needed to obtain governing equations of motion of multibody systems. Second, it leads to DAE that are at best a complicated form of the equations of motion of mechanical system dynamics and, as shown in Chapter 7, are extraordinarily difficult to solve.

On the other side of the ledger, Lagrange multipliers introduced herein, using d'Alembert's principle and the Lagrange multiplier theorem of linear algebra, yield explicit expressions for constraint reaction forces that are important in design of bearings and components of mechanical systems. The derivation presented in this section yields computable expressions for constraint reaction forces and the Lagrange multiplier form of the equations of motion that enforce all three forms of the constraint equations, called the Full DAE.

#### Key Formulas

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}) \lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = 0 \quad (4.10.13)$$

$$\Phi(\mathbf{q}) = \mathbf{0}$$

$$\Phi_q(\mathbf{q}) \dot{\mathbf{q}} = \mathbf{0} \quad (4.10.14)$$

$$\Phi_q(\mathbf{q}) \ddot{\mathbf{q}} = -(\Phi_q(\mathbf{q}) \ddot{\mathbf{q}})_q \dot{\mathbf{q}} \equiv -\gamma(\mathbf{q}, \dot{\mathbf{q}})$$

$$\begin{bmatrix} \mathbf{F}'_i \\ \mathbf{T}_i^k \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_i^T \Phi_{r_i}^{kT} \lambda^k \\ -(1/2)\mathbf{G}(\mathbf{p}_i) \Phi_{p_i}^{kT} - \tilde{\mathbf{s}}_i'^k \mathbf{A}_i^T \Phi_{r_i}^{kT} \end{bmatrix} \quad (4.10.18)$$

$$\begin{bmatrix} \mathbf{F}'_i \\ \mathbf{T}_i^k \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_i^T \Phi_{r_i}^{kT} \lambda^k \\ -(\mathbf{s}_i'^{P_k T} \mathbf{A}_i^T \mathbf{P} \Phi_{r_i}^{kT} + \Phi_{\phi_i}^{kT}) \lambda^k \end{bmatrix} \quad (4.10.25)$$

## Appendix 4.A ODE Solution Code

### 4.A.1 Code 4.8 Second Order ODE Solvers

Components of Code 4.8 that interface with the user are presented in Section 4.A.1.1, followed by an outline of the body of the code, with which the user need not interact, in Section 4.A.1.2.

#### 4.A.1.1 User Components of Code 4.8

The initial segment of Code 4.8 involves integration and error control parameters that underlie the numerical integration methods presented in Fig. 4.A.1.1. Lines 20 through 27 define parameters that control error in the numerical integration methods supported. Relatively tight error control parameters are used as defaults, which may be modified to seek greater computer efficiency or to more tightly constrain error. The initial time step is defined in line 23, followed by the maximum time step allowed in variable time step integrators in line 24. The final simulation time is defined in line 27. The integration option to be used is selected in line 34 from among six integrators shown in lines 30 through 33. Problem data are entered in lines 43 through 48. Solution data storage arrays are defined in lines 51 through 53. Finally, initial conditions are entered in lines 56 and 57.

```
1 %Code 4.8: Solution of Second Order ODE M(v)vdd=g(v,vd,t) of Eq.(4.8.43)
2 %on time interval [0,tf]
3
4 %Functions beginning with A require user input; Functions beginning with
5 %B, E, I, and O do not require user input
6
7 %Function g=Agf(t,v,vd,par) defines g(v,vd,t); required for all integrators
8
9 %Function M=AM(v,par) defines M(v); required for all integrators
10
11 %Function [gsv,gsvd] = Agfsvvd(t,v,vd,par) defines partial derivatives of
12 %g(v,vd,t) w.r.t. v and vd; required for implicit integrators
13
14 %Function M2=AM2(v,mu,par) defines derivative of (M(v)mu) w.r.t. v;
15 %required for implicit integrators
16
17 %Function [,,,]=AdatPart(dat) defines problem dependent data; may be
18 %defined if user wishes to employ named problem data in subroutines
19
20 intol=10^-6; %Tolerance in solving discretized equations
21 Atol=10^-6; %Absolute error tolerance for variable step methods
22
23 h=0.001; %Step size
24 hmax=0.01; %Maximum Allowable Step size
25 hvar=2; %hvar=1, variable step;hvar=2, fixed step
26
27 tfinal=12; %Final time
28
29 %Explicit Integration Methods:
30 %Integ=1, Nystrom4; Integ=2, RungeKutta4;
31 %Integ=3, Kutta3/8; integ=4, RKFN
```

```

32 %Implicit Integration Methods:
33 %Integ=5, Trapezoidal; Integ=6, SDIRK54
34 Integ=6; %Select from among Numerical Integration Options
35
36 nv=[]; %User define variable dimension
37
38 %Fixed Parameter Data List-Partitioning in function BparPart if
39 %use of variable names is desired in user supplied functions.
40 par=[nv;intol;Atol;hmax;hvar];
41
42 % Problem Data
43 %enter problem data by "name=value;" and enter into dat list.
44 %Problem Parameter Data List-Partitioning in function AdatPart if
45 %use of variable names is desired in user supplied functions.
46 dat=[;;;;]; %User input data parameters here and in function AdatPart,
47 %as in function BparPart
48 %Example: dat=[m1,J1,k1]
49
50 % Data Storage Arrays
51 V=zeros(nv,10);
52 Vd=zeros(nv,10);
53 Vdd=zeros(nv,10);
54
55 % Initial Conditions
56 v0=[]; %User define initial position
57 vd0=[]; %User define initial velocity
58
59 V(:,1)=v0;
60 Vd(:,1)=vd0;
61
62 % NO USER CHANGES/INPUT REQUIRED BEYOND THIS POINT

```

Figure 4.A.1.1 User Input to Integration Code

#### ***4.4.1.2 Computational Components of Code***

Computational flow in the main program beyond line 62, which requires no input from the user, is outlined in Fig. 4.A.1.2. Data for integration are initialized in lines 64 through 81. Code that implements the six numerical integration methods supported is provided in lines 83 through 120. Integration results are evaluated and recorded in lines 122 through 143.

```

64 %Integration Preparation
65 n=1; %Time step counter
66 t(1)=0; %Initial time
67
68 %Initial Parameters
69 nch=1;
70
71 % Integration
72 while t(n)<tfinal
73 %Time Step Update
74 n=n+1;
75 t(n)=t(n-1)+h;
76 tn=t(n);

```

```

77
78 % Integration
79 tnm=t(n-1);
80 vnm=V(:,n-1);
81 vdnm=Vd(:,n-1);
82
83 if Integ<5 %Explicit integrators
84 if Integ==1
85 [vn,vdn,vddn,Mcond]=ExplicitNystrom4(tnm,vnm,vdnm,h,par,dat);
86 end
87
88 if Integ==2
89 [vn,vdn,vddn,Mcond]=ExplicitRungeKutta4(tnm,vnm,vdnm,h,par,dat);
90 end
91
92 if Integ==3
93 [vn,vdn,vddn,Mcond]=ExplicitKutta38(tnm,vnm,vdnm,h,par,dat);
94 end
95
96 if Integ==4
97 [vn,vdn,vddn,Mcond,h,nch]=ExplicitRKFN45(n,tnm,vnm,vdnm,h,nch, ...
98 par,dat,hvar,ny,Atol,hmax);
99 hrpt(n)=h;
100 end
101
102 Mcondrpt(n)=Mcond; %Record condition number of mass matrix
103 end
104
105 if Integ >4 %Implicit integrators
106
107 if Integ==5
108 [vn,vdn,vddn,jiter,R1Norm,JCond]=ImplicitTrap(n,tn, ...
109 V,Vd,Vdd,intol,par,dat,h);
110 jiterrpt(n)=jiter;
111 end
112
113 if Integ==6
114 [vn,vdn,vddn,Maxjiter,R1Norm,JCond,h,err]=ImplicitSDIRK54(n,tn, ...
115 V,Vd,Vdd,par,dat,intol,Atol,ny,h,hmax,nch,hvar);
116 hrpt(n)=h;
117 R1Normrpt(n)=R1Norm;
118 errrpt(n)=err;
119 jiterrpt(n)=Maxjiter;
120 end
121
122 JCondrpt(n)=JCond; %Record Jacobian Condition Nujmber
123
124 end
125
126 %Evaluate and Record Solution
127 V(:,n)=vn;
128 vnorm(n)=norm(vn);
129 Vd(:,n)=vdn;

```

```

130 vdnorm(n)=norm(vdn);
131 Vdd(:,n)=vddn;
132 vddnorm(n)=norm(vddn);
133
134 %Report key data
135 v1(n)=vn(1);
136 v2(n)=vn(2);
137 s1=sin(vn(1));
138 s2=sin(vn(2));
139 c1=cos(vn(1));
140 c2=cos(vn(2));
141 TE(n)=0.5*vdn'*AM(vn,par,dat)*vdn+9.8*(2*s1+s2); %Total Energy
142
143 end

```

Figure 4.A.1.2 Main Code Computational Flow

Computing functions that underlie the main code are identified in Fig. 4.A.1.3. Computing functions include AdatPart and BparPart that partition data in the dat and par arrays, for use in all computing functions. The user enters  $\mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}})$  in Function Agf ,  $\mathbf{g}_v(t, \mathbf{v}, \dot{\mathbf{v}})$  and  $\mathbf{g}_{\dot{v}}(t, \mathbf{v}, \dot{\mathbf{v}})$  in Function Agfsvv,  $\mathbf{M}(\mathbf{v})$  in Function AM, and  $(\mathbf{M}(\mathbf{v})\dot{\mathbf{v}})_v$  in Function AM2. The six integration Functions listed implement the indicated numerical integration method. Finally, the ODEfunct Function computes the solution  $\dot{\mathbf{v}}$  of Eq. **Error! Reference source not found.**. Internal details of the Computing Functions listed in Fig. 4.A.1.3 are not presented here, since each of the functions is documented internally and the user need not modify these functions in applications.

```

AdatPart
Agf
Agfsvv
AM
AM2
BparPart
ExplicitKutta38
ExplicitNystrom4
ExplicitRKFN45
ExplicitRungeKutta4
ImplicitSDIRK54
ImplicitTrap
ODEfunct

```

Figure 4.A.1.3 Computing Functions in Code 4.8

#### **4.A.2 Code 4.A.1 Planar Double Pendulum**

See MATLAB Code in accompanying software file.

#### **4.A.3 Code 4.A.2 Planar Slider-Crank**

See MATLAB Code in accompanying software file.

#### **4.A.4 Code 4.A.3 Three Degree of Freedom Robotic Manipulator**

See MATLAB Code in accompanying software file.

## Appendix 4.B Key Formulas, Chapter 4

$$\Phi(\mathbf{q}, t) = 0 \quad \mathbf{E}(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{e}(\mathbf{q}, t) \quad \mathbf{C}(\mathbf{q}, t) = \begin{bmatrix} \Phi_q(\mathbf{q}, t) \\ \mathbf{E}(\mathbf{q}, t) \end{bmatrix} \quad (4.1.6) \quad (4.1.9) \quad (4.1.12)$$

$$\sum_{i=1}^{np} \delta \mathbf{q}_i^T (\mathbf{m}_i \mathbf{I}_3 \ddot{\mathbf{q}}_i - \mathbf{F}_i^A) = \delta \mathbf{q}^T (\mathbf{M} \ddot{\mathbf{q}} - \mathbf{F}^A) = 0 \quad \mathbf{C}(\mathbf{q}, t) \delta \mathbf{q} = 0 \quad (4.1.25) \quad (4.1.13)$$

$$\delta \mathbf{r}^T [m\ddot{\mathbf{r}} + m(\ddot{\phi}\mathbf{P}\mathbf{A}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi))\mathbf{s}'^c - \mathbf{F}] + \delta \phi [ms'^c \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} + \dot{\phi} \mathbf{J}' - \mathbf{n}'] = 0 \quad (4.2.18)$$

$$\delta \mathbf{r}^T [m\ddot{\mathbf{r}} - \mathbf{F}] + \delta \phi [\dot{\phi} \mathbf{J}' - \mathbf{n}'] = 0 \quad (4.2.21)$$

$$m\ddot{\mathbf{r}} + m(\ddot{\phi}\mathbf{P}\mathbf{A}(\phi) - \dot{\phi}^2 \mathbf{A}(\phi))\mathbf{s}'^c = \mathbf{F} \quad ms'^c \mathbf{A}^T(\phi) \mathbf{P}^T \ddot{\mathbf{r}} + \dot{\phi} \mathbf{J}' = \mathbf{n}' \quad (4.2.22)$$

$$m\ddot{\mathbf{r}} = \mathbf{F} \quad J' \dot{\phi} = n' \quad (4.2.23)$$

$$\begin{aligned} & \delta \mathbf{r}^T (m\ddot{\mathbf{r}} - mA(\mathbf{p})\tilde{s}'^c \dot{\omega}' + mA(\mathbf{p})\tilde{\omega}' \dot{\omega}' s'^c - \mathbf{F}) \\ & + \delta \pi'^T (m\tilde{s}'^c \mathbf{A}^T(\mathbf{p})\ddot{\mathbf{r}} + \mathbf{J}' \dot{\omega}' + \tilde{\omega}' \mathbf{J}' \omega' - \mathbf{n}') = 0 \end{aligned} \quad (4.3.15)$$

$$\delta \mathbf{r}^T (m\ddot{\mathbf{r}} - \mathbf{F}) + \delta \pi'^T (\mathbf{J}' \dot{\omega}' + \tilde{\omega}' \mathbf{J}' \omega' - \mathbf{n}') = 0 \quad (4.3.17)$$

$$\begin{aligned} & \delta \mathbf{r}^T [m\ddot{\mathbf{r}} - 2mA\tilde{s}'^c \mathbf{G}\ddot{\mathbf{p}} - 4mE\dot{\mathbf{G}}^T \dot{\mathbf{G}} \mathbf{G}^T s'^c - \mathbf{F}] \\ & + \delta \mathbf{p}^T [2m\mathbf{G}^T \tilde{s}'^c \mathbf{A}^T \ddot{\mathbf{r}} + 4\mathbf{G}^T \mathbf{J}' \mathbf{G}\ddot{\mathbf{p}} - 8\dot{\mathbf{G}}^T \mathbf{J}' \dot{\mathbf{G}} \mathbf{p} - 2\mathbf{G}^T \mathbf{n}'] = 0 \end{aligned} \quad (4.3.24)$$

$$\delta \mathbf{r}^T [m\ddot{\mathbf{r}} - \mathbf{F}] + \delta \mathbf{p}^T [4\mathbf{G}^T \mathbf{J}' \mathbf{G}\ddot{\mathbf{p}} - 8\dot{\mathbf{G}}^T \mathbf{J}' \dot{\mathbf{G}} \mathbf{p} - 2\mathbf{G}^T \mathbf{n}'] = 0 \quad (4.3.25)$$

$$\Phi^k(\mathbf{q}_i, \mathbf{q}_j, t) = 0 \quad \mathbf{E}_i^k(\mathbf{q}_i, \mathbf{q}_j, t)\dot{\mathbf{q}}_i + \mathbf{E}_j^k(\mathbf{q}_i, \mathbf{q}_j, t)\dot{\mathbf{q}}_j = \mathbf{e}^k(\mathbf{q}_i, \mathbf{q}_j, t) \quad (4.6.1) \quad (4.6.4)$$

$$\mathbf{C}_i^k \equiv \begin{cases} \Phi_q^k \text{ or } \mathbf{E}_i^k, & \text{if constraint } k \text{ involves body } i \\ 0, & \text{otherwise} \end{cases}, \quad i = 1 \dots nb \quad (4.6.6)$$

$$\mathbf{q} = [\mathbf{q}_1^T \dots \mathbf{q}_{nb}^T]^T \quad \delta \mathbf{q} = [\delta \mathbf{q}_1^T \dots \delta \mathbf{q}_{nb}^T]^T \quad \mathbf{M} = \text{diag}(\mathbf{M}_1 \dots \mathbf{M}_{nb}) \quad (4.6.15)$$

$$\mathbf{Q}^A = [\mathbf{Q}_1^{AT} \dots \mathbf{Q}_{nb}^{AT}]^T \quad \mathbf{S} = [\mathbf{S}_1^T \dots \mathbf{S}_{nb}^T]^T$$

$$\delta \mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0 \quad \mathbf{C} \delta \mathbf{q} = 0 \quad (4.6.16) \quad (4.6.17)$$

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t) \quad \mathbf{y}(t^0) = \mathbf{y}^0 \quad (4.8.1)$$

$$\mathbf{K}_i = \mathbf{f} \left( t_n + c_i h, \mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{K}_j \right), \quad i=1, \dots, s \quad \mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{K}_i \quad (4.8.2)$$

$$\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad \mathbf{v}(t^0) = \mathbf{v}^0 \quad \dot{\mathbf{v}}(t^0) = \dot{\mathbf{v}}^0 \quad (4.8.13)$$

$$\mathbf{v}_n = \mathbf{v}_{n-1} + h \dot{\mathbf{v}}_{n-1} + (h^2 / 4) (\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \quad \dot{\mathbf{v}}_n = \dot{\mathbf{v}}_{n-1} + (h / 2) (\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \quad (4.8.40)$$

$$\mathbf{k}_i = \mathbf{g} \left( t_n + c_i h, \mathbf{v}_n + h c_i \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^i A_{ij} \mathbf{k}_j, \dot{\mathbf{v}}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}_j \right) \quad (4.8.27)$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + h \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^s B_j \mathbf{k}_j \quad \dot{\mathbf{v}}_{n+1} = \dot{\mathbf{v}}_n + h \sum_{j=1}^s b_j \mathbf{k}_j \quad (4.8.28)$$

$$A_{il} = \sum_{j=1}^s a_{ij} a_{jl} \quad B_l = \sum_{j=1}^s b_j a_{jl} \quad (4.8.21) \quad (4.8.26)$$

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q})\lambda - Q^A(\mathbf{q}, \dot{\mathbf{q}}, t) - S(\mathbf{q}, \dot{\mathbf{q}}) = 0 \quad (4.10.13)$$

$$\begin{bmatrix} \mathbf{F}_i^{jk} \\ \mathbf{T}_i^{jk} \end{bmatrix} = \begin{bmatrix} -\mathbf{C}_i^{kT} \mathbf{A}_i^T \Phi_{r_i}^{kT} \lambda^k \\ -\mathbf{C}_i^{kT} ((1/2)\mathbf{G}(\mathbf{p}_i) \Phi_{p_i}^{kT} - \tilde{s}_i^{jk} \mathbf{A}_i^T \Phi_{r_i}^{kT}) \lambda^k \end{bmatrix} \quad (4.10.16)$$

$$\Phi(\mathbf{q}) = 0$$

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = 0 \quad (4.10.21)$$

$$\Phi_q(\mathbf{q})\ddot{\mathbf{q}} = -\left(\Phi_q(\mathbf{q})\dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} \equiv -\gamma(\mathbf{q}, \dot{\mathbf{q}})$$

## CHAPTER 5

### Tangent Space ODE for Holonomic Systems

#### 5.0 Introduction

As the reality that DAE formulations of mechanical system dynamics introduced in Sections 4.10.2 and 4.10.5 are very difficult to solve has matured (Petzold, 1982; Haug, 1989; Ascher and Petzold, 1998; Bauchau and Laulusa, 2008), the search for *ODE formulations* has intensified. Notably, Maggi (1896; 1901) and Kane (1985) formulations that are valid for nonholonomic systems have been used for holonomic systems by enforcing velocity constraint equations and leaving the configuration constraint equations unenforced, except at the initial time (Tseng, Ma, and Hulbert, 2003; Bauchau and Laulusa, 2008; Laulusa and Bauchau, 2008; Garcia de Jalon, Callejo, and Hidalgo, 2012). Problems with this formulation are summarized in Appendix 5.D.

Extending the Maggi/Kane approach (Haug, 2018d), parameterizations of system kinematics in terms of holonomic constraint *tangent space generalized coordinates* are presented in this chapter to reduce the equations of motion for planar and spatial systems to ODE, without relying on the use of Lagrange multipliers. ODE solution methods of Section 4.8 are applied to obtain simulation results that satisfy configuration, velocity, and acceleration constraint equations.

A spinning Top with Euler parameter orientation coordinates is used as a model problem in Section 5.1, for which globally valid independent generalized coordinates do not exist. The *tangent space parameterization* method, based on differentiable manifold theory of Sections 3.5 and 3.6, is used to reduce variational equations of motion to ODE. Numerical results obtained using Runge-Kutta integrators verify that configuration, velocity, and acceleration constraints are satisfied and that no singularities are encountered. A broadly applicable tangent space parameterization of multibody system kinematics and dynamics is presented in Section 5.2 that enforces constraints at configuration, velocity, and acceleration levels. An algorithm for computation of constraint reaction forces using ODE solutions is presented.

ODE initial-value problems of multibody system dynamics are solved in Section 5.3, using tangent space generalized coordinates. Computational algorithms for implementing the approach with explicit and implicit numerical integrators of Section 4.8 are presented and used in solution of examples in Section 5.4. Performance of the method in satisfying all three forms of kinematic constraint, based on error tolerances in the formulation, is verified.

An *Index 0 DAE formulation* of the equations of multibody dynamics is derived in Section 5.5, using tangent space generalized coordinates. The tangent space ODE formulation is modified to include Lagrange multipliers that are introduced in Section 4.10 to represent constraint reaction forces. This makes the Index 0 DAE formulation ideally suited for simulation of systems with friction, as is done in Chapter 8. Equivalence of the ODE and Index 0 DAE formulations is shown. Explicit and implicit numerical integration algorithms for numerical integration of the Index 0 DAE formulation are presented in Section 5.5 and used for numerical solution of examples in Section 5.6.

General-purpose MATLAB computer Codes 5.7 and 5.9 of Appendix 5A are presented in Sections 5.7 and 5.9 for modeling and simulation of planar and spatial systems, respectively.

Components of the codes that enable user definition of models and simulation alternatives are presented, along with an outline of computational components of the codes. Dynamic simulations of planar and spatial systems are presented in Sections 5.8 and 5.10, using Codes 5.7 and 5.9, including larger scale system simulations that would be intractable if pursued in an ad-hoc fashion.

Theorems that show the variational, tangent space ODE, Index 0 DAE, and Full DAE formulations of holonomic multibody dynamics are equivalent and well posed are proved in Section 5.11.

Derivatives of planar and spatial constraints of Sections 3.2 and 3.3 and kinetic derivatives that are required to implement the ODE and Index 0 DAE formulations are presented in Appendix 5.B, followed by a summary of key formulas for Chapter 5 in Appendix 5.C. Finally, deficiencies of the Maggi and Kane equations, when used to represent holonomic systems, are presented in Appendix 5.D. Their extension for treatment of holonomic systems, using the tangent space ODE formulation of this chapter has been outlined in (Haug, 2018d).

## 5.1 A Spatial Model Problem That Requires Local Coordinates

As shown in Section 3.5.1, even for a single particle on the two-dimensional surface of the unit sphere, it is not possible to find two generalized coordinates that parameterize the entire surface without encountering a singularity. Similarly, it is observed in Section 2.5.3 that it is not possible to parameterize the entire three-dimensional space of orientations of a rigid body with three orientation generalized coordinates. Examples presented in Sections 3.5 through 3.7 demonstrate that singularities may arise in the equations of system kinematics that preclude definition of globally valid independent generalized coordinates. Finally, it is shown in Chapter 3 that multibody system kinematic models lead to high dimensional nonlinear equations that defy practical definition of globally valid independent generalized coordinates, even if they exist. Thus, local generalized coordinates in the differential geometric setting outlined in Sections 3.5 and 3.6 are required to obtain equations of motion that are valid everywhere in the regular configuration space. A spatial rigid body model problem is treated in this section, using the tangent space parameterization introduced in Section 3.5, that resolves difficulties noted and motivates a broadly applicable approach to multibody system dynamics that is developed in Section 5.2.

It is critical in formulation of equations of motion as ODE that *independent generalized coordinates* be defined that satisfy configuration, velocity, and acceleration constraints. Otherwise, the equations of motion obtained will be in error. A tangent space *local parameterization* of the constraint manifold, following the approach in Section 3.5.2, is presented, leading to *locally defined independent generalized coordinates* that satisfy configuration, velocity, and acceleration constraint equations. Use of these coordinates and the variational equation of motion derived in Chapter 4 lead to ODE of motion that are ideally suited for digital computer implementation. The approach illustrated for the model problem is generalized in the remainder of the chapter, as a basis for computational implementation for ODE simulation of broad classes of multibody dynamic systems.

Consider the *symmetric Top* shown in Fig. 5.1.1. A body-fixed  $x'$ - $y'$ - $z'$  reference frame has its origin at the tip, with the  $z'$  axis as the axis of symmetry for the Top. The tip is attached to the origin of the inertial  $x$ - $y$ - $z$  reference frame so  $\mathbf{r} = \mathbf{0}$  and orientation of the Top is defined by *Euler parameters*, which are the generalized coordinates of the system. Admissible Euler parameter vectors  $\mathbf{p} = [e_0 \quad \mathbf{e}^T]^T \in \mathbb{R}^4$  must satisfy the normalization condition of Eq. (2.5.25) and its derivatives,

$$\begin{aligned} (\mathbf{p}) &= (\mathbf{p}^T \mathbf{p} - 1) / 2 = 0 \\ {}_{\mathbf{p}}(\mathbf{p})\dot{\mathbf{p}} &= \mathbf{p}^T \dot{\mathbf{p}} = 0 \\ {}_{\mathbf{p}}(\mathbf{p})\ddot{\mathbf{p}} &= \mathbf{p}^T \ddot{\mathbf{p}} = -\dot{\mathbf{p}}^T \dot{\mathbf{p}} \end{aligned} \tag{5.1.1}$$

The conditions of Eq. (5.1.1) are orientation, velocity, and acceleration constraints.

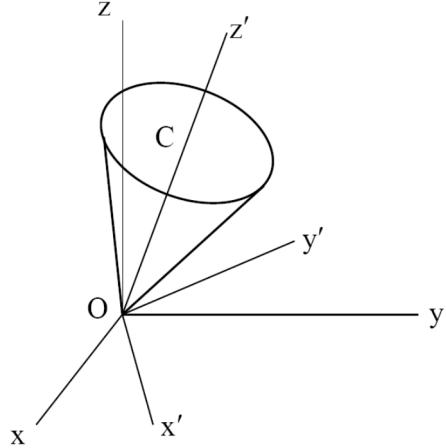


Figure 5.1.1 Symmetric Top with Tip Fixed

### 5.1.1 Tangent Space Generalized Coordinates

Let Euler parameters at the initial time  $t^0$ ,  $\mathbf{p}(t^0) = \mathbf{p}^0$ , satisfy the first of Eqs. (5.1.1). The tangent space for the regular configuration space  $\tilde{\mathcal{C}} = \{\mathbf{p} : \mathbf{p}^T \mathbf{p} = 1 \text{ and } \mathbf{p}_\parallel(\mathbf{p}) = \mathbf{p}^T \neq \mathbf{0}\}$  at  $\mathbf{p}^0$  in  $\mathbb{R}^4$  is the set of all vectors  $\dot{\mathbf{p}}$  such that  $\mathbf{p}^{0T} \dot{\mathbf{p}} = 0$ . As in Section 3.5, define

$$\mathbf{U} = \mathbf{p}(\mathbf{p}^0)^T = \mathbf{p}^0 \quad (5.1.2)$$

Equations (2.6.6) and (2.6.8) imply  $\mathbf{G}(\mathbf{p}^0)\mathbf{p}^0 = \mathbf{0}$  and  $\mathbf{G}(\mathbf{p}^0)\mathbf{G}(\mathbf{p}^0)^T = \mathbf{I}$ , so columns of  $\mathbf{G}(\mathbf{p}^0)^T$  lie in the tangent space of  $\tilde{\mathcal{C}}$  space at  $t^0$  and are orthogonal unit vectors. Thus, they form a basis for the tangent space, denoted

$$\mathbf{V} = \mathbf{G}(\mathbf{p}^0)^T \quad (5.1.3)$$

where  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T \mathbf{U} = \mathbf{V}^T \mathbf{p}^0 = \mathbf{0}$ . In this special case, singular value decomposition is not required for evaluation of  $\mathbf{V}$ . Since the unit vector  $\mathbf{U}$  is orthogonal to the three linearly independent columns of  $\mathbf{V}$ , the columns of  $\mathbf{V}$  and  $\mathbf{U}$  form a basis for  $\mathbb{R}^4$ . Thus, every vector  $\mathbf{p}$  in  $\mathbb{R}^4$  can be uniquely written in terms of tangent space coordinates  $\mathbf{v}$  and  $\mathbf{u}$  as

$$\mathbf{p} = \mathbf{p}^0 + \mathbf{V}\mathbf{v} - \mathbf{u}\mathbf{U} \quad (5.1.4)$$

where the sign of the third term on the right is selected to represent a projection onto the constraint manifold from the tangent space, as shown schematically in Fig. 5.1.2.

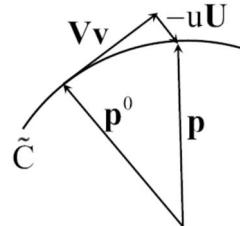


Figure 5.1.2. Orthogonal Projection onto Constraint Manifold

Multiplying Eq. (5.1.4) on the left by  $\mathbf{V}^T$  and  $\mathbf{U}^T$  and using the relations  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ ,  $\mathbf{V}^T\mathbf{U} = \mathbf{0}$ , and  $\mathbf{U}^T\mathbf{U} = \mathbf{1}$  yields the inverse relations

$$\begin{aligned}\mathbf{v} &= \mathbf{V}^T(\mathbf{p} - \mathbf{p}^0) \\ u &= -\mathbf{U}^T(\mathbf{p} - \mathbf{p}^0)\end{aligned}\quad (5.1.5)$$

In particular, Eq. (5.1.5) implies  $\mathbf{v}(t^0) = \mathbf{v}^0 = \mathbf{0}$  and  $u(t^0) = u^0 = 0$ .

Requiring that  $\mathbf{p}$  of Eq. (5.1.4) satisfies the first of Eqs. (5.1.1) yields

$$\mathbf{p}^T \mathbf{p} - 1 = u^2 - 2u + \mathbf{v}^T \mathbf{v} = 0 \quad (5.1.6)$$

Using the quadratic formula,  $u = 1 \pm \sqrt{1 - \mathbf{v}^T \mathbf{v}}$ . Clearly, the parameterization of Eq. (5.1.4) is only valid if  $\mathbf{v}^T \mathbf{v} < 1$ . Geometrically, this means that if  $\mathbf{v}^T \mathbf{v} > 1$ , the line in Fig. 5.1.2 that is perpendicular to the tangent space through the point  $\mathbf{p}^0 + \mathbf{V}\mathbf{v}$  does not intersect the unit sphere in  $\mathbb{R}^4$ . Since  $u = 0$  if  $\mathbf{v} = \mathbf{0}$ , the minus sign must be selected, so

$$u = 1 - \sqrt{1 - \mathbf{v}^T \mathbf{v}} \quad (5.1.7)$$

Equation (5.1.4) is thus

$$\mathbf{p} \equiv \psi(\mathbf{v}) = \mathbf{p}^0 + \mathbf{V}\mathbf{v} - \left(1 - \sqrt{1 - \mathbf{v}^T \mathbf{v}}\right) \mathbf{U} \quad (5.1.8)$$

Substituting  $\mathbf{p}$  of Eq. (5.1.8) into the first of Eqs. (5.1.1) verifies that it is satisfied, for arbitrary  $\mathbf{v}$  in a neighborhood of  $\mathbf{v}^0 = \mathbf{0}$ .

Since  $\mathbf{V}$  and  $\mathbf{U}$  are constant, differentiation of Eq. (5.1.4) yields

$$\begin{aligned}\dot{\mathbf{p}} &= \mathbf{V}\dot{\mathbf{v}} - \dot{u}\mathbf{U} \\ \ddot{\mathbf{p}} &= \mathbf{V}\ddot{\mathbf{v}} - \ddot{u}\mathbf{U}\end{aligned}\quad (5.1.9)$$

As in the derivation of the inverse relation of Eq. (5.1.5), multiplying Eqs. (5.1.9) on the left by  $\mathbf{V}^T$  and  $\mathbf{U}^T$  and using the relations  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ ,  $\mathbf{V}^T\mathbf{U} = \mathbf{0}$ , and  $\mathbf{U}^T\mathbf{U} = \mathbf{1}$  yields the inverse relations

$$\begin{aligned}\dot{\mathbf{v}} &= \mathbf{V}^T \dot{\mathbf{p}} \\ \dot{u} &= -\mathbf{U}^T \dot{\mathbf{p}} \\ \ddot{\mathbf{v}} &= \mathbf{V}^T \ddot{\mathbf{p}} \\ \ddot{u} &= -\mathbf{U}^T \ddot{\mathbf{p}}\end{aligned}\quad (5.1.10)$$

Substituting  $\dot{\mathbf{p}}$  of Eq. (5.1.9) into the second of Eqs. (5.1.1) yields the relation  $\mathbf{p}^T \mathbf{V} \dot{\mathbf{v}} - \dot{u} \mathbf{p}^T \mathbf{U} = 0$ . Since  $\mathbf{U}^T \mathbf{U} = \mathbf{1}$ ,  $\mathbf{p}^T \mathbf{U} \neq 0$  in a neighborhood of  $\mathbf{p}^0$ ,

$$\dot{u} = \left( \frac{1}{\mathbf{p}^T \mathbf{U}} \right) \mathbf{p}^T \mathbf{V} \dot{\mathbf{v}} \quad (5.1.11)$$

in that neighborhood. Substituting this relation into the first of Eqs. (5.1.9),

$$\dot{\mathbf{p}} = \mathbf{V}\dot{\mathbf{v}} - \left( \frac{1}{\mathbf{p}^T \mathbf{U}} \right) \mathbf{U} \mathbf{p}^T \mathbf{V} \dot{\mathbf{v}} = \left[ \mathbf{I} - \left( \frac{1}{\mathbf{p}^T \mathbf{U}} \right) \mathbf{U} \mathbf{p}^T \right] \mathbf{V} \dot{\mathbf{v}} = \mathbf{D}(\mathbf{p}) \dot{\mathbf{v}} \quad (5.1.12)$$

where

$$\mathbf{D}(\mathbf{p}) \equiv \left[ \mathbf{I} - \left( \frac{1}{\mathbf{p}^T \mathbf{U}} \right) \mathbf{U} \mathbf{p}^T \right] \mathbf{V} \quad (5.1.13)$$

Substituting  $\mathbf{p}$  of Eq. (5.1.8) and  $\dot{\mathbf{p}}$  of Eq. (5.1.12) into the second of Eqs. (5.1.1) verifies that it is satisfied for all  $\mathbf{v}$  in a neighborhood of  $\mathbf{v}^0 = \mathbf{0}$  and arbitrary  $\dot{\mathbf{v}}$ . A result identical in form to Eq. (5.1.12) holds for *differentials*; i.e.,

$$\delta \mathbf{p} = \mathbf{D} \delta \mathbf{v} \quad (5.1.14)$$

Substituting from the second of Eqs. (5.1.9) into the third of Eqs. (5.1.1) yields the relation  $\mathbf{p}^T \mathbf{V} \ddot{\mathbf{v}} - \dot{\mathbf{u}} \mathbf{p}^T \mathbf{U} + \dot{\mathbf{p}}^T \mathbf{p} = 0$ . Thus, in a neighborhood of  $\mathbf{p}^0$ ,

$$\ddot{\mathbf{u}} = \left( \frac{1}{\mathbf{p}^T \mathbf{U}} \right) (\mathbf{p}^T \mathbf{V} \ddot{\mathbf{v}} + \dot{\mathbf{p}}^T \dot{\mathbf{p}}) \quad (5.1.15)$$

Substituting this relation into the second of Eqs. (5.1.9) and manipulating,

$$\ddot{\mathbf{p}} = \mathbf{D}(\mathbf{p}) \ddot{\mathbf{v}} - \left( \frac{\dot{\mathbf{p}}^T \dot{\mathbf{p}}}{\mathbf{p}^T \mathbf{U}} \right) \mathbf{U} \quad (5.1.16)$$

Substituting  $\mathbf{p}$  of Eq. (5.1.8),  $\dot{\mathbf{p}}$  of Eq. (5.1.12), and  $\ddot{\mathbf{p}}$  of Eq. (5.1.16) into the third of Eqs. (5.1.1) verifies that it is satisfied for all  $\mathbf{v}$  in a neighborhood of  $\mathbf{v}^0 = \mathbf{0}$  and arbitrary  $\dot{\mathbf{v}}$  and  $\ddot{\mathbf{v}}$ . Equations (5.1.8), (5.1.12), and (5.1.16) thus yield orientation, velocity, and acceleration that satisfy all three forms of constraint in Eqs. (5.1.1), as long as  $\|\mathbf{v}\| < 1$ .

### 5.1.2 ODE of Motion

With gravitational acceleration  $\mathbf{g}$  acting in the negative  $\mathbf{z}$  direction on the Top of Fig. 5.1.1,  $\mathbf{n}' = -mg\tilde{\mathbf{u}}'_z \mathbf{A}(\mathbf{p})^T \mathbf{u}_z + \mathbf{n}'^A$ , where  $\mathbf{u}'_z$  and  $\mathbf{u}_z$  are unit vectors along the positive  $\mathbf{z}'$  and  $\mathbf{z}$  axes of the body fixed and inertial reference frames, respectively, and  $\mathbf{n}'^A$  is an *externally applied torque*. Since  $\mathbf{r} = \delta \mathbf{r} = \ddot{\mathbf{r}} = \mathbf{0}$ , the *variational equation of motion* of Eq. (4.3.22) is

$$\delta \mathbf{p}^T \left[ 2\mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \ddot{\mathbf{p}} - 4\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p} - \mathbf{G}(\mathbf{p})^T (-mg\tilde{\mathbf{u}}'_z \mathbf{A}(\mathbf{p})^T \mathbf{u}_z + \mathbf{n}'^A) \right] = 0 \quad (5.1.17)$$

which must hold for all  $\mathbf{p}$  such that  $\Phi_p(\mathbf{p}) \mathbf{p} = \mathbf{p}^T \mathbf{p} = 0$ . Substituting  $\mathbf{p}$  and  $\ddot{\mathbf{p}}$  from Eqs. (5.1.14) and (5.1.16) into Eq. (5.1.17),

$$\delta \mathbf{v}^T \mathbf{D}(\mathbf{p})^T \left[ \begin{array}{l} 2\mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \left( \mathbf{D}(\mathbf{p}) \ddot{\mathbf{v}} - \left( \frac{\dot{\mathbf{p}}^T \dot{\mathbf{p}}}{\mathbf{p}^T \mathbf{U}} \right) \mathbf{U} \right) - 4\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p} \\ - \mathbf{G}(\mathbf{p})^T (-mg\tilde{\mathbf{u}}'_z \mathbf{A}(\mathbf{p})^T \mathbf{u}_z + \mathbf{n}'^A) \end{array} \right] = 0$$

Since  $\mathbf{p}^T \mathbf{p} = \mathbf{p}^T \mathbf{D}(\mathbf{p}) \mathbf{v} = \left[ \mathbf{p}^T - \frac{\mathbf{p}^T \mathbf{U}}{\mathbf{p}^T \mathbf{U}} \mathbf{p}^T \right] \mathbf{V} \mathbf{v} = 0$  for arbitrary  $\mathbf{v}$ , this equation must hold for arbitrary  $\delta \mathbf{v}$ . Thus,

$$\begin{aligned} 2\mathbf{D}(\mathbf{p})^T \mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \mathbf{D}(\mathbf{p}) \ddot{\mathbf{v}} &= 2 \left( \frac{\dot{\mathbf{p}}^T \dot{\mathbf{p}}}{\mathbf{p}^T \mathbf{U}} \right) \mathbf{D}(\mathbf{p})^T \mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \mathbf{U} \\ &+ 4\mathbf{D}(\mathbf{p})^T \mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p} + \mathbf{D}(\mathbf{p})^T \mathbf{G}(\mathbf{p})^T \left( -mg \tilde{\mathbf{u}}_z' \mathbf{A}(\mathbf{p})^T \mathbf{u}_z + \mathbf{n}'^A \right) \end{aligned} \quad (5.1.18)$$

At  $\mathbf{p}^0$ ,  $\mathbf{D} = \mathbf{V} = \mathbf{G}(\mathbf{p}^0)^T$  and, since  $\mathbf{G}(\mathbf{p}^0) \mathbf{G}(\mathbf{p}^0)^T = \mathbf{I}$ , the reduced mass matrix  $2\mathbf{D}^T \mathbf{G}^T \mathbf{J}' \mathbf{G} \mathbf{D}$  on the left of Eq. (5.1.18) at  $\mathbf{p}^0$  equals  $2\mathbf{J}'$ , which is positive definite. Thus, in a neighborhood of  $\mathbf{p}^0$ , the symmetric matrix  $2\mathbf{D}^T(\mathbf{p}) \mathbf{G}^T(\mathbf{p}) \mathbf{J}' \mathbf{G}(\mathbf{p}) \mathbf{D}(\mathbf{p})$  is positive definite, hence nonsingular. The first of Eqs. (5.1.5) and (5.1.10) yield initial conditions

$$\begin{aligned} \mathbf{v}^0 &= \mathbf{0} \\ \dot{\mathbf{v}}^0 &= \mathbf{V}^T \dot{\mathbf{p}}^0 \end{aligned} \quad (5.1.19)$$

Equation (5.1.18) is a *second order ODE* of the form of Eq. (4.8.45),

$$\bar{\mathbf{M}}(\mathbf{v}) \ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (5.1.20)$$

where, from Eqs. (5.1.8) and (5.1.12),  $\mathbf{p}(\mathbf{v}) = \mathbf{p}^0 + \mathbf{V}\mathbf{v} - \left( 1 - \sqrt{1 - \mathbf{v}^T \mathbf{v}} \right) \mathbf{p}^0$ ,

$$\dot{\mathbf{p}}(\mathbf{v}, \dot{\mathbf{v}}) = \left[ \mathbf{I} - \left( \frac{1}{\mathbf{p}(\mathbf{v})^T \mathbf{p}^0} \right) \mathbf{p}^0 \mathbf{p}(\mathbf{v})^T \right] \mathbf{V} \dot{\mathbf{v}}, \text{ and}$$

$$\begin{aligned} \bar{\mathbf{M}}(\mathbf{v}) &= 2\mathbf{D}(\mathbf{p})^T \mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \mathbf{D}(\mathbf{p}) \\ \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}}, t) &= 2 \left( \frac{\dot{\mathbf{p}}^T \dot{\mathbf{p}}}{\mathbf{p}^T \mathbf{p}^0} \right) \mathbf{D}(\mathbf{p})^T \mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \mathbf{p}^0 \\ &+ 4\mathbf{D}(\mathbf{p})^T \mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p} \\ &+ \mathbf{D}(\mathbf{p})^T \mathbf{G}(\mathbf{p})^T \left( -mg \tilde{\mathbf{u}}_z' \mathbf{A}(\mathbf{p})^T \mathbf{u}_z + \mathbf{n}'^A \right) \end{aligned} \quad (5.1.21)$$

The arguments of all functions of  $\mathbf{p}$  and  $\dot{\mathbf{p}}$  involved are shown, to emphasize that they are functions of  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ . If Eq. (5.1.18), with initial conditions of Eq. (5.1.19), is numerically integrated using an ODE solver, all three forms of constraint of Eqs. (5.1.1) will be satisfied, to within the accuracy of the integrator.

Since  $\mathbf{v}$  is only a valid parameterization of the constraint manifold for  $\mathbf{v}^T \mathbf{v} < 1$ , bounds on the magnitude of  $\mathbf{v}$  must be monitored during numerical integration. Further, the *condition number*, defined in Section 2.2.8, of  $\bar{\mathbf{M}}(\mathbf{v})$  in Eq. (5.1.21) must be bounded to avoid singularity in Eq. (5.1.20). If an assigned tolerance on norm of  $\mathbf{v}$  or condition number of  $\bar{\mathbf{M}}(\mathbf{v})$  is exceeded, the current value of  $\mathbf{p}$  is designated  $\bar{\mathbf{p}}^0$ ,  $\bar{\mathbf{U}} = \bar{\mathbf{p}}^0$ , and  $\bar{\mathbf{V}} = \mathbf{G}(\bar{\mathbf{p}}^0)^T$  are redefined. The integration

process is then continued with new initial conditions on  $\bar{\mathbf{v}}$  and  $\dot{\bar{\mathbf{v}}}$ , namely from Eqs. (5.1.5) and (5.1.10),  $\bar{\mathbf{v}}^0 = \mathbf{0}$  and  $\dot{\bar{\mathbf{v}}}^0 = \mathbf{V}^T \dot{\mathbf{p}}^0$ . The process is continued to a specified final time, with results reported in terms of Euler parameters and their derivatives, using Eqs. (5.1.8), (5.1.12), and (5.1.16). The user of the process need not know values of  $\mathbf{v}$  and its derivatives that have been computed. This process can be continued, as shown in Fig. 3.6.2, since with  $\mathbf{p}^T \mathbf{p} = 1$ ,  $\mathbf{p}(\mathbf{p}) = \mathbf{p}^T$  never fails to have full rank.

### 5.1.3 Numerical Solution

Inertia properties used for simulation are  $\mathbf{J}' = \text{diag}(120, 120, 30) \text{ kg} \cdot \text{m}^2$ , relative to the *noncentroidal body fixed reference frame* at the tip,  $m = 30 \text{ kg}$ , and  $g = 9.8 \text{ m/sec}^2$ . The *initial configuration* is with the Top vertically upward; i.e.,  $\mathbf{p}^0 = [1 \ 0 \ 0 \ 0]^T$ . The *initial angular velocity* in the body fixed reference frame is  $\omega'^0 = [\epsilon \ \epsilon \ \text{omegaz}0]^T$ , where initial angular velocity components  $\epsilon = 10^{-12}$  about the  $x'$  and  $y'$  axes play the role of perturbations from the vertical configuration. From Eq. (2.6.64),  $\dot{\mathbf{p}}^0 = 0.5\mathbf{G}(\mathbf{p}^0)^T \omega'^0$ . As a check on *conservation of energy, total energy* is  $\text{TE} = 2\dot{\mathbf{p}}^T \mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \dot{\mathbf{p}} + mgs_z$ , where  $s_z$  is the  $z$  component of the vector  $\mathbf{s} = \mathbf{A}(\mathbf{p})\mathbf{u}_z$  from the tip to the centroid of the Top. Code 4.8 of Appendix 4.A was adapted for solution of this initial-value problem as Code 5.1.2 of Appendix 5.A. Four explicit numerical integration algorithms of Section 4.8 were implemented.

Plots in Fig. 5.1.3 show the  $x$  and  $y$  coordinates of the centroid as time progresses over a 100 sec interval, obtained with four initial values of  $z'$  initial angular velocity  $\text{omegaz}0$ . As  $\text{omegaz}0$  is increased from 12 rad/sec at the upper left to 13.5 rad/sec at the lower right, the *radius of precession* reduces from 0.5 m to  $2 \times 10^{-12}$  m. Thus, *stability of the spinning Top* is achieved for the higher initial angular velocity.

In all simulations, *total energy* and  *$z'$  momentum* were constant to eight places for this *conservative system*. Numerical results reported were obtained using the *Nystrom4 explicit integrator* of Table 4.8.8, with constant step size 0.001 sec. Comparable results were obtained with three other explicit integrators. During integration, 793 *reparameterizations* in 100,000 time- steps (130 time steps per reparameterization) were required, all due to a limit of 0.75 on the norm of  $\mathbf{v}$ . Computing cost to reparameterize the equations of motion was a negligible 0.03% of total CPU time.

The maximum error over the simulation interval in satisfying each of the constraints in Eqs. (5.1.1) was less than  $10^{-14}$ . The *tangent space equations of motion* thus account for nonlinear effects of Euler's equation of motion and the Euler parameter normalization, velocity, and acceleration constraints of Eqs. (5.1.1), as predicted in the case of this analytical reduction, to machine precision.

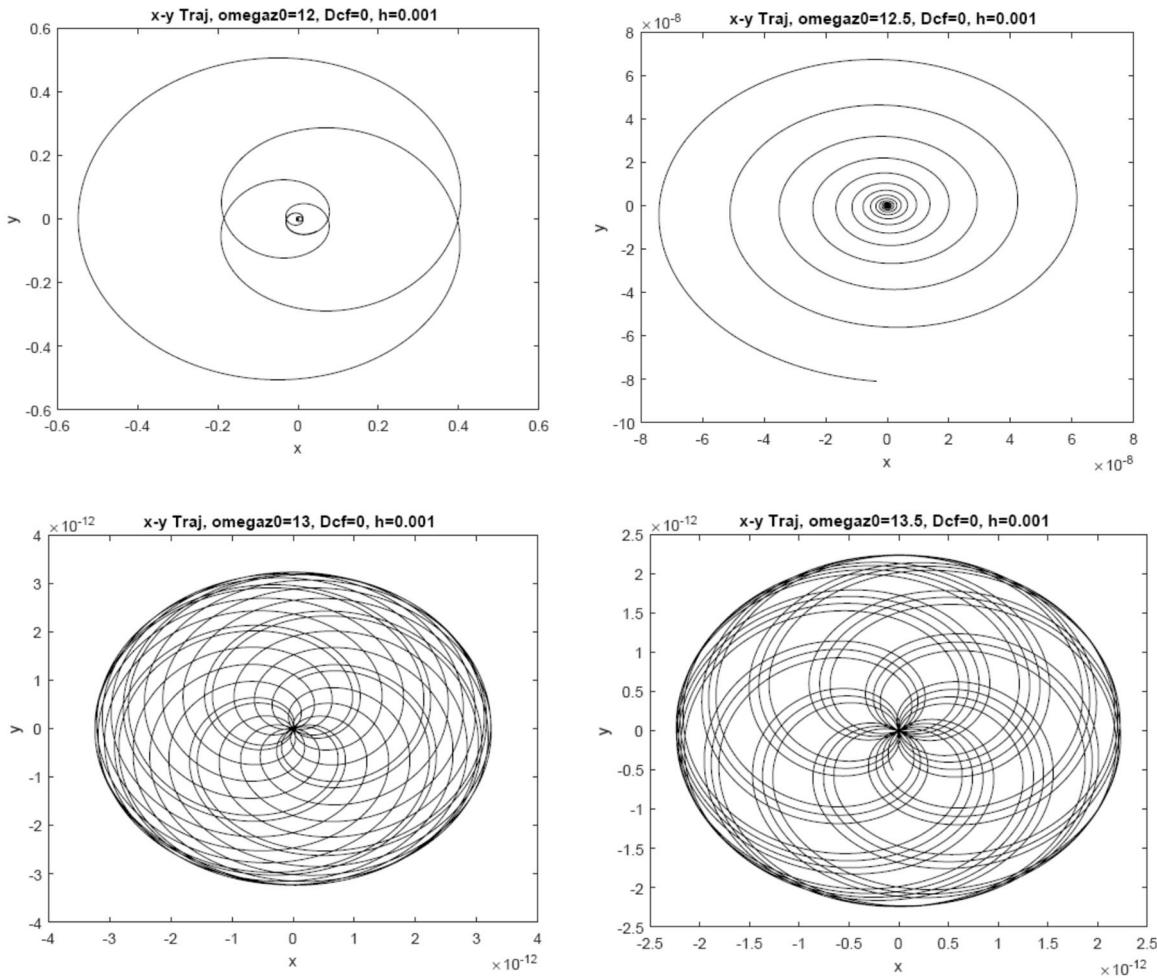


Figure 5.1.3 x-y Trajectory of Centroid

Adding a torque  $-Dcf \|\omega'\|^2 = -4Dcf \|\mathbf{G}(\mathbf{p})\dot{\mathbf{p}}\|^2$  due to *aerodynamic drag*, applied to the Top about the body-fixed  $z'$  axis leads to a loss of kinetic energy and an increasing magnitude in the radius of precession over the simulation interval. With aerodynamic drag coefficient  $Dcf = 0.005$ , x-y trajectories and plots of norm of angular velocity (omegnorm) are shown in Fig. 5.1.4, with initial vertical angular velocities of 14 and 16 rad/sec. For both initial angular velocities, there was a loss in total energy of approximately 35% over the 100 sec simulation. During integration, 786 *reparameterizations* in 100,000 time steps (127 time steps per reparameterization) were required, all due to a limit of 0.75 on the norm of  $\mathbf{v}$ . Constraint errors were comparable to those reported for the simulation without damping.

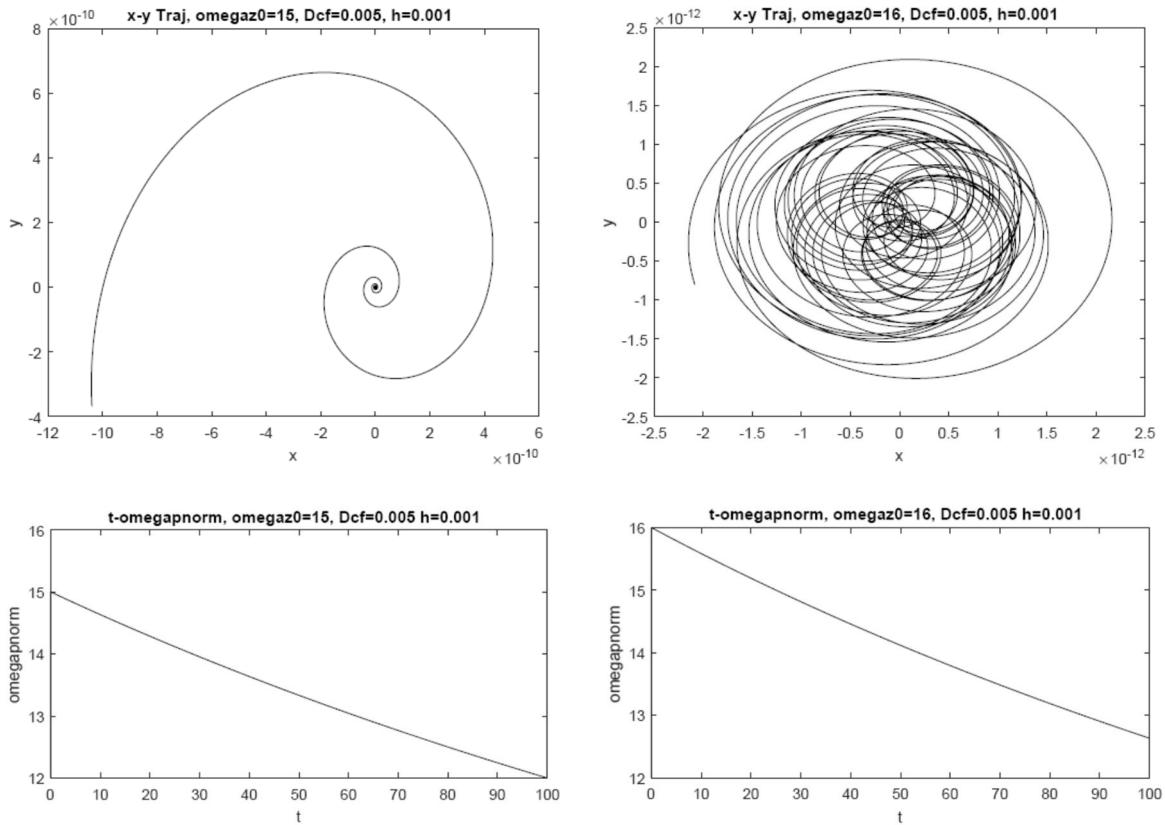


Figure 5.1.4 x-y Trajectory of Centroid and Norm of Angular Velocity with Drag

The computational approach presented for the model problem treated in this section (1) employs independent generalized coordinates that satisfy all three forms of kinematic constraint, (2) defines initial conditions on independent position and velocity, (3) forms ODE of motion with no singularities, and (4) integrates the equations of motion using numerical methods to an accuracy specified by the user. This methodology is based on rigorous foundations of analytical dynamics, mathematics, numerical analysis, and computational science. These statements serve as a challenge in subsequent sections to maintain rigor in generalizing the approach presented for this model problem to more realistic applications.

Numerical solution methods of Section 4.8 are implemented in Codes 5.1.1 and 5.1.2 of Appendix 5.A and demonstrate the ability to reliably and accurately solve the second order ODE of motion.

## 5.2 Tangent Space ODE

*Tangent space parameterization* of the constraint manifold presented in Sections 3.5 and 3.6 is extended to the realm of dynamics, including holonomic constraints with explicit time dependence associated with kinematic drivers that are defined in Sections 3.2.2 and 3.3.5. Kinematic relations among displacements, virtual displacements, velocities, and accelerations are used with the variational equations of motion of Section 4.6 to obtain second order ODE of dynamics and associated initial conditions in independent *tangent space generalized coordinates*.

### 5.2.1 Tangent Space Kinematics with Time Dependent Constraints

A time dependent holonomically constrained multibody system is represented by nhc *holonomic constraint equations* in ngc generalized coordinates  $\mathbf{q}$  and time  $t$ , of the form

$$\bar{\Phi}(\mathbf{q}, t) \equiv \Phi(\mathbf{q}) - \mathbf{f}(t) = \mathbf{0} \quad (5.2.1)$$

where the *decoupled dependence on  $\mathbf{q}$  and  $t$*  is associated with *driving constraints* presented in Sections 3.2 and 3.3. With this form of constraint, the constraint Jacobian with respect to  $\mathbf{q}$  and numerous other derivative expressions do not involve explicit time dependence. Constraint time derivatives yield velocity and acceleration constraint equations,

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{f}}(t) \quad (5.2.2)$$

$$\Phi_q(\mathbf{q})\ddot{\mathbf{q}} = -\left(\Phi_q(\mathbf{q})\hat{\dot{\mathbf{q}}}\right)_q \dot{\mathbf{q}} + \ddot{\mathbf{f}}(t) \equiv -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (5.2.3)$$

which, along with Eq. (5.2.1), comprise three forms of constraint equations that must be satisfied. The *constraint Jacobian*  $\Phi_q(\mathbf{q})$  at a configuration  $\mathbf{q}^0$  that satisfies Eq. (5.2.1) at  $t^0$ ; i.e.,  $\Phi(\mathbf{q}^0) - \mathbf{f}(t^0) = \mathbf{0}$ , is evaluated as  $\bar{\Phi}_q(\mathbf{q}^0, t^0) = \Phi_q(\mathbf{q}^0)$ . With time held fixed, Eq. (5.2.2) yields  $\Phi_q(\mathbf{q})\delta\mathbf{q} = \mathbf{0}$ , defining *kinematically admissible virtual displacements*  $\delta\mathbf{q}$  associated with the *regular configuration space*,

$$\tilde{C}(t) = \left\{ \mathbf{q} : \Phi(\mathbf{q}) - \mathbf{f}(t) = \mathbf{0} \text{ and } \text{rank}(\Phi_q(\mathbf{q})) = \text{nhc} \right\} \quad (5.2.4)$$

The fact that the constraint Jacobian  $\Phi_q(\mathbf{q})$  does not depend explicitly on time in this formulation minimizes the complexity of system kinematics.

For  $\mathbf{q}^0 \in \tilde{C}(t^0)$ , the constraint Jacobian  $\Phi_q(\mathbf{q}^0)$  defines a matrix  $\mathbf{U}$  whose column vectors are linearly independent; i.e.,

$$\mathbf{U} \equiv \Phi_q^{0T} = \Phi_q(\mathbf{q}^0)^T \quad (5.2.5)$$

and  $\mathbf{U}^T \mathbf{U}$  is positive definite, hence nonsingular. A second matrix  $\mathbf{V}$  is defined as the solution of

$$\begin{aligned} \Phi_q^0(\mathbf{q}^0)\mathbf{V} &= \mathbf{U}^T \mathbf{V} = \mathbf{0} \\ \mathbf{V}^T \mathbf{V} &= \mathbf{I} \end{aligned} \quad (5.2.6)$$

using *singular value decomposition* (Strang, 1980). The second of Eqs. (5.2.6) implies that  $\mathbf{V}$  has full rank. Since its columns are orthogonal to the columns of  $\mathbf{U}$ , the columns of  $\mathbf{U}$  and  $\mathbf{V}$  span  $\mathbb{R}^{\text{nnc}}$ . As shown in Section 3.5.2, Eq. (3.5.23),  $\mathbf{V}$  and  $\mathbf{U}$  are related by the identity

$$\mathbf{V}\mathbf{V}^T + \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T = \mathbf{I} \quad (5.2.7)$$

A *tangent space parameterization* of the regular configuration space of Eq. (5.2.4), in a neighborhood of  $(\mathbf{q}^0, t^0)$ , is defined by

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u} \quad (5.2.8)$$

The variables  $\mathbf{v}$  and  $\mathbf{u}$  that are introduced in Eq. (5.2.8) comprise a set of *local generalized coordinates* that are in one to one correspondence with  $\mathbf{q}$  in a neighborhood of  $\mathbf{q}^0$ ; i.e., they are equivalent to  $\mathbf{q}$ . To see this, multiply Eq. (5.2.8) by  $\mathbf{V}^T$  and  $\mathbf{U}^T$  to obtain

$$\begin{aligned} \mathbf{v} &= \mathbf{V}^T(\mathbf{q} - \mathbf{q}^0) \\ \mathbf{u} &= -(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T(\mathbf{q} - \mathbf{q}^0) \end{aligned} \quad (5.2.9)$$

Thus, the parameterization of Eq. (5.2.8) is a *diffeomorphism*; i.e., it is one to one, onto, differentiable, and with a differentiable inverse in a neighborhood of  $(\mathbf{q}^0, t^0)$ . It is clear from Eq. (5.2.9) that at  $\mathbf{q} = \mathbf{q}^0$ ,

$$\begin{aligned} \mathbf{v}^0 &= \mathbf{0} \\ \mathbf{u}^0 &= \mathbf{0} \end{aligned} \quad (5.2.10)$$

To assure that  $\mathbf{q}$  of Eq. (5.2.8) satisfies the holonomic constraints of Eq. (5.2.1), it is required that

$$\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}) - \mathbf{f}(t) = \mathbf{0} \quad (5.2.11)$$

The derivative of the left side of this equation with respect to  $\mathbf{u}$ , with  $\mathbf{v}$  and  $t$  held constant, is  $\Phi'(\ddot{\mathbf{q}}^0 + \dot{\mathbf{V}}\mathbf{v} - \dot{\mathbf{U}}\mathbf{u})_{\mathbf{u}} = -\Phi'_u(\mathbf{q})\mathbf{U}$ , or at  $\mathbf{q}^0$ ,

$$\Phi'_u(\mathbf{q}^0) = -\Phi'_u(\mathbf{q}^0)\mathbf{U} = -\mathbf{U}^T\mathbf{U} \equiv -(\mathbf{B}^0)^{-1} \quad (5.2.12)$$

which is nonsingular. Since  $\Phi'_u(\mathbf{q})$  is a continuously differentiable matrix function of  $\mathbf{q}$ ,

$$\mathbf{B}(\mathbf{q}) \equiv (\Phi'_u(\mathbf{q})\mathbf{U})^{-1} \quad (5.2.13)$$

is nonsingular and continuously differentiable with respect to its arguments in a neighborhood  $X_0$  of  $\mathbf{q}^0$ .

Since  $\Phi'_u(\mathbf{q}) = -\Phi'_u(\mathbf{q})\mathbf{U} = -\mathbf{B}^{-1}(\mathbf{q})$  is nonsingular in a neighborhood of  $\mathbf{q}^0$  and  $\mathbf{u} = \mathbf{0}$  is a solution of Eq. (5.2.11) with  $\mathbf{v} = \mathbf{0}$  and  $t = t^0$ , the *implicit function theorem* of Section 2.2.5 guarantees that Eq. (5.2.11) has a unique continuously differentiable solution

$$\mathbf{u} = \mathbf{h}(\mathbf{v}, t) \quad (5.2.14)$$

in a neighborhood  $(V_0, T_0)$  of  $(\mathbf{v}^0, t^0)$ . Thus, Eq. (5.2.8) becomes

$$\mathbf{q}(\mathbf{v}, t) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) \quad (5.2.15)$$

which satisfies Eq. (5.2.1) for all  $\mathbf{v}$  and  $t$  in a neighborhood  $(V_0, T_0)$  of  $(\mathbf{v}^0, t^0)$ .

To see geometrically what Eq. (5.2.15) means and why the extent of the neighborhood  $V_0$  is important, the geometry of choosing  $\mathbf{u}$  to stay in the regular configuration space of Eq. (5.2.4) is shown in Fig. 5.2.1. The vector  $\mathbf{q}^0 + \mathbf{V}\mathbf{v}$  may be viewed as movement tangent to the space, which must be modified by vector  $-\mathbf{U}\mathbf{h}(\mathbf{v}, t)$  to yield  $\mathbf{q}(\mathbf{v}, t)$  in the space. Equation (5.2.15) consolidates the two vectors into the continuously differentiable vector function  $\mathbf{q}(\mathbf{v}, t)$ . It is clear from Fig. 5.2.1, however, that if  $\mathbf{v}$  is large, the vector  $-\mathbf{U}\mathbf{h}(\mathbf{v}, t)$  may not intersect the  $\tilde{C}(t)$  and the parameterization fails; i.e., generalized coordinates  $\mathbf{v}$  are no longer valid. Fortunately, a large *condition number* of  $\Phi_q(\mathbf{q})\mathbf{U}$ , defined in Section 2.2.7, is an effective warning that such a situation is impending.

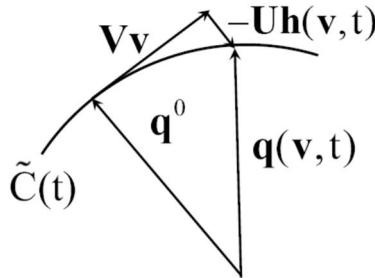


Figure 5.2.1 Projection onto Regular Configuration Space

The quantities  $\mathbf{B}(\mathbf{q})$  and  $\mathbf{h}(\mathbf{q}, t)$  are shown to exist and be continuously differentiable functions of their arguments. While they cannot be evaluated as explicit functions of their arguments, they can be evaluated numerically as accurately as desired using iterative algorithms. The vector  $\mathbf{h}(\mathbf{q}, t)$  can be evaluated using the *Newton-Raphson method* of Section 2.2.8 to solve Eq. (5.2.11) for  $\mathbf{u}$ , with given  $\mathbf{v}$  and  $t$ ; i.e.,

$$\Phi_u(\mathbf{q})\Delta\mathbf{u}^i = -\Phi_q(\mathbf{q}^0)\mathbf{U}\Delta\mathbf{u}^i = -\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i) + \mathbf{f}(t^0) \quad (5.2.16)$$

Using Eq. (5.2.13), the solution of Eq. (5.2.16) is  $\Delta\mathbf{u}^i = \mathbf{B}(\mathbf{q})(\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i) - \mathbf{f}(t^0))$ , yielding the iterative algorithm

$$\begin{aligned} \mathbf{u}^{i+1} &= \mathbf{u}^i + \mathbf{B}(\mathbf{q})(\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i) - \mathbf{f}(t^0)), \\ i &= 1, 2, \dots \text{ until } \|\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^{i+1}) - \mathbf{f}(t^0)\| \leq \text{utol} \end{aligned} \quad (5.2.17)$$

where utol is a specified *error tolerance*. Since the Newton-Raphson method does not require an exact Jacobian, the matrix  $\mathbf{B}(\mathbf{q})$  is held constant for the process. Since only matrix multiplication is required, this is an efficient computation.

The matrix  $\mathbf{B}(\mathbf{q})$  must be updated with changes in  $\mathbf{u}$  and  $\mathbf{v}$ , hence changes in  $\mathbf{q}$ . To enhance computational efficiency,  $\mathbf{B}^0 = (\Phi_q^0 \Phi_q^{0T})^{-1}$  is evaluated at  $(\mathbf{q}^0)$  and an iterative method is used to update  $\mathbf{B}(\mathbf{q}) = (\Phi_q(\mathbf{q})\mathbf{U})^{-1}$  as the computation process proceeds. For  $\mathbf{q} \neq \mathbf{q}^0$ ,  $\mathbf{B}(\mathbf{q}, t)$

must satisfy Eq. (5.2.13). In residual form, suppressing arguments  $\mathbf{q}$  for notational convenience, this is

$$\bar{\mathbf{R}} = \Phi_q \mathbf{U} \mathbf{B} - \mathbf{I} = \mathbf{0} \quad (5.2.18)$$

With an approximation  $\mathbf{B}^i$  of the solution, *Newton-Raphson iteration* can be carried out,

$$(\Phi_q \mathbf{U}) \Delta \mathbf{B}^i = -\bar{\mathbf{R}}^i = -\Phi_q \mathbf{U} \mathbf{B}^i + \mathbf{I} \quad i = 1, 2, \dots \quad (5.2.19)$$

Since the matrix  $(\Phi_q \mathbf{U})$  need not be inverted with great precision for use in the Newton-Raphson process and  $\mathbf{B}^i \approx (\Phi_q \mathbf{U})^{-1}$ , Eq. (5.2.19) yields  $\Delta \mathbf{B}^i = -\mathbf{B}^i \Phi_q \mathbf{U} \mathbf{B}^i + \mathbf{B}^i$ . This is the efficient iterative algorithm

$$\mathbf{B}^{i+1} = 2\mathbf{B}^i - \mathbf{B}^i \Phi_q \mathbf{U} \mathbf{B}^i, \quad i=1,2,\dots, \text{ until } \|\Phi_q \mathbf{U} \mathbf{B}^{i+1} - \mathbf{I}\| \leq \text{Btol} \quad (5.2.20)$$

where Btol is a specified error tolerance. Equation (5.2.20) may be executed whenever the test  $\|\Phi_q \mathbf{U} \mathbf{B}^i - \mathbf{I}\| \leq \text{Btol}$  fails, an efficient computation that requires only matrix multiplication. In this way, the foregoing kinematic equations and numerous more to follow with coefficient matrix  $\Phi_u = -\Phi_q \mathbf{U} = -\mathbf{B}^{-1}$  can be solved using only multiplication by  $\mathbf{B}$ .

### 5.2.2 Kinematic Relations

Differentiating Eq. (5.2.15) with respect to  $t$ , recalling that  $\mathbf{V}$  and  $\mathbf{U}$  are constant,

$$\dot{\mathbf{q}} = \mathbf{V} \dot{\mathbf{v}} - \mathbf{U} \dot{\mathbf{h}} \quad (5.2.21)$$

To evaluate  $\dot{\mathbf{h}}$ , the derivative of Eq. (5.2.11) with respect to  $t$ , with  $\mathbf{u} = \mathbf{h}(\mathbf{q}, t)$  and suppressing arguments, is  $\Phi_q (\mathbf{V} \dot{\mathbf{v}} - \mathbf{U} \dot{\mathbf{h}}) - \dot{\mathbf{f}}(t) = \mathbf{0}$ . Using Eq. (5.2.13), this yields

$$\dot{\mathbf{h}} = \mathbf{B} (\Phi_q \mathbf{V} \dot{\mathbf{v}} - \dot{\mathbf{f}}(t)) \quad (5.2.22)$$

and Eq. (5.2.21) is

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{V} \dot{\mathbf{v}} - \mathbf{U} \mathbf{B}(\mathbf{q}) \Phi_q(\mathbf{q}) \mathbf{V} \dot{\mathbf{v}} + \mathbf{U} \mathbf{B}(\mathbf{q}) \dot{\mathbf{f}}(t) \\ &= \mathbf{D}(\mathbf{q}) \dot{\mathbf{v}} + \mathbf{U} \mathbf{B}(\mathbf{q}) \dot{\mathbf{f}}(t) \end{aligned} \quad (5.2.23)$$

where

$$\mathbf{D}(\mathbf{q}) \equiv (\mathbf{I} - \mathbf{U} \mathbf{B}(\mathbf{q}) \Phi_q(\mathbf{q})) \mathbf{V} \quad (5.2.24)$$

At  $t^0$ , multiplying Eq. (5.2.23) on the left by  $\mathbf{V}^T$  and using the facts that  $\mathbf{V}^T \mathbf{U} = \mathbf{0}$  and  $\mathbf{D}(\mathbf{q}^0) = \mathbf{V} - \mathbf{U} \Phi_q(\mathbf{q}^0) \mathbf{V} = \mathbf{V}$ ,

$$\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0 \quad (5.2.25)$$

An important property of  $\mathbf{D}(\mathbf{q})$  is obtained using Eq. (5.2.13), suppressing arguments,

$$\Phi_q \mathbf{D} = (\Phi_q - \Phi_q \mathbf{U} \mathbf{B} \Phi_q) \mathbf{V} = (\Phi_q - \Phi_q) \mathbf{V} = \mathbf{0} \quad (5.2.26)$$

This shows that  $\mathbf{D}$  is in the null space of the constraint Jacobian; i.e., its columns comprise a *continuously differentiable basis* for the *null space* of  $\Phi_q$ . Multiplying both sides of Eq. (5.2.23) on the left by the constraint Jacobian,  $\Phi_q \dot{\mathbf{q}} = \Phi_q \mathbf{D}\dot{\mathbf{v}} + \Phi_q \mathbf{U}\mathbf{B}\dot{\mathbf{f}} = \dot{\mathbf{f}}$ , so  $\dot{\mathbf{q}}$  of Eq. (5.2.23) satisfies the velocity constraint equation of Eq. (5.2.2), for all  $\mathbf{v} \in V_0$ ,  $t \in T_0$ , and  $\dot{\mathbf{v}}$ .

Differentiating Eq. (5.2.23) with respect to  $t$ ,

$$\ddot{\mathbf{q}} = \mathbf{D}\ddot{\mathbf{v}} + \left( \mathbf{D}(\mathbf{q})\ddot{\mathbf{v}} \right)_q \dot{\mathbf{q}} + \mathbf{U} \left( \mathbf{B}(\mathbf{q})\ddot{\mathbf{f}} \right)_q \dot{\mathbf{q}} + \mathbf{U}\ddot{\mathbf{B}}\dot{\mathbf{f}} \quad (5.2.27)$$

Writing Eq. (5.2.13) in the form  $\Phi_q \mathbf{U}\mathbf{B}\mathbf{c} = \mathbf{c}$ , for a constant vector  $\mathbf{c}$ , and differentiating with respect to  $\mathbf{q}$ ,  $\left( \Phi_q(\mathbf{q})\ddot{\mathbf{U}}\ddot{\mathbf{B}}\ddot{\mathbf{c}} \right)_q + \Phi_q \mathbf{U} \left( \mathbf{B}(\mathbf{q})\ddot{\mathbf{c}} \right)_q = \mathbf{0}$ . Using Eq. (5.2.13),

$$\left( \mathbf{B}(\mathbf{q})\ddot{\mathbf{c}} \right)_q = -\mathbf{B} \left( \Phi_q(\mathbf{q})\ddot{\mathbf{U}}\ddot{\mathbf{B}}\ddot{\mathbf{c}} \right)_q \quad (5.2.28)$$

Differentiating  $\mathbf{D}(\mathbf{q})\mathbf{a}$  of Eq. (5.2.24) with respect to  $\mathbf{q}$ , where  $\mathbf{a}$  is a constant vector and using Eq. (5.2.28),

$$\begin{aligned} \left( \mathbf{D}(\mathbf{q})\ddot{\mathbf{a}} \right)_q &= -\mathbf{U} \left( \mathbf{B}(\mathbf{q})\ddot{\Phi}_q \ddot{\mathbf{V}}\ddot{\mathbf{a}} \right)_q - \mathbf{U}\mathbf{B} \left( \Phi_q(\mathbf{q})\ddot{\mathbf{V}}\ddot{\mathbf{a}} \right)_q \\ &= \mathbf{U}\mathbf{B} \left( \Phi_q(\mathbf{q})\ddot{\mathbf{U}}\ddot{\mathbf{B}}\ddot{\Phi}_q \ddot{\mathbf{V}}\ddot{\mathbf{a}} \right)_q - \mathbf{U}\mathbf{B} \left( \Phi_q(\mathbf{q})\ddot{\mathbf{V}}\ddot{\mathbf{a}} \right)_q \\ &= -\mathbf{U}\mathbf{B} \left( \Phi_q(\mathbf{q}) \left( \ddot{\mathbf{V}}\ddot{\mathbf{a}} - \ddot{\mathbf{U}}\ddot{\mathbf{B}}\ddot{\Phi}_q \ddot{\mathbf{V}}\ddot{\mathbf{a}} \right) \right)_q \\ &= -\mathbf{U}\mathbf{B} \left( \Phi_q(\mathbf{q})\ddot{\mathbf{D}}\ddot{\mathbf{a}} \right)_q \end{aligned} \quad (5.2.29)$$

Substituting from Eqs. (5.2.28) and (5.2.29) into Eq. (5.2.27) and collecting terms,

$$\begin{aligned} \ddot{\mathbf{q}} &= \mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B} \left( \left( \Phi_q(\mathbf{q})\ddot{\mathbf{D}}\ddot{\mathbf{v}} \right)_q \dot{\mathbf{q}} - \ddot{\mathbf{f}} \right) = \mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B} \left( \left( \Phi_q(\mathbf{q})\ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}} - \ddot{\mathbf{f}} \right) \\ &= \mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\gamma \end{aligned} \quad (5.2.30)$$

where Eq. (5.2.3) has been used. Multiplying on the left by  $\Phi_q$  and using Eqs. (5.2.13) and (5.2.26),

$$\Phi_q \ddot{\mathbf{q}} = \Phi_q \mathbf{D}\ddot{\mathbf{v}} - \Phi_q \mathbf{U}\mathbf{B}\gamma = -\gamma \quad (5.2.31)$$

Thus,  $\ddot{\mathbf{q}}$  of Eq. (5.2.27) satisfies Eq. (5.2.3), for all  $\mathbf{v} \in V_0$ ,  $t \in T_0$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ .

The importance of the fact that generalized coordinates  $\mathbf{q}$  given by Eq. (5.2.15) satisfy the holonomic constraint of Eq. (5.2.1) and  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  given by Eqs. (5.2.23) and (5.2.30) satisfy the velocity and acceleration constraints of Eqs. (5.2.2) and (5.2.3) cannot be overemphasized. This aspect of the *tangent space formulation* is critical in accurately satisfying holonomic configuration, velocity, and acceleration constraints, avoiding so called *constraint drift* that is associated with many multibody dynamics formulations and their numerical implementations (Bauchau and Laulusa, 2008). The cost of achieving this important benefit is the computational

burden of carrying out iterative solutions for  $\mathbf{u}$  and  $\mathbf{B}$  in Eqs. (5.2.17) and (5.2.20). This computational cost will be assessed in numerical solution of the equations of motion.

Showing the functional relationships of Eqs. (5.2.15), (5.2.23), and (5.2.30),

$$\mathbf{q}(\mathbf{v}, t) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) \quad (5.2.32)$$

$$\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}))\dot{\mathbf{v}} + \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}), t)\dot{\mathbf{f}}(t) \quad (5.2.33)$$

$$\ddot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}, t))\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, t), t)\gamma(\mathbf{q}(\mathbf{v}, t), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t), t) \quad (5.2.34)$$

Eqs. (5.2.32) through (5.2.34) demonstrate that  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  are differentiable functions of  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  and provide mappings to determine  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  that satisfy Eqs. (5.2.1) through (5.2.3), once independent coordinates  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  are determined as a solution of some form of the equations of dynamics.

### 5.2.3 Tangent Space Kinematics with Time Independent Constraints

The case of time independent holonomic constraints of the form

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (5.2.35)$$

is common in applications. Since the foregoing analysis that accounts for time dependence in Eq. (5.2.1) is intricate, it is helpful to summarize results for constraints of the form of Eq. (5.2.35).

Clearly,  $\Phi_t(\mathbf{q}) = \mathbf{0}$ ,  $\Phi_{tq}(\mathbf{q}) = \mathbf{0}$ , and  $\Phi_{tt}(\mathbf{q}) = \mathbf{0}$ . Thus, the term  $\gamma$  in Eq. (5.2.3) reduces to

$$\gamma(\mathbf{q}, \dot{\mathbf{q}}) = \left( \Phi_q(\mathbf{q}, \hat{t}) \hat{\dot{\mathbf{q}}} \right)_q \dot{\mathbf{q}} \quad (5.2.36)$$

The matrix  $\mathbf{B}(\mathbf{q})$  of Eq. (5.2.13) is

$$\mathbf{B}(\mathbf{q}) = \left( \Phi_q(\mathbf{q}) \mathbf{U} \right)^{-1} \quad (5.2.37)$$

and the iterative algorithm for its evaluation remains as Eq. (5.2.20). The vector function of Eq. (5.2.14) reduces to

$$\mathbf{u} = \mathbf{h}(\mathbf{v}) \quad (5.2.38)$$

and the iterative algorithm for its evaluation remains as Eq. (5.2.17), with  $\mathbf{f}(t^0) = \mathbf{0}$ . The matrix  $\mathbf{D}(\mathbf{q})$  of Eq. (5.2.24) that appears often is

$$\mathbf{D}(\mathbf{q}) \equiv \left( \mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q}) \right) \mathbf{V} \quad (5.2.39)$$

and the results of Eqs (5.2.32) through (5.2.34) reduce to

$$\mathbf{q}(\mathbf{v}) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}) \quad (5.2.40)$$

$$\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}) = \mathbf{D}(\mathbf{q}(\mathbf{v}))\dot{\mathbf{v}} \quad (5.2.41)$$

$$\ddot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}) = \mathbf{D}(\mathbf{q}(\mathbf{v}))\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}))\gamma(\mathbf{q}(\mathbf{v}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}})) \quad (5.2.42)$$

Thus, results of the derivation of Section 5.2.2 for the common special case of Eq. (5.2.35) reduce gracefully to Eqs. (5.2.36) through (5.2.42).

### 5.2.4 ODE of Motion

Using the tangent space kinematics formulation, the *variational equations of motion* for holonomic systems of Section 4.6 are to be reduced to a system of ODE. The variational equation of motion for a multibody system of Eq. (4.6.16) is

$$\mathbf{q}^T \left( \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) \right) = 0 \quad (5.2.43)$$

which must hold for all  $\mathbf{q}$  that satisfy  $\Phi_q(\mathbf{q}) = \mathbf{0}$ , for a system with holonomic constraints of Eq. (5.2.1) that include Euler parameter normalization conditions for spatial bodies. As in Eq. (5.2.23), with time held fixed,

$$\delta\mathbf{q} = \mathbf{D}(\mathbf{q})\delta\mathbf{v} \quad (5.2.44)$$

Using Eqs. (5.2.42) and (5.2.44), Eq. (5.2.43) reduces to

$$\mathbf{v}^T \mathbf{D}^T(\mathbf{q}) \left( \mathbf{M}(\mathbf{q}) \mathbf{D}(\mathbf{q}) \ddot{\mathbf{v}} - \mathbf{M}(\mathbf{q}) \mathbf{U} \mathbf{B}(\mathbf{q}) \gamma(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) \right) = 0 \quad (5.2.45)$$

From Eqs. (5.2.26) and (5.2.44),

$$\Phi_q(\mathbf{q}) \mathbf{q} = \Phi_q(\mathbf{q}) \mathbf{D}(\mathbf{q}) \delta\mathbf{v} = \mathbf{0} \quad (5.2.46)$$

for arbitrary  $\delta\mathbf{v}$ . Thus, Eq. (5.2.45) holds for arbitrary  $\delta\mathbf{v}$  and, suppressing arguments,

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} = \mathbf{D}^T \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A \right) \quad (5.2.47)$$

where Eqs. (5.2.32) and (5.2.33) show that  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  are functions of  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ ; i.e.,

$$\begin{aligned} \mathbf{M}(\mathbf{q}(\mathbf{v}, t)) &= \mathbf{M}(\mathbf{v}, t) \\ \gamma(\mathbf{q}(\mathbf{v}, t), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t), t) &= \gamma(\mathbf{v}, \dot{\mathbf{v}}, t) \\ \mathbf{S}(\mathbf{q}(\mathbf{v}, t), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t), t) &= \mathbf{S}(\mathbf{v}, \dot{\mathbf{v}}, t) \\ \mathbf{Q}^A(\mathbf{q}(\mathbf{v}, t), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t), t) &= \mathbf{Q}^A(\mathbf{v}, \dot{\mathbf{v}}, t) \\ \mathbf{B}(\mathbf{q}(\mathbf{v}, t)) &= \mathbf{B}(\mathbf{v}, t) \\ \mathbf{D}(\mathbf{q}(\mathbf{v}, t)) &= \mathbf{D}(\mathbf{v}, t) \end{aligned} \quad (5.2.48)$$

Equation (5.2.47) is thus a *second order ODE* in  $\mathbf{v}$ .

At an initial configuration  $(\mathbf{q}^0, t^0)$  that satisfies Eq. (5.2.1),  $\Phi_q(\mathbf{q}^0) \mathbf{V} = \mathbf{0}$ , so  $\mathbf{D}^0 = \mathbf{D}(\mathbf{q}^0) = \mathbf{V} - \mathbf{U} \mathbf{B}(\mathbf{q}^0) \Phi_q(\mathbf{q}^0) \mathbf{V} = \mathbf{V}$ . Thus,  $\mathbf{D}^{0T} \mathbf{M}(\mathbf{q}^0) \mathbf{D}^0 = \mathbf{V}^T \mathbf{M}(\mathbf{q}^0) \mathbf{V}$ . Since the columns of  $\mathbf{V}$  form a basis for the null space of  $\Phi_q(\mathbf{q}^0)$ , on which  $\mathbf{M}(\mathbf{q}^0)$  is positive definite, the symmetric *reduced mass matrix*  $\mathbf{D}^T(\mathbf{v}, t) \mathbf{M}(\mathbf{v}, t) \mathbf{D}(\mathbf{v}, t)$  of Eq. (5.2.47), which is a continuous function of  $\mathbf{v}$  and  $t$ , is *positive definite*, hence nonsingular, in a neighborhood of  $(\mathbf{v}^0, t^0)$ . Multiplying both sides of Eq. (5.2.47) by the inverse of  $\mathbf{D}^T \mathbf{M} \mathbf{D}$ , suppressing arguments of functions involved for simplicity, yields the traditional form of a second order ODE,

$$\ddot{\mathbf{v}} = (\mathbf{D}^T \mathbf{M} \mathbf{D})^{-1} \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) \equiv \mathbf{g}(\mathbf{v}, \dot{\mathbf{v}}, t) \quad (5.2.49)$$

Given initial values of  $\mathbf{q}(t^0)$  and  $\dot{\mathbf{q}}(t^0)$  that satisfy Eqs. (5.2.1) and (5.2.2) at  $t^0$ , the first of Eqs. (5.2.10) and Eq. (5.2.25) yield initial conditions on  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ ,

$$\begin{aligned}\mathbf{v}(t^0) &= \mathbf{0} \\ \dot{\mathbf{v}}(t^0) &= \mathbf{V}^T \dot{\mathbf{q}}(t^0)\end{aligned}\quad (5.2.50)$$

Assuming all functions appearing in Eq. (5.2.48) are continuously differentiable with respect to their arguments, it is shown in Section 4.7.3 that the initial-value problem of Eqs. (5.2.49) and (5.2.50) has a unique solution for  $\mathbf{v}(t)$  in a neighborhood of  $(\mathbf{v}^0, t^0)$ . It is shown in Section 3.6 that this solution can be continued, as shown schematically in Fig. 5.2.2, until a singular configuration or the desired final time  $t_f$  is encountered. Further, if terms in Eq. (5.2.48) are  $k$ -times differentiable with respect to design parameters  $\mathbf{b}$ , so is the solution. With these existence, uniqueness, and differentiability properties, the initial-value problem of Eqs. (5.2.49) and (5.2.50) is *well posed*.

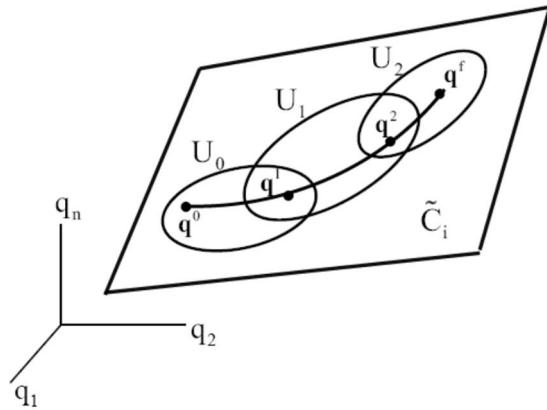


Figure 5.2.2 Continuation of Solution on Constraint Manifold

It is important to note that Eq. (5.2.47) is a *second order ODE*. It is not a DAE such as Eq. (4.10.8) that is encountered if Lagrange multipliers are introduced, nor is it a differential equation on a manifold (Hairer, Lubich, and Wanner, 2006). The tangent space formulation thus avoids analytic and computational difficulties associated with both of these classes of equations. It is interesting to note that the ODE formulation has been derived using d'Alembert's principle, without introducing Lagrange multipliers. It is shown in Section 5.11 that the tangent space ODE formulation is equivalent to variational and DAE formulations, all of which are well-posed.

Unlike the analytical reduction for the spinning Top of Section 5.1, numerical reduction of the variational equations of motion to ODE depends on tolerances used in approximating  $\mathbf{u}$  and  $\mathbf{B}$  in Eqs. (5.2.17) and (5.2.20). Thus, after numerical solution of reduced equations of motion for  $\mathbf{v}$  and its derivatives, Eqs. (5.2.36) through (5.2.38) yield  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  that satisfy the constraints of Eqs. (5.2.1) through (5.2.3). More specifically, the accuracy with which the configuration generalized coordinates  $\mathbf{q}$  are evaluated depends on the accuracy with which  $\mathbf{u}$  is computed in Eq. (5.2.19); i.e.,  $\text{utol}$ . As long as convergence in this iteration is achieved, the accuracy with which  $\mathbf{B}$  is computed in Eq. (5.2.22) is immaterial. In contrast, the accuracy of  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  that are computed in Eqs. (5.2.37) and (5.2.38) is directly impacted by the accuracy with

which  $\mathbf{B}$  is computed in Eq. (5.2.23); i.e.,  $\text{Btol}$ . Thus, both  $\text{utol}$  and  $\text{Btol}$  control the accuracy with which the velocity and acceleration generalized coordinates are evaluated.

Finally, it is noted that with the unique solution  $\mathbf{v}(t)$  and its derivatives known, Eqs. (5.2.32) through (5.2.34) determine the unique solution  $\mathbf{q}(t)$  and its derivatives of the multibody dynamics problem. As emphasized earlier, this solution satisfies all three forms of the holonomic constraint equations of Eqs. (5.2.1) through (5.2.3). Thus, problems associated with *drift from constraints* that arise in numerical solution of the DAE of multibody dynamics (Bauchau and Laulusa, 2008) are no longer of concern.

### 5.2.5 Calculation of Constraint Reaction Forces

A note is in order regarding the coupling of kinematics and dynamics. The Lagrange multiplier form of equations of motion of Eq. (4.10.8), with Euler parameter normalization conditions included in the holonomic constraint equations  $\Phi(\mathbf{q}) - \mathbf{f}(t) = \mathbf{0}$ , is

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q})\lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0} \quad (5.2.51)$$

It is shown in Section 4.10.4 that, once a solution for  $\mathbf{q}(t)$  and  $\lambda(t)$  is known, the *Lagrange multiplier*  $\lambda$  determines constraint reaction forces and is obtained by factoring the matrix  $\Phi_q \Phi_q^T$  at each solution time. A much more efficient calculation may be made in the tangent space formulation. Multiplying Eq. (5.2.51) by  $\mathbf{U}^T = \Phi_q^0$  yields

$$\mathbf{U}^T \Phi_q^T(\mathbf{q}) \lambda = \mathbf{U}^T (-\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) \quad (5.2.52)$$

Writing Eq. (5.2.13) in the form  $\Phi_q(\mathbf{q})\mathbf{U}\mathbf{B}(\mathbf{q}) = \mathbf{I}$ , the transpose is  $\mathbf{B}^T(\mathbf{q})(\mathbf{U}^T \Phi_q^T(\mathbf{q})) = \mathbf{I}$ . Thus,  $\mathbf{B}^T(\mathbf{q})$  is the inverse of the coefficient matrix of  $\lambda$  in Eq. (5.2.52) and, by a simple multiplication,

$$\lambda = \mathbf{B}^T(\mathbf{q}, t) \mathbf{U}^T (-\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) \quad (5.2.53)$$

This calculation can be performed at each time step in numerical integration of the equations of motion to inexpensively determine *constraint reaction forces*.

Broadly applicable tangent space generalized coordinates are shown to satisfy all three forms of kinematic constraint, for arbitrary values of independent coordinates. The inverse  $\mathbf{B}$  of the holonomic constraint subJacobian with respect to dependent coordinates is efficiently updated for evaluation of dependent coordinates to prescribed accuracy. It also provides for efficient calculation of numerous quantities in this and subsequent sections of the chapter. Finally, it enables efficient evaluation of constraint reaction forces in the Lagrange multiplier form of equations of motion derived in Section 4.10.

Efficient means of updating the tangent space parameterization yield a family of local parameterizations, or charts, that enable systematic continuation of solution trajectories, as outlined in Section 3.6. Since a global parameterization is generally not possible with a single set of independent generalized coordinates, the tangent space approach is among the only options available to obtain a globally valid ODE formulation for mechanical system dynamics.

## Key Formulas

$$\bar{\Phi}(\mathbf{q}, t)\Phi(\mathbf{q}) - \mathbf{f}(t) = \mathbf{0} \quad \Phi_q(\mathbf{q})\dot{\mathbf{q}} = \mathbf{f}_t(\mathbf{q}, t) \quad \Phi_q(\mathbf{q})\ddot{\mathbf{q}} = -\gamma \quad (5.2.1) \quad (5.2.2) \quad (5.2.3)$$

$$\mathbf{U} \equiv \Phi_q^T(\mathbf{q}^0) \quad \Phi_q(\mathbf{q}^0)\mathbf{V} = \mathbf{0} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I} \quad \mathbf{U}^T\mathbf{V} = \mathbf{0} \quad (5.2.5) \quad (5.2.6)$$

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u} \quad (5.2.8)$$

$$\begin{aligned} \mathbf{u} &= \mathbf{h}(\mathbf{v}, t) & \Delta\mathbf{u}^i &= \mathbf{B}(\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i) - \mathbf{f}(t_0)) & i &= 1, 2, \dots & (5.2.14) & (5.2.17) \\ & & \mathbf{u}^{i+1} &= \mathbf{u}^i + \Delta\mathbf{u}^i & & & & \end{aligned}$$

$$\begin{aligned} \mathbf{B} &\equiv (\Phi_q \mathbf{U})^{-1} & \Delta\mathbf{B}^i &= -\mathbf{B}^i \Phi_q \mathbf{U} \mathbf{B}^i + \mathbf{B}^i & i &= 1, 2, \dots & (5.2.13) & (5.2.20) \\ & & \mathbf{B}^{i+1} &= \mathbf{B}^i + \Delta\mathbf{B}^i & & & & \end{aligned}$$

$$\mathbf{q}(\mathbf{v}, t) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) \quad \mathbf{D}(\mathbf{q}) \equiv (\mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q}))\mathbf{V} \quad (5.2.32) \quad (5.2.24)$$

$$\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}, t))\dot{\mathbf{v}} + \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, t))\mathbf{f}_t(t) \quad (5.2.33)$$

$$\ddot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}, t))\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, t))\gamma(\mathbf{q}(\mathbf{v}, t), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t), t) \quad (5.2.34)$$

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} = \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) \quad (5.2.47)$$

$$\mathbf{v}(t^0) = \mathbf{0} \quad \dot{\mathbf{v}}(t^0) = \mathbf{V}^T \dot{\mathbf{q}}(t^0) \quad (5.2.50)$$

## 5.3 Numerical Solution of Tangent Space ODE

Explicit Runge-Kutta numerical integration algorithms are presented in Section 5.3.1, using results of Section 4.8, for solution of the ODE initial-value problem of Section 5.2. Derivatives required to implement implicit integration methods of Section 4.8 are derived in Sections 5.3.2 and 5.3.3 and implicit numerical integration algorithms are presented in Sections 5.3.4 through 5.3.6 for solution of the ODE initial-value problem, including computation of third derivatives for use in error control. The occurrence of stiff behavior in some multibody dynamics applications dictates that any modern multibody dynamics formulation must support the implicit integration option.

### 5.3.1 Explicit Numerical Integration Algorithm

The process of explicit numerical integration of the ODE initial-value problem of Eqs. (5.2.49) and (5.2.50), repeated here as

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} = \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) \quad (5.3.1)$$

$$\begin{aligned} \mathbf{v}(t^0) &= \mathbf{0} \\ \dot{\mathbf{v}}(t^0) &= \mathbf{V}^T \dot{\mathbf{q}}^0 \end{aligned} \quad (5.3.2)$$

requires evaluation of all terms in Eq. (5.3.1), including  $\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) = \left( \Phi_q(\mathbf{q}) \ddot{\dot{\mathbf{q}}} \right)_q \dot{\mathbf{q}} + \ddot{\mathbf{f}}(t)$  that is defined in Eq. (5.2.3). Evaluation of the dominant term  $\mathbf{P}_2(\mathbf{q}, \dot{\mathbf{q}}) = \left( \Phi_q(\mathbf{q}) \ddot{\dot{\mathbf{q}}} \right)_q$  for planar and spatial constraints is carried out using differentiation results presented in Appendix 5.B. Once a solution for  $\mathbf{v}(t)$  and its time derivatives is obtained, Eqs. (5.2.33) through (5.2.35), repeated here as

$$\mathbf{q}(\mathbf{v}, t) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) \quad (5.3.3)$$

$$\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}, t))\dot{\mathbf{v}} + \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, t))\dot{\mathbf{f}}(t) \quad (5.3.4)$$

$$\ddot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}, t))\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, t))\gamma(\mathbf{q}(\mathbf{v}, t), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t), t) \quad (5.3.5)$$

are used to recover the solution  $\mathbf{q}(t)$  and its time derivatives.

**Explicit numerical integration** of the initial-value problem of Eqs. (5.3.1) and (5.3.2), using Runge-Kutta methods, is as follows:

- (1) Define initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t^0$  that satisfy kinematic configuration and velocity constraints of Eqs. (5.2.1) and (5.2.2). Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0)$ , matrices  $\mathbf{U}$  and  $\mathbf{V}$  of Eqs. (5.2.5) and (5.2.6), and  $\gamma$  of Eq. (5.2.3). Obtain initial conditions on  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  from Eqs. (5.3.2) for integration of Eq. (5.3.1).
- (2) Solve Eq. (5.3.1) for  $\ddot{\mathbf{v}}$  and apply an explicit numerical integrator of Section 4.8.2.1 to proceed on a time grid with step size  $h$ , yielding  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  on the time grid. Use Eqs. (5.3.3) through (5.3.5) to evaluate  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  on the time grid.

(3) Monitor the *condition numbers* of  $\Phi_q \mathbf{U}$  and  $\mathbf{D}^T \mathbf{M} \mathbf{D}$ , the *norm of  $\mathbf{v}$* , and the number of iterations required to evaluate  $\mathbf{u}$  and  $\mathbf{B}$  in Eqs. (5.2.17) and (5.2.20). If tolerances are satisfied, continue the process. If a tolerance is exceeded, define  $\bar{\tau}^0$  as the current time and associated  $\bar{\mathbf{q}}^0$  and  $\bar{\dot{\mathbf{q}}}^0$ . Repeat calculations in Step (1) to define a new parameterization and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\dot{\mathbf{v}}}^0$  of Eq. (5.3.2) to restart integration. This process follows the trajectory shown in Fig. 5.2.2, moving smoothly across *charts* in the *regular configuration space*.

(4) Continue the process until the final time  $tf$  is reached, or a singular configuration associated with a faulty design or model occurs.

### 5.3.2 Derivatives for Implicit Integration

To use an *implicit numerical integration* method in solving the initial-value problem of Eqs. (5.3.1) and (5.3.2), derivatives of all terms that appear with respect to  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  are required. For the reader who is satisfied with the use of explicit numerical integration methods, the somewhat intricate derivation of these terms in the remainder of this section can be bypassed.

Since arguments of functions that appear in Eq. (5.3.1) are  $\mathbf{q}(\mathbf{v}, t)$  and  $\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t)$ , derivatives of  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  with respect to  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  are required. The partial derivative of Eq. (5.3.3) with respect to  $\mathbf{v}$  is

$$\mathbf{q}_v = \mathbf{V} - \mathbf{U} \mathbf{h}_v \quad (5.3.6)$$

To evaluate  $\mathbf{h}_v$ , the derivative of Eq. (5.2.11) with respect to  $\mathbf{v}$ , with  $\mathbf{u} = \mathbf{h}$ , is  $\Phi_q (\mathbf{V} - \mathbf{U} \mathbf{h}_v) = \mathbf{0}$ , so  $\mathbf{h}_v = \mathbf{B} \Phi_q \mathbf{V}$ . Substituting this into Eq. (5.3.6),

$$\mathbf{q}_v = \mathbf{V} - \mathbf{U} \mathbf{B} \Phi_q \mathbf{V} = \mathbf{D} \quad (5.3.7)$$

Differentiating Eq. (5.3.4) with respect to  $\dot{\mathbf{v}}$ ,

$$\dot{\mathbf{q}}_{\dot{\mathbf{v}}} = \mathbf{D} \quad (5.3.8)$$

and with respect to  $\mathbf{v}$ , using Eq. (5.3.6),

$$\dot{\mathbf{q}}_v = \left( (\mathbf{D} \ddot{\mathbf{v}})_q + \mathbf{U} (\mathbf{B} \ddot{\mathbf{f}})_q \right) \mathbf{D} \quad (5.3.9)$$

Recall the definition of Eq. (3.1.19); i.e.,

$$\mathbf{P}2(\mathbf{q}, \chi) \equiv \left( \Phi_q(\mathbf{q}) \ddot{\chi} \right)_q \quad (5.3.10)$$

where  $\chi \in \mathbb{R}^{ngc}$  and Eqs. (5.2.24), (5.2.28), and (5.2.29),

$$\mathbf{D}(\mathbf{q}) \equiv \left( \mathbf{I} - \mathbf{U} \mathbf{B}(\mathbf{q}) \Phi_q(\mathbf{q}) \right) \mathbf{V} \quad (5.3.11)$$

$$(\mathbf{B}(\mathbf{q}) \ddot{\mathbf{e}})_q = -\mathbf{B} \left( \Phi_q(\mathbf{q}) \ddot{\mathbf{U}} \ddot{\mathbf{B}} \ddot{\mathbf{e}} \right)_q \quad (5.3.12)$$

$$(\mathbf{D}(\mathbf{q}) \ddot{\mathbf{a}})_q = -\mathbf{U} \mathbf{B} \left( \Phi_q(\mathbf{q}) \ddot{\mathbf{D}} \ddot{\mathbf{a}} \right)_q \quad (5.3.13)$$

Using these relations and Eq. (5.3.4), Eq. (5.3.9) may be written as

$$\begin{aligned}\dot{\mathbf{q}}_v &= \left( -\mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, \mathbf{D}\dot{\mathbf{v}}) - \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, \mathbf{U} \mathbf{B} \dot{\mathbf{f}}) \right) \mathbf{D} = -\mathbf{U} \mathbf{B} \left( \mathbf{P}2(\mathbf{q}, \mathbf{D}\dot{\mathbf{v}} + \mathbf{U} \mathbf{B} \dot{\mathbf{f}}) \right) \mathbf{D} \\ &= -\mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{D}\end{aligned}\quad (5.3.14)$$

where linearity of  $\mathbf{P}2(\mathbf{q}, \cdot)$  in its second argument and Eq. (5.3.4) have been used.

From Eq. (5.2.3),

$$\begin{aligned}\gamma_q &= \left( (\Phi_q(\mathbf{q}) \ddot{\mathbf{q}})_q \ddot{\mathbf{q}} \right)_q \\ \gamma_{\dot{\mathbf{q}}} &= \left( (\Phi_q(\mathbf{q}) \ddot{\mathbf{q}})_q \dot{\mathbf{q}} \right)_{\dot{\mathbf{q}}}\end{aligned}\quad (5.3.15)$$

Using matrix product expansion,

$$\begin{aligned}\left( \Phi_q \ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}}^2 &= \sum_j \left( \sum_i \Phi_{q_i} \ddot{q}_i^1 \right)_{q_j} \ddot{q}_j^2 = \sum_j \left( \sum_i \Phi_{q_i q_j} \dot{q}_i^1 \right) \dot{q}_j^2 \\ &= \sum_j \left( \sum_i \Phi_{q_j q_i} \dot{q}_j^2 \right) \dot{q}_i^1 = \sum_i \left( \sum_j \Phi_{q_j} \ddot{q}_j^2 \right)_{q_i} \dot{q}_i^1 = \left( \Phi_q \ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}}^1\end{aligned}$$

so the quantity  $\left( (\Phi_q \ddot{\mathbf{q}})_q \dot{\mathbf{q}} \right)$  is symmetric in its  $\dot{\mathbf{q}}$  arguments. Thus,

$$\left( \left( \Phi_q \ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}} \right)_{\dot{\mathbf{q}}} = 2 \left( \Phi_q \ddot{\mathbf{q}} \right)_q = 2 \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \quad (5.3.16)$$

Define the triple derivative with respect to  $\mathbf{q}$  in the first of Eqs. (5.3.15) as

$$\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) \equiv \left( \left( \Phi_q(\mathbf{q}) \ddot{\mathbf{q}} \right)_q \ddot{\mathbf{q}} \right)_q = \left( \left( \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} \right)_q \ddot{\mathbf{q}} \right)_q \quad (5.3.17)$$

Substituting Eqs. (5.3.16) and (5.3.17) into Eqs. (5.3.5) and (5.3.15), with the definition of Eq. (5.3.10),

$$\begin{aligned}\gamma &= \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \ddot{\mathbf{f}} \\ \gamma_q &= \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) \\ \gamma_{\dot{\mathbf{q}}} &= 2 \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\end{aligned}\quad (5.3.18)$$

Writing Eq. (5.2.13) in the form  $\mathbf{B} \Phi_q \mathbf{U} = \mathbf{I}$ , taking its transpose, and multiplying on the right by a vector  $\mathbf{e}$  yields  $\mathbf{U}^T \Phi_q^T \mathbf{B}^T \mathbf{e} = \mathbf{e}$ . Differentiating both sides with respect to  $\mathbf{q}$ , with  $\mathbf{e}$  constant,

$$\mathbf{U}^T \left( \Phi_q^T(\mathbf{q}) \ddot{\mathbf{e}} \right)_q + \mathbf{U}^T \Phi_q^T \left( \mathbf{B}^T(\mathbf{q}) \ddot{\mathbf{e}} \right)_q = \mathbf{0} \quad (5.3.19)$$

The transpose of Eq. (5.2.13) is  $\mathbf{B}^T = (\mathbf{U}^T \Phi_q^T)^{-1}$ . Using this relation in Eq. (5.3.19) and defining

$$\mathbf{P}4(\mathbf{q}, \dot{\mathbf{q}}) \equiv \left( \Phi_q^T(\mathbf{q}) \ddot{\mathbf{q}} \right)_q \quad (5.3.20)$$

where  $\eta \in R^{n_{hc}}$ , Eq. (5.3.19) yields

$$(\mathbf{B}^T(\mathbf{q})\ddot{\mathbf{e}})_q = -\mathbf{B}^T \mathbf{U}^T \left( \Phi_q^T(\mathbf{q}) \dot{\mathbf{B}}^T \ddot{\mathbf{e}} \right)_q = -\mathbf{B}^T \mathbf{U}^T \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{e}) \quad (5.3.21)$$

Taking the transpose of Eq. (5.3.11) and multiplying on the right by a constant vector  $\mathbf{d}$  yields  $\mathbf{D}^T \mathbf{d} = \mathbf{V}^T \mathbf{d} - \mathbf{V}^T \Phi_q^T \mathbf{B}^T \mathbf{U}^T \mathbf{d}$ . Differentiating and using Eqs. (5.3.19) and (5.3.20),

$$\begin{aligned} (\mathbf{D}^T(\mathbf{q})\ddot{\mathbf{d}})_q &= -\mathbf{V}^T \left( \left( \Phi_q^T(\mathbf{q}) \dot{\mathbf{B}}^T \mathbf{U}^T \ddot{\mathbf{d}} \right)_q + \Phi_q^T \left( \mathbf{B}^T(\mathbf{q}) \mathbf{U}^T \ddot{\mathbf{d}} \right)_q \right) \\ &= -\mathbf{V}^T \left( \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T \mathbf{d}) - \Phi_q^T \mathbf{B}^T \mathbf{U}^T \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T \mathbf{d}) \right) \\ &= -\mathbf{V}^T \left( \mathbf{I} - \Phi_q^T \mathbf{B}^T \mathbf{U}^T \right) \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T \mathbf{d}) \\ &= -\mathbf{D}^T \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T \mathbf{d}) \end{aligned} \quad (5.3.22)$$

The differential equation of Eq. (5.3.1) may be written in residual form as

$$\mathbf{R}(\ddot{\mathbf{v}}, \dot{\mathbf{v}}, \mathbf{v}) \equiv \mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} - \mathbf{D}^T \left( \mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A \right) = \mathbf{0} \quad (5.3.23)$$

Numerical integration, using an implicit ODE integration formula, requires the derivatives  $\mathbf{R}_v$ ,  $\mathbf{R}_{\dot{v}}$ , and  $\mathbf{R}_{\ddot{v}}$ . Since terms on the right of Eq. (5.3.23), with the exception of  $\ddot{\mathbf{v}}$ , depend on  $\mathbf{q}(\mathbf{v}, t)$  and  $\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t)$ , the *chain rule of differentiation* may be used with the foregoing identities to calculate the needed derivatives.

The derivative of the first term on the right of Eq. (5.3.23) with respect to  $\mathbf{q}$  is

$$\left( \mathbf{D}^T(\mathbf{q}) \mathbf{M}(\mathbf{q}) \mathbf{D}(\mathbf{q}) \ddot{\mathbf{v}} \right)_q = \left( \mathbf{D}^T(\mathbf{q}) \ddot{\mathbf{M}} \ddot{\mathbf{D}} \ddot{\mathbf{v}} \right)_q + \mathbf{D}^T \left( \mathbf{M}(\mathbf{q}) \ddot{\mathbf{D}} \ddot{\mathbf{v}} \right)_q + \mathbf{D}^T \mathbf{M} \left( \mathbf{D}(\mathbf{q}) \ddot{\mathbf{v}} \right)_q \quad (5.3.24)$$

Since  $\mathbf{M}(\mathbf{q})$  is symmetric and using Eq. (5.3.13),

$$\left( \mathbf{D}^T(\mathbf{q}) \ddot{\mathbf{M}} \ddot{\mathbf{D}} \ddot{\mathbf{v}} \right)_q = \left( \dot{\mathbf{D}}^T \ddot{\mathbf{M}} \ddot{\mathbf{D}}(\mathbf{q}) \ddot{\mathbf{v}} \right)_q = \mathbf{D}^T \mathbf{M} \left( \mathbf{D}(\mathbf{q}) \ddot{\mathbf{v}} \right)_q = -\mathbf{D}^T \mathbf{M} \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, \mathbf{D} \ddot{\mathbf{v}})$$

Defining

$$\mathbf{M}2(\mathbf{q}, \boldsymbol{\mu}) \equiv \left( \mathbf{M}(\mathbf{q}) \ddot{\boldsymbol{\mu}} \right)_q \quad (5.3.25)$$

where  $\boldsymbol{\mu} \in R^{n_{gc}}$ , Eq. (5.3.24) reduces to

$$\left( \mathbf{D}^T(\mathbf{q}) \mathbf{M}(\mathbf{q}) \mathbf{D}(\mathbf{q}) \ddot{\mathbf{v}} \right)_q = -2 \mathbf{D}^T \mathbf{M} \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, \mathbf{D} \ddot{\mathbf{v}}) + \mathbf{D}^T \mathbf{M}2(\mathbf{q}, \mathbf{D} \ddot{\mathbf{v}}) \quad (5.3.26)$$

Using the foregoing derivative formulas and suppressing arguments, derivatives of the second term on the right of Eq. (5.3.23) with respect to  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  are

$$\begin{aligned}
& \left( \mathbf{D}^T(\mathbf{q}) \left( \mathbf{M}(\mathbf{q}) \mathbf{U} \mathbf{B}(\mathbf{q}) \gamma(\mathbf{q}, \ddot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \ddot{\mathbf{q}}) + \mathbf{Q}^A(\mathbf{q}, \ddot{\mathbf{q}}, t) \right) \right)_q \\
&= \left( \mathbf{D}^T(\mathbf{q}) \left( \widehat{\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A} \right) \right)_q + \mathbf{D}^T \left( \mathbf{M}(\mathbf{q}) \mathbf{U} \mathbf{B}(\mathbf{q}) \gamma(\mathbf{q}, \ddot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \ddot{\mathbf{q}}) + \mathbf{Q}^A(\mathbf{q}, \ddot{\mathbf{q}}, t) \right)_q \quad (5.3.27) \\
&= -\mathbf{D}^T \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A)) \\
&\quad + \mathbf{D}^T \left( \mathbf{M}2(\mathbf{q}, \mathbf{U} \mathbf{B} \gamma) - \mathbf{M} \mathbf{U} \mathbf{B} \left( \mathbf{P}2(\mathbf{q}, \mathbf{U} \mathbf{B} \gamma) - \gamma_q \right) + \mathbf{S}_q + \mathbf{Q}_q^A \right)
\end{aligned}$$

and

$$\mathbf{D}^T \left( \mathbf{M}(\ddot{\mathbf{q}}) \mathbf{U} \mathbf{B}(\mathbf{q}) \gamma(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, t) + \mathbf{S}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}) + \mathbf{Q}^A(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, t) \right)_{\dot{\mathbf{q}}} = \mathbf{D}^T \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma_{\dot{\mathbf{q}}} + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A \right) \quad (5.3.28)$$

With the results of Eqs. (5.3.26), (5.3.27), (5.3.28), (5.3.6), (5.3.8), and (5.3.13), and suppressing arguments,

$$\begin{aligned}
\mathbf{R}_{\ddot{\mathbf{v}}} &= \mathbf{D}^T \mathbf{M} \mathbf{D} \\
\mathbf{R}_{\dot{\mathbf{v}}} &= -\mathbf{D}^T \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma_{\dot{\mathbf{q}}} + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A \right) \mathbf{D} \\
\mathbf{R}_v &= \mathbf{D}^T \left\{ \begin{array}{l} -\mathbf{M} \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, 2\mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \gamma) + \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A)) \\ + \mathbf{M}2(\mathbf{q}, \mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \gamma) - \mathbf{M} \mathbf{U} \mathbf{B} \gamma_q - \mathbf{S}_q - \mathbf{Q}_q^A \\ + (\mathbf{M} \mathbf{U} \mathbf{B} \gamma_{\dot{\mathbf{q}}} + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A) \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \end{array} \right\} \mathbf{D} \quad (5.3.29)
\end{aligned}$$

### 5.3.3 Organization of Derivative Calculation

The foregoing derivative calculations are expanded, as in Section 3.1, for constraints  $\Phi(\mathbf{q}_i, \mathbf{q}_j) - \mathbf{f}(t) = \mathbf{0}$  between pairs of bodies. From Eq. (5.3.17),

$$\mathbf{P}3(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \left( \mathbf{P}2(\mathbf{q}_{ij}, \ddot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \right)_{\dot{\mathbf{q}}_{ij}} = \left[ \left( \mathbf{P}2(\mathbf{q}_{ij}, \ddot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \right)_{\mathbf{q}_i} \quad \left( \mathbf{P}2(\mathbf{q}_{ij}, \ddot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \right)_{\mathbf{q}_j} \right] \quad (5.3.30)$$

and, from Eqs. (3.1.16) and (3.1.21),

$$\begin{aligned}
\left( \mathbf{P}2(\mathbf{q}_{ij}, \ddot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \right)_{\mathbf{q}_i} &= \left( \left( \Phi_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i \right)_{\dot{\mathbf{q}}_{ij}} + 2 \left( \left( \Phi_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j \right)_{\dot{\mathbf{q}}_{ij}} + \left( \left( \Phi_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} \dot{\mathbf{q}}_i \right)_{\dot{\mathbf{q}}_{ij}} \quad (5.3.31) \\
\left( \mathbf{P}2(\mathbf{q}_{ij}, \ddot{\mathbf{q}}_{ij}) \dot{\mathbf{q}}_{ij} \right)_{\mathbf{q}_j} &= \left( \left( \Phi_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_i \right)_{\dot{\mathbf{q}}_{ij}} + 2 \left( \left( \Phi_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j \right)_{\dot{\mathbf{q}}_{ij}} + \left( \left( \Phi_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \dot{\mathbf{q}}_j \right)_{\dot{\mathbf{q}}_{ij}}
\end{aligned}$$

To evaluate the operator of Eq. (5.3.20),

$$\mathbf{P}4(\mathbf{q}_{ij}, \eta) = \left( \Phi_{\mathbf{q}_{ij}}^T \ddot{\mathbf{q}} \right)_{\dot{\mathbf{q}}_{ij}} = \left[ \begin{array}{c} \left( \Phi_{\mathbf{q}_i}^T \ddot{\mathbf{q}} \right)_{\mathbf{q}_i} \\ \left( \Phi_{\mathbf{q}_j}^T \ddot{\mathbf{q}} \right)_{\mathbf{q}_j} \end{array} \right] = \left[ \begin{array}{cc} \left( \Phi_{\mathbf{q}_i}^T \ddot{\mathbf{q}} \right)_{\mathbf{q}_i} & \left( \Phi_{\mathbf{q}_j}^T \ddot{\mathbf{q}} \right)_{\mathbf{q}_j} \\ \left( \Phi_{\mathbf{q}_j}^T \ddot{\mathbf{q}} \right)_{\mathbf{q}_i} & \left( \Phi_{\mathbf{q}_j}^T \ddot{\mathbf{q}} \right)_{\mathbf{q}_j} \end{array} \right] \quad (5.3.32)$$

where four derivative expressions on the right are identified. Expanding matrix products and derivatives, with row index m and column index n,

$$\left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_j} = \sum \left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_j} = \left[ \sum \left(\Phi_{q_{im}}^T \ddot{\eta}\right)_{q_{jn}} \right] = \left[ \sum \Phi_{q_{im} q_{jn}}^T \ddot{\eta} \right] = \left[ \sum \left(\Phi_{q_{jn}}^T \ddot{\eta}\right)_{q_{im}} \right] \quad (5.3.33)$$

Interchanging row and column indices m and n yields the transpose relation

$$\left( \left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_j} \right)^T = \left[ \sum \left(\Phi_{q_{jm}}^T \ddot{\eta}\right)_{q_{in}} \right] = \sum \left(\Phi_{q_j}^T \ddot{\eta}\right)_{q_i} = \left(\Phi_{q_j}^T \ddot{\eta}\right)_{q_i} \quad (5.3.34)$$

Thus, only three of the four terms on the right of Eq. (5.3.32) are independent; i.e.,

$$\mathbf{P4}(q_{ij}, \eta) = \left(\Phi_{q_{ij}}^T \ddot{\eta}\right)_{q_{ij}} = \begin{bmatrix} \left(\Phi_{q_i}^T \ddot{\eta}\right) \\ \left(\Phi_{q_j}^T \ddot{\eta}\right) \end{bmatrix}_{q_{ij}} = \begin{bmatrix} \left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_i} & \left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_j} \\ \left(\left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_j}\right)^T & \left(\Phi_{q_j}^T \ddot{\eta}\right)_{q_j} \end{bmatrix} \quad (5.3.35)$$

The first equality in Eq. (5.3.33) provides a helpful identity in evaluating terms in  $\mathbf{P4}(q_{ij}, \eta)$  of Eq. (5.3.35). For constraints that are comprised of a column of building block constraint expressions  $\Phi^\alpha(q_i, q_j) - f^\alpha(t) = \mathbf{0}$  and associated factors  $\eta_\alpha$ , entries in the matrix of Eq. (5.3.35) are comprised of the sum of terms that are calculated for individual building block constraints; i.e.,

$$\left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_j} = \sum \left(\Phi_{q_i}^T \ddot{\eta}\right)_{q_j} \quad (5.3.36)$$

Evaluation of kinetic derivatives is less intricate. Since

$$\mathbf{M}(q) = \text{diag}(\mathbf{M}_1(q_1), \dots, \mathbf{M}_{nb}(q_{nb})) \quad (5.3.37)$$

is block diagonal in terms associated with individual bodies,

$$\mathbf{M2}(q, \dot{q}) = \text{diag}\left(\left(\mathbf{M}_1 \dot{q}^1\right)_{q_1}, \dots, \left(\mathbf{M}_{nb} \dot{q}^{nb}\right)_{q_{nb}}\right) \quad (5.3.38)$$

Similarly, but less structured in generalized coordinate dependence, the virtual work definition of *generalized applied force*,

$$\delta W^A = \sum_i \mathbf{q}_i^T \mathbf{Q}_i^A(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (5.3.39)$$

identifies components of generalized force with bodies, but each component may depend on the full range of generalized coordinates and velocities. Thus,

$$\mathbf{Q}_q^A = \begin{bmatrix} \mathbf{Q}_{1q}^A \\ \vdots \\ \mathbf{Q}_{nbq}^A \end{bmatrix} \quad \mathbf{Q}_{\dot{q}}^A = \begin{bmatrix} \mathbf{Q}_{1\dot{q}}^A \\ \vdots \\ \mathbf{Q}_{nbd\dot{q}}^A \end{bmatrix} \quad (5.3.40)$$

where nb is the number of bodies in the system. Velocity coupling terms are associated with individual bodies and their generalized coordinates and velocities; i.e.,  $\mathbf{S}^i = \mathbf{S}^i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ . Thus,

$$\begin{aligned} \mathbf{S}_q &= \text{diag}\left(\mathbf{S}_{q_1}^1, \dots, \mathbf{S}_{q_{nb}}^{nb}\right) \\ \mathbf{S}_{\dot{q}} &= \text{diag}\left(\mathbf{S}_{\dot{q}_1}^1, \dots, \mathbf{S}_{\dot{q}_{nb}}^{nb}\right) \end{aligned} \quad (5.3.41)$$

Derivatives of Eqs. (5.3.30), (5.3.31), and (5.3.32) for constraints that are defined in Sections 3.2 and 3.3 are summarized in Appendix 5.B. Derivatives of kinetic terms in Eqs. (5.3.38) and (5.3.41) and derivatives of generalized forces are presented in the same appendix.

### 5.3.4 Trapezoidal Implicit Numerical Integration

*Second order trapezoidal integration formulas* of Eq. (4.8.40) for  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  are

$$\begin{aligned} \mathbf{v}_n &= \mathbf{v}_{n-1} + h\dot{\mathbf{v}}_{n-1} + (h^2/4)(\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \\ \dot{\mathbf{v}}_n &= \dot{\mathbf{v}}_{n-1} + (h/2)(\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \end{aligned} \quad (5.3.42)$$

Equations (5.2.15) and (5.2.23) are used to evaluate  $\mathbf{q}_n$  and  $\dot{\mathbf{q}}_n$ , where  $\mathbf{h}_n = \mathbf{u}_n$  is evaluated using Eq. (5.2.17) and  $\mathbf{B}(\mathbf{q}_n)$  is evaluated using Eq. (5.2.20). The residual  $\mathbf{R}$  of Eq. (5.3.23) is thus a function of only  $\ddot{\mathbf{v}}_n$  and the chain rule of differentiation yields the Jacobian of the trapezoidal residual with respect to  $\ddot{\mathbf{v}}_n$ ,

$$\mathbf{J}^{\text{trap}} = \frac{d\mathbf{R}}{d\ddot{\mathbf{v}}_n} = \mathbf{R}_{\ddot{\mathbf{v}}} + \mathbf{R}_{\dot{\mathbf{v}}} \frac{\partial \dot{\mathbf{v}}_n}{\partial \ddot{\mathbf{v}}_n} + \mathbf{R}_{\mathbf{v}} \frac{\partial \mathbf{v}_n}{\partial \ddot{\mathbf{v}}_n} = \mathbf{R}_{\ddot{\mathbf{v}}} + (h/2)\mathbf{R}_{\dot{\mathbf{v}}} + (h^2/4)\mathbf{R}_{\mathbf{v}} \quad (5.3.43)$$

With initial conditions  $\mathbf{q}(t^0) = \mathbf{q}^0$  and  $\dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0$  that satisfy  $\Phi(\mathbf{q}^0) - \mathbf{f}(t^0) = \mathbf{0}$  and  $\Phi_q(\mathbf{q}^0)\dot{\mathbf{q}}^0 = \mathbf{f}_t(t^0)$ , initial values  $\mathbf{v}^0 = \mathbf{0}$  and  $\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0$  are obtained from Eqs. (5.3.2). To obtain an estimate of  $\ddot{\mathbf{v}}(t^0)$  that is needed to begin iterative solution of the residual equation, Eq. (5.3.23) may be solved with given initial conditions for  $\ddot{\mathbf{v}}(t^0)$ . In subsequent time steps, the estimate  $\ddot{\mathbf{v}}_n^0 = \ddot{\mathbf{v}}_{n-1}$  is used.

To carry out Newton-Raphson iteration for  $\ddot{\mathbf{v}}_n$ ,

$$\begin{aligned} \mathbf{J}^{\text{trap}}(\ddot{\mathbf{v}}_{n-1})\Delta\ddot{\mathbf{v}}_n^i &= -\mathbf{R}(\ddot{\mathbf{v}}_n^i) \quad i = 0, 1, \dots \text{ until } \|\mathbf{R}(\ddot{\mathbf{v}}_n^i)\| \leq \text{intol} \\ \ddot{\mathbf{v}}_n^{i+1} &= \ddot{\mathbf{v}}_n^i + \Delta\ddot{\mathbf{v}}_n^i \end{aligned} \quad (5.3.44)$$

where intol is the *numerical solution error tolerance*.

Since a precise Jacobian is not required in Newton iteration, criteria based on the number of iterations to achieve convergence in Eq. (5.3.44) may be used to determine when the relatively costly process required to compute a new value of the integration Jacobian  $\mathbf{J}^{\text{trap}}$  is carried out. If the integration Jacobian is to be updated, partial derivatives of the residual in Eqs. (5.3.29) must be evaluated and the Jacobian of Eq. (5.3.43) computed.

Much as  $\mathbf{B} = (\Phi_q(\mathbf{q})\mathbf{U})^{-1}$  is evaluated at  $\mathbf{q}^0$  as  $\mathbf{B}^0 = (\mathbf{U}^T \mathbf{U})^{-1}$  and iteratively updated using the Newton-Raphson method applied to  $(\Phi_q(\mathbf{q})\mathbf{U})\mathbf{B}(\mathbf{q}) = \mathbf{I}$  in Eq. (5.2.22), the inverse of the Jacobian may be evaluated at each time step. At  $t^0$ ,  $\mathbf{J}_{\text{inv}}^0 \equiv \mathbf{J}(\mathbf{q}(t^0))^{-1}$  may be evaluated and  $\mathbf{J}_{\text{inv}}$

updated at time step  $t_n$  as the solution of  $\mathbf{J}\mathbf{J}_{\text{inv}} = \mathbf{I}$ , where  $\mathbf{J} = \mathbf{J}(\mathbf{q}(t_n))$ , using Newton-Raphson iteration with the estimate  $\mathbf{J}_{\text{inv}}^0 = \mathbf{J}_{\text{inv}}(\mathbf{q}(t_{n-1}))$ ; i.e., with  $\mathbf{R} = \mathbf{J}\mathbf{J}_{\text{inv}} - \mathbf{I}$  and  $\Delta\mathbf{J}_{\text{inv}}^j = \mathbf{J}_{\text{inv}}^{j+1} - \mathbf{J}_{\text{inv}}^j$ ,

$$\mathbf{J}\Delta\mathbf{J}_{\text{inv}}^j = -(\mathbf{J}\mathbf{J}_{\text{inv}}^j - \mathbf{I}), j = 0, 1, \dots \quad (5.3.45)$$

Since  $\mathbf{J}^{-1} \approx \mathbf{J}_{\text{inv}}^j$ ,  $\Delta\mathbf{J}_{\text{inv}}^j = -\mathbf{J}_{\text{inv}}^j (\mathbf{J}\mathbf{J}_{\text{inv}}^j - \mathbf{I}) = \mathbf{J}_{\text{inv}}^j - \mathbf{J}_{\text{inv}}^j \mathbf{J}\mathbf{J}_{\text{inv}}^j$ , and

$$\mathbf{J}_{\text{inv}}^{j+1} = 2\mathbf{J}_{\text{inv}}^j - \mathbf{J}_{\text{inv}}^j \mathbf{J}\mathbf{J}_{\text{inv}}^j, j = 0, 1, \dots \text{ until } \|\mathbf{J}\mathbf{J}_{\text{inv}}^j - \mathbf{I}\| \leq \text{intol} \quad (5.3.46)$$

With  $\mathbf{J}_{\text{inv}} \approx (\mathbf{J}^{\text{trap}}(\ddot{\mathbf{v}}_n))^{-1}$  so determined, Eq. (5.3.43) yields  $\Delta\ddot{\mathbf{v}}_n^i = -\mathbf{J}_{\text{inv}} \mathbf{R}(\ddot{\mathbf{v}}_n^i)$ , or

$$\ddot{\mathbf{v}}_n^{i+1} = \ddot{\mathbf{v}}_n^i - \mathbf{J}_{\text{inv}} \mathbf{R}(\ddot{\mathbf{v}}_n^i), i = 1, \dots, \text{ until } \|\mathbf{R}(\ddot{\mathbf{v}}_n^i)\| \leq \text{intol} \quad (5.3.47)$$

To implement the integration error control approach presented in Section 4.8.5, in the present case,

$$\bar{\mathbf{v}}_n = \mathbf{v}_{n-1} + h\dot{\mathbf{v}}_{n-1} + (h^2/2)\ddot{\mathbf{v}}_{n-1} + (h^3/6)\dddot{\mathbf{v}}_{n-1}$$

the third derivative  $\ddot{\mathbf{v}}_n$  must be computed. For systems in which the variable  $t$  does not appear explicitly, Eq. (5.3.1) may be differentiated to obtain

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}}_{n-1} = -(\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}})_q \dot{\mathbf{q}}_{n-1} + (\mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A))_q \dot{\mathbf{q}}_{n-1} + \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A)_q \ddot{\mathbf{q}}_{n-1}$$

where all terms are evaluated at  $t_{n-1}$ . Substituting from Eqs. (5.3.26) and (5.3.28) and collecting terms, this yields

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}}_{n-1} = \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma_q + \mathbf{S}_q + \mathbf{Q}_q^A) \ddot{\mathbf{q}}_{n-1} + \bar{\bar{\mathbf{A}}} \dot{\mathbf{q}}_{n-1} \equiv \mathbf{Rhs} \quad (5.3.48)$$

where

$$\bar{\bar{\mathbf{A}}} \equiv \mathbf{D}^T \begin{bmatrix} \mathbf{M} \mathbf{U} \mathbf{B} \gamma_2 (\mathbf{q}, 2\mathbf{D} \ddot{\mathbf{v}}_{n-1} - \mathbf{U} \mathbf{B} \gamma) - \mathbf{M} \gamma_2 (\mathbf{q}, \mathbf{D} \ddot{\mathbf{v}}_{n-1} - \mathbf{U} \mathbf{B} \gamma) \\ - \mathbf{P} \gamma_4 (\mathbf{q}, \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A)) + \mathbf{M} \mathbf{U} \mathbf{B} \gamma_q + \mathbf{S}_q + \mathbf{Q}_q^A \end{bmatrix}$$

While there are numerous terms in Eq. (5.3.48), all must be evaluated in each iteration of Eq. (5.3.44), so the only cost in solving for  $\ddot{\mathbf{v}}_{n-1}$  is the solution of Eq. (5.3.48). This is not a significant additional cost, as indicated in Section 4.8.5.

In addition to providing information for error control, the third derivative  $\ddot{\mathbf{v}}_{n-1}$  enables improved accuracy of the estimate for the solution variable  $\ddot{\mathbf{v}}_n$ , namely

$$\ddot{\mathbf{v}}_n \approx \ddot{\mathbf{v}}_{n-1} + h\ddot{\mathbf{v}}_{n-1} \quad (5.3.49)$$

### 5.3.5 Runge-Kutta Implicit Numerical Integration

Stage i of the Runge-Kutta method of Eq. (4.8.27) is applied to Eq. (5.3.1) to obtain

$$\begin{aligned}
\mathbf{v}'_i &\equiv \mathbf{v}_n + hc_i \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^{i-1} A_{ij} \mathbf{k}_j \\
\dot{\mathbf{v}}'_i &\equiv \dot{\mathbf{v}}_n + h \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j \\
\mathbf{k}_i &= \mathbf{g}\left(t_n + c_i h, \mathbf{v}'_i + h^2 A_{ii} \mathbf{k}_i, \dot{\mathbf{v}}'_i + ha_{ii} \mathbf{k}_i\right) \\
&= \left(\left(\mathbf{D}^T \mathbf{M} \mathbf{D}\right)^{-1} \mathbf{D}^T \left(\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A\right)\right) \left(\mathbf{v}'_i + h^2 A_{ii} \mathbf{k}_i, \dot{\mathbf{v}}'_i + ha_{ii} \mathbf{k}_i, t_n + c_i h\right)
\end{aligned} \tag{5.3.50}$$

where  $\mathbf{q}_i$  and  $\dot{\mathbf{q}}_i$  are evaluated as outlined for the trapezoidal method. Multiplying both sides on the left by  $\mathbf{D}^T \mathbf{M} \mathbf{D}$  yields the residual form of the third of Eqs. (5.3.50),

$$\begin{aligned}
\mathbf{R} &\equiv \left(\left(\mathbf{D}^T \mathbf{M} \mathbf{D}\right)(\mathbf{v}'_i + h^2 A_{ii} \mathbf{k}_i, t_n + c_i h)\right) \mathbf{k}_i \\
&\quad - \left(\mathbf{D}^T \left(\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A\right)\right) (\mathbf{v}'_i + h^2 A_{ii} \mathbf{k}_i, \dot{\mathbf{v}}'_i + ha_{ii} \mathbf{k}_i, t_n + c_i h) = \mathbf{0}
\end{aligned} \tag{5.3.51}$$

The Jacobian of Eq. (5.3.51) is thus

$$\mathbf{J}^{RK} = \frac{d\mathbf{R}}{dk_i} = \mathbf{R}_{\dot{\mathbf{v}}} + ha_{ii} \mathbf{R}_{\mathbf{v}} + h^2 A_{ii} \mathbf{R}_{\mathbf{k}} \tag{5.3.52}$$

With initial conditions  $\mathbf{q}(t^0) = \mathbf{q}^0$  and  $\dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0$  that satisfy  $\Phi(\mathbf{q}^0, t^0) = \mathbf{0}$  and  $\Phi_q(\mathbf{q}^0, t^0) \dot{\mathbf{q}}^0 = -\Phi_t(\mathbf{q}^0, t^0)$ , initial values  $\mathbf{v}^0 = \mathbf{0}$  and  $\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0$  are obtained from Eqs. (5.3.2). To obtain an estimate for  $\mathbf{k}_i^0$  that is needed to begin iterative solution of the residual equation, Eq. (5.3.51) may be solved with initial conditions and  $h = 0$ . Newton-Raphson iteration for  $\mathbf{k}_i$  is then carried out as in Eq. (5.3.44); i.e., at time  $t_i$ , with an estimate  $\mathbf{k}_i^0 = \mathbf{k}_{i-1}$ ,

$$\begin{aligned}
\mathbf{J}^{RK}(\mathbf{v}_{n-1}) \Delta \mathbf{k}_i^j &= -\mathbf{R}(\mathbf{k}_i^j) \quad j = 0, 1, \dots \text{ until } \|\mathbf{R}(\mathbf{k}_i^{j+1})\| \leq \text{intol} \\
\mathbf{k}_i^{j+1} &= \mathbf{k}_i^j + \Delta \mathbf{k}_i^j
\end{aligned} \tag{5.3.53}$$

where intol is a *solution error tolerance*. To evaluate  $\mathbf{R}(\mathbf{k}_i^j)$  of Eq. (5.3.53), Eqs. (4.8.28) are used to obtain  $\mathbf{v}_n^i$  and  $\dot{\mathbf{v}}_n^i$ , Eq. (5.2.17) is solved for  $\mathbf{u}_n^i$ , Eq. (5.3.3) is used to obtain  $\mathbf{q}_n^i$ , Eq. (5.2.25) is used to update  $\mathbf{B}$ , and Eqs. (5.3.4) and (5.3.5) are used to determine  $\dot{\mathbf{q}}_i^n$  and  $\ddot{\mathbf{q}}_i^n$ .

Since a precise Jacobian is not required in Newton iteration, criteria based on the number of iterations to achieve convergence in Eq. (5.3.53) may be used to determine when the relatively costly process required to compute a new value of the integration Jacobian  $\mathbf{J}$  is justified. If the integration Jacobian  $\mathbf{J}$  is to be updated, partial derivatives of the residual in Eqs. (5.3.29) must be evaluated and the Jacobian of Eq. (5.3.52) computed. As in the trapezoidal algorithm of Section 5.3.4, the inverse of  $\mathbf{J}^{RK}$  can be iteratively evaluated. This leads to the modified iterative algorithm of Eq. (5.3.47). The integration error control and step size algorithm of Eqs. (4.8.35) to (4.8.37) in Section 4.8 may be used in implementing the Runge-Kutta method.

### 5.3.6 Implicit Numerical Integration Algorithm

**Implicit numerical integration** of the initial-value problem of Eqs. (5.3.1) and (5.3.2), using trapezoidal and Runge-Kutta methods, is as follows:

- (1) Define initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t^0$  that satisfy kinematic configuration and velocity constraints. Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0)$  and matrices  $\mathbf{U}$  and  $\mathbf{V}$  in Eqs. (5.2.5) and (5.2.6). Evaluate initial conditions  $\mathbf{v}^0 = \mathbf{0}$  and  $\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0$ .
- (2) Apply an implicit numerical integrator to proceed stepwise on a time grid with step size  $h$ , using a factored form of the integration Jacobian of Eq. (5.3.43) or (5.3.52) to iteratively determine  $\ddot{\mathbf{v}}_n$  or  $\mathbf{k}_n$ . Use Eq. (5.3.42) or (4.8.28) to determine  $\mathbf{v}_n$  and  $\dot{\mathbf{v}}_n$ . Use Eqs. (5.3.3) through (5.3.5) to evaluate  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ , and  $\ddot{\mathbf{q}}_n$  on the time grid.
- (3) Monitor the *condition numbers* of  $\Phi_q \mathbf{U}$  and  $\mathbf{D}^T \mathbf{M} \mathbf{D}$ , as measures of *ill conditioning* of the equations. If a condition number exceeds a limit, say  $10^3$ , define a new time  $\bar{t}^0$  and associated  $\bar{\mathbf{q}}^0$ . Repeat calculations in Step (1) to define a new parameterization and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\mathbf{v}}^0$ , and restart the simulation. As secondary indicators of the need to reparameterize, monitor the number of Newton-Raphson iterations required in Step 2, the norm of  $\mathbf{v}$ , and the number of iterations required to evaluate  $\mathbf{u}$  and  $\mathbf{B}$ .
- (4) Continue the process until the final time  $tf$  is reached, or a singularity is encountered due to a faulty design or model.

ODE of motion in independent tangent space coordinates are integrated using explicit Runge-Kutta numerical integration methods presented in Section 4.8, with minimal coding effort.

In order to support implicit numerical integration, derivatives of terms in the ODE of motion are evaluated. While the formulas derived for this purpose are intricate, they have been systematically programmed to implement the desired implicit numerical integration method. Any modern multibody dynamics formulation must support the implicit integration option.

### Key Formulas

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} = \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) \quad \mathbf{v}(t^0) = \mathbf{0} \quad \dot{\mathbf{v}}(t^0) = \mathbf{V}^T \dot{\mathbf{q}}^0 \quad (5.3.1) \quad (5.3.2)$$

$$\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) \equiv (\Phi_q \ddot{\boldsymbol{\chi}})_q \quad \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) \equiv \left( \left( \Phi_q(\mathbf{q}) \ddot{\mathbf{q}} \right)_q \ddot{\mathbf{q}} \right)_q \quad (5.3.10) \quad (5.3.17)$$

$$\mathbf{P}4(\mathbf{q}, \boldsymbol{\eta}) \equiv (\Phi_q^T \ddot{\boldsymbol{\eta}})_q \quad \mathbf{M}2(\mathbf{q}, \boldsymbol{\mu}) \equiv (\mathbf{M} \ddot{\boldsymbol{\mu}})_q \quad (5.3.20) \quad (5.3.25)$$

$$\gamma = \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{f}_{tt}$$

$$\gamma_q = \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) \quad (5.3.18)$$

$$\gamma_{\dot{q}} = 2\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})$$

$$\mathbf{R}(\ddot{\mathbf{v}}, \dot{\mathbf{v}}, \mathbf{v}) \equiv \mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} - \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0} \quad (5.3.23)$$

$$\begin{aligned} \mathbf{R}_{\ddot{\mathbf{v}}} &= \mathbf{D}^T \mathbf{M} \mathbf{D} \quad \mathbf{R}_{\dot{\mathbf{v}}} = -\mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma}_{\dot{\mathbf{q}}} + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A) \mathbf{D} \\ \mathbf{R}_{\mathbf{v}} &= \mathbf{D}^T \left\{ \begin{array}{l} -\mathbf{M} \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, 2\mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\boldsymbol{\gamma}) + \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A)) \\ + \mathbf{M}2(\mathbf{q}, \mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\boldsymbol{\gamma}) - \mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma}_{\mathbf{q}} - \mathbf{S}_{\mathbf{q}} - \mathbf{Q}_{\mathbf{q}}^A \\ + (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma}_{\dot{\mathbf{q}}} + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A) \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \end{array} \right\} \mathbf{D} \end{aligned} \quad (5.3.29)$$

$$\mathbf{J}^{\text{trap}} = \frac{d\mathbf{R}}{d\ddot{\mathbf{v}}_n} = \mathbf{R}_{\ddot{\mathbf{v}}} + \mathbf{R}_{\dot{\mathbf{v}}} \frac{\partial \dot{\mathbf{v}}_n}{\partial \ddot{\mathbf{v}}_n} + \mathbf{R}_v \frac{\partial \mathbf{v}_n}{\partial \ddot{\mathbf{v}}_n} = \mathbf{R}_{\ddot{\mathbf{v}}} + (h/2)\mathbf{R}_{\dot{\mathbf{v}}} + (h^2/4)\mathbf{R}_v \quad (5.3.43)$$

$$\mathbf{J}^{\text{RK}} = \frac{d\mathbf{R}}{d\mathbf{k}_i} = \mathbf{R}_{\ddot{\mathbf{v}}} + h a_{ii} \mathbf{R}_{\dot{\mathbf{v}}} + h^2 A_{ii} \mathbf{R}_v \quad (5.3.52)$$

## 5.4 Numerical Examples with Tangent Space ODE

Four systems, one planar and three spatial, are simulated using the tangent space ODE solution algorithms of Section 5.3. The effectiveness of *constraint error control* mechanisms that are embedded in the formulation is verified. All functions that appear in the equations of motion and derivatives required for numerical integration are explicitly derived. MATLAB computer codes that implement the tangent space formulation and numerical solution for each of the examples are provided in Appendix 5.A. Numerical integration is carried out with Nystrom4, RK4, Kutta3/8, and RKFN explicit integrators and Trapezoidal and SDIRK54 implicit integrators.

### 5.4.1 Planar Double Pendulum

The *planar double pendulum* shown in Fig. 5.4.1 is made up of a pair of uniform bars of length two units each, moving in a vertical x-y plane, with gravity  $g$  acting in the negative y-direction. Revolute joints at the origin of the x-y frame and at points  $(1,0)$  and  $(-1,0)$  in the  $x'_1-y'_1$  and  $x'_2-y'_2$  centroidal body reference frames, respectively, define constraints. Generalized coordinates for the system are

$$\mathbf{q} = [x_1 \quad y_1 \quad \phi_1 \quad x_2 \quad y_2 \quad \phi_2]^T \quad (5.4.1)$$

For simplicity, inertia parameters are taken as  $m_1 = m_2 = 1 \text{ kg}$ ,  $J'_1 = J'_2 = 0.3 \text{ kg}\cdot\text{m}^2$ , and  $g = 9.8 \text{ m/sec}^2$ . Since the body fixed reference frames are centroidal,  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ .

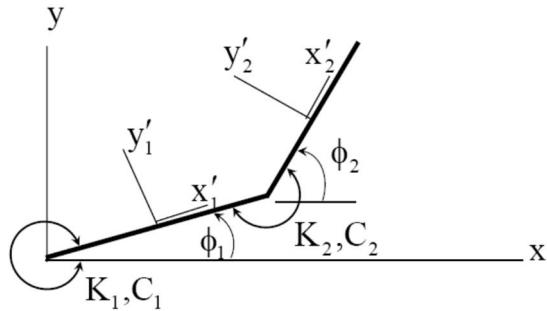


Figure 5.4.1 Planar Double Pendulum

*Torsional springs and dampers* act between ground and body one at the origin of the x-y inertial frame, with spring and damping coefficients  $K_1$  and  $C_1$ , and between bodies one and two at their common pivot point, with spring and damping coefficients  $K_2$  and  $C_2$ . In order that equilibrium is achieved with  $\phi_1 = \phi_2 = -\pi/2$ , the torque in the spring acting between ground and body one is  $K_1(\phi_1 + \pi/2)$  and the torque in the spring between bodies one and two is  $K_2(\phi_2 - \phi_1)$ . To derive *generalized forces* that act on the system due to the springs, dampers, and gravity, the virtual work done on the system by these forces is

$$\begin{aligned}
W &= -g y_1 - g y_2 + n_1 \phi_1 + n_2 \phi_2 - K_1(\phi_1 + \pi/2) \dot{\phi}_1 - C_1 \dot{\phi}_1 \dot{\phi}_1 \\
&\quad - K_2(\phi_2 - \phi_1)(\dot{\phi}_2 - \dot{\phi}_1) - C_2(\dot{\phi}_2 - \dot{\phi}_1)(\dot{\phi}_2 - \dot{\phi}_1) \\
&= -g y_1 - \left( -n_1 + K_1(\phi_1 + \pi/2) + C_1 \dot{\phi}_1 - K_2(\phi_2 - \phi_1) - C_2(\dot{\phi}_2 - \dot{\phi}_1) \right) \dot{\phi}_1 \quad (5.4.2) \\
&\quad - g y_2 - \left( -n_2 + K_2(\phi_2 - \phi_1) + C_2(\dot{\phi}_2 - \dot{\phi}_1) \right) \dot{\phi}_2 \\
&\equiv Q^A \cdot q
\end{aligned}$$

where  $n_1$  and  $n_2$  are externally applied torques on bodies one and two. Thus,

$$Q^A = \begin{bmatrix} 0 \\ -g \\ -\left( -n_1 + K_1(\phi_1 + \pi/2) + C_1 \dot{\phi}_1 - K_2(\phi_2 - \phi_1) - C_2(\dot{\phi}_2 - \dot{\phi}_1) \right) \\ 0 \\ -g \\ -\left( -n_2 + K_2(\phi_2 - \phi_1) + C_2(\dot{\phi}_2 - \dot{\phi}_1) \right) \end{bmatrix} \quad (5.4.3)$$

Constraint equations and their Jacobian are

$$\Phi(q) = \begin{bmatrix} x_1 - \cos \phi_1 \\ y_1 - \sin \phi_1 \\ x_1 + \cos \phi_1 - x_2 + \cos \phi_2 \\ y_1 + \sin \phi_1 - y_2 + \sin \phi_2 \end{bmatrix} = \mathbf{0} \quad (5.4.4)$$

$$\Phi_q(q) = \begin{bmatrix} 1 & 0 & \sin \phi_1 & 0 & 0 & 0 \\ 0 & 1 & -\cos \phi_1 & 0 & 0 & 0 \\ 1 & 0 & -\sin \phi_1 & -1 & 0 & -\sin \phi_2 \\ 0 & 1 & \cos \phi_1 & 0 & -1 & \cos \phi_2 \end{bmatrix} \quad (5.4.5)$$

The six nonzero terms in the  $4 \times 6$  matrix  $P2(q, \chi)$  of Eq. (5.3.10) are

$$\begin{aligned}
P2(q, \chi)_{13} &= \chi_{\phi_1} \cos \phi_1; P2(q, \chi)_{23} = \chi_{\phi_1} \sin \phi_1; P2(q, \chi)_{33} = -\chi_{\phi_1} \cos \phi_1; \\
P2(q, \chi)_{36} &= \chi_{\phi_2} \cos \phi_2; P2(q, \chi)_{43} = \chi_{\phi_1} \sin \phi_1; P2(q, \chi)_{46} = \chi_{\phi_2} \sin \phi_2;
\end{aligned} \quad (5.4.6)$$

These quantities are sufficient for explicit numerical integration.

For implicit numerical integration, the six nonzero terms in the  $4 \times 6$  matrix  $P3(q, \dot{q})$  of Eq. (5.3.17) are

$$\begin{aligned}
P3(q, \dot{q})_{13} &= -\dot{\phi}_1^2 \sin \phi_1; P3(q, \dot{q})_{23} = \dot{\phi}_1^2 \cos \phi_1; P3(q, \dot{q})_{33} = \dot{\phi}_1^2 \sin \phi_1 \\
P3(q, \dot{q})_{34} &= \dot{\phi}_2^2 \sin \phi_2; P3(q, \dot{q})_{43} = -\dot{\phi}_1^2 \cos \phi_1; P3(q, \dot{q})_{46} = -\dot{\phi}_2^2 \cos \phi_2
\end{aligned} \quad (5.4.7)$$

the two nonzero terms in the  $6 \times 6$  matrix  $\mathbf{P}4(\mathbf{q}, \boldsymbol{\eta})$  of Eq. (5.3.20) are

$$\begin{aligned}\mathbf{P}4(\mathbf{q}, \boldsymbol{\eta})_{33} &= (\eta_1 - \eta_3) \cos \phi_1 + (\eta_2 - \eta_4) \sin \phi_1 \\ \mathbf{P}4(\mathbf{q}, \boldsymbol{\eta})_{66} &= -(\eta_3 + \eta_4) \sin \phi_2\end{aligned}\quad (5.4.8)$$

and the four nonzero terms in the  $6 \times 6$   $\mathbf{Q}_q^A$  and  $\dot{\mathbf{Q}}_q^A$  matrices are

$$\begin{aligned}\mathbf{Q}_{q33}^A &= -K_1 - K_2; \mathbf{Q}_{q36}^A = K_2; \mathbf{Q}_{q63}^A = K_2; \mathbf{Q}_{q66}^A = -K_2 \\ \dot{\mathbf{Q}}_{q33}^A &= -C_1 - C_2; \dot{\mathbf{Q}}_{q36}^A = C_2; \dot{\mathbf{Q}}_{q63}^A = C_2; \dot{\mathbf{Q}}_{q66}^A = -C_2\end{aligned}\quad (5.4.9)$$

All other derivative terms are zero, so the matrices are sparse.

*Initial conditions* that satisfy position and velocity constraints are that both bodies hang vertically downward at  $t^0 = 0$ ,  $\mathbf{q}^0 = [0 \ -1 \ -\pi/2 \ 0 \ -3 \ -\pi/2]^T$  with velocity  $\dot{\mathbf{q}}^0 = [0 \ 0 \ 0 \ 10 \ 0 \ 10]^T$ ; i.e., body one is at rest and the angular velocity of body two is 10 rad/sec. Ten second simulations are carried out. The plot on the left of Fig. 5.4.2 with  $K_1 = K_2 = 20 \text{ N}\cdot\text{m}/\text{rad}$ , no damping, and no external torques was obtained with Code 5.4.1 of Appendix 5.A, using the *Nystrom4 explicit integrator* with a constant step size of 0.001 sec. Total energy for this conservative system was constant to nine decimal places. Seventy nine *reparameterizations* in 10,000 time steps (127 time steps per reparameterization) were required, based on a unit limit on the norm of  $\mathbf{v}$ . Timing results showed that less than one percent of CPU time was required for reparameterization. The plot on the right of Fig. 5.4.2 was obtained with the same data, except that damping was  $C_1 = C_2 = 10 \text{ N}\cdot\text{m}\cdot\text{sec}/\text{rad}$ . Clearly, energy is dissipated. Just 12 *reparameterizations* in 10,000 time steps (833 time steps per reparameterization) were required, based on a unit limit on the norm of  $\mathbf{v}$ . Comparable results were obtained with all six numerical integration methods.

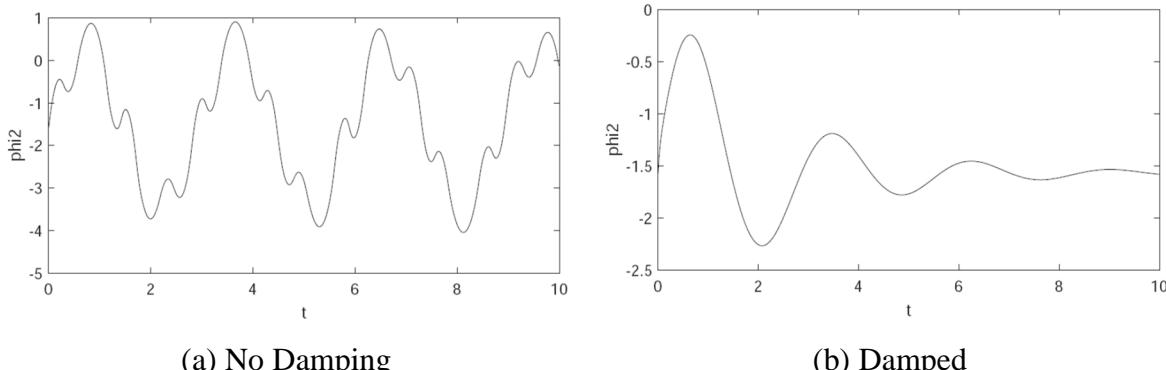


Figure 5.4.2 Rotation of Body 2 vs. Time

As a check on *constraint error control*, the convergence tolerance  $\text{Tol}=\text{utol}=\text{Btol}$  in Eqs. (5.2.17) and (5.2.20) was varied to evaluate its influence on constraint error. Results presented in Table 5.4.1, obtained with the damped simulation on the right of Fig. 5.4.2, show that maximum norms of position, velocity, and acceleration error over the simulation interval are driven toward

zero to computer precision, as convergence tolerances are tightened. Little increase in compute time was associated with the tighter tolerances.

Table 5.4.1 Maximum Constraint Error

Tol	Position Err.	Velocity Err.	Acceleration Err.
e-6	e-9	3e-11	3e-10
e-9	e-9	e-14	e-13
e-12	2e-14	3e-16	2e-15

To simulate dynamics of a stiff flexible bar, body one hangs vertically downward and body 2 is offset from the vertical by 0.01 radians to simulate an initial deformation of the bar; e.g., due to impact by a third body. The magnitude of stiffness and damping between the bars is varied to simulate transient dynamics due to the initial impact event. It is anticipated that transient response of the system will represent a stiff dynamic event as K and C are increased.

Initial conditions of the system are  $\mathbf{q}^0 = [0 \ -1 \ -\pi/2 \ 0.01 \ -3 \ -\pi/2 + 0.01]^T$  and  $\dot{\mathbf{q}}^0 = \mathbf{0}$ .

A sequence of values  $K = C = e3, e4, e5$ , and  $e6$  are imposed in simulations carried out with explicit and implicit integrators. As shown in Fig. 5.4.3, the SDIRK54 implicit integrator performed well for even quite *stiff system characteristics*. In contrast, the explicit variable time step explicit integrator RKF45 struggled with even modest stiffness and failed for the stiffest simulation performed.

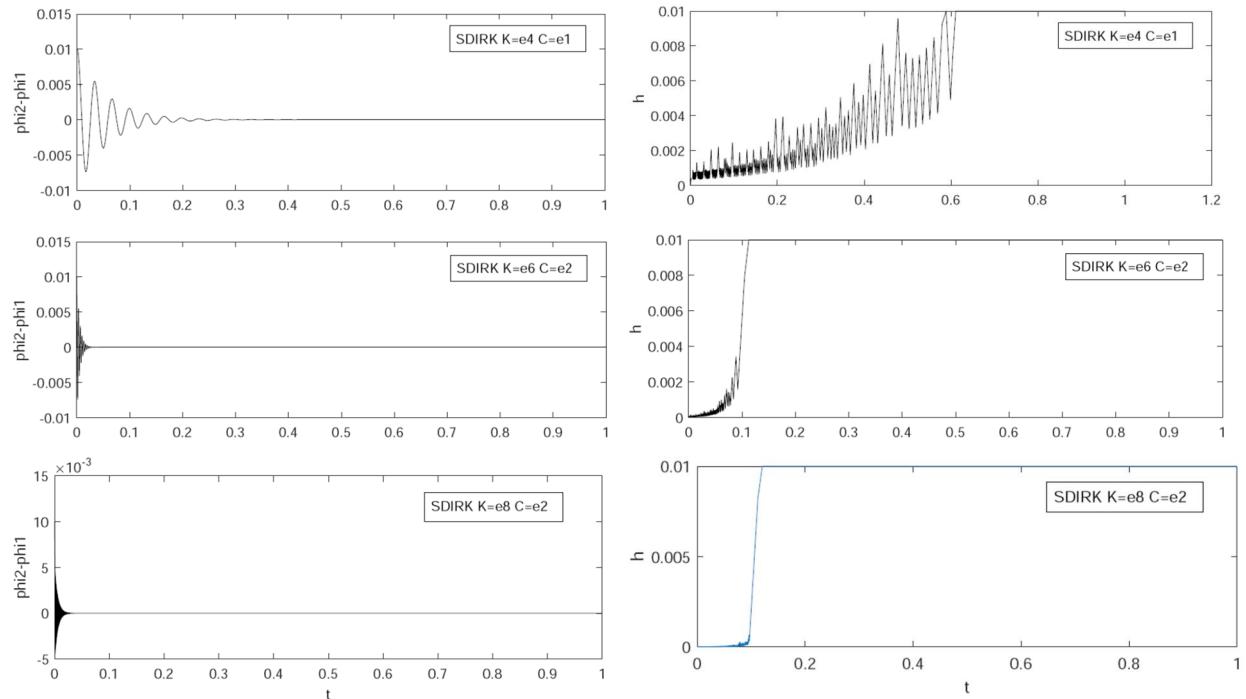


Figure 5.4.3 SDIRK54 Implicit Integrator Performance

### 5.4.2 Top with Tip Fixed

The *symmetric Top* presented as a special case to illustrate the tangent space concept in Section 5.1 is treated here as a constrained body, using the tangent space ODE formulation of Sections 5.2 and 5.3. Both the spin stabilized version studied in Section 5.1 and a transient dynamic variant used in the literature as a test problem are analyzed, using Codes 5.4.2a and 5.4.2b of Appendix 5.A.

#### 5.4.2.1 Spin Stabilized Top

The *spin stabilized Top* shown in Fig. 5.4.4 is constrained so that the tip is fixed at the origin of the inertial reference frame. A centroidal body fixed  $x'$ - $y'$ - $z'$  frame, with  $z'$  axis along the axis of symmetry of the Top and origin 1 m from the tip, is used to model the Top, so generalized coordinates include the vector  $\mathbf{r}$  from the origin of the global  $x$ - $y$ - $z$  frame to origin of the centroidal frame. As in Section 5.1, the Top is to be stabilized by an initial angular velocity  $\omega_{\text{gaz}0}$  about the body fixed  $z'$  axis. The Top is initially oriented so that body fixed and global axes are aligned; i.e.,  $\mathbf{r}^0 = \mathbf{u}_z$  and  $\mathbf{p}^0 = [1 \ 0 \ 0]^T$ . Initial angular velocity is

$\boldsymbol{\omega}'^0 = [10^{-12} \ 10^{-12} \ \omega_{\text{gaz}0}]^T$  rad/sec, where initial angular velocities of  $10^{-12}$  about the body fixed  $x'$  and  $y'$  axes are perturbations to cause *precession* of the Top relative to the global  $z$  axis. Using Eq. (2.4.44), the initial velocity of the centroid is  $\dot{\mathbf{r}}^0 = \mathbf{A}(\mathbf{p}^0)\tilde{\boldsymbol{\omega}}'^0\mathbf{u}'_z$  m/sec, where  $\mathbf{u}'_z$  is the unit vector along the positive  $z'$  axis. The mass of the Top is 30 kg and the inertia matrix relative to the centroidal  $x'$ - $y'$ - $z'$  frame is  $\mathbf{J}' = \text{diag}(90, 90, 30)$  kg·m<sup>2</sup>. Note that the parallel axis theorem of Section 4.4.2 has been used to transform the inertia matrix from the noncentroidal frame at the tip used in Section 5.1 to the centroidal frame used here. A gravitational acceleration of 9.8 m / sec<sup>2</sup> acts in the negative global  $z$  direction.

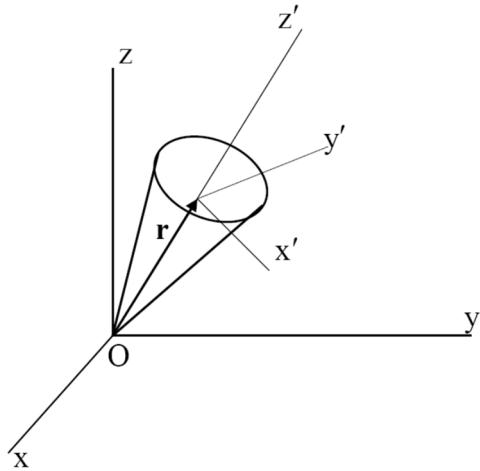


Figure 5.4.4 Symmetric Top with Tip Fixed

Kinematic and Euler parameter normalization constraints on *Cartesian generalized coordinates*  $\mathbf{q} = [\mathbf{r}^T \ \mathbf{p}^T]^T$  are

$$\Phi(\mathbf{q}) = \begin{bmatrix} \mathbf{r} - \mathbf{A}(\mathbf{p})\mathbf{u}'_z \\ (\mathbf{p}^T \mathbf{p} - 1)/2 \end{bmatrix} = \mathbf{0} \quad (5.4.10)$$

Derivatives of constraints required for analysis are

$$\begin{aligned} \Phi_q &= \begin{bmatrix} \mathbf{I}_3 & -\mathbf{B}(\mathbf{p}, \mathbf{u}'_z) \\ \mathbf{0} & \mathbf{p}^T \end{bmatrix} \\ \mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) &\equiv (\Phi_q \hat{\boldsymbol{\chi}})_q = \begin{bmatrix} \mathbf{0} & -\mathbf{B}(\boldsymbol{\chi}_p, \mathbf{u}'_z) \\ \mathbf{0} & \boldsymbol{\chi}_p^T \end{bmatrix} \\ \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) &\equiv \left( (\Phi_q \hat{\mathbf{q}})_q \right)_q = \mathbf{0} \\ \mathbf{P}4(\mathbf{q}, \boldsymbol{\eta}) &\equiv (\Phi_q^T \hat{\boldsymbol{\eta}})_q = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{K}(\mathbf{u}'_z, \boldsymbol{\eta}_k) + \boldsymbol{\eta}_p \mathbf{I}_4 \end{bmatrix} \end{aligned} \quad (5.4.11)$$

where  $\mathbf{B}(\mathbf{p}, \mathbf{u}'_z)$  is given by Eq. (2.6.25) and  $\mathbf{K}(\mathbf{a}', \mathbf{b})$  is given by Eq. (2.6.37). Derived terms in Eq. (5.3.18) are

$$\begin{aligned} \gamma &= \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ \gamma_q &= \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0} \\ \gamma_{\dot{q}} &= 2\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \end{aligned} \quad (5.4.12)$$

With the centroidal reference frame,

$$\begin{aligned} \mathbf{M}(\mathbf{q}) &= \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \end{bmatrix} \\ \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} \mathbf{0} \\ 8\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p} \end{bmatrix} \\ \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) &= \begin{bmatrix} -mg\mathbf{u}_z \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (5.4.13)$$

where  $\mathbf{G}(\mathbf{p})$  is given by Eq. (2.6.2). Kinetic derivatives that are required for analysis are

$$\mathbf{Q}_q^A = \mathbf{Q}_{\dot{q}}^A = \mathbf{0}$$

$$\begin{aligned}
\mathbf{M}2(\mathbf{q}, \boldsymbol{\mu}) \equiv (\mathbf{M}(\mathbf{q})\hat{\boldsymbol{\mu}})_q &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -4\mathbf{G}(\mathbf{p})^T \mathbf{J}'\mathbf{G}(\boldsymbol{\mu}_p) + 4\mathbf{T}(\mathbf{J}'\mathbf{G}(\mathbf{p})\boldsymbol{\mu}_p) \end{bmatrix} \\
\mathbf{S}_q &= \begin{bmatrix} \mathbf{0} \\ 8\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}'\mathbf{G}(\dot{\mathbf{p}}) \end{bmatrix} \\
\mathbf{S}_{\dot{q}} &= \begin{bmatrix} \mathbf{0} \\ -8\mathbf{G}(\dot{\mathbf{p}})\mathbf{J}'\mathbf{G}(\mathbf{p}) + 8\mathbf{T}(\mathbf{J}'\mathbf{G}(\dot{\mathbf{p}})\mathbf{p}) \end{bmatrix}
\end{aligned} \tag{5.4.14}$$

where  $\boldsymbol{\mu}_p$  is comprised of the last four components of the 7-vector  $\boldsymbol{\mu}$  and  $\mathbf{T}(\mathbf{a})$  is given by Eq. (2.6.20).

The residual form of the tangent space ODE of Eq. (5.3.1) is

$$\mathbf{R} \equiv \mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} - \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0} \tag{5.4.15}$$

For implicit integration, derivatives of the residual given by Eq. (5.3.29) are

$$\begin{aligned}
\mathbf{R}_{\ddot{\mathbf{v}}} &= \mathbf{D}^T \mathbf{M} \mathbf{D} \\
\mathbf{R}_{\dot{\mathbf{v}}} &= -\mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma}_q + \mathbf{S}_q + \mathbf{Q}_q^A) \mathbf{D} \\
\mathbf{R}_v &= \mathbf{D}^T \left\{ \begin{array}{l} -\mathbf{M} \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, 2\mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \boldsymbol{\gamma}) + \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A)) \\ + \mathbf{M}2(\mathbf{q}, \mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \boldsymbol{\gamma}) - \mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma}_q - \mathbf{S}_q - \mathbf{Q}_q^A \\ - (\mathbf{M} \mathbf{U} \mathbf{B} \boldsymbol{\gamma}_q + \mathbf{S}_q + \mathbf{Q}_q^A) \mathbf{U} \mathbf{B} (\mathbf{P}2(\mathbf{q}, \mathbf{U} \mathbf{B} \boldsymbol{\Phi}_t - \mathbf{D} \dot{\mathbf{v}}) - \boldsymbol{\Phi}_{tq}) \end{array} \right\} \mathbf{D}
\end{aligned} \tag{5.4.16}$$

where  $\mathbf{D} = (\mathbf{I} - \mathbf{U} \mathbf{B} \boldsymbol{\Phi}_q) \mathbf{V}$ .

Six numerical integration algorithms of Section 4.8 are implemented in tangent space ODE Code 5.4.2a of Appendix 5.A, with data provided here. Numerical simulations with this formulation were carried out over a simulation interval of 100 sec, with integration error tolerances  $\text{Intol} = \text{Atol} = e-6$  for constant and variable step size integrators, an iteration limit of 8 in implicit integration, and a bound of 0.75 on the norm of  $\mathbf{v}$ . Results were essentially identical to those shown in Fig. (5.1.6). Total energy for this conservative system was constant to seven decimal places in simulations with  $\omega_0 = 13.5$ . For this Top, gravitational and constraint reaction forces at the tip create no torque about the z-axis, so the  $z'$  component of angular momentum must be conserved (Arnold, 1989). The  $z'$  component of angular momentum over the 100 sec simulation was constant to nine decimal places.

As a check on constraint error control, the convergence tolerance  $\text{Tol} = \text{utol} = \text{Btol}$  in Eqs. (5.2.17) and (5.2.20) was varied to evaluate its influence on constraint error. The maximum norms of constraint errors encountered with the trapezoidal integrator and  $\omega_0 = 13.5$  over the simulation interval are presented in Table 5.4.2. These results show that maximum norms of position, velocity, and acceleration constraint error are driven toward zero, to computer precision, as convergence tolerances are tightened. An insignificant increase in compute time was associated with the tighter tolerances.

Table 5.4.2 Maximum Norms of Constraint Error For Trapezoidal Integrator

Tol	Position Err.	Velocity Err.	Acceleration Err.
e-6	e-6	2.5e-8	1.5e-7
e-9	e-9	3e-9	3e-8
e-12	e-12	e-14	1.4e-13

### 5.4.2.2 Transient Top

A test problem that is used in the literature (Bruls and Cardona, 2010; Terze, Muller, and Zlatar, 2014) to evaluate formulations and numerical methods is transient behavior of a constrained Top, with gravitational acceleration  $9.81 \text{ m/sec}^2$  acting in the negative  $y$  direction, a mass of 15 kg, and an inertia matrix  $\mathbf{J}' = \text{diag}(0.234357, 0.234357, 0.46875) \text{ kg}\cdot\text{m}^2$ . The initial position of the centroid is  $\mathbf{r}^0 = \mathbf{u}_z$ , initial angular velocity is  $\boldsymbol{\omega}'^0 = [0 \ -4.61538 \ 150]^T$ , and the associated initial velocity of the centroid is  $\dot{\mathbf{r}}^0 = \tilde{\boldsymbol{\omega}}'^0 \mathbf{u}_z$ . Integration error tolerance, iteration limits, and  $\mathbf{v}$  norm limits are as in Section 5.4.2.1. Simulations were carried out with constant step size  $h = 0.0001$ , since transient motion of this Top is severe.

Plots of coordinates of the centroid vs time in Fig. 5.4.5, obtained with implicit trapezoidal integration in Code 5.4.2b of Appendix 5.A, are close to results presented in the papers cited. The total energy of the system in this simulation is 5,583 N·m, with a maximum variation of 0.018% over the 2 sec simulation. This contrasts with significant energy loss predicted by the second order *generalized- $\alpha$*  integration method used by Bruls and Cardona (2010), which imposes numerical damping to achieve stability of their integrator. Constraint errors were similar to those cited in Table 5.4.2. Identical results were obtained with explicit Nystrom4 and implicit SDIRK54 integration methods.

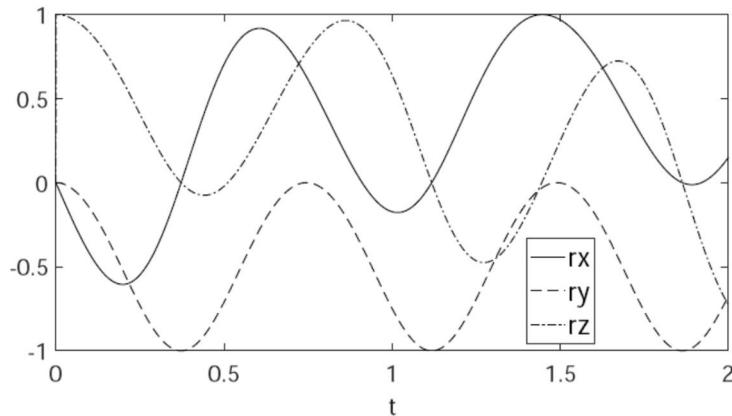


Figure 5.4.5 Coordinates of Centroid for Transient Top

### 5.4.3 Top with Tip Sliding on x-y Plane

A variation on the spin stabilized Top of Sections 5.1 and 5.4.2.1 is to relax the constraint on the tip T, allowing it to move without friction in the x-y plane, as shown in Fig. 5.4.6. A spring-damper with coefficients K and C acts between the tip T and the origin O of the x-y plane.

Generalized coordinates are  $\mathbf{q} = [\mathbf{r}^T \quad \mathbf{p}^T]^T$ , where  $\mathbf{r}$  is the vector from the origin of the x-y-z inertial frame in Fig. 5.4.5 to the centroid of the Top that is a unit distance from the tip. With the unit vector  $\mathbf{u}_z = [0 \ 0 \ 1]^T$  in the positive z-direction, constraint equations and their Jacobian are

$$\Phi(\mathbf{q}) = \begin{bmatrix} \mathbf{u}_z^T (\mathbf{r} - \mathbf{A}(\mathbf{p})\mathbf{u}'_z) \\ 1/2(\mathbf{p}^T \mathbf{p} - 1) \end{bmatrix} = \mathbf{0} \quad (5.4.17)$$

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} \mathbf{u}_z^T & -\mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{u}'_z) \\ \mathbf{0} & \mathbf{p}^T \end{bmatrix} \quad (5.4.18)$$

This Top has five degrees of freedom. The Top with tip fixed has only three degrees of freedom.

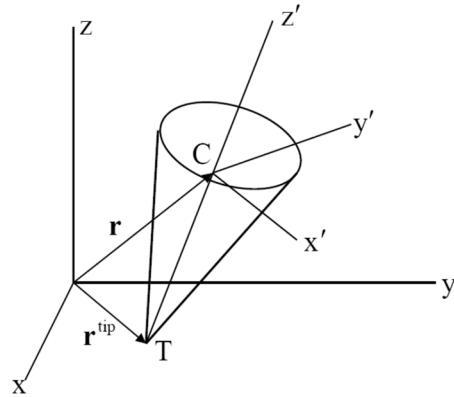


Figure 5.4.56 Top with Tip Sliding on the x-y Plane

The variational equation of motion of Eq. (4.3.24) for the Top, with its centroidal body reference frame, is

$$m \mathbf{r}^T \ddot{\mathbf{r}} + \mathbf{p}^T [4\mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \ddot{\mathbf{p}} - 8\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p}] - \delta \mathbf{q}^T \mathbf{Q}^A = 0 \quad (5.4.19)$$

which must hold for all  $\mathbf{r}$  and  $\mathbf{p}$  that satisfy  $\Phi_q \delta \mathbf{q} = \mathbf{0}$ . Terms in Eq. (4.3.28) are

$$\begin{aligned} \mathbf{M}(\mathbf{q}) &= \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}(\mathbf{p})^T \mathbf{J}' \mathbf{G}(\mathbf{p}) \end{bmatrix} \\ \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} \mathbf{0} \\ 8\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p} \end{bmatrix} \end{aligned} \quad (5.4.20)$$

With a spring force  $\mathbf{F} = -K\mathbf{r}^{tip} - C\dot{\mathbf{r}}^{tip}$  that acts on the tip of the Top at point  $\mathbf{r}^{tip} = \mathbf{r} - \mathbf{A}(\mathbf{p})\mathbf{u}'_z$  and gravity acting in the negative z-direction, the virtual work of forces acting on the Top is

$$\begin{aligned}
W &= -mg\delta r^T \mathbf{u}_z - K \cdot r^{\text{tip}T} \mathbf{r}^{\text{tip}} - C \cdot r^{\text{tip}T} \dot{\mathbf{r}}^{\text{tip}} \\
&= -mg\delta r^T \mathbf{u}_z - K(r - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z) \cdot \mathbf{p})^T(r - \mathbf{A}(\mathbf{p})\mathbf{k}') \\
&\quad - C(r - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z) \cdot \mathbf{p})^T(\dot{\mathbf{r}} - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)\dot{\mathbf{p}}) \\
&= \begin{bmatrix} \mathbf{r}^T & \mathbf{p}^T \end{bmatrix} \begin{bmatrix} -mg\mathbf{u}_z - K(r - \mathbf{A}(\mathbf{p})\mathbf{u}'_z) - C(\dot{\mathbf{r}} - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)\dot{\mathbf{p}}) \\ \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)^T(K(r - \mathbf{A}(\mathbf{p})\mathbf{u}'_z) + C(\dot{\mathbf{r}} - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)\dot{\mathbf{p}})) \end{bmatrix} \equiv \begin{bmatrix} \mathbf{r}^T & \mathbf{p}^T \end{bmatrix} \mathbf{Q}^A
\end{aligned}$$

Thus, the *applied generalized force* is

$$\mathbf{Q}^A = \begin{bmatrix} -mg\mathbf{u}_z - K(r - \mathbf{A}(\mathbf{p})\mathbf{u}'_z) - C(\dot{\mathbf{r}} - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)\dot{\mathbf{p}}) \\ \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)^T(K(r - \mathbf{A}(\mathbf{p})\mathbf{u}'_z) + C(\dot{\mathbf{r}} - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)\dot{\mathbf{p}})) \end{bmatrix} \quad (5.4.21)$$

For implicit numerical integration, terms from Eqs. (5.3.7), (5.3.10), (5.3.17), (5.3.20), (5.3.25), (5.3.40), and (5.3.41) are

$$\begin{aligned}
\mathbf{D} &= (\mathbf{I} - \mathbf{U}\mathbf{B}\Phi_q)\mathbf{V} \\
\mathbf{P}2(\mathbf{q}, \chi) &= (\Phi_q \ddot{\chi})_q = \begin{bmatrix} \mathbf{0} & -\mathbf{u}_z^T \mathbf{B}(\chi_p, \mathbf{u}'_z) \\ \mathbf{0} & \chi_p^T \end{bmatrix} \\
\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) &= (\mathbf{P}2(\mathbf{q}, \ddot{\mathbf{q}})\ddot{\mathbf{q}})_q = \mathbf{0} \\
\mathbf{P}4(\mathbf{q}, \eta) &= (\Phi_q^T \ddot{\eta})_q = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\eta_1 \mathbf{K}(\mathbf{u}_z, \mathbf{u}'_z) + \eta_2 \mathbf{I}_4 \end{bmatrix} \\
\mathbf{M}2(\mathbf{q}, \mu) &= (\mathbf{M}\ddot{\mu})_q = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 4\mathbf{T}(\mathbf{J}'\mathbf{G}(\mathbf{p})\mu_p) - 4\mathbf{G}(\mathbf{p})^T \mathbf{J}'\mathbf{G}(\mu_p) \end{bmatrix} \\
\mathbf{Q}_q^A &= \begin{bmatrix} -K\mathbf{I}_3 & \mathbf{K}\mathbf{B}(\mathbf{p}, \mathbf{u}'_z) + \mathbf{C}\mathbf{B}(\dot{\mathbf{p}}, \mathbf{u}'_z) \\ \mathbf{K}\mathbf{B}(\mathbf{p}, \mathbf{u}'_z)^T & \mathbf{a} \end{bmatrix} \\
\mathbf{a} &= \mathbf{K}(\mathbf{u}'_z, (K(r - \mathbf{A}(\mathbf{p})\mathbf{u}'_z) + C(\dot{\mathbf{r}} - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)\dot{\mathbf{p}})) - \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)^T(\mathbf{K}\mathbf{B}(\mathbf{p}, \mathbf{u}'_z) + \mathbf{C}\mathbf{B}(\dot{\mathbf{p}}, \mathbf{u}'_z))) \\
\mathbf{Q}_{\dot{\mathbf{q}}}^A &= \mathbf{C} \begin{bmatrix} -\mathbf{I}_3 & \mathbf{B}(\mathbf{p}, \mathbf{u}'_z) \\ \mathbf{B}(\mathbf{p}, \mathbf{u}'_z)^T & -\mathbf{B}(\mathbf{p}, \mathbf{u}'_z)^T \mathbf{B}(\mathbf{p}, \mathbf{u}'_z) \end{bmatrix} \\
\mathbf{S}_q &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 8\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}'\mathbf{G}(\dot{\mathbf{p}}) \end{bmatrix} \\
\mathbf{S}_{\dot{\mathbf{q}}} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 8\mathbf{T}(\mathbf{J}'\mathbf{G}(\dot{\mathbf{p}})\mathbf{p}) - 8\mathbf{G}(\dot{\mathbf{p}})^T \mathbf{J}'\mathbf{G}(\mathbf{p}) \end{bmatrix} \quad (5.4.22)
\end{aligned}$$

Since the constraint of Eq. (5.4.17) does not depend explicitly on t and  $\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ , Eq. (5.3.18) reduces to

$$\begin{aligned}\gamma &= \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ \gamma_q &= \mathbf{0} \\ \gamma_{\dot{q}} &= 2\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\end{aligned}\tag{5.4.23}$$

Inertia properties are  $m = 30 \text{ kg}$ ,  $\mathbf{J}' = \text{diag}(90, 90, 30) \text{ kg} \cdot \text{m}^2$  relative to the centroidal  $x'$ - $y'$ - $z'$  frame, and  $g = 9.8 \text{ m/sec}^2$ . The Tops of this and Sections 5.1 and 5.4.2.1 are identical, with the exception that here a centroidal inertia matrix is used, whereas in Section 5.1 the inertia matrix is with respect to the tip. The initial configuration is with the top vertically upward;  $\mathbf{r}^0 = [0 \ 0 \ 1]^T$  and  $\mathbf{p}^0 = [1 \ 0 \ 0 \ 0]^T$ . The initial angular velocity in the body fixed reference frame is  $\omega'^0 = [\varepsilon \ \varepsilon \ \text{omegaz}0]^T$ , where  $x'$  and  $y'$  angular velocities  $= 10^{-12}$  are perturbations from the vertical. Initial generalized coordinate velocities are  $\dot{\mathbf{r}}^0 = \mathbf{0}$  and  $\dot{\mathbf{p}}^0 = 0.5\mathbf{G}(\mathbf{p}^0)^T\omega'^0$ . Total energy  $TE = 2\dot{\mathbf{p}}^T\mathbf{G}(\mathbf{p})^T\mathbf{J}'\mathbf{G}(\mathbf{p})\dot{\mathbf{p}} + mgz$  should be constant for this conservative system.

The plots in Fig. 5.4.7 were obtained with Code 5.4.3 of Appendix 5.A, using the *trapezoidal implicit integrator* with  $h = 0.001 \text{ sec}$  and  $K = C = 0$ . The 40 sec simulations reported show  $tx$  and  $ty$  coordinates of the tip, for simulations with three initial values  $\text{omegaz}0$  of  $z'$ -angular velocity. As shown in the fourth plot at the lower right, the horizontal displacement of the centroid is zero to computer precision, unlike the Top with tip fixed in Sections 5.1 and 5.4.2.1, for which the centroid moves with significant amplitude. The plots in Fig. 5.4.6 indicate that the Top stabilizes at a slightly lower initial angular velocity than the Top with tip fixed. Total energy and angular momentum are constant to seven decimal places in each simulation and 200 *reparameterizations* in 40,000 time steps were required (200 time steps per reparameterization), all due to a limit of 50 on condition number of the reduced mass matrix. Less than one percent of computation effort was utilized for reparameterization.

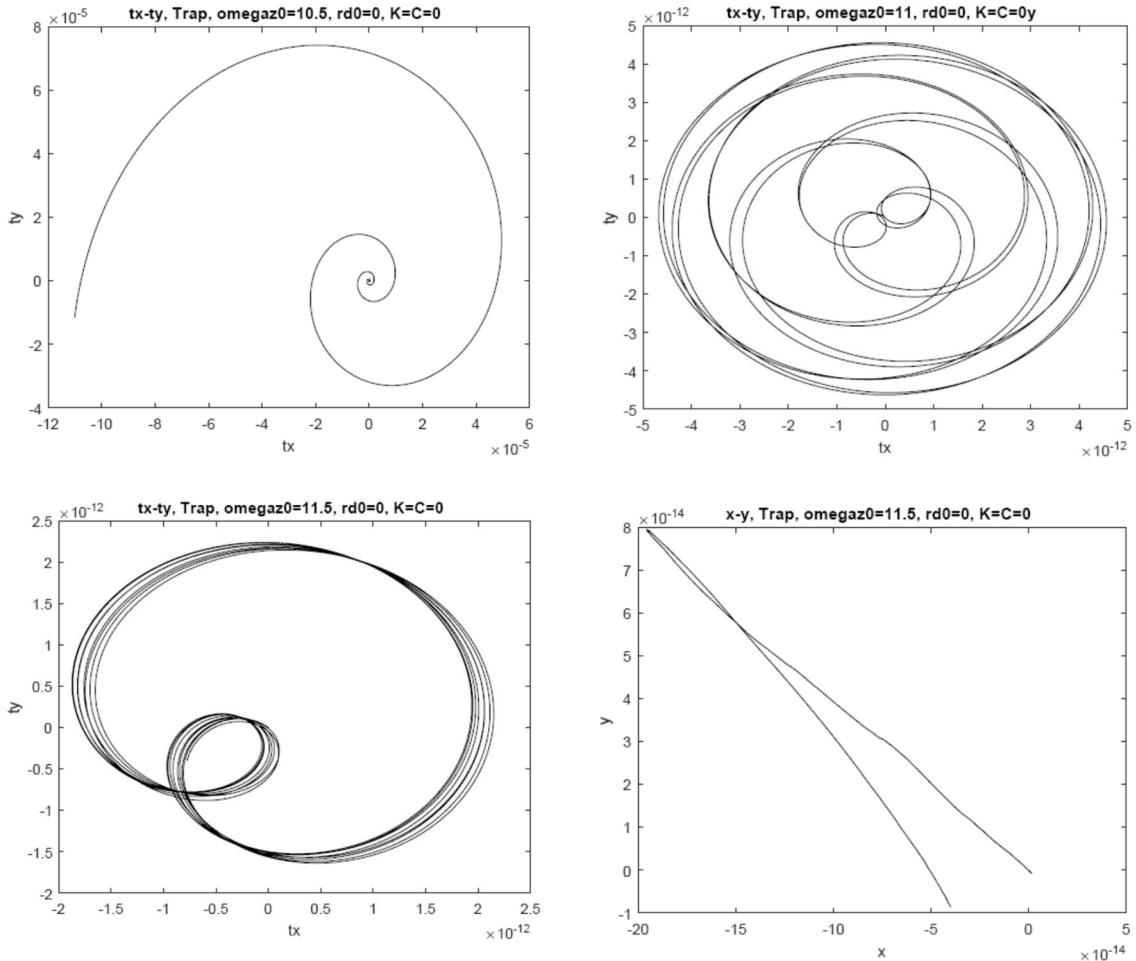


Figure 5.4.7 tx-ty Trajectories for Tip and x-y Trajectory of Centroid,  $K = C = 0$

Simulations carried out with the four explicit integrators of Section 4.8 yielded essentially identical results as those reported. Simulations carried out with the implicit SDIRK54 algorithm of Section 4.8 yielded the same results, but required somewhat greater CPU time.

To investigate the influence of a nonzero spring coefficient  $K$  on motion of the Top, simulations over a 40 sec period were carried out with  $K = 50$  N/m (a stiffness characteristic of a rubber band acting on the tip of the 30 kg Top),  $C = 0$ , and initial angular velocities shown in Fig. 5.4.8. The horizontal trajectories in Fig. 5.4.8 show a marked difference in character from those of the top with  $K = 0$ . The trajectories on the left are displacement of the tip and those on the right are displacement of the centroid. Unlike the case with  $K = 0$  and zero centroidal displacement shown at the lower right of Fig. 5.4.7, with the nonzero but small spring constant, the centroid and tip move with similar radii in horizontal planes. It is interesting that the Top with spring attached to the moving tip requires a significantly higher initial vertical angular velocity to achieve stable motion. As noted above, all six integration algorithms give essentially identical results.

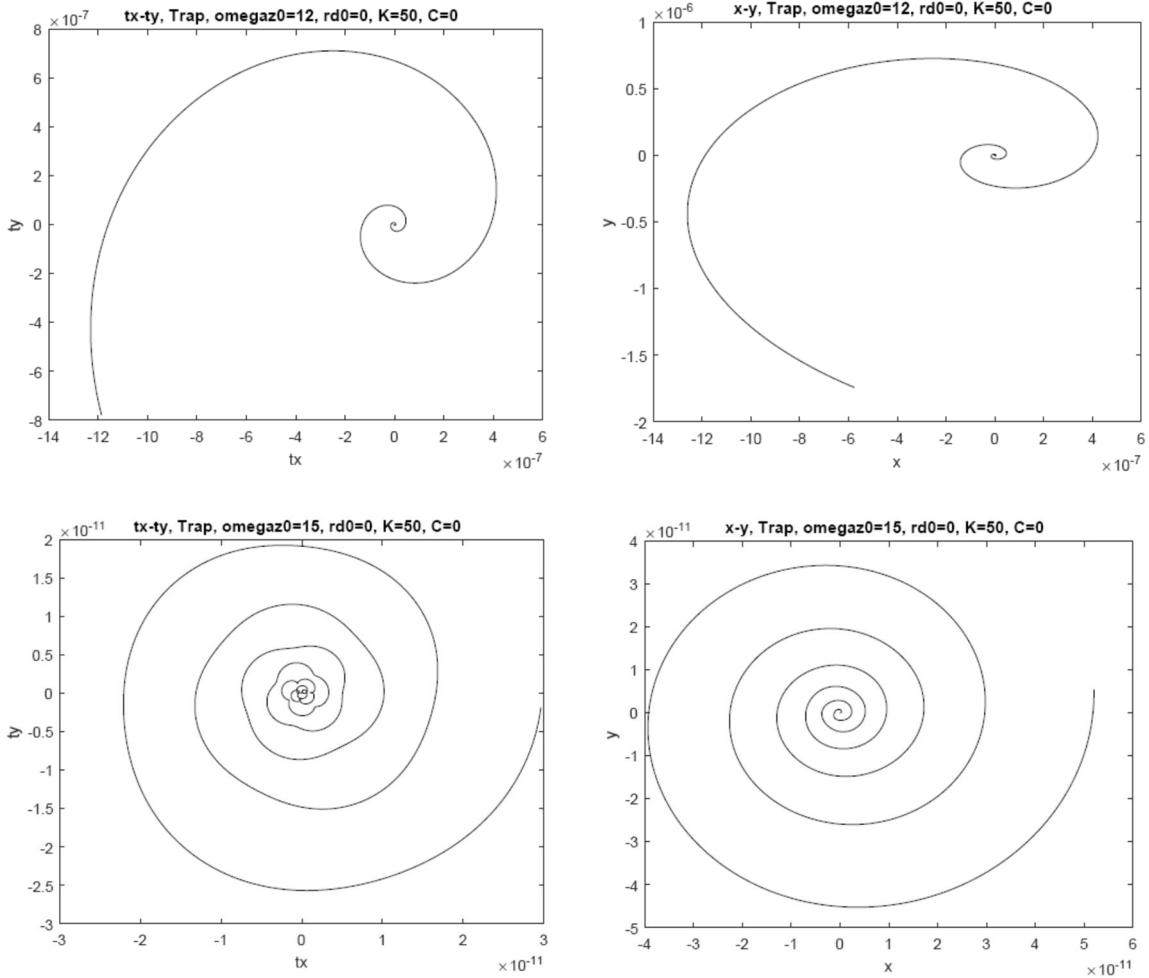


Figure 5.4.8 tx-ty and x-y Trajectories of Tip and Centroid,  $K = 50$ ,  $C = 0$

As a numerical experiment that demonstrates favorable attributes of implicit integration algorithms, very large values of  $K$  and  $C$  are used, to simulate the Top with fixed tip studied in Sections 5.1 and 5.4.2.1. To be consistent with initial conditions for the Top with tip fixed, the same initial configuration was used and initial velocities were  $\boldsymbol{\omega}^0 = \begin{bmatrix} 10^{-12} & 10^{-12} & \text{omegapr}0 \end{bmatrix}^T$  and  $\dot{\mathbf{r}}^0 = \tilde{\boldsymbol{\omega}}^0 \mathbf{u}_z$ . With  $K = C = 10^6$ , the three fixed step size explicit algorithms (Nystrom4, Runge-Kutta4, and Kutta3/8) failed during the first few time steps, as expected due to *stiffness* associated with large physical stiffness and damping in this example. The *variable step size* RKFN54 explicit algorithm with tight absolute error tolerance Atol was able to reduce step size and calculate a numerical solution, but results did not comport with reality. Simulations carried out with the implicit constant step trapezoidal and variable step SDIRK54 algorithms provided meaningful solutions. Results presented in Fig. 5.4.8, using the implicit trapezoidal algorithm and  $\text{omegaz}0 = 13.5$ , indicate that implicit algorithms can indeed handle stiff problems and yield results close to those in Fig. 5.1.3. While the tip displacement at the right of Fig. 5.4.8 shows high frequency activity, its magnitude is zero to computer precision. The trajectory at the left of

Fig. 5.4.89 is remarkably similar to the plot at the lower right of Fig 5.1.3, for the same input data.

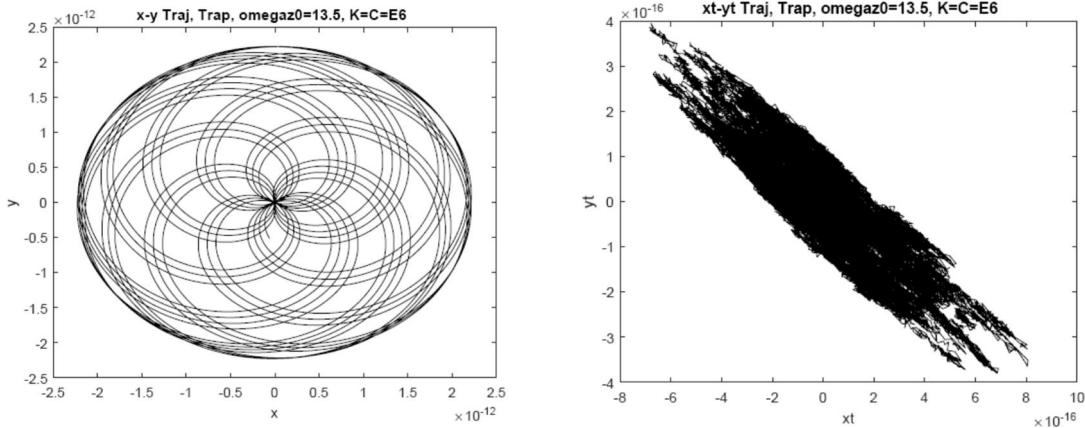


Figure 5.4.9 Simulation of Top with Tip Fixed, Using  $K = C = 10^6$

As a check on constraint error control, the convergence tolerance  $Tol = utol = Btol$  in Eqs. (5.2.17) and (5.2.20) is varied to evaluate its influence on constraint error in the simulation at the upper right of Fig. 5.4.6. Results presented in Table 5.4.3 show that the maximum norm of position, velocity, and acceleration errors over the simulation interval are driven toward zero to computer precision, as convergence tolerances are tightened. Little increase in compute time is associated with the tighter tolerances.

Table 5.4.3 Maximum Constraint Error

Tol	Position Err.	Velocity Err.	Acceleration Err.
e-6	e-6	6e-7	3e-5
e-9	e-9	2.5e-9	1.5e-8
e-12	e-12	8e-13	3e-11

#### 5.4.4 Spatial Double Pendulum

The *spatial double pendulum* of Section 3.3.7.1 is made up of two spheres of unit radius shown in Fig. 5.4.10. The origin of body 1 is fixed to the origin of the global x-y-z frame, permitting the body to rotate freely about that point. A unit distance constraint acts between points  $P_1$  on body 1 at  $-\mathbf{u}'_{1z}$  and  $P_2$  on body 2 at  $\mathbf{u}'_{2z}$ , where  $\mathbf{u}'_{1z}$  and  $\mathbf{u}'_{2z}$  are unit vectors along the positive body fixed  $z'_1$  and  $z'_2$  axes. The mass of each sphere is  $m = 75$  kg, the inertia matrix of each sphere is  $\mathbf{J}' = (2m/5)\mathbf{I}$  kg · m<sup>2</sup>, and a gravitational acceleration of  $g = 9.8$  m / sec<sup>2</sup> acts in the negative z-direction. Euler parameter and position initial conditions are

$\mathbf{p}_1^0 = \mathbf{p}_2^0 = [1 \ 0 \ 0 \ 0]^T$  and  $\mathbf{r}_2^0 = [0 \ 0 \ -3]^T$ ; i.e., the body  $z'$  axes coincide with the inertial  $z$  axis and body two hangs vertically downward. The initial velocity of body two is  $\dot{\mathbf{r}}_2 = \mathbf{0}$  and

initial angular velocities are  $\omega_1' = [\text{Omeg1} \ 0 \ 0]^T$  and  $\omega_2' = [0 \ \text{Omeg2} \ 0]^T$ , so  $\dot{\mathbf{p}}_i^0 = 0.5\mathbf{G}(\mathbf{p}_i^0)^T \omega_i'$ ,  $i = 1, 2$ .

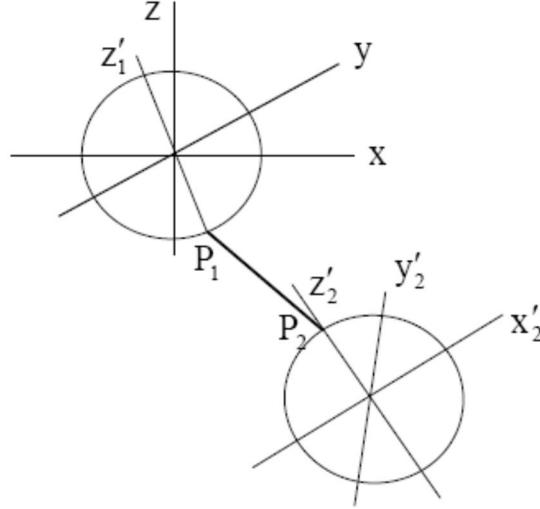


Figure 5.4.10 Spatial Double Pendulum

Generalized coordinates for this system are  $\mathbf{q} = [\mathbf{p}_1^T \ \mathbf{p}_2^T \ \mathbf{r}_2^T]^T \in \mathbb{R}^{11}$ . The vector  $\mathbf{r}^{21}$  from point  $P_2$  to  $P_1$  is

$$\mathbf{r}^{21} = \mathbf{r}_2 + \mathbf{A}(\mathbf{p}_2)\mathbf{u}'_{2z} + \mathbf{A}(\mathbf{p}_1)\mathbf{u}'_{1z} \quad (5.4.24)$$

Constraints and their required derivatives are

$$\Phi(\mathbf{q}) = \begin{bmatrix} 1/2(\mathbf{p}_1^T \mathbf{p}_1 - 1) \\ 1/2(\mathbf{p}_2^T \mathbf{p}_2 - 1) \\ 1/2(\mathbf{r}^{21T} \mathbf{r}^{21} - 1) \end{bmatrix} = \mathbf{0} \quad (5.4.25)$$

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} \mathbf{p}_1^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_2^T & \mathbf{0} \\ \mathbf{r}^{21T} \bar{\mathbf{B}}(\mathbf{p}_1, \mathbf{u}'_{1z}) & \mathbf{r}^{21T} \bar{\mathbf{B}}(\mathbf{p}_2, \mathbf{u}'_{2z}) & \mathbf{r}^{21T} \end{bmatrix}$$

$$\mathbf{P2}(\mathbf{q}, \boldsymbol{\chi}) = \begin{bmatrix} \boldsymbol{\chi}_{\mathbf{p}_1}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\chi}_{\mathbf{p}_2}^T & \mathbf{0} \\ \mathbf{a}^T \mathbf{B}(\mathbf{p}_1, \mathbf{u}'_{1z}) + \mathbf{r}^{21T} \mathbf{B}(\boldsymbol{\chi}_{\mathbf{p}_1}, \mathbf{u}'_{1z}) & \mathbf{a}^T \mathbf{B}(\mathbf{p}_2, \mathbf{u}'_{2z}) + \mathbf{r}^{21T} \mathbf{B}(\boldsymbol{\chi}_{\mathbf{p}_2}, \mathbf{u}'_{2z}) & \mathbf{a}^T \end{bmatrix}$$

where  $\boldsymbol{\chi}_{\mathbf{p}_1}$ ,  $\boldsymbol{\chi}_{\mathbf{p}_2}$ , and  $\boldsymbol{\chi}_{\mathbf{r}_2}$  are components of  $\boldsymbol{\chi} \in \mathbb{R}^{11}$  that correspond to  $\mathbf{p}_1$ ,  $\mathbf{p}_2$ , and  $\mathbf{r}_2$  and  $\mathbf{a} = \mathbf{B}(\mathbf{p}_1, \mathbf{u}'_{1z})\boldsymbol{\chi}_{\mathbf{p}_1} + \mathbf{B}(\mathbf{p}_2, \mathbf{u}'_{2z})\boldsymbol{\chi}_{\mathbf{p}_2} + \boldsymbol{\chi}_{\mathbf{r}_2}$ ,

$$\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}$$

where  $\mathbf{c} = \dot{\mathbf{p}}_1^T \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}'_{1z}) + \dot{\mathbf{p}}_2^T \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}'_{2z}) + \dot{\mathbf{r}}_2^T$ ,  $\mathbf{e} = \mathbf{B}(\mathbf{p}_1, \mathbf{u}'_{1z})\dot{\mathbf{p}}_1 + \mathbf{B}(\mathbf{p}_2, \mathbf{u}'_{2z})\dot{\mathbf{p}}_2 + \dot{\mathbf{r}}_2$ ,

$\mathbf{d} = \dot{\mathbf{p}}_1^T \mathbf{B}^T(\dot{\mathbf{p}}_1, \mathbf{u}'_{1z}) + \dot{\mathbf{p}}_2^T \mathbf{B}^T(\dot{\mathbf{p}}_2, \mathbf{u}'_{2z})$ ,  $\mathbf{b}_1 = (\mathbf{e}^T + \mathbf{c})\mathbf{B}(\dot{\mathbf{p}}_1, \mathbf{u}'_{1z}) + \mathbf{d}\mathbf{B}(\mathbf{p}_1, \mathbf{u}'_{1z})$ ,

$\mathbf{b}_2 = (\mathbf{e}^T + \mathbf{c})\mathbf{B}(\dot{\mathbf{p}}_2, \mathbf{u}'_{2z}) + \mathbf{d}\mathbf{B}(\mathbf{p}_2, \mathbf{u}'_{2z})$ , and  $\mathbf{b}_3 = \mathbf{d}$ ,

$$\mathbf{P}4(\mathbf{q}, \dot{\mathbf{q}}) = (\Phi_q^T \ddot{\mathbf{q}})_q = \begin{bmatrix} \mathbf{c}_1 & \eta_3 \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}'_{1z})\mathbf{B}(\mathbf{p}_2, \mathbf{u}'_{2z}) & \eta_3 \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}'_{1z}) \\ \eta_3 \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}'_{2z})\mathbf{B}(\mathbf{p}_1, \mathbf{u}'_{1z}) & \mathbf{c}_2 & \eta_3 \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}'_{2z}) \\ \eta_3 \mathbf{B}(\mathbf{p}_1, \mathbf{u}'_{1z}) & \eta_3 \mathbf{B}(\mathbf{p}_2, \mathbf{u}'_{2z}) & \eta_3 \mathbf{I}_3 \end{bmatrix}$$

where  $\mathbf{c}_1 = \eta_1 \mathbf{I}_4 + \eta_3 \mathbf{K}(\mathbf{u}'_{1z}, \mathbf{r}^{21}) + \eta_3 \mathbf{B}^T(\mathbf{p}_1, \mathbf{u}'_{1z})\mathbf{B}(\mathbf{p}_1, \mathbf{u}'_{1z})$  and

$\mathbf{c}_2 = \eta_2 \mathbf{I}_4 + \eta_3 \mathbf{K}(\mathbf{u}'_{2z}, \mathbf{r}^{21}) + \eta_3 \mathbf{B}^T(\mathbf{p}_2, \mathbf{u}'_{2z})\mathbf{B}(\mathbf{p}_2, \mathbf{u}'_{2z})$ . Finally, from Eq. (5.3.18),

$$\gamma = \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$$

$$\gamma_q = \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}})$$

$$\gamma_{\dot{\mathbf{q}}} = 2\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})$$

The variational equation of motion for the system is obtained by summing variational equations for each of the bodies,

$$m \mathbf{r}_2^T [\ddot{\mathbf{r}}_2 + g\mathbf{k}] + \sum_{i=1}^2 \mathbf{p}_i^T [4\mathbf{G}^T(\mathbf{p}_i)\mathbf{J}'\mathbf{G}(\mathbf{p}_i)\ddot{\mathbf{p}}_i - 8\mathbf{G}^T(\dot{\mathbf{p}}_i)\mathbf{J}'\mathbf{G}(\dot{\mathbf{p}}_i)\mathbf{p}_i] = 0$$

which must hold for all  $\delta\mathbf{q}$  such that  $\Phi_q \delta\mathbf{q} = \mathbf{0}$ . The mass matrix, velocity coupling vector, and generalized force vector are

$$\mathbf{M}(\mathbf{q}) = m \begin{bmatrix} (8/5)\mathbf{G}^T(\mathbf{p}_1)\mathbf{G}(\mathbf{p}_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (8/5)\mathbf{G}^T(\mathbf{p}_2)\mathbf{G}(\mathbf{p}_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix}$$

$$\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{16m}{5} \begin{bmatrix} \mathbf{G}^T(\dot{\mathbf{p}}_1)\mathbf{G}(\dot{\mathbf{p}}_1)\mathbf{p}_1 \\ \mathbf{G}^T(\dot{\mathbf{p}}_2)\mathbf{G}(\dot{\mathbf{p}}_2)\mathbf{p}_2 \\ \mathbf{0} \end{bmatrix}$$

$$\mathbf{Q}^A = [\mathbf{0} \quad \mathbf{0} \quad -mg\mathbf{u}_z^T]^T$$

Derivatives of kinetic quantities for implicit integration are  $\mathbf{Q}_q^A = \mathbf{Q}_{\dot{\mathbf{q}}}^A = \mathbf{0}$  and

$$\mathbf{S}_q = \frac{16m}{5} \begin{bmatrix} \mathbf{G}^T(\dot{\mathbf{p}}_1)\mathbf{G}(\dot{\mathbf{p}}_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T(\dot{\mathbf{p}}_2)\mathbf{G}(\dot{\mathbf{p}}_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{S}_{\dot{q}} = \frac{16m}{5} \begin{bmatrix} -\mathbf{G}^T(\dot{\mathbf{p}}_1)\mathbf{G}(\mathbf{p}_1) + \mathbf{T}(\mathbf{G}(\dot{\mathbf{p}}_1)\mathbf{p}_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{G}^T(\dot{\mathbf{p}}_2)\mathbf{G}(\mathbf{p}_2) + \mathbf{T}(\mathbf{G}(\dot{\mathbf{p}}_2)\mathbf{p}_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{M2}(q, \mu) = \frac{8m}{5} \begin{bmatrix} -\mathbf{G}^T(\mathbf{p}_1)\mathbf{G}(\boldsymbol{\mu}_{\mathbf{p}_1}) + \mathbf{T}(\mathbf{G}(\mathbf{p}_1)\boldsymbol{\mu}_{\mathbf{p}_1}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{G}^T(\mathbf{p}_2)\mathbf{G}(\boldsymbol{\mu}_{\mathbf{p}_2}) + \mathbf{T}(\mathbf{G}(\mathbf{p}_2)\boldsymbol{\mu}_{\mathbf{p}_2}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where  $\boldsymbol{\mu}_{\mathbf{p}_1}$  and  $\boldsymbol{\mu}_{\mathbf{p}_2}$  are components of  $\boldsymbol{\mu} \in \mathbb{R}^{11}$  that correspond to  $\mathbf{p}_1$  and  $\mathbf{p}_2$ .

The plot in Fig 5.4.11 is obtained using MATLAB Code 5.4.4 of Appendix 5.A. It shows the x-value of the centroid vs time for body 2, for  $\Omega_{10} = 0$  and  $\Omega_{20} = 5$  rad/sec. Total energy is constant to eight decimal places for this conservative system and only one parameterization in 20,000 time steps was required. Constraint reaction force in the massless bar that connects the bodies, in Newtons, calculated using the third component of the Lagrange multiplier of Eq. (5.2.54), which corresponds to the third constraint equation of Eq. (5.4.25), is shown at the left of Fig. 5.4.12. To see the effect of reducing the severity of input, constraint reaction force for input  $\Omega_{20} = 2.5$  is shown at the right of Fig. 5.4.12.

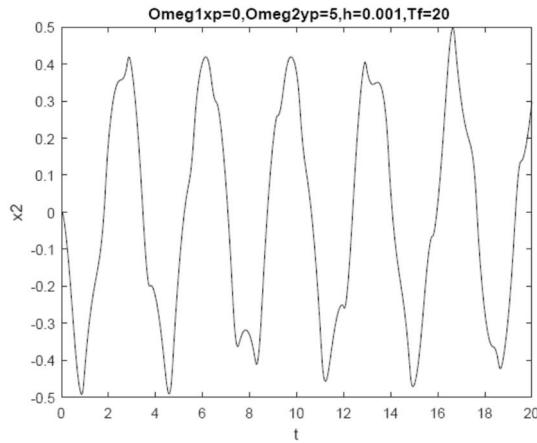


Figure 5.4.11 x-Coordinate of Centroid of Body 2 vs Time

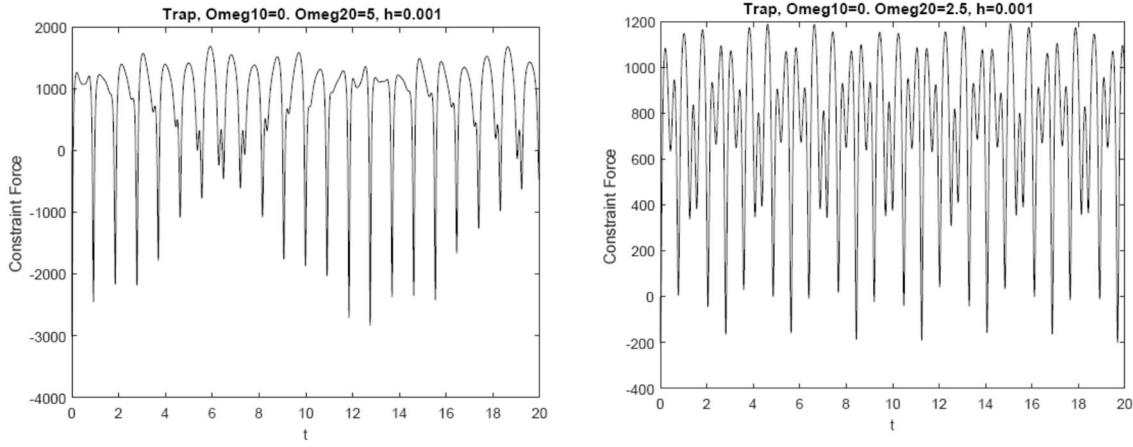


Figure 5.4.12 Constraint Reaction Force in the Bar Between Points  $P_1$  and  $P_2$  vs Time

As a check on *constraint error control*, the convergence tolerance  $Tol = utol = Btol$  in Eqs. (5.2.19) and (5.2.22) is varied to evaluate its influence on constraint error in the simulation of Fig. 5.4.10. Results presented in Table 5.4.4 show that the maximum norms of position, velocity, and acceleration error over the simulation interval are driven toward zero to near computer precision as the convergence tolerance is tightened. Little increase in compute time is associated with the tighter tolerances.

Table 5.4.4 Maximum Constraint Error

Tol	Position Err.	Velocity Err.	Acceleration Err.
e-6	e-6	e-7	e-6
e-9	e-9	e-12	e-11
e-12	e-12	e-13	e-12

#### 5.4.5 Need for Automated Equation Formulation and Solution

While accurate solution of multibody system dynamics with the tangent space ODE formulation is demonstrated, via ad-hoc derivation of governing equations and ad-hoc programming of numerical solution methods, a great deal of person time and effort is required. Motivation is thus provided for effort in later sections of the chapter to automate the formulation and solution process. This topic is addressed in Sections 5.7 through 5.10.

Numerical examples presented using MATLAB codes in Appendix 5.A illustrate the effectiveness of the tangent space ODE formulation and provide confidence that computer implementation for more realistic applications is possible. Examples show that the tangent space parameterization is efficiently updated, based on criteria provided, and yields a unique solution throughout the time interval of interest. Parametric excursions show that all three forms of kinematic constraint are satisfied to specified accuracy, a property that very few computational approaches in mechanical system dynamics achieve. An example that models a spherical joint as a very stiff spring shows that implicit numerical integration of tangent space ODE for a stiff model yields accurate results, which is not possible with explicit integration methods. This is the best definition of a stiff system. Finally, it is observed that automated methods for formulation and solution of equations of kinematics and dynamics are required to avoid oppressive ad-hoc analysis,

## 5.5 Tangent Space Index 0 DAE

The ODE of Eq. (5.2.47) is transformed to an equivalent Index 0 form of DAE that introduces Lagrange multipliers as the derivative of a variable that is determined in numerical integration. This formulation is thus applicable for *systems with friction*, since the Lagrange multipliers may be used to define *constraint reaction forces* and associated *friction forces*. Just as in the ODE formulation of Section 5.3, the Index 0 formulation is based on the tangent space kinematics of Section 5.2 that includes embedded tolerances to assure satisfaction of configuration, velocity, and acceleration constraints. Explicit and implicit numerical integration formulas are applied to the underlying ODE and inflated to the *Index 0 DAE* for numerical solution. As shown in Section 5.11, Index 0 DAE inherit attractive theoretical properties associated with ODE.

### 5.5.1 Index 0 DAE Formulation

Holonomic constraints treated are of the form

$$\bar{\Phi}(\mathbf{q}, t) \equiv \Phi(\mathbf{q}) - \mathbf{f}(t) = \mathbf{0} \in \mathbb{R}^{n_{hc}} \quad (5.5.1)$$

in generalized coordinates  $\mathbf{q} \in \mathbb{R}^{n_{gc}}$ , including Euler parameter normalization conditions for spatial bodies. In addition to satisfying the constraints of Eq. (5.5.1),  $\mathbf{q}$  and its derivatives must satisfy velocity and acceleration constraint equations that are obtained by differentiating Eq. (5.5.1) with respect to time,

$$\Phi_q \dot{\mathbf{q}} = \dot{\mathbf{f}} \equiv \mathbf{v} \quad (5.5.2)$$

$$\Phi_q \ddot{\mathbf{q}} = -\left( \left( \Phi_q \ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}} - \ddot{\mathbf{f}} \right) \equiv -\gamma \quad (5.5.3)$$

Using the *tangent space kinematics* formulation of Section 5.2, the ODE of Eq. (5.2.47) that governs system dynamics is

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} = \mathbf{D}^T \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A \right) \quad (5.5.4)$$

with initial conditions of Eq. (5.2.50),

$$\begin{aligned} \mathbf{v}(t^0) &= \mathbf{0} \\ \dot{\mathbf{v}}(t^0) &= \mathbf{V}^T \dot{\mathbf{q}}^0 \end{aligned} \quad (5.5.5)$$

Using the definition  $\mathbf{D}(\mathbf{q}) \equiv (\mathbf{I} - \mathbf{U} \mathbf{B} \Phi_q) \mathbf{V}$  in Eq. (5.2.24), Eq. (5.5.4) may be written in the form

$$\begin{aligned} &\mathbf{V}^T \left\{ \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} - \Phi_q^T \mathbf{B}^T \mathbf{U}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} + \Phi_q^T \mathbf{B}^T \mathbf{U}^T \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A \right) - \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A \right) \right\} \\ &= \mathbf{V}^T \left\{ \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} + \Phi_q^T \mathbf{B}^T \mathbf{U}^T \left[ \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A \right) - \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} \right] - \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A \right) \right\} = \mathbf{0} \end{aligned} \quad (5.5.6)$$

Defining

$$\begin{aligned} \dot{\boldsymbol{\mu}} &= \mathbf{B}^T \mathbf{U}^T \left[ \left( \mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A \right) - \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} \right] \\ \boldsymbol{\mu}(t^0) &= \mathbf{0} \end{aligned} \quad (5.5.7)$$

Eq. (5.5.6) becomes

$$\mathbf{V}^T \left\{ \mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \dot{\boldsymbol{\mu}} - (\mathbf{M}\mathbf{U}\mathbf{B}\gamma + \mathbf{S} + \mathbf{Q}^A) \right\} = \mathbf{0} \quad (5.5.8)$$

Taking the transpose of Eq. (5.2.13), written in the form  $\mathbf{B}\Phi_q \mathbf{U} = \mathbf{I}$ , yields the identity  $\mathbf{U}^T \Phi_q^T \mathbf{B}^T = \mathbf{I}$ . Multiplying the first of Eqs. (5.5.7) on the left by the nonsingular matrix  $\mathbf{U}^T \Phi_q^T$  yields

$$\mathbf{U}^T \Phi_q^T \dot{\boldsymbol{\mu}} = \mathbf{U}^T \Phi_q^T \mathbf{B}^T \mathbf{U}^T (\mathbf{M}\mathbf{U}\mathbf{B}\gamma + \mathbf{S} + \mathbf{Q}^A) - \mathbf{U}^T \Phi_q^T \mathbf{B}^T \mathbf{U}^T \mathbf{M}\mathbf{D}\ddot{\mathbf{v}}$$

With the foregoing identity, this is

$$\mathbf{U}^T \left\{ \mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \dot{\boldsymbol{\mu}} - (\mathbf{M}\mathbf{U}\mathbf{B}\gamma + \mathbf{S} + \mathbf{Q}^A) \right\} = \mathbf{0} \quad (5.5.9)$$

Combining Eqs. (5.5.8) and (5.5.9),

$$\begin{bmatrix} \mathbf{V}^T \\ \mathbf{U}^T \end{bmatrix} \left( \mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \dot{\boldsymbol{\mu}} - (\mathbf{M}\mathbf{U}\mathbf{B}\gamma + \mathbf{S} + \mathbf{Q}^A) \right) = \mathbf{0}$$

Since the columns of  $\mathbf{V}$  and  $\mathbf{U}$  span  $\mathbf{R}^{ngc}$ , the matrix  $\begin{bmatrix} \mathbf{V}^T \\ \mathbf{U}^T \end{bmatrix} = [\mathbf{V} \quad \mathbf{U}]^T$  is nonsingular and

$$\mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \dot{\boldsymbol{\mu}} - \mathbf{M}\mathbf{U}\mathbf{B}\gamma - \mathbf{S} - \mathbf{Q}^A = \mathbf{0} \quad (5.5.10)$$

If, instead of defining the term on the right of Eqs. (5.5.7) as  $\dot{\boldsymbol{\mu}}$  it were defined as  $\lambda$ , Eq. (5.5.10) would be

$$\mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \lambda - \mathbf{M}\mathbf{U}\mathbf{B}\gamma - \mathbf{S} - \mathbf{Q}^A = \mathbf{0} \quad (5.5.11)$$

which has been called an Index 0 DAE (Haug, 2017b), because it is a DAE that reduces to the ODE of Eq. (5.5.10) with no differentiations (Ascher and Petzold, 1998), simply defining  $\lambda = \dot{\boldsymbol{\mu}}$ . Equation (5.5.11) is thus called an *Index 0 DAE*.

The physical significance of the vector  $\boldsymbol{\mu}$  in Eq. (5.5.7) is not clear, so assignment of the initial condition of Eq. (5.5.7) may be of no consequence, other than assuring there is a unique solution of the initial-value problem.

The steps in Eq. (5.5.4) in deriving Eqs. (5.5.10) and (5.5.11) are reversible, so Eq. (5.5.11) is equivalent to the ODE of Eq. (5.5.4). The ODE theory summarized in Section 4.7.3 shows that the initial-value problem of Eq. (5.5.4) with the associated initial values is *well posed*. Since the initial-value problems of Eqs. (5.5.4) and (5.5.10), with the same initial conditions, are equivalent, the Index 0 DAE with these initial conditions is also *well posed*.

Writing Eq. (5.5.10) in matrix form,

$$\begin{bmatrix} \mathbf{M}\mathbf{D} & \Phi_q^T \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{v}} \\ \dot{\boldsymbol{\mu}} \end{bmatrix} = \mathbf{M}\mathbf{U}\mathbf{B}\gamma + \mathbf{S} + \mathbf{Q}^A \quad (5.5.12)$$

it is clear that for existence of a unique solution, the coefficient matrix on the left must be nonsingular. At a parameterization point  $(\mathbf{q}^0, t^0)$  that satisfies Eq. (5.5.1), Eqs. (5.2.6) and

(5.2.24) show that  $\mathbf{D}^0 = \mathbf{V}$ . To show that the coefficient matrix on the left of Eq. (5.5.12) at  $(\mathbf{q}^0, t^0)$ , with  $\mathbf{D}^0 = \mathbf{V}$  and  $\Phi_q^{T0} = \mathbf{U}$ , is nonsingular, set

$$[\mathbf{MV} \quad \mathbf{U}] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{MV}\boldsymbol{\alpha} + \mathbf{U}\boldsymbol{\beta} = \mathbf{0} \quad (5.5.13)$$

If it can be shown that this equation holds only for  $\boldsymbol{\alpha} = \mathbf{0}$  and  $\boldsymbol{\beta} = \mathbf{0}$ , then the coefficient matrix at  $(\mathbf{q}^0, t^0)$  is nonsingular. Multiplying Eq. (5.5.13) on the left by  $\mathbf{V}^T$ ,

$$\mathbf{V}^T \mathbf{MV}\boldsymbol{\alpha} + \mathbf{V}^T \mathbf{U}\boldsymbol{\beta} = \mathbf{V}^T \mathbf{MV}\boldsymbol{\alpha} = \mathbf{0}$$

Even though  $\mathbf{M}$  is often singular, it is shown in Section 4.6.2 to be positive definite on the null space of  $\Phi_q$ ; i.e.,  $\mathbf{V}^T \mathbf{MV}$  is *positive definite*, hence nonsingular. Thus,  $\boldsymbol{\alpha} = \mathbf{0}$  and, from Eq. (5.5.13),  $\mathbf{U}\boldsymbol{\beta} = \mathbf{0}$ . Since  $\mathbf{U}$  has full column rank,  $\boldsymbol{\beta} = \mathbf{0}$ . This shows that the coefficient matrix on the left of Eq. (5.5.13) is nonsingular. The coefficient matrix of Eq. (5.5.12) is thus nonsingular at  $(\mathbf{q}^0, t^0)$  and, since functions in the matrix are continuous in  $\mathbf{q}$  and  $t$ , the coefficient matrix of Eq. (5.5.10) is nonsingular in a neighborhood of  $(\mathbf{q}^0, t^0)$ .

Finally, since Eq. (5.5.10) involves  $\ddot{\mathbf{v}}$  and  $\dot{\mu}$ , it is neither a first nor a second order ODE. Defining  $y_1 = \mathbf{v}$ ,  $y_2 = \dot{\mathbf{v}}$ , and  $y_3 = \mu$  Eq. (5.5.10) may be written as an equivalent system of first order ODE in  $\mathbf{y} = [y_1 \quad y_2 \quad y_3]^T$ ,

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{MD} & \Phi_q^T \end{bmatrix} \dot{\mathbf{y}} = \begin{bmatrix} y_2 \\ \mathbf{MUB}\gamma + \mathbf{S} + \mathbf{Q}^A \end{bmatrix} \quad (5.5.14)$$

As in the case of the coefficient matrix in Eq. (5.5.12) the coefficient matrix of Eq. (5.5.14) is nonsingular. The system of Eq. (5.5.14) with initial conditions from Eqs.(5.5.5) and (5.5.7),

$$\mathbf{y}(t^0) = [\mathbf{0} \quad \dot{\mathbf{q}}^{0T} \mathbf{V} \quad \mathbf{0}]^T \quad (5.5.15)$$

is a first order initial-value problem that can be numerically integrated using the methods of Section 4.8.

### 5.5.2 Explicit Integration of Index 0 DAE

**Explicit integration of the Index 0 DAE initial-value problem** of Eqs. (5.5.11) and Eq. (5.5.5), using explicit Runge-Kutta methods, is as follows:

(1) Define initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t^0$  that satisfy kinematic configuration and velocity constraints of Eqs. (5.5.1) and (5.5.2). Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{U}$  and  $\mathbf{V}$  of Eqs. (5.2.5) and (5.2.6) and the initial conditions of Eq. (5.5.5). Evaluate  $\mathbf{B}^0$  of Eq. (5.2.13) at  $t^0$  and updated at  $t$  as  $\mathbf{B}(\mathbf{v})$  using Eq. (5.2.20). Define  $\mathbf{u} = \mathbf{h}(\mathbf{v}, t)$  of Eq. (5.2.14), and  $\mathbf{q}$  of Eq. (5.2.8). Evaluate  $\mathbf{D}$  of Eq. (5.2.24), and  $\dot{\mathbf{q}}$  of Eq. (5.2.34). Evaluate  $\gamma$  of Eq. (5.5.3).

(2) Solve Eq. (5.5.11) by factoring  $[\mathbf{MD} \quad \Phi_q^T]$  at time step  $t_i$  to determine  $\ddot{\mathbf{v}}_i$  and  $\lambda_i$ .

Apply an explicit numerical integrator to  $\ddot{\mathbf{v}}_i$  on a time grid with step size  $h$  to determine

$\mathbf{v}_{i+1}$  and  $\dot{\mathbf{v}}_{i+1}$ . This determines  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  on the time grid. Use Eqs. (5.2.33) through (5.2.35) to evaluate  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  on the time grid.

(3) Monitor the condition number of  $[\mathbf{MD} \quad \Phi_q^T]$ , the norm of  $\mathbf{v}$ , and the number of iterations required to evaluate  $\mathbf{u}$  and  $\mathbf{B}$ . If tolerances are exceeded, define a new time  $\bar{t}^0$  and associated  $\bar{\mathbf{q}}^0$  and  $\bar{\mathbf{q}}^0$ . Repeat calculations in Step (1) to define a new parameterization and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\mathbf{v}}^0$ . This process follows the trajectory shown in Fig. 5.2.2, moving smoothly across *charts* in the regular configuration space.

(4) Continue the process until the final time  $tf$  is reached, or a singular configuration associated with a faulty design or model occurs.

While *Lagrange multipliers* are not used in application of the explicit integrator, they are available to determine constraint reaction forces in the system, using results of Section 4.10.

### 5.5.3 Derivatives for Implicit Integration of Index 0 DAE

To use an *implicit numerical integration method* to solve Eqs. (5.5.11) and (5.5.5), derivatives of all terms that appear with respect to  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  are required. These terms are actually a simplified form of those appearing in the ODE of Section 5.3, so the needed derivatives may be obtained using relations derived in Section 5.3

If Eq. (5.5.11) is to be solved numerically with an implicit numerical integration method, a convenient form is the *residual equation*

$$\mathbf{R} \equiv \mathbf{MD}\ddot{\mathbf{v}} + \Phi_q^T \lambda - \mathbf{MUB}\gamma - \mathbf{S} - \mathbf{Q}^A = \mathbf{0} \quad (5.5.16)$$

This form of the equations of motion in terms of  $\dot{\mathbf{v}}$ ,  $\lambda$ ,  $\mathbf{q}(\mathbf{v}, t)$ , and  $\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t)$  is used to iteratively solve a discretized form of the equation, based on *implicit numerical integration* formulas. To implement this approach, expressions for derivatives of  $\mathbf{R}$  with respect to  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  are required. Derivatives of Eq. (5.5.16) are simpler than those for the ODE of Eq. (5.3.1), since the later equation has factors  $\mathbf{D}^T$  on the left that do not appear in Eq. (5.5.16).

Expanding the term  $\mathbf{MD}\ddot{\mathbf{v}} = \mathbf{MV}\ddot{\mathbf{v}} - \mathbf{MUB}\Phi_q V\ddot{\mathbf{v}}$ , using Eq. (5.3.11), the *chain rule of differentiation* can be used to obtain derivatives of the residual  $\mathbf{R}$  of Eq. (5.5.16) with respect to  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ . Using Eqs. (5.3.7), (5.3.14), (5.3.9), and (5.3.8),

$$\begin{aligned} \mathbf{R}_v &= \left( \mathbf{MV}\hat{\mathbf{V}} - \mathbf{MUB}\Phi_q V\hat{\mathbf{V}} - \mathbf{MUB}\gamma + \Phi_q^T \hat{\lambda} - \mathbf{S} - \mathbf{Q}^A \right)_q \mathbf{D} \\ &\quad + \left( \mathbf{MUB}\gamma + \mathbf{S} + \mathbf{Q}^A \right)_{\dot{q}} \mathbf{UBP2}(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{D} \\ \mathbf{R}_{\dot{v}} &= - \left( \mathbf{MUB}\gamma + \mathbf{S} + \mathbf{Q}^A \right)_{\dot{q}} \mathbf{D} \\ \mathbf{R}_{\ddot{v}} &= \mathbf{MD} \end{aligned} \quad (5.5.17)$$

Derivatives called for in Eqs. (5.5.17), evaluated using Eqs. (5.3.25), (5.3.12), (5.3.10), and (5.3.20) are

$$\begin{aligned}
(\mathbf{M}\mathbf{V}\hat{\mathbf{v}})_q &= \mathbf{M}2(\mathbf{q}, \mathbf{V}\hat{\mathbf{v}}) \\
(\mathbf{MUB}\Phi_q \mathbf{V}\hat{\mathbf{v}})_q &= (\mathbf{M}\hat{\mathbf{U}}\hat{\mathbf{B}}\hat{\Phi}_q \hat{\mathbf{V}}\hat{\mathbf{v}})_q + (\hat{\mathbf{M}}\hat{\mathbf{U}}\hat{\mathbf{B}}\hat{\Phi}_q \hat{\mathbf{V}}\hat{\mathbf{v}})_q + (\hat{\mathbf{M}}\hat{\mathbf{U}}\hat{\mathbf{B}}\Phi_q \hat{\mathbf{V}}\hat{\mathbf{v}})_q \\
&= \mathbf{M}2(\mathbf{q}, \mathbf{UB}\Phi_q \mathbf{V}\hat{\mathbf{v}}) - \mathbf{MUBP}2(\mathbf{q}, \mathbf{UB}\Phi_q \mathbf{V}\hat{\mathbf{v}}) + \mathbf{MUBP}2(\mathbf{q}, \mathbf{V}\hat{\mathbf{v}}) \\
(\mathbf{MUB}\gamma)_q &= (\mathbf{M}\hat{\mathbf{U}}\hat{\mathbf{B}}\hat{\gamma})_q + (\hat{\mathbf{M}}\hat{\mathbf{U}}\hat{\mathbf{B}}\hat{\gamma})_q + (\hat{\mathbf{M}}\hat{\mathbf{U}}\hat{\mathbf{B}}\gamma)_q \\
&= \mathbf{M}2(\mathbf{q}, \mathbf{UB}\gamma) - \mathbf{MUBP}2(\mathbf{q}, \mathbf{UB}\gamma) + \mathbf{MUB}\gamma_q \\
(\Phi_q^T \hat{\lambda})_q &= \mathbf{P}4(\mathbf{q}, \lambda) \\
(\hat{\mathbf{M}}\hat{\mathbf{U}}\hat{\mathbf{B}}\gamma)_q &= \mathbf{MUB}\gamma_q
\end{aligned}$$

Using these relations, the fact that  $\mathbf{M}2(\mathbf{q}, \mathbf{b})$  and  $\mathbf{P}2(\mathbf{q}, \chi)$  of Eqs. (5.3.25) and (5.3.10) are linear in their second arguments, and using Eqs. (5.3.34) and (5.3.35), Eqs. (5.5.17) become

$$\begin{aligned}
\mathbf{R}_v &= \left[ \begin{array}{l} \mathbf{M}2(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUBP}2(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUB}\gamma_q + \mathbf{P}4(\mathbf{q}, \lambda) \\ -\mathbf{S}_q - \mathbf{Q}_q^A + (\mathbf{MUB}\gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \mathbf{UBP}2(\mathbf{q}, \dot{\mathbf{q}}) \end{array} \right] \mathbf{D} \\
\mathbf{R}_{\dot{v}} &= -(\mathbf{MUB}\gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \mathbf{D} \\
\mathbf{R}_{\ddot{v}} &= \mathbf{MD}
\end{aligned}$$

Using the relations  $\gamma_q = \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}})$  and  $\gamma_{\dot{q}} = 2\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})$  of Eq. (5.3.18), this is

$$\begin{aligned}
\mathbf{R}_v &= \left[ \begin{array}{l} \mathbf{M}2(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUBP}2(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUBP}3(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{P}4(\mathbf{q}, \lambda) \\ -\mathbf{S}_q - \mathbf{Q}_q^A + (2\mathbf{MUBP}2(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \mathbf{UBP}2(\mathbf{q}, \dot{\mathbf{q}}) \end{array} \right] \mathbf{D} \\
\mathbf{R}_{\dot{v}} &= -(2\mathbf{MUBP}2(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \mathbf{D} \\
\mathbf{R}_{\ddot{v}} &= \mathbf{MD}
\end{aligned} \tag{5.5.18}$$

Finally, from Eq. (5.5.16),

$$\partial \mathbf{R} / \partial \lambda = \Phi_q^T \tag{5.5.19}$$

#### 5.5.4 Implicit Trapezoidal Integration of Index 0 DAE

Implicit trapezoidal integration formulas of Eqs. (4.8.40) are

$$\begin{aligned}
\mathbf{v}_n &= \mathbf{v}_{n-1} + h\dot{\mathbf{v}}_{n-1} + (h^2/4)(\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n) \\
\dot{\mathbf{v}}_n &= \dot{\mathbf{v}}_{n-1} + (h/2)(\ddot{\mathbf{v}}_{n-1} + \ddot{\mathbf{v}}_n)
\end{aligned} \tag{5.5.20}$$

These integration formulas are technically valid only for use with ODE. Without justification, which follows in Section 5.5.5, substituting Eqs. (5.5.20) into Eq. (5.5.16), the residual  $\mathbf{R}_n$  is a function of  $\ddot{\mathbf{v}}_n$  and the chain rule of differentiation yields the derivative of  $\mathbf{R}$  with respect to  $\ddot{\mathbf{v}}_n$ ,

$$\partial \mathbf{R} / \partial \ddot{\mathbf{v}}_n = \mathbf{R}_{\dot{\mathbf{v}}} + \mathbf{R}_{\ddot{\mathbf{v}}} \frac{\partial \dot{\mathbf{v}}_n}{\partial \ddot{\mathbf{v}}_n} + \mathbf{R}_v \frac{\partial \mathbf{v}_n}{\partial \ddot{\mathbf{v}}_n} = \mathbf{R}_{\dot{\mathbf{v}}} + (h/2) \mathbf{R}_{\ddot{\mathbf{v}}} + (h^2/4) \mathbf{R}_v \quad (5.5.21)$$

with all arguments evaluated at  $\mathbf{v}_{n-1}$ ,  $\dot{\mathbf{v}}_{n-1}$ ,  $\ddot{\mathbf{v}}_{n-1}$ , and  $\lambda_{n-1}$ . Thus, the *trapezoidal Jacobian* of the residual of  $\mathbf{R}$  of Eq. (5.5.16), evaluated at  $t_n$ , is

$$\mathbf{J}_n^{trap} = \begin{bmatrix} \partial \mathbf{R} / \partial \ddot{\mathbf{v}}_n & \Phi_q^T \end{bmatrix} \quad (5.5.22)$$

With initial conditions  $\mathbf{q}(t^0) = \mathbf{q}^0$  and  $\dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0$  that satisfy  $\Phi(\mathbf{q}^0, t^0) = \mathbf{0}$  and  $\Phi_q(\mathbf{q}^0, t^0) \dot{\mathbf{q}}^0 = \mathbf{f}_t(t^0)$ , initial values  $\mathbf{v}^0 = \mathbf{0}$  and  $\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0$  are obtained from Eq. (5.3.2). To obtain estimates of  $\ddot{\mathbf{v}}^0$  and  $\lambda^0$ , that are needed to begin iterative solution of Eq. (5.5.16), Eq. (5.5.12) may be solved at  $t^0$  with the initial conditions.

To carry out Newton-Raphson iteration for  $\ddot{\mathbf{v}}_n$  and  $\lambda_n$ , using the Jacobian of Eq. (5.5.22),

$$\begin{aligned} \mathbf{J}_n^{trap} \begin{bmatrix} \Delta \ddot{\mathbf{v}}_n^i \\ \Delta \lambda_n^i \end{bmatrix} &= -\mathbf{R}(\ddot{\mathbf{v}}_n^i, \lambda_n^i) \\ \begin{bmatrix} \ddot{\mathbf{v}}_n^{i+1} \\ \lambda_n^{i+1} \end{bmatrix} &= \begin{bmatrix} \ddot{\mathbf{v}}_n^i \\ \lambda_n^i \end{bmatrix} + \begin{bmatrix} \Delta \ddot{\mathbf{v}}_n^i \\ \Delta \lambda_n^i \end{bmatrix} \quad i=1,2,\dots \text{ until } \|\mathbf{R}\| \leq \text{intol} \end{aligned} \quad (5.5.23)$$

is formed and solved until convergence is achieved. To evaluate the right side of the first of Eqs. (5.5.23), Eqs. (5.5.20) are used to obtain  $\mathbf{v}_n^i$  and  $\dot{\mathbf{v}}_n^i$ , Eq. (5.2.19) is used to determine  $\mathbf{u}_n^i$ , Eq. (5.2.20) is used to determine  $\mathbf{B}_n$ , and Eqs. (5.2.33) and (5.2.34) are used to determine  $\mathbf{q}_n^i$  and  $\dot{\mathbf{q}}_n^i$ . Since a precise Jacobian is not required to achieve convergence in Eq. (5.5.23), the integration Jacobian  $\mathbf{J}_n^{trap}$  of Eq. (5.5.22) is held constant over iterations in Eq. (5.5.23).

### 5.5.5 Implicit Runge-Kutta Integration of Index 0 DAE

For a general second order ODE of the form  $\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}})$ , the Runge-Kutta integrator of Eqs. (4.8.27) and (4.8.28) is

$$\begin{aligned} \mathbf{k}_i &= \mathbf{g} \left( t_{n-1} + c_i h, \mathbf{v}_{n-1} + h c_i \dot{\mathbf{v}}_{n-1} + h^2 \sum_{j=1}^i A_{ij} \mathbf{k}_j, \dot{\mathbf{v}}_{n-1} + h \sum_{j=1}^i a_{ij} \mathbf{k}_j \right), \quad i=1 \dots s \\ \mathbf{v}_n &= \mathbf{v}_{n-1} + h \dot{\mathbf{v}}_{n-1} + h^2 \sum_{j=1}^s B_j \mathbf{k}_j \\ \dot{\mathbf{v}}_n &= \dot{\mathbf{v}}_{n-1} + h \sum_{j=1}^s b_j \mathbf{k}_j \end{aligned} \quad (5.5.24)$$

Applying this formula to the conventional ODE reformulation of Eq. (5.5.4),

$$\ddot{\mathbf{v}} = \bar{\mathbf{M}}^{-1} \mathbf{F} \quad (5.5.25)$$

where  $\bar{\mathbf{M}} \equiv \mathbf{D}^T \mathbf{M} \mathbf{D}$  and  $\mathbf{F} \equiv \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A)$ ,

$$\mathbf{k}_i = \check{\bar{\mathbf{M}}}^{-1} \check{\mathbf{F}} \quad (5.5.26)$$

The over-score notation  $\check{g}$  for a function or matrix  $\mathbf{f}(t, v, \dot{v})$  is defined as

$$\check{g}(t, v, \dot{v}) \equiv g \left( t_{n-1} + c_i h, v_{n-1} + hc_i \dot{v}_{n-1} + h^2 \sum_{j=1}^i A_{ij} \mathbf{k}_j, \dot{v}_{n-1} + h \sum_{j=1}^i a_{ij} \mathbf{k}_j \right) \quad (5.5.27)$$

Multiplying both sides of Eq. (5.5.26) on the left by  $\check{\bar{\mathbf{M}}}$ ,

$$\check{\bar{\mathbf{M}}} \mathbf{k}_i = \check{\mathbf{F}} \quad (5.5.28)$$

Expanding this equation, using the definitions of  $\bar{\mathbf{M}}$  and  $\mathbf{F}$ , yields

$$\mathbf{V}^T \check{\bar{\mathbf{M}}} \check{\mathbf{D}} \mathbf{k}_i - \mathbf{V}^T \check{\Phi}_q^T \check{\mathbf{B}}^T \mathbf{U}^T \check{\bar{\mathbf{M}}} \check{\mathbf{D}} \mathbf{k}_i + \mathbf{V}^T \check{\Phi}_q^T \check{\mathbf{B}}^T \mathbf{U} \check{\mathbf{Q}} - \mathbf{V}^T \check{\mathbf{Q}} = \mathbf{0} \quad (5.5.29)$$

Defining

$$\lambda_i \equiv \check{\mathbf{B}}^T \mathbf{U} \check{\mathbf{Q}} - \check{\mathbf{B}}^T \mathbf{U} \check{\bar{\mathbf{M}}} \check{\mathbf{D}} \mathbf{k}_i \quad (5.5.30)$$

Eq. (5.5.29) becomes

$$\mathbf{V}^T (\check{\bar{\mathbf{M}}} \check{\mathbf{D}} \mathbf{k}_i + \check{\Phi}_q^T \lambda_i - \check{\mathbf{Q}}_i) = \mathbf{0} \quad (5.5.31)$$

Multiplying both sides of Eq. (5.5.30) on the left by  $\mathbf{U}^T \check{\Phi}_q^T$  and using  $\mathbf{B}^T = (\mathbf{U}^T \check{\Phi}_q^T)^{-1}$  from Eq. (5.2.13) yields

$$\mathbf{U}^T (\check{\bar{\mathbf{M}}} \check{\mathbf{D}} \mathbf{k}_i + \check{\Phi}_q^T \lambda_i - \check{\mathbf{Q}}_i) = \mathbf{0} \quad (5.5.32)$$

Since the columns of  $\mathbf{U}$  and  $\mathbf{V}$  span  $\mathbb{R}^n$ , their common coefficient in Eqs. (5.5.31) and (5.5.32) must be zero; i.e.,

$$\mathbf{R}_n \equiv \check{\bar{\mathbf{M}}} \check{\mathbf{D}} \mathbf{k}_i + \check{\Phi}_q^T \lambda_i - \check{\mathbf{Q}}_i = \mathbf{0} \quad (5.5.33)$$

This system of nonlinear algebraic equations may be solved using the *Newton-Raphson method*. Partial derivatives of Eqs. (5.5.33) and the chain rule of differentiation with the arguments of Eq. (5.5.27) and the second and third of Eqs. (5.5.24) yield

$$\partial \mathbf{R}_n / \partial \mathbf{k}_i = \mathbf{MD} + ha_{ii} \mathbf{R}_v + h^2 A_{ii} \mathbf{R}_v \quad (5.5.34)$$

For the SDIRK54 algorithm of Eqs. (5.5.24) and (4.8.43),  $a_{ii} = 1/4$  and  $A_{ii} = a_{ii}^2 = 1/16$ . Thus,

$$\partial \mathbf{R}_n / \partial \mathbf{k}_i = \mathbf{R}_v + 0.25h \mathbf{R}_v + 0.0625h^2 \mathbf{R}_v \quad (5.5.35)$$

and, with  $\partial \mathbf{R} / \partial \lambda_i = \check{\Phi}_q^T$ , the *SDIRK54 integration Jacobian* is

$$\mathbf{J}_n^{RK} \equiv \begin{bmatrix} \partial \mathbf{R}_n / \partial \mathbf{k}_i & \check{\Phi}_q^T \end{bmatrix} \quad (5.5.36)$$

The solution algorithm at iteration  $j$  is thus

$$\begin{aligned} \mathbf{J}_n^{RK} \begin{bmatrix} \Delta \mathbf{k}_i^j \\ \Delta \lambda_i^j \end{bmatrix} &= -\mathbf{R}(\mathbf{k}_i^j, \lambda_i^j) \\ \begin{bmatrix} \mathbf{k}_i^{j+1} \\ \lambda_i^{j+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{k}_i^j \\ \lambda_i^j \end{bmatrix} + \begin{bmatrix} \Delta \mathbf{k}_i^j \\ \Delta \lambda_i^j \end{bmatrix} \quad j=1,2,\dots \text{ until } \|\mathbf{R}\| \leq \text{intol} \end{aligned} \quad (5.5.37)$$

for stages  $i = 1, \dots, s$ . The solution  $\mathbf{v}_n$  and  $\dot{\mathbf{v}}_n$  is given by the second and third of Eqs. (5.5.24).

This procedure may be viewed as applying the Runge-Kutta algorithm to the DAE of Eq. (5.5.16). Since the manipulations are reversible, as shown in Section 5.2, Eq. (5.5.33) is equivalent to Eq. (5.5.26). The solutions of these equations, with the same initial conditions on  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ , must be identical, to within numerical precision. This result justifies the formal calculation with the trapezoidal integrator in Section 5.5.4 and shows that the Index 0 DAE solution algorithm has the same convergence properties as the ODE integrator that is used.

### 5.5.6 Implicit Index 0 DAE Integration Algorithm

**Implicit Index 0 DAE integration** in the residual DAE form of Eq. (5.5.16) is as follows:

- (1) Define initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t^0$  that satisfy kinematic configuration and velocity constraints. Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{U}$  and  $\mathbf{V}$  in Eqs. (5.2.5) and (5.2.6). Obtain initial conditions  $\mathbf{v}^0 = \mathbf{0}$  and  $\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0$  from Eq. (5.3.2). Evaluate  $\mathbf{B}^0$  of Eq. (5.2.13) and updated in Eq. (5.2.20). Evaluate  $\mathbf{u} = \mathbf{h}(\mathbf{v}, t)$  of Eq. (5.2.14) and  $\mathbf{q}$  of Eq. (5.2.33). Evaluate  $\mathbf{D}$  of Eq. (5.2.24), and  $\ddot{\mathbf{q}}$  of Eq. (5.2.34). Evaluate  $\mathbf{P}2(\mathbf{q}, \chi)$ ,  $\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}})$ ,  $\mathbf{P}4(\mathbf{q}, \eta)$ ,  $\mathbf{M}2(\mathbf{q}, \mu)$ ,  $\mathbf{R}_{\dot{\mathbf{v}}}$ ,  $\mathbf{R}_{\ddot{\mathbf{v}}}$ , and  $\mathbf{R}_v$  of Section 5.5.3.
- (2) Apply the implicit numerical integrator of Eq. (5.5.23) for the trapezoidal method or of Eq. (5.5.37) for the Runge-Kutta method to proceed stepwise on a time grid with step size  $h$ . Use Eqs. (5.5.20) or the second and third of Eqs. (5.5.24) to evaluate  $\mathbf{v}_n$  and  $\dot{\mathbf{v}}_n$ . In either case, use Eqs. (5.2.33) through (5.2.35) to evaluate  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ , and  $\ddot{\mathbf{q}}_n$ .
- (3) Monitor the *condition number* of  $\mathbf{J}_n$ , the number of Newton-Raphson iterations required in Step (2), the norm of  $\mathbf{v}$ , and the number of iterations required to evaluate  $\mathbf{u}$  and  $\mathbf{B}$ . If tolerances are exceeded, define a new time  $\bar{t}^0$  and associated  $\bar{\mathbf{q}}^0$  and repeat calculations in Step (1) to define a new *parameterization* and *initial conditions*  $\bar{\mathbf{v}}^0$  and  $\dot{\bar{\mathbf{v}}}^0$ . Otherwise, continue with the same parameterization.
- (4) Continue the process until the final time  $tf$  is reached, or a singularity due to a faulty design or model is encountered.

### 5.5.7 Third Derivatives for Trapezoidal Error Control

To implement the integration error control approach presented in Section 4.8.5, for problems with no explicit time dependent constraints,

$$\bar{\mathbf{v}}_n = \mathbf{v}_{n-1} + h \dot{\mathbf{v}}_{n-1} + (h^2 / 2) \ddot{\mathbf{v}}_{n-1} + (h^3 / 6) \dddot{\mathbf{v}}_{n-1}$$

the third derivative  $\ddot{\mathbf{v}}_n$  must be computed. For systems in which the variable  $t$  does not appear explicitly, Eq. (5.5.10) may be differentiated to obtain

$$\begin{aligned} \mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \dot{\lambda} - \left\{ \mathbf{M}\mathbf{U}\mathbf{B}\gamma_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A \right\} \ddot{q} \\ + \left\{ \left( \mathbf{M}\hat{\mathbf{D}}\hat{\mathbf{v}} \right)_q + \mathbf{M} \left( \mathbf{D}\hat{\mathbf{v}} \right)_q + \left( \Phi_q^T \hat{\lambda} \right)_q - \left( \mathbf{M}\hat{\mathbf{U}}\hat{\mathbf{B}}\hat{\gamma} \right)_q - \mathbf{M}\mathbf{U} \left( \mathbf{B}\hat{\gamma} \right)_q - \mathbf{M}\mathbf{U}\mathbf{B}\gamma_q - \mathbf{Q}_q^A \right\} \dot{q} = \mathbf{0} \end{aligned}$$

where all terms are evaluated at  $t_{n-1}$ . Substituting from Eqs. (5.3.18), (5.3.12), and (5.3.13), using the fact that operators  $M_2$  and  $P_2$  are linear in their second arguments and collecting terms,

$$\mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \dot{\lambda} \equiv \mathbf{Rhs} \quad (5.5.38)$$

where

$$\mathbf{Rhs} = \left\{ \mathbf{M}\mathbf{U}\mathbf{B}\gamma_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A \right\} \ddot{q} + \left\{ \mathbf{M}\mathbf{U}\mathbf{B}\mathbf{P}_2(\mathbf{q}, \ddot{\mathbf{q}}) - \mathbf{M}\mathbf{2}(\mathbf{q}, \ddot{\mathbf{q}}) - \mathbf{P}\mathbf{4}(\mathbf{q}, \lambda) + \mathbf{M}\mathbf{U}\mathbf{B}\gamma_q + \mathbf{Q}_q^A \right\} \dot{q}$$

is evaluated at  $t_{n-1}$ . While there are numerous terms in Eqs. (5.5.38), all must be evaluated in each iteration of Eq. (5.5.23) for an implicit method, so the additional cost in solving for  $\ddot{\mathbf{v}}_{n-1}$  is the solution of Eq. (5.5.38). This is not a significant additional cost, as indicated in Section 4.8.5. For an explicit method such as Nystrom4, the cost of these calculations may be justified if sufficiently large step sizes can be taken and error control is of value.

In addition to providing information for error control, the third derivative  $\ddot{\mathbf{v}}_{n-1}$  enables improved accuracy of the estimate for  $\ddot{\mathbf{v}}_n$ , namely

$$\ddot{\mathbf{v}}_n \approx \ddot{\mathbf{v}}_{n-1} + h\ddot{\mathbf{v}}_{n-1} \quad (5.5.39)$$

A redefinition of terms in the tangent space ODE formulation yields an equivalent set of equations that include Lagrange multipliers of the DAE derived in Section 4.10, which provide constraint reaction forces. The resulting Index 0 DAE formulation is equivalent to the ODE formulation and enables direct application of explicit integration formulas and implicit trapezoidal and Runge-Kutta numerical integration formulas to the underlying ODE, which is then inflated for solution in the DAE setting. This justifies the name “Index 0 DAE” and provides a practical foundation for modeling and simulating the effects of friction in Chapter 8, using constraint reaction forces that are defined by the Lagrange multipliers.

Numerical implementation of explicit and implicit numerical integration formulas proceeds as in the ODE, with somewhat simplified derivative expressions.

### Key Formulas

$$\mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T \lambda - \mathbf{MUB}\gamma - \mathbf{S} - \mathbf{Q}^A = \mathbf{0} \quad \mathbf{v}(t^0) = \mathbf{0}, \quad \dot{\mathbf{v}}(t^0) = \mathbf{V}^T \dot{\mathbf{q}}^0 \quad (5.5.11)$$

$$\begin{aligned} \mathbf{R}_v &= \left[ \begin{array}{c} \mathbf{M2}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUBP2}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUBP3}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{P4}(\mathbf{q}, \lambda) \\ -\mathbf{S}_q - \mathbf{Q}_q^A + (2\mathbf{MUBP2}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A) \mathbf{UBP2}(\mathbf{q}, \dot{\mathbf{q}}) \end{array} \right] \mathbf{D} \\ \mathbf{R}_{\dot{v}} &= -(2\mathbf{MUBP2}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A) \mathbf{D} \end{aligned} \quad (5.5.18)$$

$$\mathbf{R}_{\dot{v}} = \mathbf{MD}$$

$$\partial \mathbf{R} / \partial \lambda = \Phi_q^T \quad \mathbf{J}_n = \left[ \begin{array}{cc} \partial \mathbf{R}_n / \partial \ddot{\mathbf{v}} & \Phi_q^T \end{array} \right] \quad (5.5.19) \quad (5.5.22)$$

## 5.6 Numerical Examples with Tangent Space Index 0 DAE

Two versions of a spatial spinning Top and a spatial double pendulum are used as examples to test and evaluate the Index 0 DAE formulation of Section 5.5.

### 5.6.1 Top with Tip Fixed

#### 5.6.1.1 Spin Stabilized Top

The *spin stabilized Top* of Section 5.4.2.1 is treated here, using the Index 0 DAE formulation. Data, constraint equations, kinetic expressions, and derivative operators are as presented in Section 5.4.2.1. For implicit Index 0 DAE integration, using the residual of Eq. (5.5.16), derivatives of the residual in Eq. (5.5.18) are

$$\begin{aligned}\mathbf{R}_v &= \left[ \begin{array}{l} \mathbf{M2}(\mathbf{q}, \mathbf{D}\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}\gamma) - \mathbf{MUBP2}(\mathbf{q}, \mathbf{D}\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}\gamma) - \mathbf{MUB}\gamma_q + \mathbf{P4}(\mathbf{q}, \lambda) \\ -\mathbf{S}_q - \mathbf{Q}_q^A + (\mathbf{MUB}\gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A)\mathbf{UBP2}(\mathbf{q}, \dot{\mathbf{q}}) \end{array} \right] \mathbf{D} \\ \mathbf{R}_{\dot{v}} &= -(\mathbf{MUB}\gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A)\mathbf{D} \\ \mathbf{R}_{\ddot{v}} &= \mathbf{MD}\end{aligned}$$

The implicit *trapezoidal algorithm* of Section 5.5.4 is implemented in Code 5.6.1a of Appendix 5.A, using data summarized here. Numerical results for this formulation over a simulation interval of 100 sec, with an *integration error tolerance* Intol =  $10^{-4}$ , an *iteration limit* of 6, and a bound of 0.7 on the norm of  $\mathbf{v}$  are identical to Fig. 5.1.3 in Section 5.1. As indicated, the transition from 12 to 13.5 rad/sec initial  $z'$  angular velocity results in a tight spiral; i.e., stable vertical spin of the Top. The *Nystrom4* method with the explicit integration algorithm of Section 5.5.2 and the *SDIRK54* method with the implicit integration algorithm of Section 5.5.6 yielded essentially identical results.

As a check on *constraint error control*, the convergence tolerance Tol=utol=Btol in Eqs. (5.2.17) and (5.2.20) is varied to evaluate its influence on constraint error. Maximum norms of constraint error over the simulation interval presented in Table 5.6.1 show that position, velocity, and acceleration constraint errors are driven to zero, to near computer precision, as convergence tolerances in the trapezoidal method are tightened. No significant increase in compute time was associated with the tighter tolerances. Similar results were obtained with the Nystrom4 and SDIRK54 methods.

Table 5.6.1 Maximum Constraint Error for Trapezoidal Integrator

Tol.	Position Err.	Velocity Err.	Acceleration Err.
e-6	e-9	e-14	e-12
e-9	e-9	e-14	e-12
e-12	e-12	e-15	e-13

In addition to demonstrating that the formulation presented satisfies the equations of motion and all three forms of kinematic constraint, the total energy predicted in simulation of this conservative system, using the SDIRK54 integrator with a step size  $h = 0.001$  sec, had a

maximum variation of  $4.66 \times 10^{-5}$  over the 100 sec simulation with  $\text{omegaz0} = 13$ . For this Top, gravitational force and constraint reaction forces at the tip create no torque about the z-axis, so the z component of angular momentum must be conserved (Arnold, 1989). Maximum deviation of the z component of angular momentum in simulations was  $7.25 \times 10^{-7}$ .

### 5.6.1.2 Transient Top

The *transient Top* of Section 5.4.2.2 is treated here, using the Index 0 DAE formulation. Data, constraint equations, kinetic expressions, and derivative operators are as presented in Section 5.4.2.2. Integration Jacobians are as in Section 5.6.1.1.

The plot in Fig. 5.6.1 of the y coordinate of the centroid vs time, obtained with trapezoidal integration in Code 5.6.1b of Appendix 5.A, agrees with corresponding results in (Bruls and Cardona, 2010). Constraint errors were similar to those cited in Table 5.6.1.1. Identical results were obtained with Nystrom4 and SDIRK54 methods.

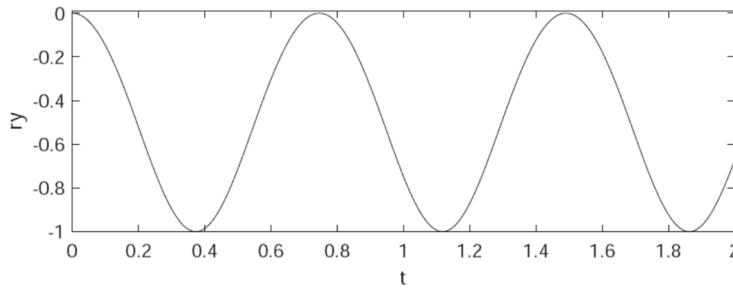


Figure 5.6.2 y-Coordinate of Centroid for Transient Top

Dynamic response of the transient Top is somewhat more extreme than the spin stabilized top, and there was modest variation in total energy and y component of angular momentum relative to the tip, both of which are constant in the exact solution. Simulations were carried out with step size  $h = 0.0001$ . Mean values of y-angular momentum and total energy were  $-70.3$  and  $5,435$ , respectively. Maximum variations in y-angular momentum over the two second simulation were  $3 \times 10^{-3}$  ( $2.7 \times 10^{-2}$ ),  $8.3 \times 10^{-5}$  ( $1.4 \times 10^{-4}$ ), and  $8.3 \times 10^{-5}$  ( $1.9 \times 10^{-5}$ ) for the trapezoidal, Nystrom4, and SDIRK54 integrators, respectively. These results indicate that the implicit five stage SDIRK54 method preserves invariants more accurately than the explicit four stage Nystrom4 method or the implicit one stage trapezoidal method. This contrasts with energy loss predicted by the second order generalized- $\alpha$  integration method used by Bruls and Cardona (2010), which imposes numerical damping on the solution.

## 5.6.2 Spatial Double Pendulum

The spatial double pendulum of Section 5.4.4 is treated here, using the Index 0 DAE formulation. Data, constraint equations, kinetic expressions, and derivative operators are as presented in Section 5.4.4. Integration Jacobians are as in Section 5.6.1.1.

The plot in Fig 5.6.2 is obtained with the implicit SDIRK54 integrator in Code 5.6.2 of Appendix 5.A, using the Index 0 DAE formulation with  $h = 0.001$ , a limit of 1 on the norm of  $v$ , and an integration convergence tolerance of 0.01. It shows the x-value of the centroid vs time for body 2, with  $\text{Omeg2} = 5$  rad/sec and  $\text{Omeg1} = 0$ . In explicit Nystrom4 integration, with a unit norm limit on  $v$  and a limit of 500 on condition number of the coefficient matrix, only one

*parameterization* in 20,000 time steps was required. Likewise, in implicit SDIRK54 and trapezoidal integration, just one parameterization in 20,000 time steps was required.

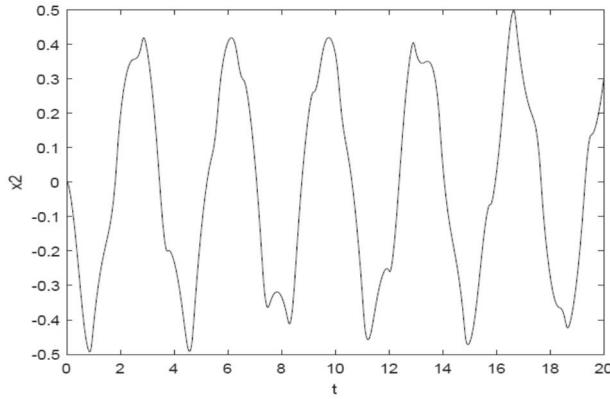


Figure 5.6.2 x-Coordinate of Centroid of Body 2 vs Time

As a check on constraint error control, the convergence tolerance Tol=utol=Btol in Eqs. (5.2.19) and (5.2.22) was varied in Nystrom4, trapezoidal, and SDIRK54 methods to evaluate its influence on constraint error. Maximum norms of constraint error over the simulation period presented in Table 5.6.2 show that position, velocity, and acceleration constraint errors are driven toward zero, to computer precision, as convergence tolerances are tightened. No significant increase in compute time was associated with the tighter tolerances.

Table 5.6.2 Maximum Constraint Error For Nystrom4 Explicit Integrator

Tol.	Position Err.	Velocity Err.	Acceleration Err.
e-6	e-10	e-7	e-7
e-9	e-12	e-11	e-9
e-12	e-15	e-14	e-13

In addition to demonstrating that the formulation presented satisfies the equations of motion and all three forms of kinematic constraint, the total energy predicted in simulation of this conservative system using the SDIRK54 integrator had a maximum variation of  $2.3 \times 10^{-6}$  over the 20 sec simulation. Further evidence of accuracy of solutions is based on a comparison of results with the ODE formulation of Section 5.3 that is implemented in Section 5.4.4 with numerical methods that assure accurate solutions. The maximum deviation over the 20 sec simulation of solutions obtained using the SDIRK integrator in the present formulation and in the ODE formulation was  $2.5 \times 10^{-9}$  for  $\mathbf{q}$ ,  $5.7 \times 10^{-8}$  for  $\dot{\mathbf{q}}$ , and  $7.3 \times 10^{-7}$  for  $\ddot{\mathbf{q}}$ . These data support the validity of the Index 0 DAE formulation applied to the underlying ODE.

Simulations using MATLAB codes of Appendix 5.A confirm that accurate results are obtained with the Index 0 formulation. Numerical performance in obtaining solutions that satisfy all three forms of the constraint equations is shown to be comparable to the ODE formulation.

As with implementation of the ODE formulation in Section 5.4, ad-hoc derivation of equations of motion and programming of solution methods is oppressive. This motivates effort in the chapter to create computational implementations of both equation formulation and solution.

## 5.7 Code 5.7 for Tangent Space Simulation of Planar Systems

The general-purpose MATLAB computer Code 5.7 in noncentroidal coordinates of Appendix 5.A is presented to implement the *tangent space ODE* and *Index 0 DAE* formulations of Sections 5.3 and 5.5 for planar multibody systems that are modelled with noncentroidal body reference frames. Revolute, translational, and distance kinematic constraints of Section 3.2 are implemented, using derivatives presented in Section 3.2 and Appendix 5.B that are required for explicit Nystrom4 and RKFN45 and implicit Trapezoidal and SDIRK54 numerical integration methods of Section 4.8. Applied, gravitational, and internal forces defined by translational- and rotational-spring-damper-actuators (TSDA and RSDA) that are presented in Section 4.5 are implemented, using derivatives of Appendix 5.B. *Noncentroidal body fixed reference frames* for equations of motion derived in Section 4.2 are employed, with derivative expressions presented in Appendix 5.B. Fixed time step explicit *Nystrom4* and variable time step *RKFN45* algorithms and implicit variable time step *trapezoidal* and *SDIRK54* algorithms are implemented for numerical integration of tangent space ODE and Index 0 DAE of planar system dynamics.

Following an explanation of Code 5.7 in this section, numerical examples are presented in Section 5.8, including those treated with ad-hoc derivations and computer implementations in Sections 5.4 and 5.6. The essence of computer-aided kinematics and dynamics of mechanical systems is to enable computer formulation and solution of the equations of motion, without the painful detail of ad-hoc derivation of equations of motion and associated ad-hoc coding of numerical solution algorithms that are illustrated in Sections 5.4 and 5.6. The computer implementation presented in this section and applied in Section 5.8 is intended to introduce the reader to methods that are now available in commercial dynamic simulation software and advanced software that is likely to appear in the foreseeable future.

Components of Code 5.7 that interface with the user are presented in Section 5.7.1, followed by an outline of the body of the code, with which the user need not interact, in Section 5.7.2.

### 5.7.1 User Components of Code

The initial segment of Code 5.7 involves integration and error control parameters that underlie the tangent space formulation and associated numerical integration methods, as presented in Fig. 5.7.1. Lines 3 through 16 define parameters that control error in the *tangent space formulations* of Sections 5.3 and 5.5 and numerical integration methods of Section 4.8. Relatively tight *error control parameters* are used as defaults, which may be modified to seek greater computer efficiency or to more tightly constrain error. The maximum time step that is allowed in variable time step integrators is specified in line 17, followed by the initial time step that is defined in line 19. The index *hvar* is set in line 20 to select variable or constant time step. The final simulation time is defined in line 21. The integration option to be used is selected in line 26, from among four integrators and two formulations (ODE and Index 0 DAE) shown in lines 22 through 25. This is followed by selection of an option in line 27 to invert or factor the integration Jacobian for implicit integrators in the ODE formulation.

1 %AA Planar Tangent Space Multibody Simulation, Noncentroidal With Drivers

2 %Integration and Error Control Parameters

3 utol=10^-6; %Tolerance in solving for u

4 Btol=10^-6; %Convergence criteria in B iteration

5 intol=10^-6; %Tolerance in solving discretized equations of motion

```

6 Atol=10^-4; %Absolute error tolerance for variable step methods
7 Maxv=10; %Limit on norm of v
8 Maxu=10; %Limit on norm of u
9 MaxImpSoliter=20; %Limit on number of implicit integration iterations
10 MaxUiter=8; %Limit on number of U iterations
11 MaxJcond=3000; %Limit on magnitude of Jcond in Trap
12 R1nmax=30000; %Limit on initial residual in implicit integration
13 MaxECond=200; %Limit on magnitude of ECond
14 MaxPhiqUCond=1000; %Limit on magnitude of PhiqUCond
15 MaxBnormRat=10; %Maximum ratio of Bnorm to Bnorm at parameterization
16 MaxBCondRat=10; %Maximum ratio of BCond to BCond0 at parameterization
17 hmax=0.01; %Maximum time step
18 h0=0.001; %Initial time step
19 h=h0; %Value of time step to start simulation
20 hvar=1; %hvar=1, variable h;hvar=2, constant h
21 tfinal=2; %Final simulation time
22 %Integration Options %integ=1-Impl ODE Trap; integ=2-Impl ODE SDIRK54;
23 %integ=3-Impl Ind0 Trap; integ=4-Impl Ind0 SDIRK54;
24 %integ=5-Expl ODE Nystrom4; integ=6-Expl ODE RKFN45;
25 %integ=7-Expl Ind0 Nystrom4; integ=8-Expl Ind0 RKFN45
26 integ=2;
27 InvJ=2; % InvJ=1, Invert J; InvJ=2, Factor J

```

Figure 5.7.1 Integration and Error Control Parameters

Data for 16 applications that are implemented in Code 5.7 are indexed in lines 31 through 45 of Fig. 5.7.2 to application data files that are defined in the AppData function presented in Fig 5.7.3. The declaration in line 46 defines which application is implemented in the simulation to be carried out. The application data and parameter definitions of lines 47 through 50 are used throughout the code to pass application data for each simulation. If required for definition of initial conditions that are consistent with constraints, the user may enter code following line 51. Finally, output data desired are defined for each application, as illustrated for the double pendulum of app = 1 in lines 419 to 427.

```

30 %Applications
31 %app=1, Double Pendulum- Centroidal Data
32 %app=2, Quick Return-Centroidal Data
33 %app=3, Lumped Mass Coil Spring-5 masses
34 %app=4, Lumped Mass Coil Spring-10 masses
35 %app=5, Three Body Translational Model
36 %app=6, Slider-Crank
37 %app=7, Rotating Disk with Translating Body
38 %app=8, Multiple Slider-Crank
39 %app=9, Flywheel-Spring
40 %app=10, Loader
41 %app=11, Slider-Crank-2
42 %app=12, Triple Slider-Crank
43 %app=13, 3-body Slider-Crank-2 with C0 applied force
44 %app=15, Double Pendulum-noncentroidal coordinates
45 %app=16, Quick Return-noncentroidal coordinates
46 app=15; %Application selected for simulation
47 [nb,ngc,nh,nhc,nd,nc,nv,nu,NTSDA,NRSDA,PJDT,PMRD,PTSDAT, ...
48 PRSDAT,q0,qd0]=AppData(app); %Data from AppData(app) Function

```

```

49 par=[nb;ngc;nh;nhc;nd;nc;nv;nu;g;utol;Btol;intol;Atol;h0;hvar;...
50 NTSDA;NRSDA;app]; %Data available in functions
51 %Initial condition calculation, if required

417 %Data of Interest (Enter for each application)
419 if app==1 %Double Pendulum
420 x1(n)=q(1);
421 y1(n)=q(2);
422 phi1(n)=q(3);
423 x2(n)=q(4);
424 y2(n)=q(5);
425 phi2(n)=q(6);
426 phi2m1(n)=phi2(n)-phi1(n);
427 end

```

Figure 5.7.2 Application Data in Main Code

The third component of user entered code is the *AppData* function shown for the double pendulum in Fig. 5.7.3. The *AppData Function* definition of lines 8 through 46, based on data entered for each application, is passed to the main code and functions that are executed in calls throughout the code. Lines 8 through 18 define the dimension of variables that characterize the model. If entered incorrectly, the code will fail in ways that are difficult to understand from MATLAB error messages.

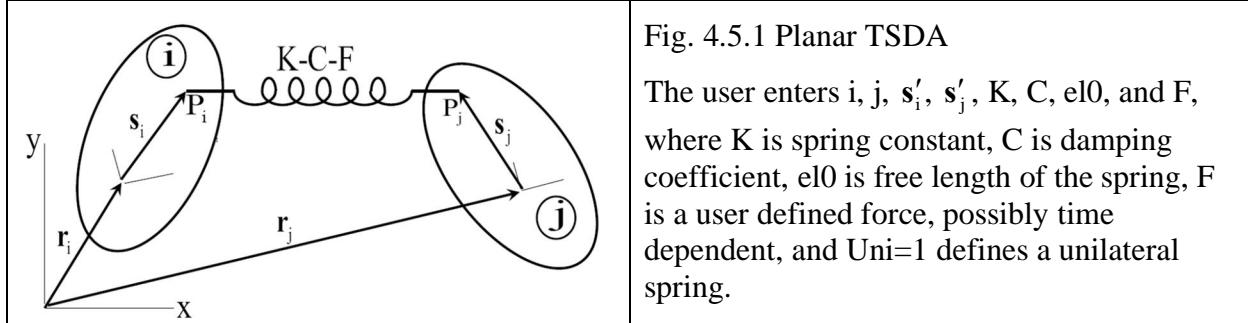
The *Planar Joint Data Table (PJDT)* is defined in lines 19 through 24. A detailed guide to entering data into the PJDT is provided in Section 3.9.1 and is not repeated here. It contains data defined in lines 24 and 25, as entered for the planar double pendulum.

The *Planar Mass Data Table (PMDT)* is defined in lines 26 through 30. It enables entry of mass, moment of inertia relative to the origin of the body reference frame, and the *vector locating the centroid* in the body reference frame for each body in the system, as follows:

```
PMDT=[[m1;J1;sc1],[m2;J2],...,[mnb;Jnb];scnb]
```

The *Planar TSDA Data Table (PTSDAT)* is defined in lines 32 through 36. More specifically, a template for TSDAT defined in Fig. 4.5.1 is repeated here for clarity.

```
PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F;Uni];
```



The *Planar RSDA Data Table (PRSDAT)* is defined in lines 34 through 39. More specifically, a template for RSDA R defined in Fig. 4.5.2 is repeated here for clarity.

```
PRSDAT(:,R)=[i;j;K;C;phi0;T];
```

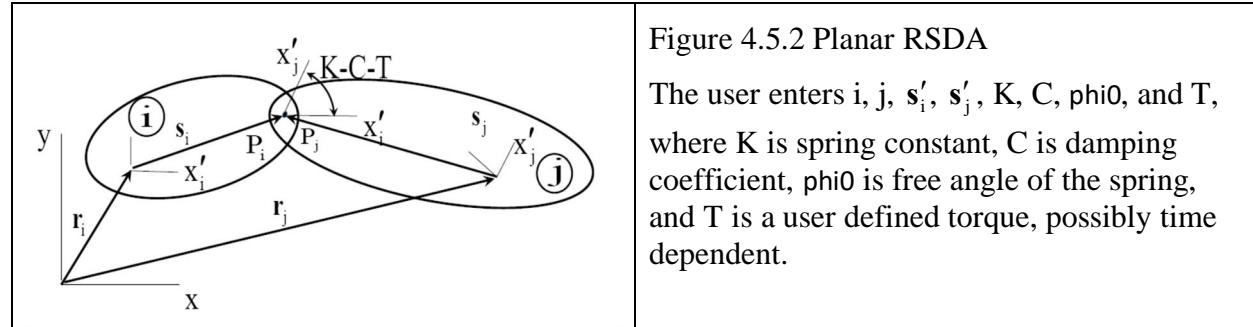


Figure 4.5.2 Planar RSDA

The user enters  $i$ ,  $j$ ,  $s'_i$ ,  $s'_j$ ,  $K$ ,  $C$ ,  $\phi_0$ , and  $T$ , where  $K$  is spring constant,  $C$  is damping coefficient,  $\phi_0$  is free angle of the spring, and  $T$  is a user defined torque, possibly time dependent.

Finally, *initial generalized coordinates* and *initial velocities* are defined in lines 44 through 46. If computation is required to define consistent initial data, code doing so is entered in the main code, following line 51 in Fig. 5.7.2.

```

8 if app==1 %Double Pendulum-centroidal model
9 nb=2; %Number of bodies
10 ngc=3*nb; %number of generalized coordinates
11 nh=2; %Number of time independent holonomic constraints
12 nhc=4; %Number of time independent holonomic constraint equations
13 nd=0; %Number of drivers
14 nc=nhc+nd; %Number of constraint equations, including drivers
15 nv=ngc-nc;
16 nu=nc;
17 NTSDA=1; %Number of TSDA force elements
18 NRSDA=2; %Number of RSDA force elements
19 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
20 %PJDT(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
21 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD),
22 %i&j=bodies connected, sipr&sjpr=vectors to Pi&Pj,
23 %d=dist., vipr&vjpr=vectors along translational axis
24 PJDT(:,1)=[1;1;0;-ux;zer;0;zer;zer]; %Rev-Body 1 to Ground
25 PJDT(:,2)=[1;1;2;ux;-ux;0;zer;zer]; %Rev-Body 1 to Body 2
26 %PMDT(4,nb): Mass Data Table
27 %PMDT=[[m1;J1;s1c],[m2;J2;s2c],...,[mn;Jn;snc]] (mi is mass of body i,
28 %ji is polar moment of inertia relative to the origin of the body fixed frame,
29 %and sic is the body fixed vector from the origin of the body fixed frame to
30 %the centroid)
31 PMDT=[[1;0.3;zer],[1;0.3;zer]];
32 %PTSDAT(11,NTSDA) TSDA Data Table
33 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F;Uni]; T=TSDA No.,
34 %i&j=bodies conn.,sipr&sjpr=vectors to Pi&j, K=spring constant,
35 %C=damping coefficient,el0=spring free length,F=const. force
36 %Unilateral spring is defined by setting Uni=1
37 PTSDAT(:,1)=[1;0;zer;-ux-uy;10^5;0;1;0;1];
38 %PRSDAT(6,NRSDA): RSDA Data Table
39 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=RSDA No.,
40 %i&j=bodies connected, K=spring constant,
41 %C=damping coefficient,phi0=spring free angle,T=constant torque

```

```

42 PRSDAT(:,1)=[1,0,0,0,pi/2,0];
43 PRSDAT(:,2)=[1,2,10^5,10^4,0,0];
44 %Initial generalized coordinates
45 q0=[1;0;0;3;0;0];
46 qd0=[0;0;0;0;0;0];
47 end

```

Figure 5.7.3 AppData Function, Double Pendulum

### 5.7.2 Computational Components of Code

Computational flow in the main program, which requires no input from the user, is outlined in Fig. 5.7.4. Data for integration that are defined in Section 5.7.1 are initialized following line 144. *Criteria for reparameterization* for both implicit and explicit integrators are defined following lines 149 and 171. Code to implement reparameterization is called following line 170, if *error control tolerances* are exceeded. Code that implements the eight numerical integration methods supported is outlined following line 212. Integration results for tangent space coordinates ( $v$ ,  $vd$ , and  $vdd$ ) that are computed using the selected integration method are processed to obtain generalized coordinates ( $q$ ,  $qd$ ,  $qdd$ , and  $Lam$ ) following line 267. Output data of general interest and for each of the applications discussed in Section 5.7.1 are defined following lines 335 and 417, respectively. Finally, position, velocity, and acceleration constraint errors are computed following line 600.

```

114 %Initialize Data For Integration

148 %Reparameterization Criteria
149 if integ<5 %Start implicit reparameterization criteria
176 if integ>4 %Start explicit reparameterization criteria

194 if Cr>1 %Reparameterization

212 % Integration
213 if integ==1 %Implicit ODE trapezoidal
214 [v,vd,vdd,R1n,ImpSoliter,JCond,Jinv,Jinvriter,h,nch]=...
215 ImplicitODETrap(n,tn,npar,Vv,Vvd,Vvdd,Uu,q0,V,U,B, ...
216 h,hmax,nch,PMDT,PTSDAT,PRSDAT,PJDT,par,InvJ,Jinv);

257 [v, vd, vdd, ECond]=ExplicitODENyström4(n,tn,Vv,Vvd,Uu,V,U,B,q0, ...
258 h,npar,PMDT,PTSDAT,PRSDAT,PJDT,par);

287 %Process Results
295 %Evaluate q
311 %Evaluate B and qd
325 % Evaluate qdd and Lam (Lam evaluation is a postprocessing step)

335 %Calculate output data
417 %Data of Interest (Enter for each application)
600 %Calculate constraint error

```

Figure 5.7.4 Main Code Computational Flow

*Computing functions* that underlie the main code that is outlined in Fig. 5.7.4 are identified in Fig. 5.7.5. Computing functions include the *Add function* that enables adding nonzero submatrices to *sparse matrices*, below and to the right of the address of the (1,1) term in

the submatrix, which is added to the underlying matrix that was initialized to zero. The function *ATran* evaluates the orientation transformation matrix. Functions *BEval* and *usolve* evaluate the matrix **B** in Eq. (5.2.20) and dependent generalized coordinates in Eq. (5.2.17) and *Param* evaluates terms that are required for the tangent space parameterization of Section 5.2.

Vector partition functions *parPart*, *qPart*, and *xPart* support *partitioning of vectors* into components that are used throughout the code. Similarly, the Constraint and Data Table Partition Functions provide access to elements of the data tables that are required to implement computation.

Constraint and derivative evaluation functions listed evaluate kinematic constraint expressions and derivatives defined in Section 3.2 and Appendix 5.B that are needed in implementing the tangent space ODE and Index 0DAE formulations and numerical integration of the associated equations of motion. Care is taken to account for the fact that ground is designated by  $j = 0$  and its generalized coordinates are constant, yielding no derivative contribution, but to include geometric quantities that define the constraint of body  $i$  with ground. Similarly, the Kinetic and Derivative Evaluation Functions listed evaluate inertial and generalized force terms and derivatives defined in Sections 4.2, 5.3, 5.5, and Appendix 5.B that are required in formulation and solution of the equations of motion.

Finally, the *numerical integration functions* listed carry out the numerical integration process, where *ODEfunct* evaluates the *tangent space ODE* of Section 5.3 and *Ind0ODEfunct* evaluates the *tangent space Index 0 DAE* of Section 5.5, for use in the two explicit numerical integrators. Similarly, *JacobODE* and *JacobInd0* evaluate the implicit numerical integration Jacobians defined in Sections 5.3 and 5.5 and *ResidODE* and *ResidInd0* evaluate the associated residuals of the equations of motion for use in the implicit numerical integrators.

Internal details of the Computing Functions listed in Fig. 5.7.5 are not presented here, since each of the functions is documented internally and the user need not modify these functions in applications. In comparison to computing functions for planar kinematic analysis in Fig. 3.9.5, however, it is noted that many additional functions are required. Basic kinematic computing functions are identical in both codes, but additional kinematic computing functions appear that support implicit numerical integration; e.g., *GamsqqdEval*, *P3Eval*, and *P4Eval*. New computing functions, of course, appear for dynamics.

#### Computing Functions

- Add
- ATran
- BEval
- usolv
- Param

#### User Input Function

- AppData

#### Vector Partition Functions

- parPart
- qPart
- xPart

Constraint and Data Table Partition Functions

- DistPart
- RevPart
- TranPart
- PTSDATPart
- PRSDATPart

Constraint and Derivative Evaluation Functions

- PhiEval
- PhiqEval
- GamEval
- GamsqqdEval
- P2Eval
- P3Eval
- P4Eval
- P5Eval

Kinetic and Derivative Evaluation Functions

- MEval
- M2Eval
- QAEval
- QAsqqd
- SEval
- Ssqqd

Numerical Integration Functions

- ExplicitNystrom4
- ExplicitRKF45
- ODEfunct
- IndODEfunct
- ImplicitTrap
- ImplicitSDIRK54
- JacobODE
- JacobInd0
- ResidODE
- ResidInd0

Figure 5.7.5 Computing Functions

### 5.7.3 *Code Output*

In addition to output defined for each application in the AppData function, the code reports the following arrays of values at each time step that shed light on the application and performance of the code:

- PosConstrNorm; norm of position constraint error
- VelConstrNorm; norm of position constraint error
- AccConstrNorm; norm of acceleration constraint error
- Biterrpt; number of iterations required to update  $\mathbf{B}$
- Iterurpt; number of iterations required to update  $\mathbf{u}$
- hrpt; stepsize  $h$  for variable step integrators
- ImpSoliterrpt; number of iterations required in implicit integration
- JCondrpt; Condition number of matrix in implicit equation solution
- jRepar; total number of reparameterizations in simulation
- $Q$ ; array of values of  $\mathbf{q}$
- $Qd$ ; array of values of  $\dot{\mathbf{q}}$
- $Qdd$ ; array of values of  $\ddot{\mathbf{q}}$

$Vv$ ; array of values of  $v$   
 $Vvd$ ; array of values of  $\dot{v}$   
 $Vvdd$ ; array of values of  $\ddot{v}$   
 $LLam$ ; array of values of Lagrange multiplier vector  $\lambda$   
 $TE$ ; total energy of system

Code 5.7 of Appendix 5.A is a tool for numerical simulation and experimentation with the tangent space ODE and Index 0 DAE formulations of planar multibody dynamics. It is typical of code that enables the engineer to create models of mechanical systems, carry out simulations, and investigate the effects of model and computational variations, without investing extensive time and effort in ad-hoc derivation and coding in specific applications. Development and use of such codes is the essence of computer aided kinematics and dynamics of mechanical systems.

## 5.8 Planar System Simulation Using Code 5.7

Six planar systems are simulated using the general-purpose tangent space Code 5.7 of Appendix 5.A. Kinematic and kinetic characteristics of the models are defined and entered in the AppData function defined in Section 5.7. Code 5.7 is then run and results are analyzed. The technical complexity of ad-hoc derivation of equations and ad-hoc coding that are characteristic of examples presented earlier in the chapter is contrasted with the ease of use of a general-purpose computer code. This is the essence of the field of *computational dynamics*.

### 5.8.1 Double Pendulum

#### 5.8.1.1 Model and Data Set

The *planar double pendulum* that is treated with ad-hoc derivation and numerical implementation in Section 5.4.1 is shown in Fig. 5.8.1, with a *unilateral spring*  $K_3$  between body 1 and ground. The *AppData Function* for this model, using data of Section 5.4.1, is shown in Fig. 5.8.2 for the bodies in an initial configuration along the global x axis and a free angle of RSDA1 of  $\pi/2$  radians.

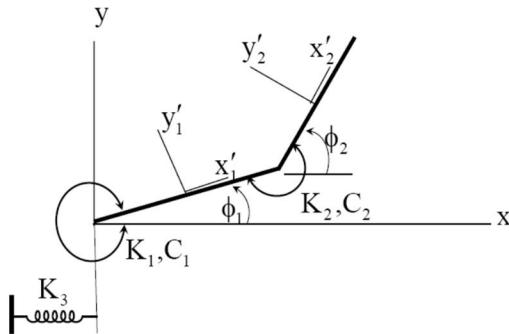


Figure 5.8.1 Double Pendulum

```

8 if app==1 %Double Pendulum-centroidal model
9 nb=2; %Number of bodies
10 ngc=3*nb; %number of generalized coordinates
11 nh=2; %Number of time independent holonomic constraints
12 nhc=4; %Number of time independent holonomic constraint equations
13 nd=0; %Number of drivers
14 nc=nhc+nd; %Number of constraint equations, including drivers
15 nv=ngc-nc;
16 nu=nc;
17 NTSDA=1; %Number of TSDA force elements
18 NRSDA=2; %Number of RSDA force elements
19 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
20 %PJDT(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
21 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD),
22 %i&j=bodies connected, sipr&sjpr=vectors to Pi&Pj,
23 %d=dist., vipr&vjpr=vectors along translstional axis
24 PJDT(:,1)=[1;1;0;-ux;zer;0;zer;zer]; %Rev-Body 1 to Ground
25 PJDT(:,2)=[1;1;2;ux;-ux;0;zer;zer]; %Rev-Body 1 to Body 2
26 %PMDT(4,nb); Mass Data Table
27 %PMDT=[[m1;J1;s1c],[m2;J2;s2c],...,[mnb;Jnb;snb]] (mi is mass of body i,

```

```

28 %Ji is polar moment of inertia relative to the origin of the body fixed frame,
29 %and sic is the body fixed vector from the origin of the body fixed frame to
30 %the centroid)
31 PMDT=[[1;0.3;zer],[1;0.3;zer]];
32 %PTSDAT(11,NTSDA) TSDA Data Table
33 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F;Uni]; T=TSDA No.,
34 %i&j=bodies conn.,si&jpr=vectors to Pi&j, K=spring constant,
35 %C=damping coefficient,el0=spring free length,F=const. force
36 %Unilateral spring is defined by setting Uni=1
37 PTSDAT(:,1)=[1;0;zer;-ux-uy;10^5;0;1;0;1];
38 %PRSDAT(6,NRSDA): RSDA Data Table
39 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=RSDA No.,
40 %i&j=bodies connected, K=spring constant,
41 %C=damping coefficient,phi0=spring free angle,T=constant torque
42 PRSDAT(:,1)=[1,0,0,0,pi/2,0];
43 PRSDAT(:,2)=[1,2,10^5,10^4,0,0];
44 %Initial generalized coordinates
45 q0=[1;0;0;3;0;0];
46 qd0=[0;0;0;0;0;0];
47 end

```

Figure 5.8.2 AppData Function, Double Pendulum

### 5.8.1.2 Nominal Results

Numerical results for 10 sec simulations, carried out with  $K_3 = 0$ , other data as in Section 5.4.1, and all four integrators, are identical to those reported in Section 5.4.1. Timing results for the simulation shown on the right of Fig. 5.4.2 indicate that CPU time using the general-purpose Code 5.7 on a laptop workstation was approximately three times that required by the hand derived and programmed Code 5.4.1 in Appendix 5.A, for the same simulation data and integrator. Specifically, for the 10 sec simulation reported on the right of Fig. 5.4.2 using the trapezoidal integrator, Code 5.7 required 34 CPU sec and Code 5.4.1 required 11 CPU sec. The 23 CPU sec. additional cost with the general-purpose code more than pays for the enormous difference in person hours required in the ad-hoc implementation. See Section 5.8.1.5 for details.

### 5.8.1.3 Results for Stiff Systems

As a test of the ability of explicit and implicit numerical integration methods to accurately simulate moderately *stiff systems*, spring and damping parameters were set to  $K_1 = 20$ ,  $C_1 = 10$ ,  $K_2 = C_2 = 1,000$ , and  $K_3 = 0$ . Initial conditions are as in Section 5.4.1. This models a relatively stiff single bar flexible pendulum. Results presented in Fig. 5.8.3 were obtained using the SDIRK54 implicit integration method with *variable step size* and an *error tolerance*  $Atol = e-5$ . This integrator is the best of the six used in handling stiff systems and yielded results that approximate swinging motion of a single flexible bar, including damping relative motion of the two bodies that make up the model and settling to a vertical downward configuration. A simulation carried out with the explicit *Nystrom4* integration method with a constant step size  $h = e-3$  yields erroneous results. This is to be expected when using an explicit integrator to simulate a *stiff system*. With this moderately stiff example, setting  $h = e-5$  yielded results with the *Nystrom4* method that are close to those in Fig. 5.8.3. Simulations carried out for this moderately stiff example with  $h = e-3$  in the implicit trapezoidal integrator yielded reasonable results, even though it is known to be less stable than SDIRK54. Finally, the explicit variable step size RKFN45 method yielded accurate results, at the cost of reducing step size well below  $e-3$ .

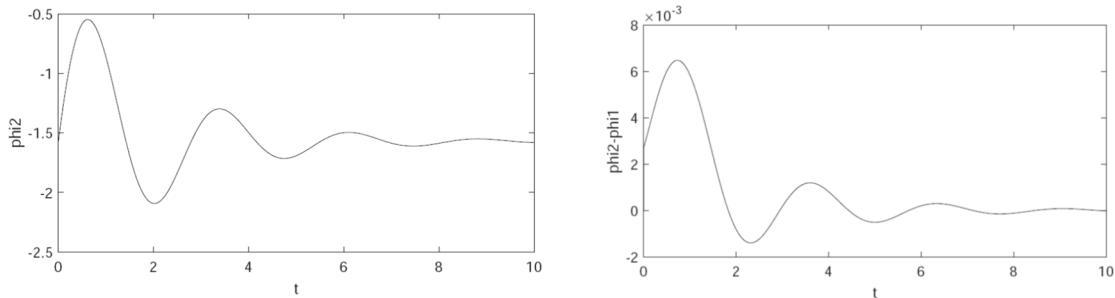


Figure 5.8.3 Simulations with Moderately Stiff System;  $K_1=20$ ,  $C_1=10$ ,  $K_2=C_2=1,000$

A more severe test of the effect of stiffness on numerical performance of solution methods is obtained by setting  $K_1 = 20$ ,  $C_1 = 10$ ,  $K_2 = C_2 = e6$ , and  $K_3 = 0$ . Initial conditions are as above. In this case, all explicit integrators failed to even complete a simulation. The SDIRK54 integrator with maximum step size  $e-3$  yielded the solution of Fig. 5.8.4. The implicit trapezoidal method with constant step sizes  $e-3$  through  $e-5$  provided qualitatively reasonable solutions over the simulation interval, but failed to accurately simulate the initial transient. The reader is encouraged to use Codes 5.4.1 and 5.7 in Appendix 5.A to carry out simulations with varying parameter values to gain an appreciation for the strengths and weaknesses of the various numerical integration methods. Constraint errors encountered in all simulations reported were comparable to values reported in Table 5.4.1 of Section 5.4.1.

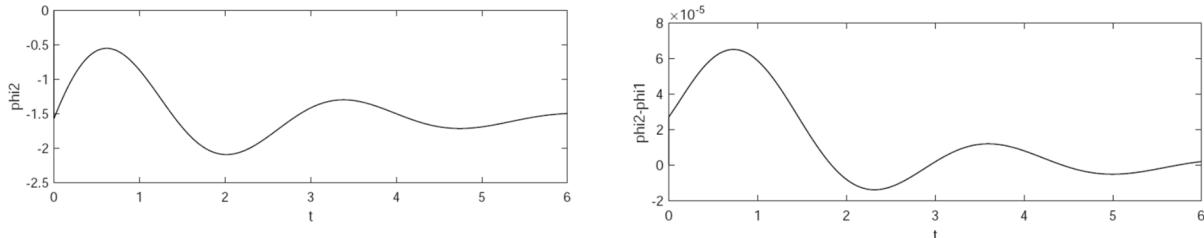


Figure 5.8.4 Simulations with Stiff System;  $K_1 = 20$ ,  $C_1 = 10$ ,  $K_2 = C_2 = 10^5$ ,  $K_3 = 0$

Finally, a spectrum of simulations was carried out using Code 5.7 with  $K_1 = C_1 = 0$ , initial conditions with the bars initially at rest along the x axis; i.e.,  $\dot{\mathbf{q}}^0 = \mathbf{0}$  and  $\mathbf{q}^0 = [1 \ 0 \ 0 \ 3 \ 0 \ 0]^T$ , constant or maximum step size  $h = 0.001$ ,  $\text{intol}=e-6$ ,  $\text{utol}=\text{Btol}=e-12$ , and  $\text{Atol}=e-5$  for a range of values of  $K_2$ ,  $C_2$ , and  $K_3$ . The data set for these simulations is given in Fig. 5.8.2. In ten second simulations, the bars fall due to gravity until the first bar is oriented vertically downward, where it encounters the *unilateral spring* shown in Fig. 5.8.1. The first bar deforms the unilateral spring, the torsional spring and damper between the bars are activated, and the bars rebound toward their initial configuration.

Table 5.8.1 is a tabulation of successful and failed simulations carried out with three variable step integration methods, for spring and damper values given in the bottom three rows of the table. A number in the first three rows of the table is the average step size for an ODE simulation that ran successfully and an x denotes a simulation that failed. The reader is cautioned that average step size does not adequately represent the ability of a *variable step integrator* to automatically reduce step size and achieve accurate results through each transient event.

The fixed step size explicit Nystrom4 method failed for even modest stiffness. The variable step size explicit *RKFN45* method adapts to treat some stiffness, but fails long before the implicit methods. As noted in the second column, it succeeds only by tracking the *high frequency response* with a small step size, even though the amplitude of oscillation is very small. The implicit *trapezoidal* method successfully simulates quite *stiff systems*, as well as the *SDIRK54* implicit method that is among the best integrators for stiff systems. Data in Table 8.5.1 are for the ODE formulation. Slightly smaller step sizes were required for the Index0 formulation, which performed comparably.

Table 5.8.1 Simulation Success (Avg. Step Size) and Failure (x)

	RKFN45	8.5e-4	8.6e-5	x	x	x
trapezoidal	9.9e-4	9.6e-4	9.3e-4	6.1e-4	x	
SDIRK54	9.9e-4	9.9e-4	9.9e-4	9.9e-4	9.8e-4	
$K_2$	e4	e5	e6	e7	e8	
$C_2$	e3	e4	e5	e6	e7	
$K_3$	e4	e5	e6	e7	e8	

To see the effect of stiffness and the transient response due to the stiff unilateral spring  $K_3$ , stiff spring  $K_2$ , and high damping  $C_2$ , 10 sec simulation results for data  $K_2 = K_3 = 10^5$ ,  $C_2 = 10^4$  and  $\text{Atol} = 10^{-3}$  with the SDIRK54 algorithm are presented in Fig. 5.8.5. Even with this level of stiffness, only a *single parameterization* was required in 32,049 time steps and maximum norms of position, velocity, and acceleration constraint errors were 2e-9, 3e-8, and e-6, respectively. The top plot in Fig. 5.8.5 shows the angle  $\phi_1$ . As the first bar encounters the unilateral spring, the pendulum rebounds, but loses amplitude due to damping and the associated reduction in total energy. The second plot shows that the relative angle  $\phi_2 - \phi_1$  between the bars is activated for short periods of time, due to *impact*, and is rapidly damped out. The *total energy* shown in the third plot confirms that energy is lost due to high damping over short periods of time following impacts. The bottom plot shows that step size  $h$  is reduced significantly in neighborhoods of impacts and increases to the maximum  $h = e-3$  that is allowed after *transient oscillations* dissipate. This illustrates how an implicit integrator with variable step size adapts to *stiff transients*, whereas explicit integrators cannot.

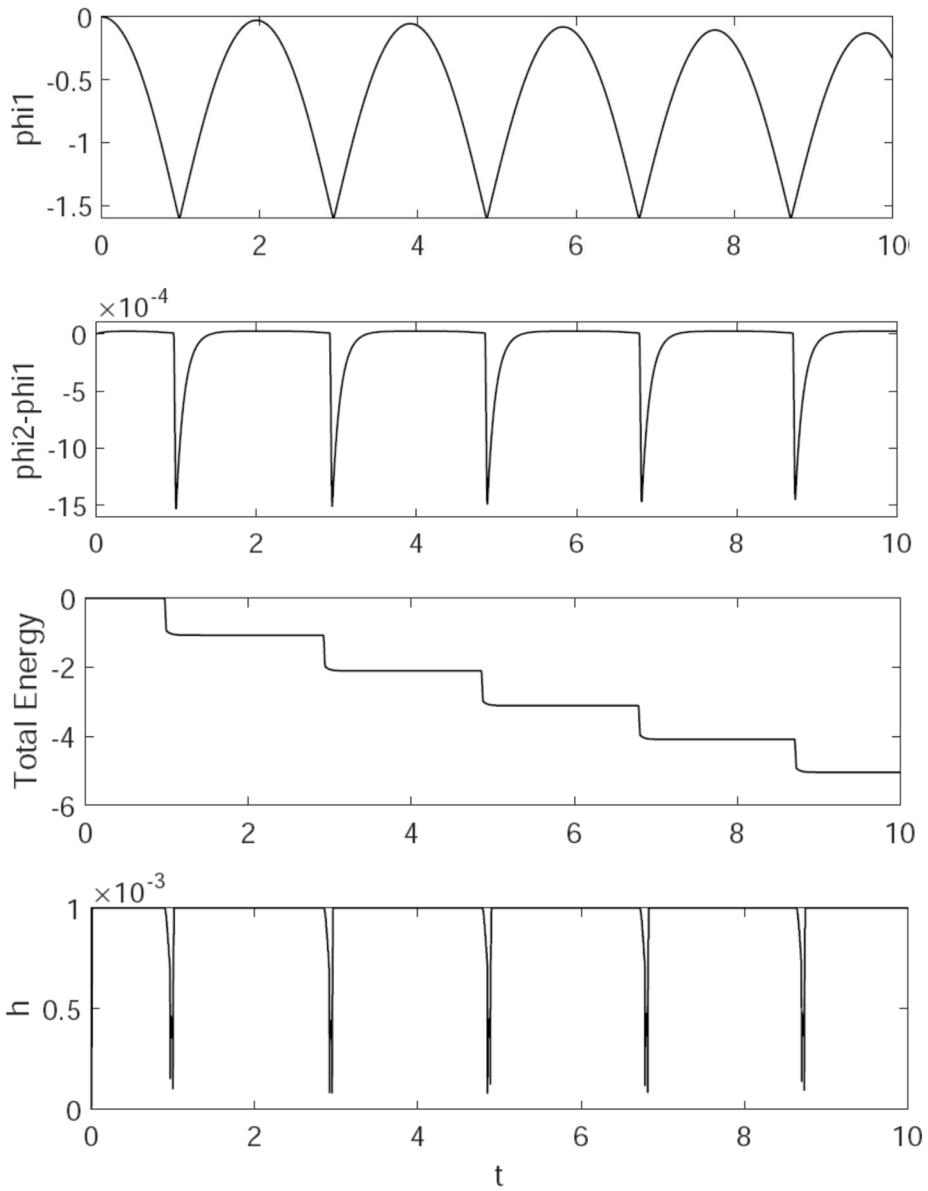


Figure 5.8.5 Simulation with  $K_1=C_1=0$ ,  $K_2=K_3=10^5$ ,  $C_2=10^4$ ,  $Atol=10^{-4}$

#### 5.8.1.4 Results for Highly Oscillatory System

To see the influence of high damping, coupled with stiff springs, in rapidly reducing transient response and enabling implicit integrators to reduce step size during transients and increase step size as transients damp out, it is instructive to retain high spring stiffness and reduce damping to zero. Setting  $K_1 = C_1 = C_2 = 0$ ,  $K_2 = K_3 = 10^5$ ,  $Atol=10^{-5}$ , and the maximum allowed step size  $h_{\max} = 10^{-3}$ , simulations are carried out with the *variable step size* explicit RKF45 implicit algorithm. In contrast with the ability of the implicit algorithm shown in Fig. 5.8.5 to reduce step size during transients and to increase step size after damping of the transient, as shown in Fig. 5.8.5, in this example the algorithm is forced to track *high frequency response* throughout the simulation. This is because the transient does not damp out. Thus, this

example is simply *highly oscillatory* and *not stiff*. It is interesting to note that the explicit RKFN45 algorithm performs as well in this situation as the implicit SDIRK54 algorithm.

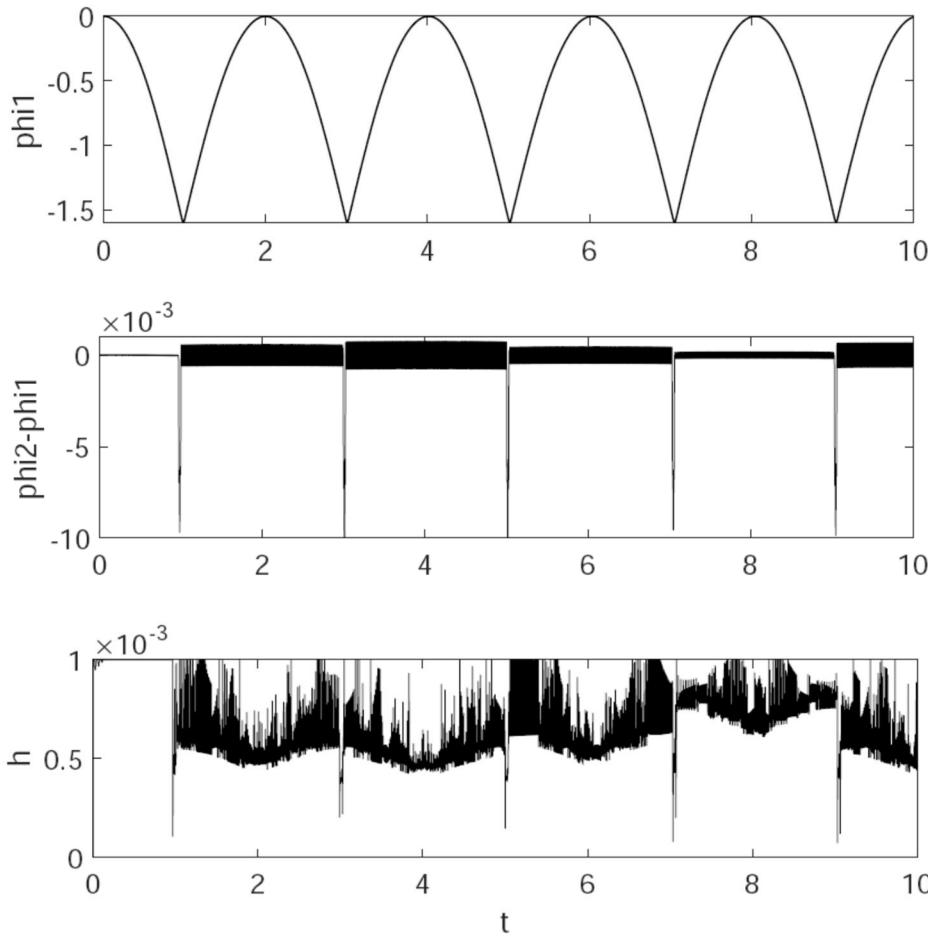


Figure 5.8.6 Simulation with RKFN45 and  $K1=C1=C2=0$ ,  $K2=K3=10^5$ ,  $Atol=10^{-5}$

### 5.8.1.5 Level of Effort Required

From the point of view of personnel time involved in the ad-hoc and computer generated equation approaches, compare the few minutes required to create the data set in Fig. 5.8.2 and launch the general-purpose code with the time required to derive the extensive expressions in Section 5.4.1 and to program and debug the ad-hoc Code 5.4.1. Even with the greatest of care, ad-hoc derivation of constrained equations of motion and the numerous derivatives that are required for numerical solution is time consuming and error prone. Finding derivation errors is time consuming and painful. Even more time consuming and painful is the process of creating and debugging ad-hoc computer code to implement numerical solution methods. A minimum of several person-days of effort is required by even the most proficient analyst/programmer for such an application, much less the extensive time required by we mere mortals. In contrast, implementing the same simulations with data presented in Fig. 5.8.2 and launching simulations with Code 5.7 in Appendix 5.A required only a few person-minutes of effort. Considering the high cost of engineering labor and the minimal cost of digital computer CPU time, the 23 CPU sec penalty in CPU time reported above is a bargain.

## 5.8.2 Quick Return Mechanism

### 5.8.2.1 Model and Data Set

The *quick return mechanism* shown in Fig. 5.8.6, whose kinematics were studied in Section 3.10.2, is driven by counterclockwise rotation of the flywheel (body 2) to cause oscillation of body 1 via the body 3 key that translates in the slot shown in body 1. The connecting rod between point  $P_1$  on body 1 and the centroid of the cutting tool (body 4) causes body 4 to move slowly to the left in a cutting stroke and to return more quickly to the right for the next cutting stroke.

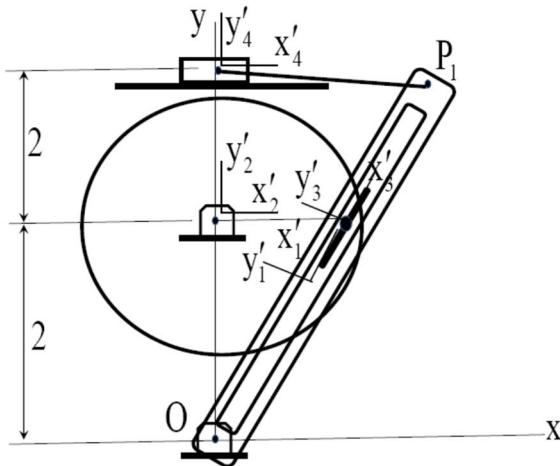


Figure 5.8.6 Quick Return Mechanism

The radius of the flywheel is 2 m, the distance between points O and  $P_1$  on body 1 is 4 m, and the length of the connecting rod that is modeled as a distance constraint is 2.5298 m. The centroid of body one is at its midpoint. The initial configuration of body 1 is  $\mathbf{r}^0 = [1.2 \quad 1.6]^T$  and  $\phi_1^0 = 0.9273$  rad. With  $m_1 = J_1 = 100$ ,  $m_2 = J_2 = 1000$ ,  $m_3 = J_3 = 1$ , and  $m_4 = J_4 = 50$  in SI units, the data set for the system in the *AppData Function* is given in Fig. 5.8.7. Initial velocities, with an initial angular velocity  $\dot{\phi}_2^0 = 6$  rad/sec of body 2, are calculated in the main code.

```

50 if app==2 %Quick Return-centroidal model
51 nb=4; %Number of bodies
52 ngc=3*nb; %number of generalized coordinates
53 nh=6; %Number of holonomic constraints
54 nhc=11; %Number of holonomic constraint equations
55 nd=0; %Number of drivers
56 nc=nhc+nd; %Number of constraint equations, including drivers
57 nv=ngc-nc; %Number of independent coordinates
58 nu=nc; %Number of dependent coordinates
59 NTSDA=0; %Number of TSDA force elements
60 NRSDA=0; %Number of RSDA force elements
61 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
62 %PJDT(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
63 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD),
64 %i&j=bodies connected, sipr&sjpr=vectors to Pi&Pj,

```

```

65 %d=dist., vipr&vjpr=vectors along translstional axis
66 PJDT(:,1)=[1;1;0;-2*ux;zer;0;zer;zer]; %Revolute-bar to ground
67 PJDT(:,2)=[1;2;0;zer;2*uy;0;zer;zer]; %Revolute-crank to ground
68 PJDT(:,3)=[1;2;3;1.5*ux;zer;0;zer;zer]; %Revolute-crank to key
69 PJDT(:,4)=[2;1;3;zer;zer;0;ux;ux]; %Trans.-bar to key
70 PJDT(:,5)=[2;4;0;zer;4*uy;0;ux;ux]; %Trans.-cutter to ground
71 PJDT(:,6)=[3;1;4;2*ux;zer;2.5298;zer;zer]; %Dist.-bar to cutter
72 %PMDT(4,nb): Mass Data Table
73 %PMDT=[[m1;J1;s1c],[m2;J2;s2c],...,[mn;Jn;snbc]] (mi is mass of body i,
74 %Ji is polar moment of inertia relative to the origin of the body fixed frame,
75 %and sic is the body fixed vector from the origin of the body fixed frame to
76 %the centroid)
77 PMDT=[[100;100;zer],[1000;1000;zer],[1;1;zer],[50;50;zer]];
78 %PTSDAT(11,NTSDA) TSDA Data Table
79 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F;Uni]; T=TSDA No.,
80 %i&j=bodies conn.,sipr&jpr=vectors to Pi&j, K=spring constant,
81 %C=damping coefficient,el0=spring free length,F=const. force
82 %Unilateral spring is defined by setting Uni=1
83 PTSDAT=zeros(11,1);
84 %PRSDAT(6,NRSDA): RSDA Data Table
85 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
86 %i&j=bodies connected, K=spring constant,
87 %C=damping coefficient,phi0=spring free angle,T=constant torque
88 PRSDAT=zeros(6,1);
89 %Initial generalized coordinates
90 q10=[1.2;1.6;0.9273];
91 q20=[0;2;0];
92 q30=[1.5;2;0.9273];
93 q40=[0;4;0];
94 q0=[q10;q20;q30;q40];
95 qd0=zeros(12,1); %Placeholder, qd0 calculated in main program
96 end

```

Figure 5.8.7 AppData Function, Quick Return Mechanism

### 5.8.2.2 Simulation Results and Analysis

Results of a simulation carried out with tolerances shown in Fig. 5.7.1, the variable step size SDIRK54 ODE integrator with tolerances  $\text{utol} = \text{Btol} = \text{e-8}$ ,  $\text{Atol} = \text{e-5}$ ,  $\text{hmax} = \text{e-3}$ , and a final time  $\text{tf} = 2.5$  sec are presented in Fig. 5.8.8. The cutting stroke that takes place over the first 0.6 sec of the stroke is at a speed less than 10 m/sec, whereas the top speed during the return stroke is 25 m/sec, thus the name “quick return”. The rapid change in velocity results in high accelerations. No cutting forces are included in this simulation, so acceleration predictions should be *conservative*. To confirm this is true, *total energy* is constant to four decimal places.

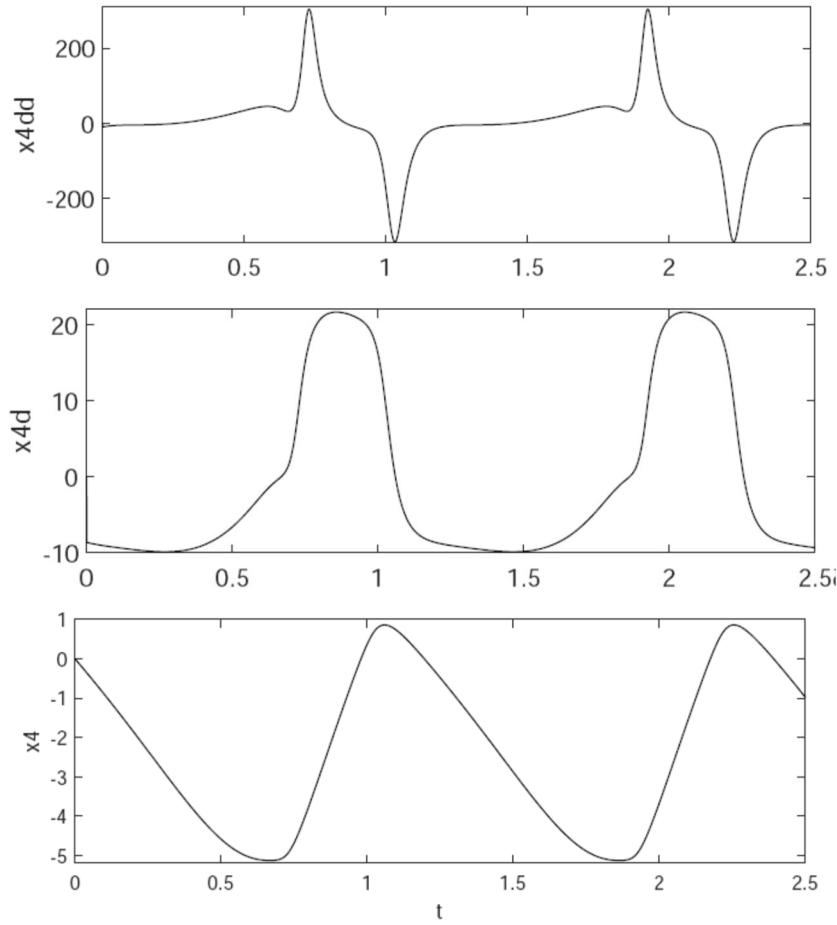


Figure 5.8.8 Quick Return Cutter Position, Velocity, and Acceleration vs Time

185 reparameterizations were required in 2,517 time steps (8 time steps per reparameterization). Maximum norms of position, velocity, and acceleration constraint errors were e-12, 2e-9, and 2e-8, respectively. Comparable results were obtained using the explicit Nystrom and RKN45 ODE integrators. Success of the explicit integrators indicates that this system is not stiff. Simulations carried out using all four integrators with the Index0 formulation yielded identical results and comparable numerical performance.

To see differences in solution obtained with the less accurate trapezoidal and SDIRK54 integrators, norms of the difference between system position ( $q$  diff), velocity ( $qd$  diff), and acceleration ( $qdd$  diff) predicted by the two integrators in the ODE formulation are shown in Fig. 5.8.9. From the  $x4$  position and velocity plots in Fig. 5.8.8, it is clear that relatively smooth behavior occurs before approximately  $t = 1$  sec, hence the small norms of position and velocity difference before this time, after which slight phase differences lead to larger norms of position and velocity differences.

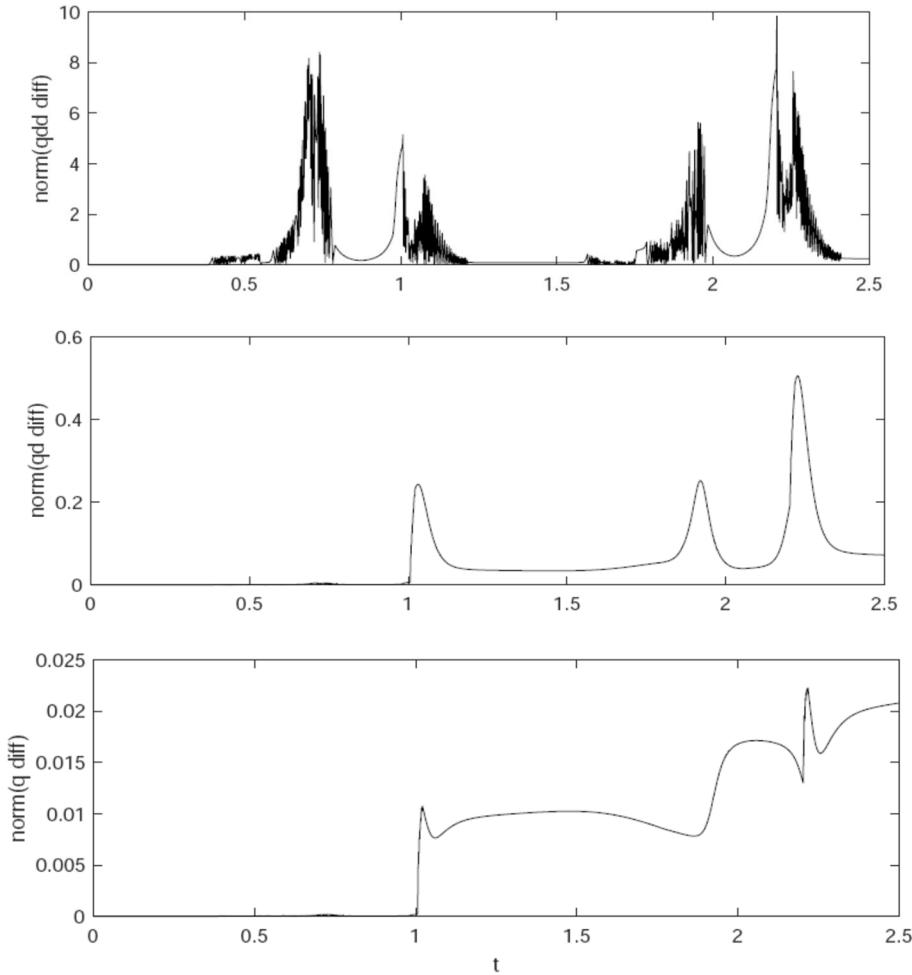


Figure 5.8.9 Quick Return, Norm Trap-SDIRK54 Solution Differences vs Time

The acceleration plot in Fig. 5.8.8 suggests that severe accelerations may lead to differences in prediction accuracies. This is born out in the top plot in Fig. 5.8.9. Recall that the norms presented in Fig. 5.8.9 are of the full vectors of 12 generalized coordinates. Individual generalized coordinate plots differ only slightly between the two approximate solutions, phase differences leading to the magnitudes of norms of differences.

### 5.8.2.3 Level of Effort Required

MATLAB timing routines reported 59 CPU seconds for each of the foregoing simulations, after less than two hours of person time to create the data set of Fig. 5.8.7 and run a simulation. Extending the argument of Section 5.8.1, numerous person days would be required in ad-hoc derivation and debugging of equations of motion and derivatives required for numerical integration. Even more time and pain would be required in creating and debugging ad-hoc computer code that might reduce the CPU time for simulation to approximately 33 cpu seconds. At essentially 0\$ per CPU minute on a laptop workstation, trading the numerous person days of effort and the uncertainty in accuracy of results for half a CPU minute increase in computer time is clearly the preferred option.

### 5.8.3 Surge Waves in a Coil Spring

#### 5.8.3.1 Model and Data Set

In many applications, a relatively long *coil spring* is used to arrest the motion of a moving body and return it to its original position. While a coil spring of length 1 m is a continuum of mass  $M$  kg and stiffness  $K$  n/m for the full spring, a reasonable model is formed by discretizing the mass and stiffness of the spring. The *lumped mass model* shown in Fig. 5.8.10 is constructed by dividing the spring into 5 equal lengths 0.2 m, each with stiffness  $k = 5K$ , and approximating the total mass as 5 lumped masses,  $m = M/5$  each. For computation,  $M = 1$  kg,  $K = 1000$  N/m, and the spring has free length  $L = 1$  m. The initial position of mass  $i$  is  $x_i = 0.2i$  m,  $i = 1, \dots, 5$ . To study *surge wave* behavior, motion of the system is initiated by giving mass 5 an initial velocity 1 m/sec in the negative x direction, which would be induced by *impact* with another body. The initial velocity of the first four masses is zero.

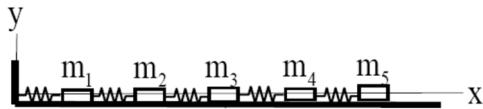


Figure 5.8.10 Lumped Mass Model of Coil Spring

Kinematics of the system is defined by prescribing a translational joint between each of the masses and ground. Each spring is attached at the centers of mass of the bodies it connects. The mass of each body is  $m = 0.2$  kg, and each spring has spring rate  $k = 5000$  N/m and free length  $\ell_0 = 0.2$  m. Data for the simulation are given in the *AppData Function* of Fig. 5.8.11.

```

98 if app==3 %Lumped Mass Coil Spring-5 masses-centroidal model
100 nb=5; %Number of bodies
101 ngc=3*nb; %number of generalized coordinates
102 nh=5; %Number of holonomic constraints
103 nhc=10; %Number of holonomic constraint equations
104 nd=0; %Number of drivers
105 nc=nhc+nd; %Number of constraint equations, including drivers
106 nv=ngc-nc;
107 nu=nc;
108 NTSDA=5; %Number of TSDA force elements
109 NRSDA=0; %Number of RSDA force elements
111 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
112 %PJDT(:,k)=[T;i;j;si(pr)sj(pr)d;vi(pr);vj(pr)]; k=joint No.,
113 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD),
114 %i&j=bodies connected, si(pr)&sj(pr)=vectors to Pi&Pj,
115 %d=dist., vi(pr)&vj(pr)=vectors along translational axis
116 PJDT(:,1)=[2;1;0;zer;zer;0;ux;ux]; %Tran.-Body 1 to Ground
117 PJDT(:,2)=[2;2;0;zer;zer;0;ux;ux]; %Tran.-Body 2 to Ground
118 PJDT(:,3)=[2;3;0;zer;zer;0;ux;ux]; %Tran.-Body 3 to Ground
119 PJDT(:,4)=[2;4;0;zer;zer;0;ux;ux]; %Tran.-Body 4 to Ground
120 PJDT(:,5)=[2;5;0;zer;zer;0;ux;ux]; %Tran.-Body 5 to Ground
122 %PMDT(4,nb): Mass Data Table
123 %PMDT=[[m1;J1;s1c],[m2;J2;s2c],...,[mn;Jn;snc]] (mi is mass of body i,
124 %Ji is polar moment of inertia relative to the origin of the body fixed frame,
125 %and sic is the body fixed vector from the origin of the body fixed frame to

```

```

126 %the centroid)
127 PMDT(:,1)=[0.2;0.1;zer];
128 PMDT(:,2)=[0.2;0.1;zer];
129 PMDT(:,3)=[0.2;0.1;zer];
130 PMDT(:,4)=[0.2;0.1;zer];
131 PMDT(:,5)=[0.2;0.1;zer];
133 %PTSDAT(11,NTSDA) TSDA Data Table
134 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F;Uni]; T=TSDA No.,
135 %i&j=bodies conn.,si&jpr=vectors to Pi&j, K=spring constant,
136 %C=damping coefficient,el0=spring free length,F=const. force
137 %Unilateral spring is defined by setting Uni=1
138 %PTSDAT=zeros(11,1);
139 PTSDAT(:,1)=[1;0;zer;zer;5000;0;0.2;0;0]; %Body 1 to Ground
140 PTSDAT(:,2)=[1;2;zer;zer;5000;0;0.2;0;0]; %Body 1 to 2
141 PTSDAT(:,3)=[2;3;zer;zer;5000;0;0.2;0;0]; %Body 2 to 3
142 PTSDAT(:,4)=[3;4;zer;zer;5000;0;0.2;0;0]; %Body 3 to 4
143 PTSDAT(:,5)=[4;5;zer;zer;5000;0;0.2;0;0]; %Body 4 to 5
145 %PRSDAT(6,NRSDA): RSDA Data Table
146 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
147 %i&j=bodies connected, K=spring constant,
148 %C=damping coefficient,phi0=spring free angle,T=constant torque
149 PRSDAT=zeros(6,1);
151 %Initial generalized coordinates
153 q0=[0.2;0;0;0.4;0;0;0.6;0;0;0.8;0;0;1;0;0];
154 qd0=zeros(12,1);-1;0;0];
156 end

```

Figure 5.8.11 AppData Function, Five Lumped Mass Coil Spring Model

### 5.8.3.2 Simulation Results and Analysis

Numerical results of simulation, carried out using trapezoidal integration in the ODE formulation with Code 5.7 of Appendix 5.A, include displacements  $\text{delxi} = \mathbf{x}_i - \mathbf{x}_i^0$ ,  $i = 1, \dots, 5$ , versus time for each lumped mass in Fig. 5.8.12. Note that with even this crude model, the *wave nature of motion* is clear. Mass 1 remains stationary for 0.017 sec, until it is encountered by the wave. It experiences reflection at the left end and begins rebound at 0.035 sec. The resulting wave to the right reaches mass 5 at 0.052 sec, the same delay as the initial wave reaching mass 1. This yields an estimate of the *wave speed* as  $1 \text{ m}/0.017 \text{ sec} = 59 \text{ m/sec}$ . The theoretical wave speed is (Goldstein, 1980)  $\sqrt{K/M} = \sqrt{1000} = 31.6 \text{ m/sec}$ . Due to the lack of sharpness in wave behavior of the lumped mass model, the wave speed predicted with this multibody model is only a rough approximation.

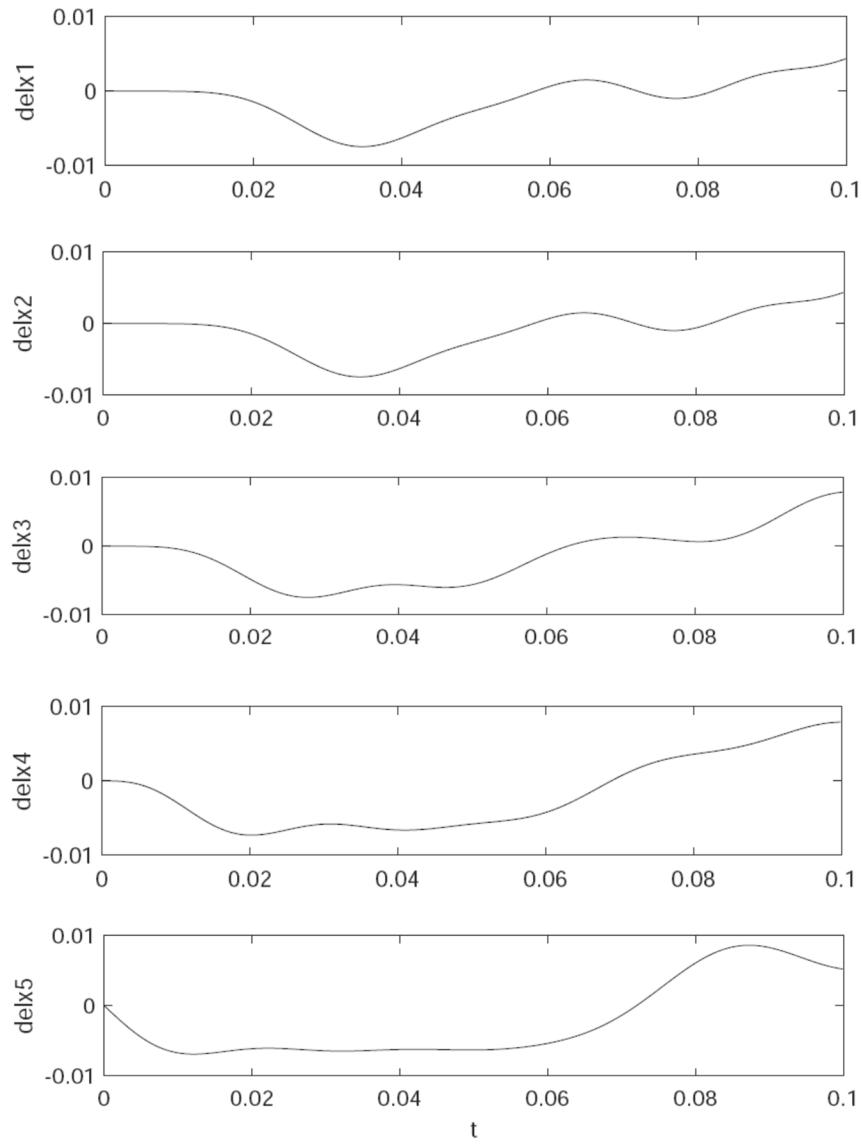


Figure 5.8.12 Lumped Mass Displacements Versus Time for 5 Mass Model

To obtain a better estimate of wave speed, a 10 *lumped mass model* is defined in App 4 of Code 5.7. In this model, each lumped mass is 0.1 kg and the springs have stiffness 10,000 N/m. Plots of displacement of masses 1 and 10 are presented in Fig. 5.8.13. In this model, mass 1 remains stationary for 0.024 sec, until it is encountered by the wave. It experiences reflection at the left end and begins rebound at 0.037 sec. The resulting wave to the right reaches mass 10 at 0.061 sec, the same delay as the initial wave reaching mass 1. This yields an estimate of the wave speed as  $1 \text{ m}/0.024 \text{ sec} = 42 \text{ m/sec}$ , a better estimate of wave speed, but still high.

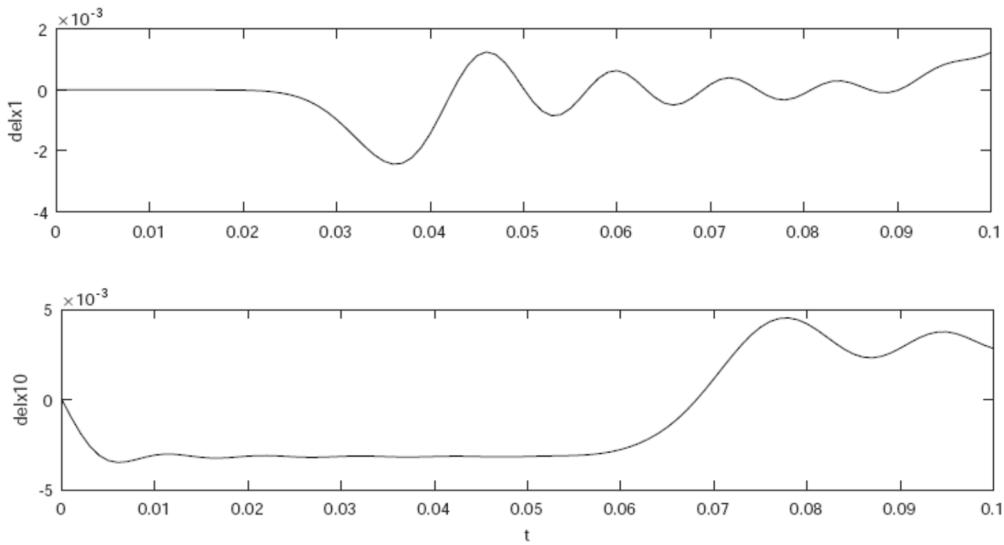


Figure 5.8.13 Lumped Mass Displacements Versus Time for 10 Mass Model

Simulations carried out with all seven of the other integrators in Code 5.7 yielded essentially identical results.

#### 5.8.4 Three-Body Mechanism with Translational Joints in Ground

##### 5.8.4.1 Model and Data Set

The *three-body mechanism* shown in Fig. 5.8.14 is comprised of translational joints between each of the bodies and one of the axes in ground, a distance constraint between bodies one and two, and springs between body one and ground and between bodies two and three. The body fixed x axis in each body is along its long dimension of the body. Vectors that define the joint between body one and ground are  $\mathbf{v}'_1 = \mathbf{u}_x$  and  $\mathbf{v}'_0 = \mathbf{u}_y$ ; vectors defining joints between bodies two and three and ground are  $\mathbf{v}'_2 = \mathbf{v}'_3 = \mathbf{u}_x$  and  $\mathbf{v}'_0 = \mathbf{u}_x$ ; the length of the distance constraint is  $d = 5$  m; spring characteristics are  $K_1 = K_2 = 10$  N/m,  $C_1 = C_2 = 0$ , and free lengths are set so there is no spring force when  $y_1 = 0$  and  $x_3 = x_2 = 1$ ; masses and moments of inertia of the bodies are  $m_1 = 5$  kg,  $m_2 = m_3 = 2$  kg, and  $J_1 = J_2 = J_3 = 1$  kgm<sup>2</sup>; and initial conditions are  $\mathbf{q}_1 = [0 \quad 5/\sqrt{2} \quad \pi/2]^T$ ,  $\dot{\mathbf{q}}_1 = [5/\sqrt{2} \quad 0 \quad 0]^T$ ,  $\mathbf{q}_3 = [(5/\sqrt{2})+1 \quad 0 \quad 0]^T$ ,  $\dot{\mathbf{q}}_3 = \mathbf{0}$ . The data set for simulation of this mechanism using Code 5.7 is presented in the *AppData Function* of Fig. 5.8.15.

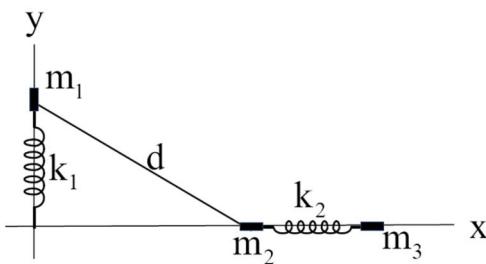


Figure 5.8.14 Mechanism with Translational Joints in Ground

```

237 if app==5 %Three Body Translational Model-centroidal model
238 nb=3; %Number of bodies
239 ngc=3*nb; %number of generalized coordinates
240 nh=4; %Number of holonomic constraints
241 nhc=7; %Number of holonomic constraint equations
242 nd=0; %Number of drivers
243 nc=nhc+nd; %Number of constraint equations, including drivers
244 nv=ngc-nc;
245 nu=nc;
246 NTSDA=3; %Number of TSDA force elements
247 NRSDA=0; %Number of RSDA force elements
249 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
250 %PJDT(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
251 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD),
252 %i&j=bodies connected, sipr&sjpr=vectors to Pi&Pj,
253 %d=dist., vipr&vjpr=vectors along translstional axis
254 PJDT(:,1)=[2;1;0;zer;zer;0;0.1*ux;uy]; %Tran-Bod1 to ground
255 PJDT(:,2)=[2;2;0;zer;zer;0;0.1*ux;ux]; %Tran-Bod2 to ground
256 PJDT(:,3)=[2;3;0;zer;zer;0;0.1*ux;ux]; %Tran-Bod3 to ground
257 PJDT(:,4)=[3;1;2;zer;zer;5;zer;zer]; %Dist.-Bod1 to Bod2
259 %PMDT(4,nb): Mass Data Table
260 %PMDT=[[m1;J1;s1c],[m2;J2;s2c],...,[mnb;Jnb;snbc]] (mi is mass of body i,
261 %Ji is polar moment of inertia relative to the origin of the body fixed frame,
262 %and sic is the body fixed vector from the origin of the body fixed frame to
263 %the centroid)
264 PMDT=[[5;1;zer],[2;1;zer],[2;1;zer]];
266 %PTSDAT(11,NTSDA) TSDA Data Table
267 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F;Uni]; T=TSDA No.,
268 %i&j=bodies conn.,si&jpr=vectors to Pi&j, K=spring constant,
269 %C=damping coefficient,el0=spring free length,F=const. force
270 %Unilateral spring is defined by setting Uni=1
271 %PTSDAT=zeros(11,1);
272 PTSDAT(:,1)=[1;0;zer;-10*uy;200;0;10;0;0]; %Bod 1 to Grnd
273 PTSDAT(:,2)=[2;3;zer;10*ux;10^3;10^2;11;0;0]; %Bod2 to bod3
274 PTSDAT(:,3)=[3;0;zer;7*ux;10^5;0;1.1;0;1]; %Bod 3 to Grnd, unilateral
276 %PRSDAT(6,NRSDA): RSDA Data Table
277 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
278 %i&j=bodies connected, K=spring constant,
279 %C=damping coefficient,phi0=spring free angle,T=constant torque
280 PRSDAT=zeros(6,1);
282 %Initial generalized coordinates
284 q0=[0;5/sqrt(2);pi/2;5/sqrt(2);0;0;(5/sqrt(2))+1;0;0];
285 qd0=[0;0;0;0;0;0;0;0];
287 end

```

Figure 5.8.15 AppData Function for Three-Body Translational Model

#### 5.8.4.2 Simulation Results and Analysis

Plots of position and velocity of the three bodies predicted by simulation using the SDIRK54 integrator with the ODE formulation in Code 5.7 are presented in Fig. 5.8.16. Five *reparameterizations* were required in 5016 time steps (1003 time steps per reparameterization) and maximum norms of position, velocity, and acceleration constraint error were 5e-15, 5e-14,

and 2.5e-13, respectively. All eight formulation and integrator combinations supported by Code 5.7 yielded essentially identical results.

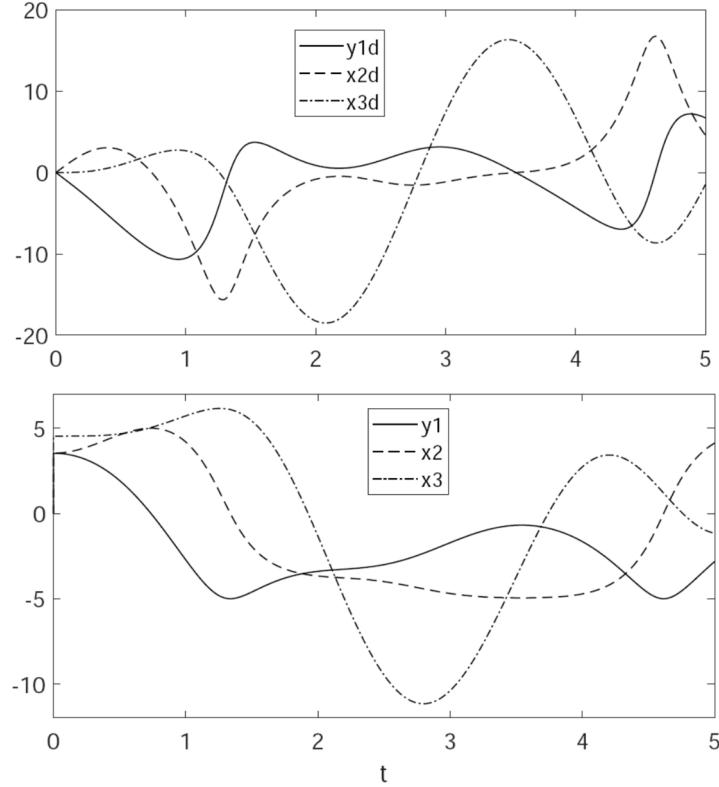


Figure 5.8.16 Position and Velocity vs Time

#### 5.8.4.3 Impact of Body 3 With a Unilateral Spring

Data for springs and dampers in the three-body mechanism are modified with  $K_1 = 200$  and  $C_1 = 0$  and  $K_2 = 10^3$  and  $C_2 = 10^2$ . A *stiff unilateral spring* is set to the right of body 3, with attachment to body 3 at its origin and to ground at  $7\mathbf{u}_x$ , free length  $\ell_0 = 1.1$ ,  $K_3 = 10^5$  and  $C_3 = 0$ . This models a *two-body impact damper* comprised of bodies 2 and 3 connected by a very stiff spring-damper to dissipate energy. Initial conditions are  $\mathbf{q}_0 = [0 \quad 5/\sqrt{2} \quad \pi/2 \quad (5/\sqrt{2})\mathbf{u}_x^T \quad 0 \quad ((5/\sqrt{2})+1)\mathbf{u}_x^T \quad 0]^T$  and  $\dot{\mathbf{q}}_0 = \mathbf{0}_{9 \times 1}$ . Plots of the position and velocity response of the three bodies, obtained with Code 5.7 using the SDIRK54 integrator, are presented in Fig. 5.8.17. This integrator effectively reduced step size to obtain an accurate solution through *stiff impact events* and increased step size after transients are damped.

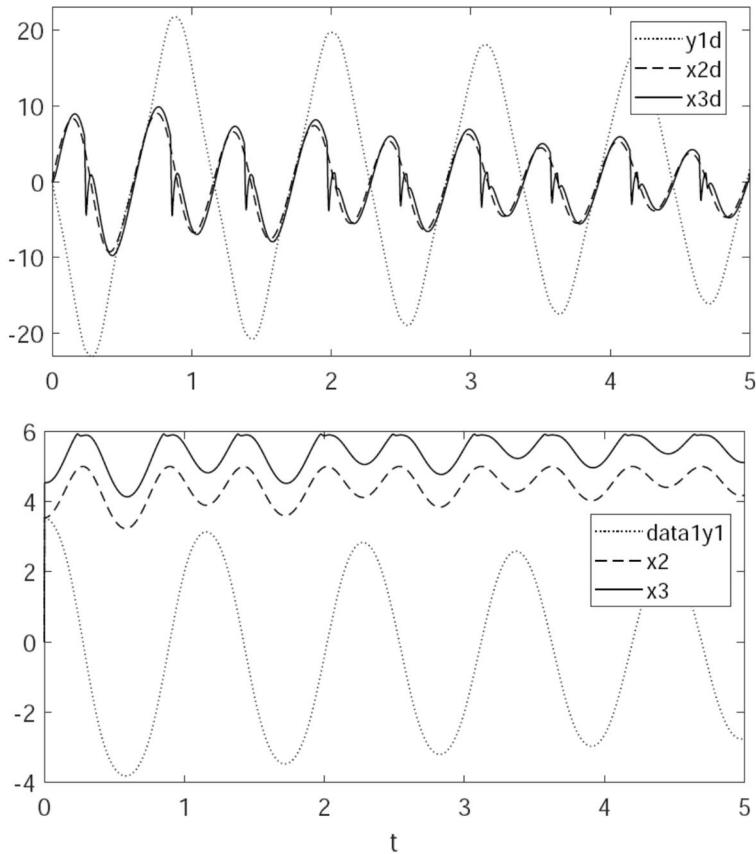


Figure 5.8.17 Body Positions and Velocities vs Time with Unilateral Spring

### 5.8.5 Slider-Crank

#### 5.8.5.1 Model and Data Set

Dynamics of the two-body model of a *slider-crank* mechanism that was analyzed kinematically in Section 3.2.3.3, shown in Fig. 5.8.18, is analyzed here. The radius of the crank is 1m and masses and moments of inertia of bodies one and two are  $m_1 = 5 \text{ kg}$ ,  $J_1 = 5 \text{ kgm}^2$ ,  $m_2 = 1 \text{ kg}$ , and  $J_2 = 1 \text{ kgm}^2$ . Initial conditions with  $d = 1.5 \text{ m}$  are  $\mathbf{q}_1 = [0 \ 0 \ 0]^T$ ,  $\mathbf{q}_2 = [2.5 \ 0 \ 0]^T$ ,  $\dot{\mathbf{q}}_1 = [0 \ 0 \ 100]^T$ , and  $\dot{\mathbf{q}}_2 = [0 \ 0 \ 0]^T$ . The data set for simulation of this mechanism using Code 5.7 is presented in the *AppData Function* of Fig. 5.8.19.

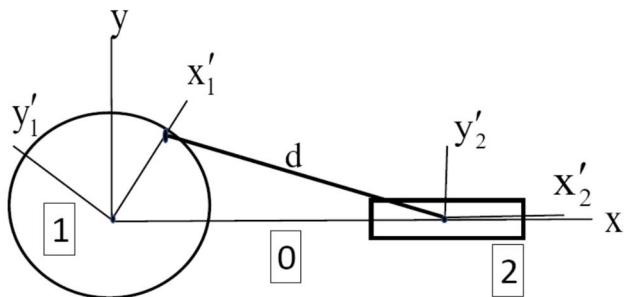


Figure 5.8.18 Two-Body Model of Slider-Crank Mechanism

```

290 if app==6 %Slider-Crank-centroidal model
292 nb=2; %Number of bodies
293 ngc=3*nb; %number of generalized coordinates
294 nh=3; %Number of holonomic constraints
295 nhc=5; %Number of holonomic constraint equations
296 nd=0; %Number of drivers
297 nc=nhc+nd; %Number of constraint equations, including drivers
298 nv=ngc-nc;
299 nu=nc;
300 NTSDA=0; %Number of TSDA force elements
301 NRSDA=0; %Number of RSDA force elements
303 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
304 %PJDT(:,k)=[T;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
305 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD),
306 %i&j=bodies connected, sipr&sjpr=vectors to Pi&Pj,
307 %d=dist., vipr&vjpr=vectors along translational axis
308 PJDT(:,1)=[1;1;0;zer;zer;0;zer;zer]; %Revolute-crank to ground
309 PJDT(:,2)=[2;2;0;zer;zer;0;ux;ux]; %Trans.-slider2 to ground
310 PJDT(:,3)=[3;1;2;ux;zer;1.01;zer;zer]; %Dist.-crank to slider2
312 %PMDT(4,nb): Mass Data Table
313 %PMDT=[[m1;J1;s1c],[m2;J2;s2c],...,[mnb;Jnb;snbc]] (mi is mass of body i,
314 %Ji is polar moment of inertia relative to the origin of the body fixed frame,
315 %and sic is the body fixed vector from the origin of the body fixed frame to
316 %the centroid)
317 PMDT=[[5;5;zer],[1;1;zer]];
319 %PTSDAT(11,NTSDA) TSDA Data Table
320 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F;Uni]; T=TSDA No.,
321 %i&j=bodies conn.,si&jpr=vectors to Pi&Pj, K=spring constant,
322 %C=damping coefficient,el0=spring free length,F=const. force
323 %Unilateral spring is defined by setting Uni=1
324 PTSDAT=zeros(11,1);
327 %PRSDAT(6,NRSDA): RSDA Data Table
328 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=RSDA No.,
329 %i&j=bodies connected, K=spring constant,
330 %C=damping coefficient,phi0=spring free angle,T=constant torque
331 PRSDAT=zeros(6,1);
333 %Initial generalized coordinates
335 q0=[0;0;0;2.01;0;0];
336 qd0=[0;0;100;0;0;0];
338 end

```

Figure 5.8.19 AppData Function, Two-Body Slider-Crank

### 5.8.5.2 Simulation Results and Analysis

Plots of angular velocity ( $\omega$ ), angular acceleration ( $\omega_{ad}$ ), and revolute joint bearing load ( $F_{Rev}$ ), computed from *Lagrange multipliers* and divided by 10 are shown in Fig. 5.8.20, for connecting rod lengths varying from 1.5 m down to 1.01 m. Since connecting rod length of 1m is a *singular configuration*, large accelerations and bearing loads are expected as connecting rod length approaches 1 m. All implicit integration formulations performed well, even in the near singular configuration, where explicit integration formulations failed. Even in the near singular configuration, implicit integrators led to displacement, velocity, and acceleration constraint error norms less than 2.5e-14, 4.5e-12, and e-9, respectively.

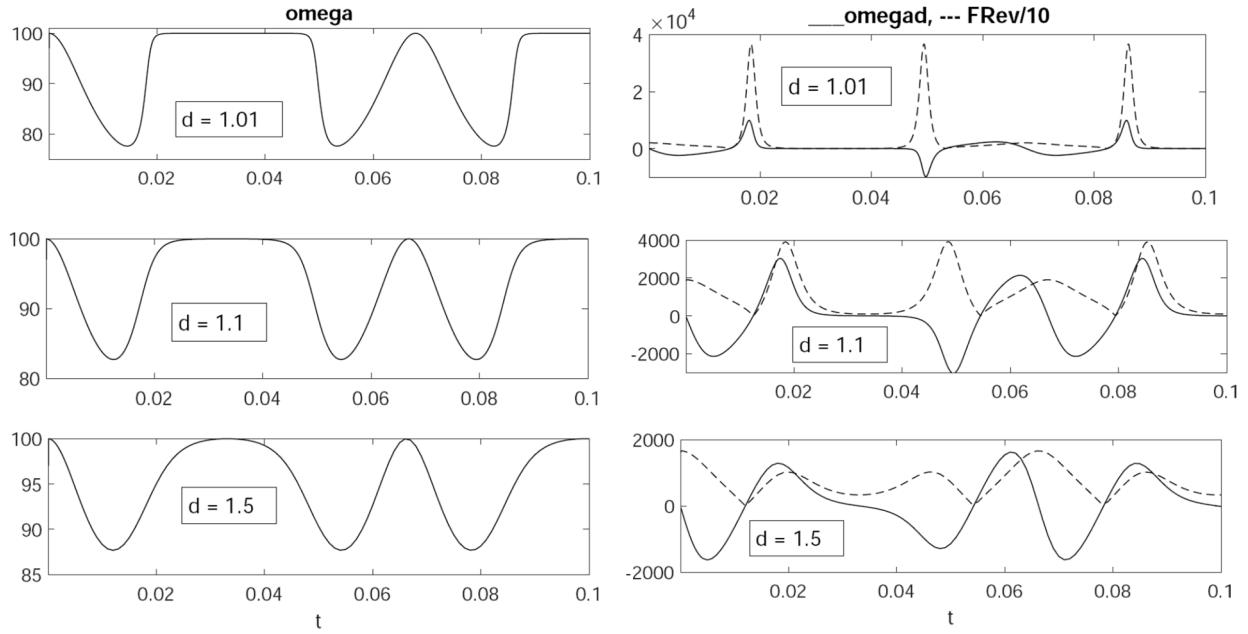


Figure 5.8.20 Angular Velocity and Acceleration and Revolute Joint Bearing Load

### 5.8.6 Rotating Disk with Translating Body

#### 5.8.6.1 Model and Data Set

The disk (body 1) in Fig. 5.8.21 rotates in ground (body 0) about the origin of the x-y inertial frame. Body 2 translates in body 1 along a guide in the disk that is parallel to the body fixed  $y_1$  axis, one unit to the right of the  $y_1$  axis. A TSDA with spring constant  $K = 10 \text{ N/m}$ , damping constant  $C = 0$ , and free length of  $10.1 \text{ m}$  acts between the origin of the  $x_2$ - $y_2$  reference frame and a point  $\mathbf{u}_x^{1'} + 10\mathbf{u}_y^{1'}$  in body 1. Initial conditions for the system are  $\mathbf{q}_1^0 = [0 \ 0 \ 0]^T$ ,  $\dot{\mathbf{q}}_1^0 = [0 \ 0 \ 0]^T$ ,  $\mathbf{q}_2^0 = [1 \ 0 \ 0]^T$ , and  $\dot{\mathbf{q}}_2^0 = [0 \ 0 \ 0]^T$ . The masses and moments of inertia of the bodies are  $m_1 = 10 \text{ kg}$ ,  $J_1 = 10 \text{ kg} \cdot \text{m}^2$ ,  $m_2 = 5 \text{ kg}$ , and  $J_2 = 5 \text{ kg} \cdot \text{m}^2$ . To simulate motion in a horizontal plane the acceleration due to gravity is set to zero; i.e.,  $g = 0$ . The *AppData Function* for simulation of this system using Code 5.7 is shown in Fig. 5.8.22.

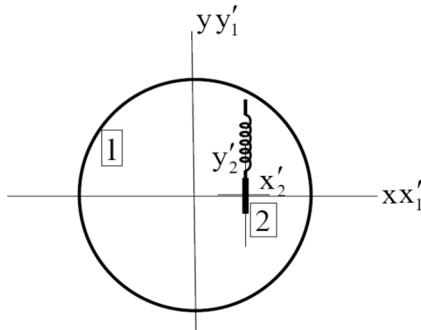


Figure 5.8.21 Rotating Disk with Translating Body

```

341 if app==7 %Rotating Disk with Translating Body-centroidal model
343 nb=2; %Number of bodies
344 ngc=3*nb; %number of generalized coordinates
345 nh=2; %Number of holonomic constraints
346 nhc=4; %Number of holonomic constraint equations
347 nd=0; %Number of drivers
348 nc=nhc+nd; %Number of constraint equations, including drivers
349 nv=ngc-nc;
350 nu=nc;
351 NTSDA=1; %Number of TSDA force elements
352 NRSDA=0; %Number of RSDA force elements
354 %PJDT(12,nh); Planar Joint Data Table (First nh joints not time dependent)
355 %PJDT(:,k)=[T;i;j;si(pr);sj(pr);d;v(pr);v(j(pr)]; k=joint No.,
356 %T=joint type(1=Rev,2=Tran,3=Dist, 4=RotD, 5=DistD),
357 %i&j=bodies connected, si(pr)&sj(pr)=vectors to Pi&Pj,
358 %d=dist., v(pr)&v(j(pr)=vectors along translational axis
359 PJDT(:,1)=[1;1;0;zer;zer;0;zer;zer]; %Rev.-Body to ground
360 PJDT(:,2)=[2;1;2;ux;zer;0;uy;uy]; %Tran.-Body 1 to 2
362 %PMDT(4,nb): Mass Data Table
363 %PMDT=[[m1;J1;s1c],[m2;J2;s2c],...,[mnb;Jnb;snbc]] (mi is mass of body i,
364 %Ji is polar moment of inertia relative to the origin of the body fixed frame,
365 %and sic is the body fixed vector from the origin of the body fixed frame to
366 %the centroid)
367 PMDT(:,1)=[10;10;zer];
368 PMDT(:,2)=[5;5;zer];
370 %PTSDAT(11,NTSDA) TSDA Data Table
371 %PTSDAT(:,T)=[i;j;si(pr);sj(pr);K;C;el0;F;Uni]; T=TSDA No.,
372 %i&j=bodies conn., si(pr)&sj(pr)=vectors to Pi&Pj, K=spring constant,
373 %C=damping coefficient, el0=spring free length, F=const. force
374 %Unilateral spring is defined by setting Uni=1
375 %PTSDAT=zeros(11,1);
376 PTSDAT(:,1)=[1;2;ux+10*uy;zer;10;1;10.1;0;0];
378 %PRSDAT(6,NRSDA): RSDA Data Table
379 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
380 %i&j=bodies connected, K=spring constant,
381 %C=damping coefficient, phi0=spring free angle, T=constant torque
382 PRSDAT=zeros(6,1);
384 %Initial generalized coordinates
386 q0=[0;0;0;ux;0];
387 qd0=[0;0;0;0;0];
389 end

```

Figure 5.8.22 AppData Function for Rotating Disk with Translating Body

### 5.8.6.2 Simulation Results and Analysis

Simulation is carried out over a 10 sec interval, with plots of rotation of body 1,  $\phi_1 = \text{phi1}$ . and translation of body 2 relative to body 1,  $y'_2 = \text{dely2pr}$ , shown in Fig. 5.8.24. As shown, since the spring is precompressed, the disk initially rotates counterclockwise and body 2 moves in the negative  $y'_2$  direction. With the same data, but damping constant  $C = 1 \text{ Nsec/m}$ , the dynamic response is shown in Fig. 5.8.24. Identical results were obtained with all eight integration formulations, for both data sets.

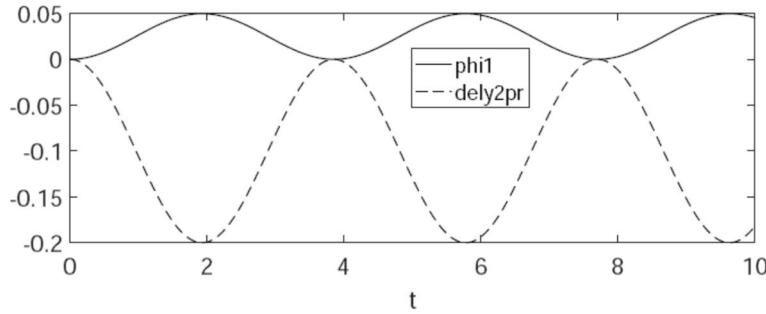


Figure 5.8.23  $\phi_1 = \text{phi1}$  and  $\delta y'_2 = \text{dely2pr}$ , No Damping

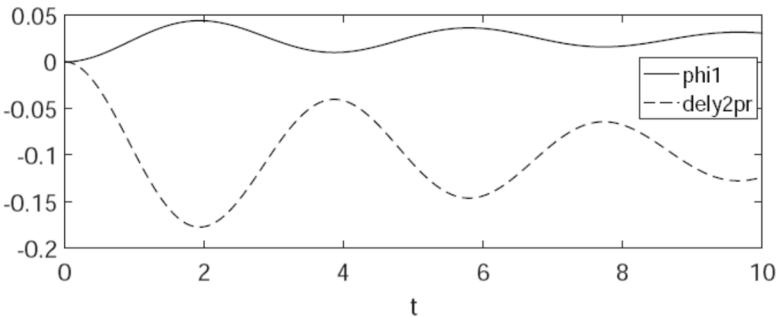


Figure 5.8.24  $\phi_1 = \text{phi1}$  and  $\delta y'_2 = \text{dely2pr}$ , Damping C = 1 Nsec/m

Modeling and simulation of planar multibody systems using Code 5.7 of Appendix 5.A illustrates properties of the tangent space formulation and numerical methods in accurately forming and solving the ODE and Index 0 DAE of planar system dynamics. Through study of the behavior of concrete examples, the engineer can gain a practical appreciation of the theoretical underpinnings of the methods. This is especially important for systems that exhibit stiff behavior.

Examples presented make clear the disparity in engineering time and effort between ad-hoc model definition, equation derivation, and coding and debugging versus use of general-purpose software as a means to achieve effective simulation and dynamic system analysis.

## 5.9 Code 5.9 for Tangent Space Simulation of Spatial Systems

The general-purpose Code 5.9 of Appendix 5.A implements the *tangent space ODE* and *Index 0 DAE* formulations of Sections 5.3 and 5.5 for *spatial multibody systems*. Building block distance, spherical, dot1, and dot2 constraints are implemented and serve as the foundation for cylindrical, revolute, translational, distance, universal, strut, and revolute-spherical constraints of Section 3.3. *Noncentroidal body reference frames* are used to make the kinematic modeling task manageable, with minimal additional computational cost associated with additional inertial terms that arise in the equations of motion. Kinematic and kinetic derivatives presented in Section 3.3 and Appendix 5.B are used to implement explicit and implicit numerical integration methods of Section 4.8 for solution of tangent space ODE and Index 0 DAE. Applied, gravitational, and internal forces defined by translational- and rotational- spring-damper-actuators (*TSDA* and *RSDA*, respectively) presented in Section 4.5 are implemented. The same fixed and variable time step explicit and implicit integration algorithms used in planar system Code 5.7 are implemented for numerical integration of tangent space ODE and Index 0 DAE of *spatial system dynamics*.

Following an explanation of Code 5.9 in this section, numerical examples are presented in Section 5.10, including those treated with ad-hoc derivations and computer implementations in Sections 5.4 and 5.6. The essence of computational methods in mechanical system dynamics is enabling computer formulation and solution of the equations of motion, without the painful detail of ad-hoc derivation of equations of motion and ad-hoc coding of numerical solution algorithms experienced in Sections 5.4 and 5.6. The computer implementation presented in this section is intended to introduce the reader to methods that are now available in commercial dynamic simulation software and advanced software that is likely to appear in the foreseeable future.

Components of Code 5.9 that interface with the user are presented in Section 5.9.1, followed by an outline of the body of the code, with which the user need not interact, in Section 5.9.2. The structure of Code 5.9 is identical to that of Code 5.7 presented in Section 5.7. References are made to that code, where appropriate, to avoid unnecessary repetition.

### 5.9.1 User Components of Code

The initial segment of Code 5.9 defines integration and *error control parameters* that underlie the tangent space formulation and associated numerical integration methods. Since it is identical to the code of Fig 5.7.1, the reader is referred to Section 5.7.1 for discussion of its use.

Application data are indexed in lines 33 to 44 of Fig. 5.9.1 to 12 applications that are defined in the AppData function presented in Fig. 5.9.2. The declaration in line 45 defines which application is implemented in the simulation. The AppData function parameter definitions of lines 46 through 49 are used throughout the code to pass data for each simulation. If required for definition of initial conditions that are consistent with constraint, the user may enter code following line 50. Finally, desired output data are defined for each application; e.g., lines 403 to 416 for the spin stabilized top of app = 1.

```
32 %Application Data
33 %app=1, Spin Stabilized Top
34 %app=2, Spatial Double Pendulum-variable length dist constr
35 %app=3, One Body in Cylindrical Joint, with Spring
36 %app=4, Spatial Slider-Crank
37 %app=5, Spin Stabilized Top on unit length bar
```

```

38 %app=6, Transient Top
39 %app=7 4-Translating Mass Model
40 %app=8 Fly-Ball Governor
41 %app=9 2 Bar Spatial Mechanism
42 %app=10 2 Bar Spatial Mechanism, cylindrical
43 %app=11 Bar on rotating shaft
44 %app=12, Spin Stabilized Top on unit length bar, centroidal formulation
45 app=5;
46 [nb,ngc,nh,nhc,nd,nc,nv,nu,SJDT,SMDT,STSDAT,SRSDAT,...]
47 NTSDA,NRSDA,q0,qd0,]=AppData(app);
48 par=[nb;ngc;nh;nhc;nd;nc;nv;nu;g;utol;Btol;intol;Atol;h0;hvar;...
49 NTSDA;NRSDA;app];
50 %Initial condition calculation, if required

403 %Data of Interest (Enter for each application)
405 if app==1 %Spin Stabilized Top
406 [r1,p1]=qPart(q,1);
407 rc=ATran(p1)*uz;
408 xc(n)=rc(1);
409 yc(n)=rc(2);
410 zc(n)=rc(3);
411 [r1d,p1d]=qPart(qd,1);
412 rcd=BTran(p1,uz)*p1d;
413 xcd(n)=rcd(1);
414 ycd(n)=rcd(2);
415 zcd(n)=rcd(3);
416 end

```

Figure 5.9.1 Application Data in Main Code

The third component of user entered code is the *AppData Function* shown for the spin stabilized top in Fig. 5.9.2. The AppData function definition, based on data entered, is passed to the main code for functions that are executed in calls throughout the code. Lines 9 through 18 define the dimension of variables in the model. If entered incorrectly, the code will fail in ways that are difficult to understand from MATLAB error messages. The *Spatial Joint Data Table* (SJDT) is defined in lines 19 through 24. It contains data defined in Line 24, as entered for the spin stabilized top. A detailed guide to entering data into the SJDT is provided in Section 3.11.1 and is not repeated here.

The *Spatial Mass Data Table* (SMDT) is defined in lines 25 through 32. The SMDT contains mass and moment of inertia data for each body in the system, as follows:

```
SMDT=[[m1;J1x;J1y;J1z;J1zy;J1xz;J1yz;s1c],...,[mn1;Jnbx;Jnby;Jnbz;Jnbzy;Jnbxz;Jnbzy;snb]]
```

where for body i, mi is its mass; Jix, Jiy, and Jiz are diagonal elements of the inertia matrix, Jixy, Jixz, and Jiyy are off-diagonal elements of the symmetric inertia matrix, and sic is the body fixed vector from the origin of the x'-y'-z' body reference frame to the centroid. In this model of the spinning top , a noncentroidal reference frame with origin at the tip is employed. A centroidal body reference frame is employed in app = 12.

The *Spatial TSDA Data Table* (STSDAT) is defined in lines 33 through 39. More specifically, a template for the STSDAT defined in Fig. 5.9.2, repeated here for clarity, is

```
STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
```

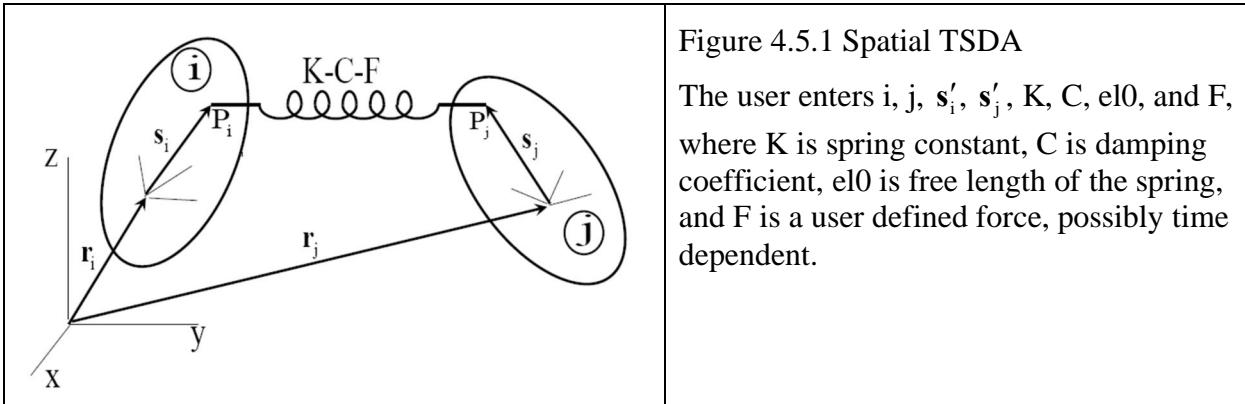


Figure 4.5.1 Spatial TSDA

The user enters  $i$ ,  $j$ ,  $s'_i$ ,  $s'_j$ ,  $K$ ,  $C$ ,  $el0$ , and  $F$ , where  $K$  is spring constant,  $C$  is damping coefficient,  $el0$  is free length of the spring, and  $F$  is a user defined force, possibly time dependent.

While there are no TSDA elements in the spinning top, a zeros table of the appropriate dimension is specified for consistent bookkeeping.

The *Spatial RSDA Data Table* (SRSDAT) is defined in lines 40 through 46. More specifically, a template for the SRSDAT defined in Fig. 5.9.2, repeated here for clarity, is

`SRSDAT(:,R)=[i;j;uipr;wipr;ujpr;wipr;K;C;thet0;T];`

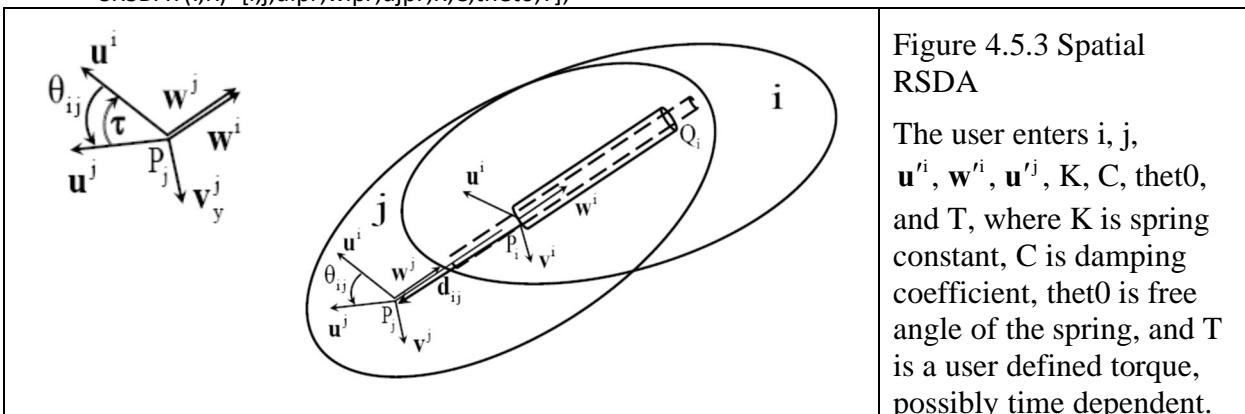


Figure 4.5.3 Spatial RSDA

The user enters  $i$ ,  $j$ ,  $u'^i$ ,  $w'^i$ ,  $u'^j$ ,  $w'^j$ ,  $K$ ,  $C$ ,  $thet0$ , and  $T$ , where  $K$  is spring constant,  $C$  is damping coefficient,  $thet0$  is free angle of the spring, and  $T$  is a user defined torque, possibly time dependent.

While there are no RSDA elements in the spinning top, a zeros table of the appropriate dimension is specified for consistent bookkeeping.

Finally, *initial generalized coordinates* and *initial velocities* are defined in lines 55 through 69, using the three-point method of Section 2.5.5 to define each body reference frame and specifying the velocity  $rOd$  of its origin and its angular velocity  $\Omega_{meg}$ , specified in the global reference frame. If computation is required to define initial conditions that are consistent with constraints, code doing so is entered in the main code, following line 50 in Fig. 5.9.1.

```

8 if app==1 %Spin Stabilized Top
9 nb=1; %Number of bodies
10 ngc=7*nb; %number of generalized coordinates
11 nh=1; %Number of holonomic constraints
12 nhc=3; %Number of holonomic constraint equations, including drivers
13 nd=0; %number of driving constraints
14 nc=nhc+nb; %Number of constraint equations
15 nv=ngc-nc;
16 nu=nc;
```

```

17 NTSDA=0; %Number of TSDA force elements
18 NRSDA=0; %Number of RSDA force elements
19 %SJDT(22,nh): Spatial Joint Data Table
20 %SJDT(:,k)=[T;i;j;si(pr);sj(pr);d;ui(pr);wi(pr);uj(pr);wj(pr)];
21 %k=joint No., T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
22 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,i>0;
23 %si&jpr=vectors to Pi&j; d=dist.; ui(pr), wi(pr), uj(pr), wj(pr)=joint vectors
24 SJDT(:,1)=[2;1;0;zer;zer;0;zer;zer;zer]; %Sph Jt - Body1 and ground
25 %SMDT(4,nb): Mass Data Table (With full inertia matrix in noncentroidal frame
26 %parallel to body reference frame)
27 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...,
28 [%[mn(b);Jnbx;Jnby;Jnbz;Jnbxy;Jnbxz;Jnbyz;snb]];
29 %mi=mass of body i,Jixy,Jixz,Jixx are diagonal of inertia matrix,
30 %Jixy, Jixz, and Jixx are products of inertia, sibc is body fixed
31 %vector from origin of body fixed frame to centroid
32 SMDT=[30;120;120;30;0;0;0;uz];
33 %STSDAT(12,1): TSDA Data Table
34 %STSDAT(:,T)=[i;j;si(pr);sj(pr);K;C;el0;F];
35 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
36 %C=damping coefficient; el0=spring free length; F=const. force
37 if NTSDA==0
38 STSDAT=zeros(12,NTSDA);
39 end
40 %SRSDAT(15,1): RSDA Data Table
41 %SRSDAT(:,R)=[i;j;ui(pr);wi(pr);uj(pr);K;C;thet0;T];
42 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
43 %C=damping coefficient; thet0=spring free angle; T=const. torque
44 if NRSDA==0
45 SRSDAT=zeros(15,NRSDA);
46 end
47 %Body Initial Condition Data Table
48 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
49 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
50 %rQ=vector in x-y plane (as close to y axis as possible),
51 %rOd=velocity of origin, Omeg=angular velocity in global frame
52 Omeg=[10^-12;10^-12;13.5];
53 rOd=atil(Omeg)*uz;
54 BPDDT(:,1)=[zer;ux;uy;rOd;Omeg];
55 %Initial generalized coordinates
56 q0=zeros(7*nb,1);
57 qd0=zeros(7*nb,1);
58 j=1;
59 while j<=nb
60 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
61 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
62 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
63 [q,p,A]=InitConfig(rO,rP,rQ);
64 q0=Add(q0,q,7*(j-1),0);
65 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
66 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
67 pd=0.5*EEval(p)'*Omeg;
68 qd=[rOd;pd];
69 qd0=Add(qd0,qd,7*(j-1),0);

```

```

70 j=j+1;
71 end
72 end

```

Figure 5.9.2 AppData Function

### 5.9.2 Computational Components of Code

Computational flow in the main program, which requires no input from the user beyond line 100, is outlined in Fig. 5.9.3. Data required for integration that are defined in Section 5.9.1 are initialized following line 100. *Criteria for reparameterization*, for both implicit and explicit integrators, are defined following lines 140 and 164. Code to implement parameterization is called following line 186, if *error control tolerances* are exceeded. Code that implements the eight supported numerical integration methods is outlined following line 206. Integration results for tangent space coordinates ( $v$ ,  $vd$ , and  $vdd$ ) that are computed using the selected integration method are processed to obtain physical generalized coordinates ( $q$ ,  $qd$ ,  $qdd$ , and  $Lam$ ) following line 287. Total energy is computed following line 340. Norms of position, velocity, and acceleration constraint error are computed following line 392. Output data of interest for each of the applications discussed in Section 5.9.1 are defined following line 403.

```

100 %Initialize Data For Integration

140 if integ<5 %Start implicit reparameterization criteria
164 if integ>4 %Start explicit reparameterization criteria

186 % Parameterization

206 % Integration
208 if integ==1 %Implicit ODE Trap
210 [v, vd, vdd, R1n, Jiter, JCond, Jinv, Jinviter, h, nch]=...
211 ImplicitODETrap(n, tn, npar, ...
212 Vv, Vvd, Vvdd, Uu, q0, V, U, B, h, hmax, nch, SMDT, STSDAT, SRSDAT, SJDT, ...
213 par, InvJ, Jinv);

256 if integ==5 %Explicit ODE Nystrom4
258 [v, vd, vdd, ECond]=ExplicitODENystrom4(n, tn, Vv, Vvd, Uu, V, U, B, q0, ...
259 h, SMDT, STSDAT, SRSDAT, SJDT, par);

287 %Process Results
297 %Evaluate q
312 %Update B and Evaluate qd
328 % Evaluate qdd and Lam (Lam evaluation a postprocessing step)
340 %Calculate Total Energy
392 %Calculate constraint error
403 %Data of Interest (Enter for each application)

```

Figure 5.9.3 Main Code Computational Flow

Computing functions that underlie the main code outlined in Fig. 5.9.3 are identified in Fig. 5.9.4. Computing subroutines include the *Add function* that enables adding nonzero submatrices to sparse matrices, below and to the right of the address of the 1-1 term in the submatrix added to the underlying matrix that was initialized to zero. *ATran* evaluates the orientation transformation matrix. Functions *CorrectB* and *usolve* evaluate the matrix B in Eq.

(5.2.22) and dependent generalized coordinates in Eq. (5.2.19). *GamEval* and *GamsqqdEval* evaluate terms in Eq. (5.3.18) and *Param* evaluates terms required for the tangent space parameterization of Section 5.2.

Vector partition functions *parPart*, *qPart*, and *xPart* support partitioning of vectors involved into components that are used in kinematic and kinetic computations. Similarly, the constraint and data table functions listed provide access to elements of the data tables that are required to implement computation.

Constraint and derivative evaluation functions listed evaluate kinematic constraint expressions and the numerous derivatives defined in Sections 3.2, 5.3, and 5.5 and Appendix 5.B that are needed in implementing the tangent space ODE and Index 0 DAE formulations and numerical integration of the associated equations of motion. The numerous *bbxxx* functions implement building block constraint derivatives that underpin the physical joint constraint equations. Care is taken to account for the fact that ground is designated by  $j = 0$  and its generalized coordinates are constant, yielding no derivative contribution, but including geometric quantities that define a constraint of body  $i$  with ground, if one is specified. Similarly, the kinetic and derivative evaluation functions listed evaluate inertial and generalized force terms and derivatives defined in Sections 4.2, 5.3, and 5.5 and in Appendix 5.B that are required in formulation and solution of the equations of motion.

Finally, numerical integration functions listed carry out the numerical integration process, where *ODEfunct* evaluates tangent space ODE and *Ind0ODEfunct* evaluates tangent space Index 0 DAE of Sections 5.3 and 5.5, for use in the two explicit integrators. Similarly, *JacobODE* and *JacobInd0* evaluate terms in implicit numerical integration Jacobians that are defined in Sections 5.3 and 5.5, and *ResidODE* and *ResidInd0* evaluate the residual of the equations of motion for use in implicit integrators.

Internal details of the MATLAB Functions listed in Fig. 5.9.4 are not presented, since each of the functions is documented internally and the user need not modify these functions in applications.

#### Computing Functions

- Add
- atil
- EEval
- GEval
- KEval
- BTran
- ATran
- CorrectB
- usolv
- GamEval
- GamsqqdEval
- Param

#### User Input Function

- AppData

```

Vector Partition Functions
  parPart
  qPart
  xPart
  pNormPart
Constraint and Data Table Partition Functions
  CylPart
  DistPart
  RevPart
  TranPart
  StrutPart
  RevSphPart
  UnivPart
  STSDATPart
Constraint and Derivative Evaluation Functions
  bbxxx
  PhiEval
  PhiqEval
  P2Eval
  P3Eval
  P4Eval
  P5Eval
Kinetic and Derivative Evaluation Functions
  MEval
  M2Eval
  QAEval
  QAsqqd
Numerical Integration Functions
  ExplicitNystrom4
  ExplicitRKF45
  ODEfunct
  Ind0ODEfunct
  ImplicitTrap
  ImplicitSDIRK54
  JacobODE
  JacobInd0
  ResidODE
  ResidInd0

```

Figure 5.9.4 Computing Functions

### 5.9.3 Code Output

In addition to output defined for each application in the AppData function, the code reports the following arrays of values at each time step that shed light on dynamics of the application and performance of the code:

- PosConstrNorm; norm of position constraint error
- VelConstrNorm; norm of position constraint error
- AccConstrNorm; norm of acceleration constraint error
- Biterrpt; number of iterations required to update **B**
- Iterurpt; number of iterations required to update **u**
- hrpt; stepsize h for variable step integrators

`ImpSoliterrpt`; number of iterations required in implicit integration  
`JCondrpt`; Condition number of matrix in implicit equation solution  
`jRepar`; total number of reparameterizations in simulation  
`Q`; array of values of  $\mathbf{q}$   
`Qd`; array of values of  $\dot{\mathbf{q}}$   
`Qdd`; array of values of  $\ddot{\mathbf{q}}$   
`Vv`; array of values of  $\mathbf{v}$   
`Vvd`; array of values of  $\dot{\mathbf{v}}$   
`Vvdd`; array of values of  $\ddot{\mathbf{v}}$   
`LLam`; array of values of Lagrange multiplier vector  $\lambda$   
`TE`; total energy of system

Much as outlined in Section 5.7, Code 5.9 of Appendix 5.A is a tool for numerical simulation and experimentation with the tangent space ODE and Index 0 DAE formulations of spatial multibody dynamics. A general-purpose formulation and computer implementation is especially important for spatial systems, due to the analytical complexity of orientation of bodies and the resulting kinematic relations and equations of motion. Development and use of such codes is the essence of the field of computational mechanical system dynamics.

## 5.10 Spatial System Simulation Using Code 5.9

Seven spatial examples are simulated using Code 5.9 of Appendix 5.A. Kinematic and kinetic characteristics of the models are defined and entered in AppData functions defined in Section 5.9. Code 5.9 is then run and results analyzed. The technical complexity of ad-hoc derivation of equations and associated coding, characteristic of examples presented earlier in the chapter, is contrasted with the ease of use of a general-purpose computer code.

### 5.10.1 Spinning Top with Tip Fixed

The *spinning Top with tip fixed*, shown in Fig. 5.10.1, was studied in Sections 5.1 and 5.4.2.1. It is simulated here as a body moving in space with a spherical joint between its tip and the origin of the x-y-z frame, using Code 5.9. The full data set is provided by the *AppData Function* of Fig. 5.10.2. This function is similar to that for spatial kinematic analysis in Code 3.11, with notable exceptions. First, it defines inertia and force elements that influence dynamics of the system and inertia data in the SMDT Mass Data Table of lines 27 through 29. Second, initial conditions for the ODE of dynamics are entered through three point orientation data and velocities and angular velocities in lines 46 through 70, or by entering explicit initial conditions from prior kinematic analysis in lines 71 through 75, under the control of the initcond index. For the single body spinning top, definition of kinematically admissible initial conditions is straight forward.

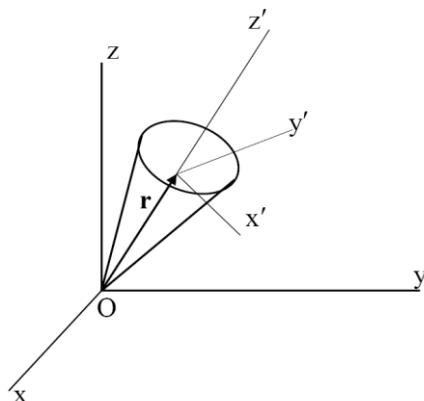


Figure 5.10.1 Symmetric Top with Tip Fixed

```

8 if app==1 %Spin Stabilized Top
9 nb=1; %Number of bodies
10 ngc=7*nb; %number of generalized coordinates
11 nh=1; %Number of holonomic constraints
12 nhc=3; %Number of holonomic constraint equations, including drivers
13 nd=0; %number of driving constraints
14 nc=nhc+nb; %Number of constraint equations
15 nv=ngc-nc;
16 nu=nc;
17 NTSDA=0; %Number of TSDA force elements
18 NRSDA=0; %Number of RSDA force elements
19 %SJDT(22,nh): Spatial Joint Data Table
20 %SJDT(:,k)=[T;i;j;si(pr;sj(pr;d;ui(pr;wi(pr;uj(pr;wj(pr];
21 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
22 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,>0;
23 %si&jpr=vectors to Pi&j; d=dist.; ui(pr, wi(pr, ui(pr, wi(pr)=joint vectors

```

```

24 SJDT(:,1)=[2;1;0;zer;zer;0;zer;zer;zer;zer]; %Sph Jt - Body1 and ground
25 %SMDT(4,nb): Mass Data Table (With full inertia matrix in noncentroidal frame
26 %parallel to body reference frame)
27 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...,
28 [%mnbs;Jnbx;Jnby;Jnbz;Jnbxy;Jnbxz;Jnbyz;snbc]];
29 %mi=mass of body i,Jix,Jiy,Jiz are diagonal of inertia matrix,
30 %Jixy, Jixz, and Jixz are products of inertia, sibc is body fixed
31 %vector from origin of body fixed frame to centroid
32 SMDT=[30;120;120;30;0;0;uz];
33 %STSDAT(12,1): TSDA Data Table
34 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
35 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
36 %C=damping coefficient; el0=spring free length; F=const. force
37 if NTSDA==0
38 STSDAT=zeros(12,NTSDA);
39 end
40 %SRSDAT(15,1): RSDA Data Table
41 %SRSDAT(:,R)=[i;j;uipr;wipr;ujpr;K;C;thet0;T];
42 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
43 %C=damping coefficient; thet0=spring free angle; T=const. torque
44 if NRSDA==0
45 SRSDAT=zeros(15,NRSDA);
46 end
47 %Body Initial Condition Data Table
48 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
49 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
50 %rQ=vector in x-y plane (as close to y axis as possible),
51 %rOd=velocity of origin, Omeg=angular velocity in global frame
52 Omeg=[10^-12;10^-12;13.5];
53 rOd=atil(Omeg)*uz;
54 BPDDT(:,1)=[zer;ux;uy;rOd;Omeg];
55 %Initial generalized coordinates
56 q0=zeros(7*nb,1);
57 qd0=zeros(7*nb,1);
58 j=1;
59 while j<=nb
60 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
61 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
62 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
63 [q,p,A]=InitConfig(rO,rP,rQ);
64 q0=Add(q0,q,7*(j-1),0);
65 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
66 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
67 pd=0.5*EEval(p)'*Omeg;
68 qd=[rOd;pd];
69 qd0=Add(qd0,qd,7*(j-1),0);
70 j=j+1;
71 end
72 end

```

Figure 5.10.2 AppData Function, Spin Stabilized Top

Plots of x vs y coordinates of the centroid of the top presented in Fig. 5.10.3 were obtained using Code 5.9 with the trapezoidal integrator and ODE formulation, with utol = Btol = e-12, intol = e-8, and Atol = e-5. These results are essentially identical to those presented in

Sections 5.1 and 5.4.2.1. Maximum position, velocity, and acceleration constraint error norms over the 100 sec simulation for  $\omega_{z0} = 13.5$  rad/sec were  $2.5e-14$ ,  $5e-15$ , and  $e-13$ , respectively. The maximum variation in *total energy* for this *conservative system* was  $5e-3\%$ . Similar results were obtained with the other 7 integrator-formulation combinations supported in Code 5.9. On average, charts were algorithmically reparametrized 625 times during e5 time steps in each simulation, or once per 160 time steps. Only 0.02% of CPU time and no person effort was devoted to *reparameterization*.

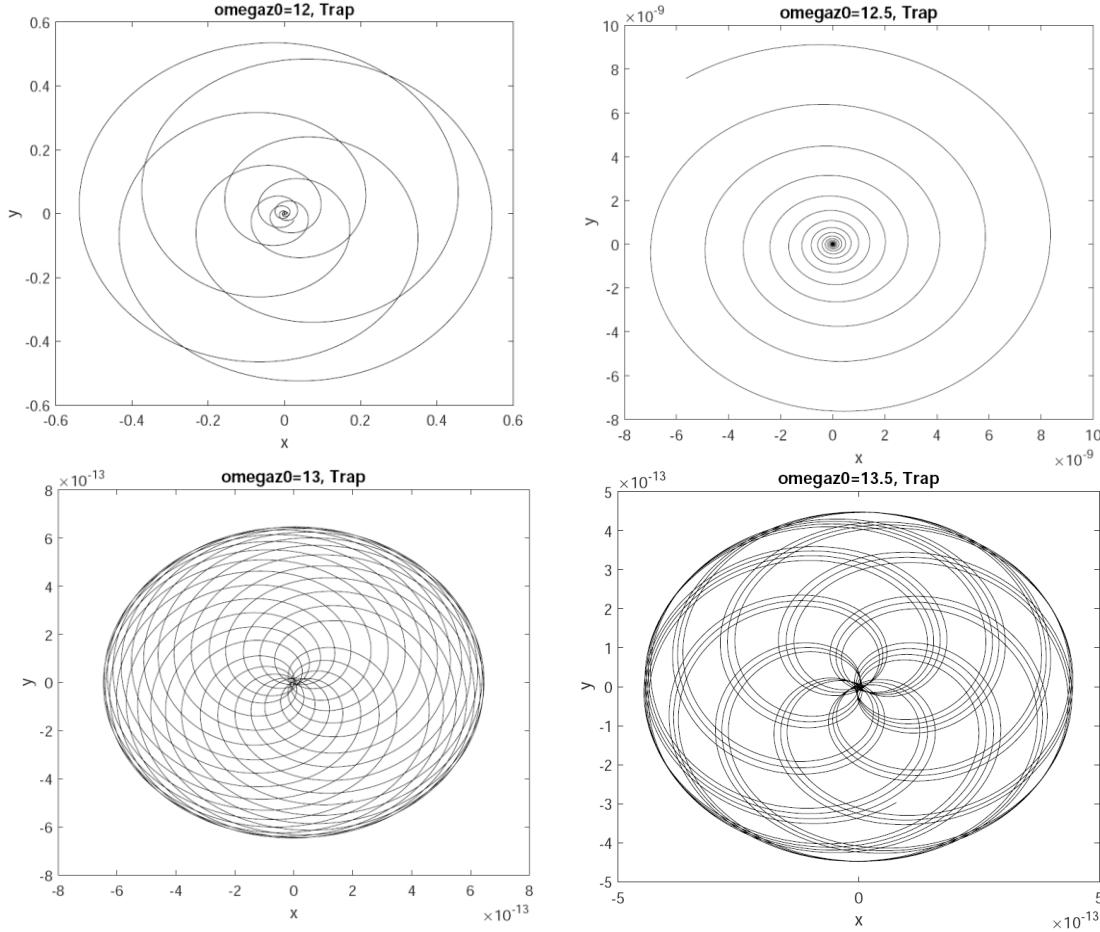


Figure 5.10.3 Centroidal x vs y for 100 sec Simulation of Spin Stabilized Top

### 5.10.2 Spinning Top with Tip Constrained to be a Unit Distance from a Fixed Point

The same symmetric Top with tip constrained by a massless bar of unit length to be one meter from the origin of the x-y-z frame is shown in Fig. 5.10.4. In this model, a unit distance constraint is imposed between the tip of the Top and the origin of the x-y-z inertial frame; i.e., vector  $\mathbf{s}$  in Fig. 5.10.4 has unit length. Whereas the Top of Section 5.10.1 has three degrees of freedom, this Top has five *degrees of freedom*. The *AppData Function* for this model, including definition of kinematically admissible initial conditions is presented in Fig. 5.10.5.

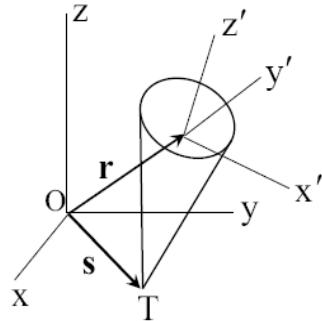


Figure 5.10.4 Symmetric Top with Tip Constrained to be Unit Distance from Origin O

```

277 if app==5 %Spin Stabilized Top With Dist Constraint from Tip to Origin
278 nb=1; %Number of bodies
279 ngc=7*nb; %number of generalized coordinates
280 nh=1; %Number of holonomic constraints
281 nhc=1; %Number of holonomic constraint equations, including drivers
282 nd=0; %number of driving constraints
283 nc=nhc+nb; %Number of constraint equations
284 nv=ngc-nc;
285 nu=nc;
286 NTSDA=0; %Number of TSDA force elements
287 NRSDA=0; %Number of RSDA force elements
288 %SJDT(22,nh): Spatial Joint Data Table
289 %SJDT(:,k)=[T;ij;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
290 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
291 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,i>0;
292 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=joint vectors
293 SJDT(:,1)=[1;1;0;zer;zer;1;zer;zer;zer]; %Dist Jt - Body1 and ground
294 %SMDT(4,nb): Mass Data Table (With full inertia matrix in centroidal frame
295 %parallel to body reference frame)
296 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...,
297 %[mnb;Jnbx;Jnby;Jnbz;Jnbxy;Jnbxz;Jnbyz;snbc]];
298 %mi=mass of body i,Jix,Jiy,Jiz are diagonal of inertia matrix,
299 %Jixy, Jixz, and Jixz are products of inertia, sibc is body fixed
300 %vector from origin of body fixed frame to centroid
301 SMDT=[30;120;120;30;0;0;0;uz];
302 %STSDAT(12,1): TSDA Data Table
303 if NTSDA==0
304 STSDAT=zeros(12,NTSDA);
305 end
306 %STSDAT(:,T)=[ij;sipr;sjpr;K;C;el0;F];
307 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
308 %C=damping coefficient; el0=spring free length; F=const. force
309 %SRSDAT(15,1): RSDA Data Table
310 %SRSDAT(:,R)=[ij;uipr;wipr;ujpr;K;C;thet0;T];
311 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
312 %C=damping coefficient; thet0=spring free angle; T=const. torque
313 if NRSDA==0
314 SRSDAT=zeros(15,NRSDA);
315 end
316 %Body Initial Condition Data Table
317 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
318 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,

```

```

319 %rQ=vector in x-y plane (as close to y axis as possible),
320 %rOd=velocity of origin, Omeg=angular velocity in global frame
321 omeg10=[10^-12;10^-12;13.5];
322 rOd=-atil(omeg10)*uz;
323 BPDDT(:,1)=[-uz;-uz+ux;-uz+uy;rOd;omeg10];
324 %Initial generalized coordinates
325 q0=zeros(7*nb,1);
326 qd0=zeros(7*nb,1);
327 j=1;
328 while j<=nb
329 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
330 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
331 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
332 [q,p,A]=InitConfig(rO,rP,rQ);
333 q0=Add(q0,q,7*(j-1),0);
334 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
335 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
336 pd=0.5*EEval(p)'*Omeg;
337 qd=[rOd;pd];
338 qd0=Add(qd0,qd,7*(j-1),0);
339 j=j+1;
340 end
341 end

```

Figure 5.10.5 AppData Function, Spining Top with Tip One Meter from O

The *initial configuration* of the system is with the tip on the z-axis one unit below the origin, the centroid at the origin, and the z'-axis coincident with the z-axis. The *initial velocity* of the centroid is  $\dot{\mathbf{r}}^0 = \mathbf{0}$  and the initial angular velocity in the body fixed x'-y'-z' reference frame is  $\boldsymbol{\omega}^0 = [\varepsilon \quad \varepsilon \quad \text{omegaz0}]^T$ , where initial angular velocity components  $\varepsilon = 10^{-12}$  about the x' and y' axes play the role of *perturbations* from the vertical configuration. Code 5.9 yielded results shown in Fig. 5.10.6 for 100 sec simulations, using the trapezoidal integrator with variable step size, showing the Top stabilizing at an initial angular velocity about the z-axis of approximately 13.5 rad/sec.

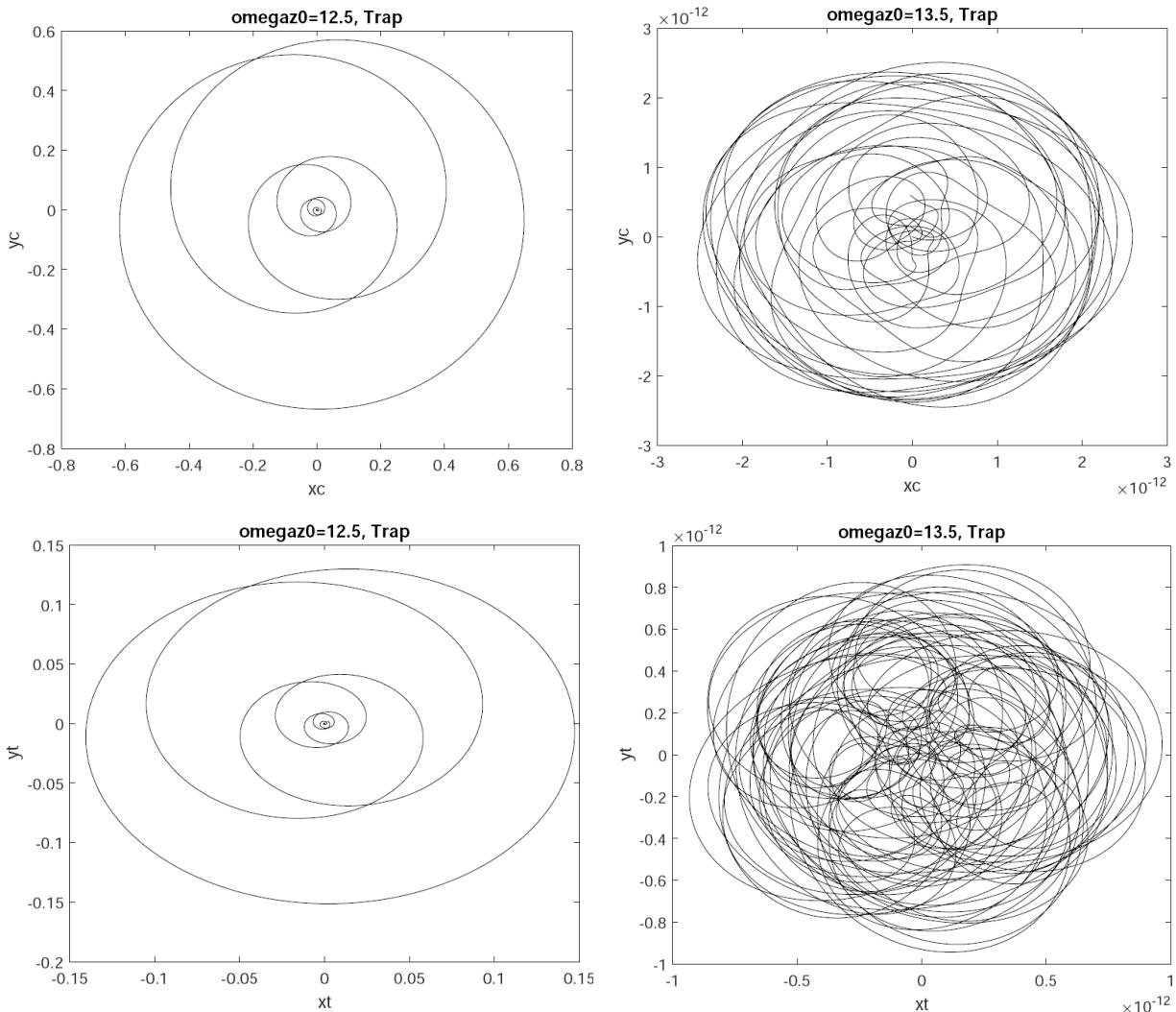


Figure 5.10.6 Plots of Centroid ( $xc$ - $y_C$ ) and Tip ( $xt$ - $yt$ ) Trajectories for Top on Bar

The upper plots for each initial angular velocity are  $xc$ - $y_C$  trajectories of the centroid and the lower plots are  $xt$ - $yt$  trajectories of the tip. *Total energy variation* for this *conservative system* was  $2e-4\%$  over the 100 sec simulation. On average, charts were algorithmically reparametrized 470 times during  $e5$  time steps in each simulation, or once per 213 time steps. Only 0.02% of CPU time and no person effort was devoted to *reparameterization*.

As evidence that the formulation provides accurate results, with varying values of  $Tol = Btol = utol$ , the maximum norm of error in position, velocity, and acceleration constraints over the 100 sec simulations are reported in Table 5.10.1. As predicted by the theory, these errors approach zero to computer precision as  $Tol$  approaches zero.

Table 5.10.1 Maximum Norm of Position, Velocity, and Acceleration Constraint Error

Tol	Pos Constr Err	Vel Constr Err	Acc Constr Err
e-6	2e-8	e-6	5e-5
e-8	2e-10	3e-8	2e-7
e-10	2e-12	2e-10	4e-9

### 5.10.3 Spatial Double Pendulum

The *spatial double pendulum* of Section 5.4.4, shown in Fig. 5.10.7, is simulated using Code 5.9, with data, including kinematically admissible initial conditions, is provided in the *AppData Function* of Fig. 5.10.8.

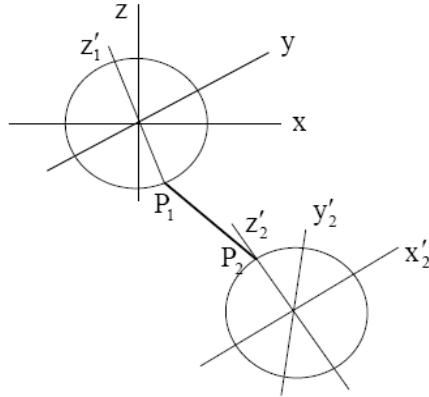


Figure 5.10.7 Spatial Double Pendulum

```

75 if app==2 %Spatial Double Pendulum
76 nb=2; %Number of bodies
77 ngc=7*nb; %number of generalized coordinates
78 nh=2; %Number of holonomic constraints
79 nhc=4; %Number of holonomic constraint equations, including drivers
80 nd=0; %number of driving constraints
81 nc=nhc+nb; %Number of constraint equations
82 nv=ngc-nc;
83 nu=nc;
84 NTSDA=0; %Number of TSDA force elements
85 NRSDA=0; %Number of RSDA force elements
86 %SJDT(22,nh); Spatial Joint Data Table
87 %SJDT(:,k)=[T;i;j;si(pr;sj(pr;d;ui(pr;wi(pr;uj(pr;wj(pr];
88 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
89 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,i>0;
90 %si&jpr=vectors to Pi&j; d=dist.; ui(pr, wi(pr, uj(pr, wj(pr=joint vectors
91 SJDT(:,1)=[2;1;0;zer;zer;0;zer;zer;zer]; %Sph. - Body 1 to Ground
92 SJDT(:,2)=[1;1;2;-uz;uz;1;zer;zer;zer;zer]; %Dist. - Body 1 to 2
93 %SMDT(4,nb); Mass Data Table (With full inertia matrix in centroidal frame
94 %parallel to body reference frame)
95 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...
96 %[mnb;Jnbx;Jnby;Jnbz;Jnbxy;Jnbxz;Jnbyz;snbc]];
97 %mi=mass of body i,Jix,Jiy,Jiz are diagonal of inertia matrix,
98 %Jixy, Jixz, and Jixz are products of inertia, sibc is body fixed
99 %vector from origin of body fixed frame to centroid
100 SMDT=[[75;30;30;30;0;0;0;zer],[75;30;30;30;0;0;0;zer]];
101 %STSDAT(12,1); TSDA Data Table
102 if NTSDA==0
103 STSDAT=zeros(12,NTSDA);
104 end
105 %STSDAT(:,T)=[i;j;si(pr;sj(pr;K;C;el0;F];
106 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
107 %C=damping coefficient; el0=spring free length; F=const. force
108 %SRSDAT(15,1); RSDA Data Table

```

```

109 %SRSDAT(:,R)=[i;j;uipr;wipr;ujpr;K;C;thet0;T];
110 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
111 %C=damping coefficient; thet0=spring free angle; T=const. torque
112 if NRSDA==0
113 SRSDAT=zeros(15,NRSDA);
114 end
115 %Body Initial Condition Data Table
116 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
117 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
118 %rQ=vector in x-y plane (as close to y axis as possible),
119 %rOd=velocity of origin, Omeg=angular velocity in global frame
120 BPDDT(:,1)=[zer;ux;uy;zer;zer];
121 BPDDT(:,2)=[-3*uz;-3*uz+ux;-3*uz+uy;zer;5*uy];
122 %Initial generalized coordinates
123 q0=zeros(7*nb,1);
124 qd0=zeros(7*nb,1);
125 j=1;
126 while j<=nb
127 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
128 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
129 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
130 [q,p,A]=InitConfig(rO,rP,rQ);
131 q0=Add(q0,q,7*(j-1),0);
132 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
133 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
134 pd=0.5*EEval(p)'*Omeg;
135 qd=[rOd;pd];
136 qd0=Add(qd0,qd,7*(j-1),0);
137 j=j+1;
138 end
139 end

```

Figure 5.10.8 AppData Function, Spatial Double Pendulum

Numerical results for x coordinate position, velocity, and acceleration of body 2 presented in Fig. 5.10.9, using the trapezoidal integrator with Atol = e-5, are for the pendulum with a distance constraint d = 1 m. Results are essentially identical to those presented in Fig. 5.4.10. This simulation was with the same *integration error tolerances* used in Section 5.4.4, with the same integrator, and satisfied kinematic constraints to the same level of precision reported in Table 5.4.4. Essentially identical results were obtained using the other 7 integrator-formulation alternatives supported in Code 5.9.

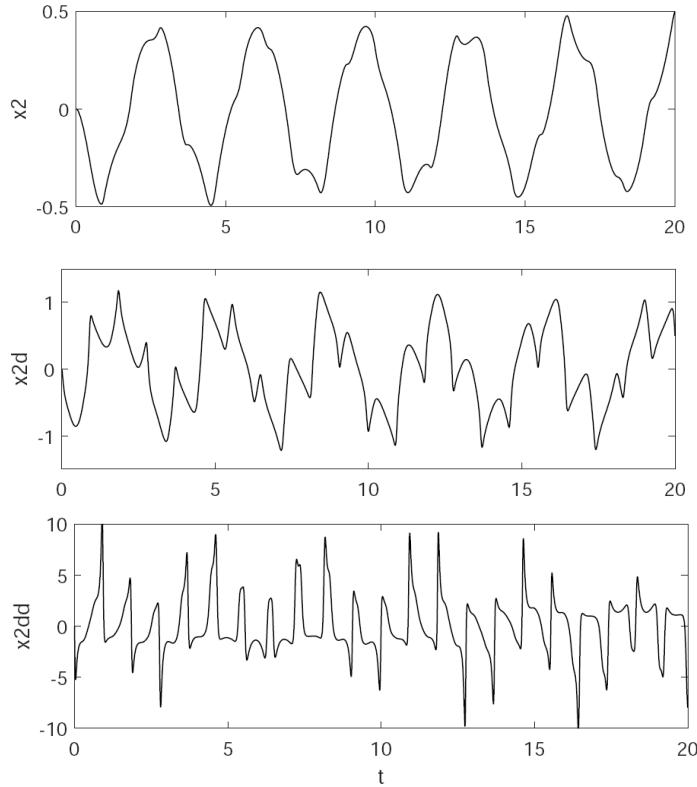


Figure 5.10.9  $x_2$ ,  $x_{2d}$ , and  $x_{2dd}$  vs Time for Spatial Double Pendulum,  $d = 1$

The plot in Fig. 5.10.10 is for the same pendulum, but with the *variable distance constraint*  $d = d(t) = 1 + 0.5\sin(t)$ , which is entered in the P5 function of Fig. 3.10.11 for evaluation of the term  $d(t)^2 / 2$  in the distance constraint. The constraint function and time derivatives are

$$\begin{aligned}\Phi^d &= (1 + 0.5\sin(t))^2 / 2 \\ \Phi_t^d &= (1 + 0.5\sin(t))\cos(t) \\ \Phi_{tt}^d &= -0.5(1 + 0.5\sin(t))\sin(t) + 0.25\cos^2(t)\end{aligned}$$

With  $d = 0$  entered into the distance constraint in the AppData Function, these time dependent functions are entered in the P5 Function as

$$\begin{aligned}Pf &= -\Phi^d \\ Pf_t &= -\Phi_t^d \\ Pf_{tt} &= -\Phi_{tt}^d\end{aligned}\tag{5.10.1}$$

This distance constraint, with a time dependent 50% variation in  $d$ , has a significant influence on dynamics of the system.

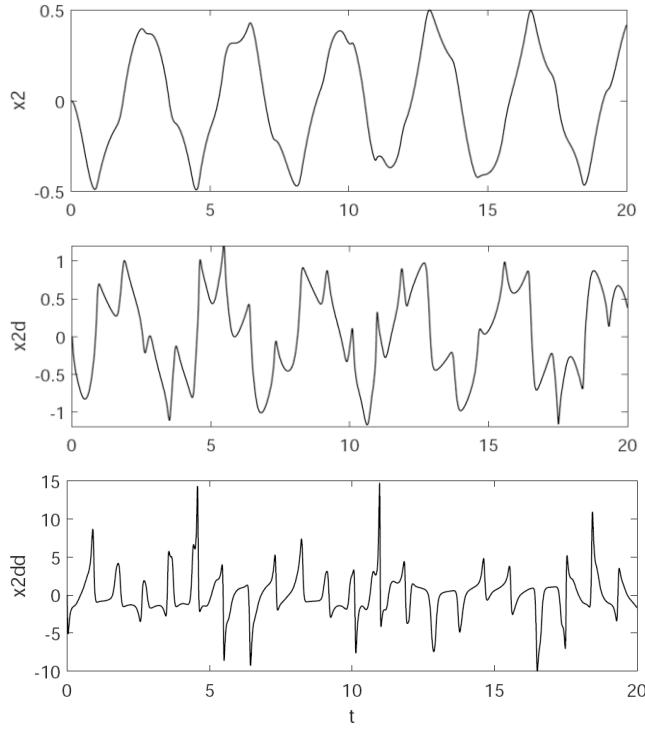


Figure 5.10.10  $x_2$ ,  $x_{2d}$ , and  $x_{2dd}$  vs Time for Double Pendulum,  $d = d(t)$

```

15 if app==2
16 if SJDT(1,2)==9
17 df=(1+0.5*sin(tn));
18 Pf=[zer;-0.5*df^2;0;0];
19 Pfst=[zer;-0.5*df*cos(tn);0;0];
20 Pfstt=[zer;-0.25*(cos(tn))^2+0.5*df*sin(tn);0;0];
21 else
22 Pf=zeros(nc,1);
23 Pfst=zeros(nc,1);
24 Pfstt=zeros(nc,1);
25 end
26 end

```

Figure 5.10.11 P5 Function for Double Pendulum with  $d = d(t)$

Use of general purpose Code 5.9 with the data set of Fig 5.10.8 provides accurate simulation results in a matter of minutes of person time to populate the data set and carry out simulations on a laptop workstation. This is contrasted with the intricate expressions in Section 5.4.4 that required several person days for derivation and creation and debugging of the ad-hoc computer Code 5.4.4 of Appendix 5.A. Timing results reported 140 CPU sec for simulation with the general-purpose Code 5.9 vs 76 CPU sec for simulation with ad-hoc Code 5.4.4. The saving of one minute of CPU time on a laptop workstation hardly compensates for the time consuming and painful process of ad-hoc derivation and programming reported in Section 5.4.4.

#### 5.10.4 Body in Cylindrical Joint with Ground

A single body shown in Fig. 5.10.12 is constrained to rotate and translate in a *cylindrical joint*, with axis along the global z axis in ground. A spring connects point A (0,1,1) on the body with point B (-1,1,1) in ground. With joint reference frames initially at the origin of and aligned with the global x-y-z frame,  $\mathbf{s}_0' = \mathbf{s}_1' = \mathbf{0}$ ,  $\mathbf{u}'^1 = \mathbf{u}_x$ ,  $\mathbf{w}'^1 = \mathbf{u}_z$ ,  $\mathbf{u}'^0 = \mathbf{u}_x$ , and  $\mathbf{w}'^0 = \mathbf{u}_z$ . The kinematically admissible initial position and orientation of body 1 are

$\mathbf{r}_1(0) = \mathbf{0}$  and  $\mathbf{p}_1(0) = [1 \ 0 \ 0]^T$ , as shown in Fig. 5.10.12. Kinematically admissible initial velocity and angular velocity are  $\dot{\mathbf{r}}_1(0) = \mathbf{0}$  and  $\omega_{\text{gaz}}(0) = 10 \text{ rad/sec}$ . The mass and moments of inertia of body 1 are 1 kg and  $1 \text{ kg} \cdot \text{m}^2$  about each axis, gravitational acceleration is  $9.8 \text{ m/sec}^2$  in the negative z direction, the free length of the spring is  $\ell_0 = 1 \text{ m}$ , and the spring constant is  $K = 100 \text{ N/m}$ .

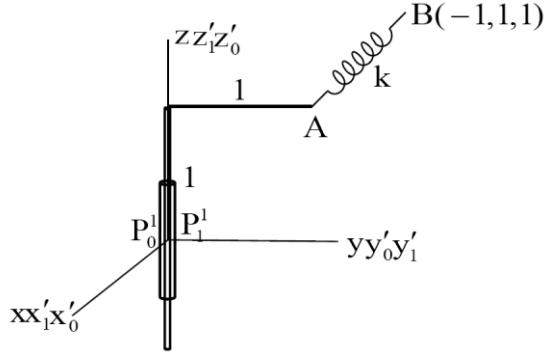


Figure 5.10.12 Body in Cylindrical Joint with Ground

A simulation is carried out using Code 5.9, with data defined in the *AppData Function* of Fig. 5.10.13, the trapezoidal ODE formulation, and *absolute error tolerance Atol = e-5*. The resulting vertical position and velocity and angular velocity about the z axis are shown in Fig. 5.10.14. The error control mechanism led to an average step size of  $7.7e-4 \text{ sec}$  and a single parameterization covered the 6,496 step simulation. Maximum position, velocity, and acceleration constraint norms were  $6e-11$ ,  $3.5e-9$ , and  $1.2e-7$ , respectively. Essentially identical results were obtained with each of the other seven formulations and numerical integrators supported in Code 5.9, with *reparameterizations* ranging from one to 17.

```

142 if app==3 %One Body in Cylindrical Joint with Spring
143 nb=1; %Number of bodies
144 ngc=7*nb; %number of generalized coordinates
145 nh=1; %Number of holonomic constraints
146 nhc=4; %Number of holonomic constraint equations, including drivers
147 nd=0; %number of driving constraints
148 nc=nhc+nb; %Number of constraint equations
149 nv=ngc-nc;
150 nu=nc;
151 NTSDA=1; %Number of TSDA force elements
152 NRSDA=0; %Number of RSDA force elements
153 %SJDT(22,nh): Spatial Joint Data Table
154 %SJDT(:,k)=[T;i;j;si(pr);sj(pr);d;ui(pr);wi(pr);uj(pr);wj(pr)];
155 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
156 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,i>0;
157 %si&jpr=vectors to Pi&j; d=dist.; ui(pr), wi(pr), uj(pr), wj(pr)=joint vectors

```

```

158 SJDT(:,1)=[3;1;0;zer;zer;0;ux;uz;ux;uz]; %Cyl - Body1 to Ground
159 %SMDT(4,nb): Mass Data Table (With full inertia matrix in centroidal frame
160 %parallel to body reference frame)
161 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...,
162 [%nb;Jnbx;Jnby;Jnzb;Jnbxy;Jnbxz;Jnbyz;snbc]];
163 %mi=mass of body i,Jix,Jiy,Jiz are diagonal of inertia matrix,
164 %Jixy, Jixz, and Jixz are products of inertia, sibc is body fixed
165 %vector from origin of body fixed frame to centroid
166 SMDT=[[1;1;1;1;0;0;0;zer]];
167 %STSDAT(12,1): TSDA Data Table
168 if NTSDA==0
169 STSDAT=zeros(12,NTSDA);
170 end
171 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
172 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
173 %C=damping coefficient; el0=spring free length; F=const. force
174 STSDAT(:,1)=[1;0;uy+uz;-ux+uy+uz;100;0;1;0];
175 %SRSDAT(15,1): RSDA Data Table
176 %SRSDAT(:,R)=[i;j;uipr;wipr;ujpr;K;C;thet0;T];
177 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
178 %C=damping coefficient; thet0=spring free angle; T=const. torque
179 if NRSDA==0
180 SRSDAT=zeros(15,NRSDA);
181 end
182 %Body Initial Condition Data Table
183 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
184 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
185 %rQ=vector in x-y plane (as close to y axis as possible),
186 %rOd=velocity of origin, Omeg=angular velocity in global frame
187 BPDDT(:,1)=[zer;ux;uy;zer;10*uz];
188 %Initial generalized coordinates
189 q0=zeros(7*nb,1);
190 qd0=zeros(7*nb,1);
191 j=1;
192 while j<=nb
193 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
194 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
195 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
196 [q,p,A]=InitConfig(rO,rP,rQ);
197 q0=Add(q0,q,7*(j-1),0);
198 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
199 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
200 pd=0.5*EEval(p)'*Omeg;
201 qd=[rOd;pd];
202 qd0=Add(qd0,qd,7*(j-1),0);
203 j=j+1;
204 end
205 end

```

Figure 5.10.13 AppData Function, Body in Cylindrical Joint with Ground

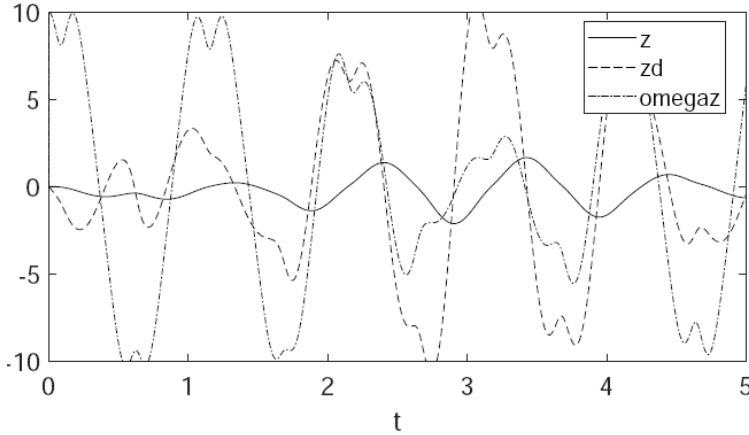


Figure 5.10.14 Vertical Position and Velocity and Angular Velocity of Body in Cylinder

### 5.10.5 Spatial Slider-Crank

The *spatial slider-crank* mechanism shown in Fig. 5.10.15, whose kinematics were studied in Section 3.12.3, consists of two bodies and a distance constraint to model the connecting rod between them. It involves potentially large *constraint contact forces* and a near *singular configuration*. The crank (body 1) rotates about an axis parallel to and offset from the global  $z$  axis in ground (body 0), with a crank in the  $x$ - $y$  plane of radius 0.08 m. The slider (body 2) translates in a guide along the global  $z$  axis in ground. It has an offset of length  $c_2$  in the positive  $Z_2$  direction for distance constraint attachment, which increases constraint reaction forces in the translational joint. A distance constraint of length  $d$  connects the top of the offset in body 2 with the end of the crank in body 1. Data are given in the mks system of units.

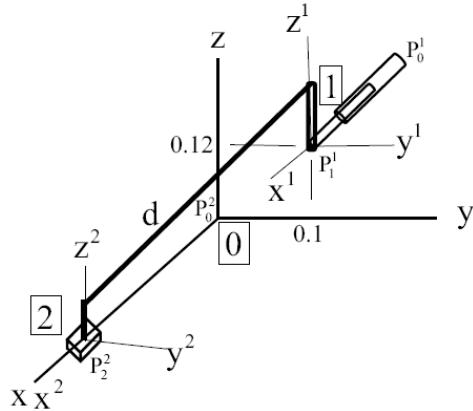


Figure 5.10.15 Spatial Slider-Crank

Since the  $x'$ - $y'$ - $z'$  reference frames in both bodies and ground are parallel in the initial configuration shown, unit vectors used in definition of the joints are  $\mathbf{u}_x^{rik} = \mathbf{u}_z$ ,  $\mathbf{v}_y^{rik} = \mathbf{u}_y$ , and  $\mathbf{w}_z^{rik} = \mathbf{u}_x$ ,  $i = 0, 1, 2$  and  $k = 1$  and 2. For the revolute joint,  $\mathbf{s}'_1 = \mathbf{0}$  and  $\mathbf{s}'_0 = [-1 \quad 0.1 \quad 0.12]^T$ . For the translational joint,  $\mathbf{s}'_2 = \mathbf{s}'_0 = \mathbf{0}$ . For the distance constraint,  $\mathbf{s}'_1 = 0.08\mathbf{u}_z$ ,  $c_2 = 0.02$ ,  $\mathbf{s}'_2 = c_2\mathbf{u}_z$ , and  $d = 0.3$ . Masses and moments of inertia are  $m_1 = 5$  and  $I_{aa} = 5$  for body one and  $m_2 = 5$  and  $I_{\beta\beta} = 0.2$  for body two. Data for simulation, using Code 5.9 of Appendix 5.A, are

presented in the *AppData Function* of Fig. 5.10.16. Due to the complexity of constraint geometry, *initial conditions* are obtained using *kinematic simulation* of the model for two time steps with kinematics Code 3.11.

```

208 if app==4 %Spatial Slider-Crank
209 nb=2; %Number of bodies
210 ngc=7*nb; %number of generalized coordinates
211 nh=3; %Number of holonomic constraints
212 nhc=11; %Number of holonomic constraint equations, including drivers
213 nd=0; %number of driving constraints
214 nc=nhc+nb; %Number of constraint equations
215 nv=ngc-nc;
216 nu=nc;
217 NTSDA=0; %Number of TSDA force elements
218 NRSDA=0; %Number of RSDA force elements
219 %SJDT(22,nh): Spatial Joint Data Table
220 %SJDT(:,k)=[T;i;j;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
221 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
222 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,i>0;
223 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=joint vectors
224 SJDT(:,1)=[4;1;0;zer;0.1*uy+0.12*uz;0;uz;ux;uz;ux]; %Rev-Body1 to Ground
225 SJDT(:,2)=[5;2;0;zer;zer;0;uz;ux;uz;ux]; %Tran-Body2 to Ground
226 SJDT(:,3)=[1;1;2;0.08*uz;0.02*uz;0.24;zer;zer;zer]; %Dist-Body 1 to
227 %SMDT(4,nb): Mass Data Table (With full inertia matrix in centroidal frame
228 %parallel to body reference frame)
229 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...,
230 %[mnb;Jnbx;Jnby;Jnbz;Jnbxy;Jnbxz;Jnbyz;snbc]];
231 %mi=mass of body i,Jix,Jiy,Jiz are diagonal of inertia matrix,
232 %Jixy, Jixz, and Jixz are products of inertia, sibc is body fixed
233 %vector from origin of body fixed frame to centroid
234 SMDT=[[0.5;0.2;0.2;0.2;0;0;zer],[5;0.2;0.2;0.2;0;0;zer]];
235 %STSDAT(12,1): TSDA Data Table
236 if NTSDA==0
237 STSDAT=zeros(12,NTSDA);
238 end
239 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
240 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
241 %C=damping coefficient; el0=spring free length; F=const. force
242 %SRSDAT(15,1): RSDA Data Table
243 %SRSDAT(:,R)=[i;j;uipr;wipr;ujpr;K;C;thet0;T];
244 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
245 %C=damping coefficient; thet0=spring free angle; T=const. torque
246 if NRSDA==0
247 SRSDAT=zeros(15,NRSDA);
248 end
249 %Body Initial Condition Data Table
250 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
251 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
252 %rQ=vector in x-y plane (as close to y axis as possible),
253 %rOd=velocity of origin, Omeg=angular velocity in global frame
254 BPDDT(:,1)=[0.1*uy+0.12*uz;0.1*uy+0.12*uz+ux;0.1*uy+0.12*uz+uy;zer;4*ux];
255 BPDDT(:,2)=[0.1233*ux;0.1233*ux+ux;0.1233*ux+uy;0.2596*ux;zer];
256 %Initial generalized coordinates

```

```

257 q0=zeros(7*nb,1);
258 qd0=zeros(7*nb,1);
259 j=1;
260 while j<=nb
261 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
262 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
263 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
264 [q,p,A]=InitConfig(rO,rP,rQ);
265 q0=Add(q0,q,7*(j-1),0);
266 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
267 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
268 pd=0.5*EEval(p)*Omeg;
269 qd=[rOd;pd];
270 qd0=Add(qd0,qd,7*(j-1),0);
271 j=j+1;
272 end
273 end

```

Figure 5.10.16 AppData Function, Spatial Slider-Crank

Numerical results for angular velocity omega and angular acceleration omegad of the crank for the data set in Fig. 5.10.16 with  $d = 0.3$  m and  $\omega_{10} = 4$  rad/sec are shown in Fig. 5.10.17. Slider  $x_2$ ,  $x_{2d}$ , and  $x_{2dd}$  for the same simulation are shown in Fig. 5.10.18. Plots of angular velocity omega and angular acceleration omegad of the crank in Fig. 5.10.19 for the near singular case with  $d = 0.24$  show more extreme variations, especially in angular acceleration. This is due to *approaching a singular configuration* at  $d = 0.22$

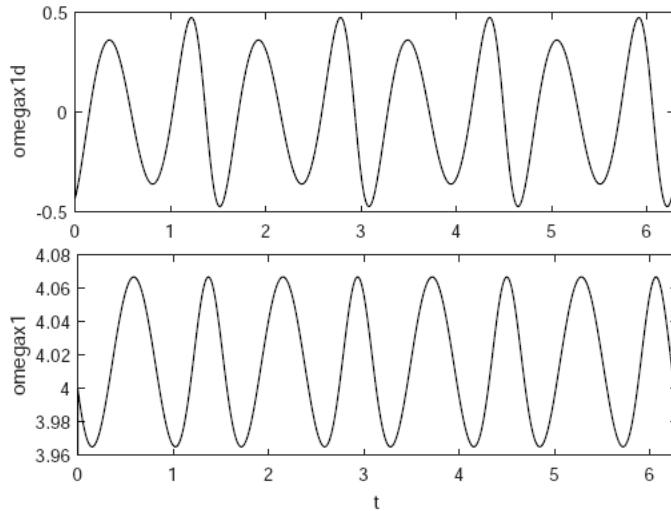
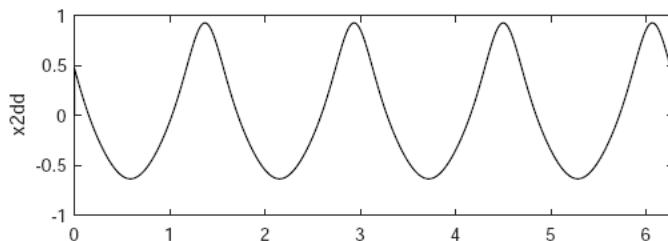


Figure 5.10.17  $\omega_{x1}$  and  $\omega_{x1d}$  vs  $t$  for Spatial Slider-Crank,  $d = 0.3$



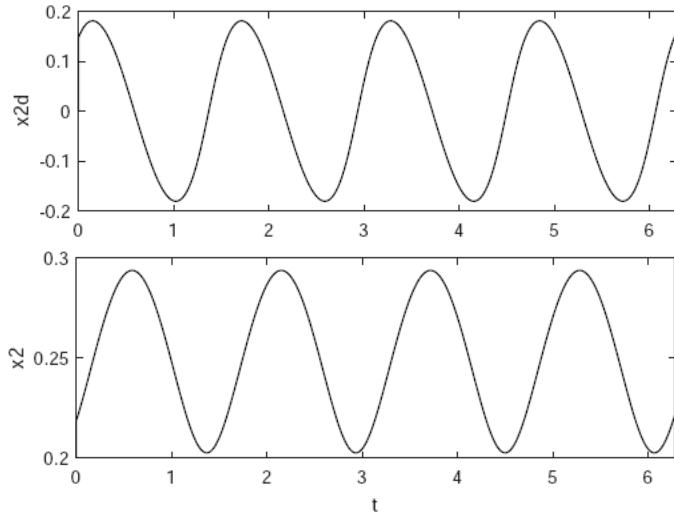


Figure 5.10.18  $x_2$ ,  $x_{2d}$ , and  $x_{2dd}$  vs  $t$  for Spatial Slider-Crank,  $d = 0.3$

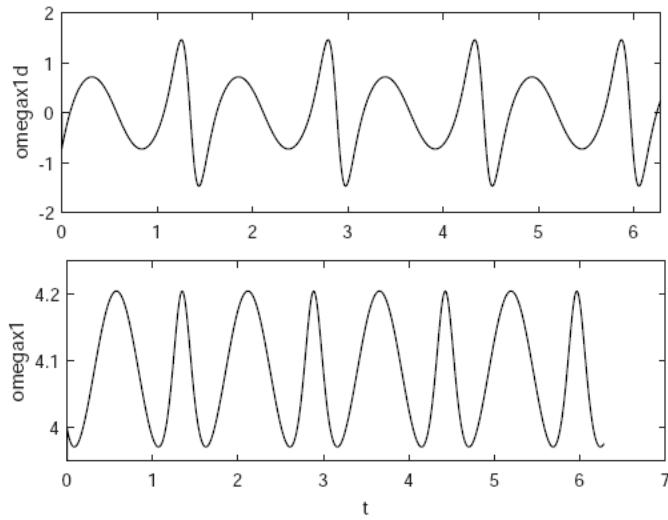


Figure 5.10.19  $\omega$  and  $\omega_{ad}$  vs  $t$  for Spatial Slider-Crank,  $d = 0.24$

Even for the most extreme simulation, the *maximum total energy variation* is  $2e-3\%$  and the maximum position, velocity, and acceleration error norms over the simulation are  $e-9$ ,  $2e-11$ , and  $2e-8$ , respectively. These results are obtained using the trapezoidal integrator with the ODE formulation and Atol =  $e-5$ . Essentially identical results are obtained using the other 7 integrator-formulation alternatives supported in Code 5.9.

The ease with which this system model was created using Code 5.9 in about two person hours and accurate results obtained in an 8 CPU sec simulation on a laptop workstation belies the extreme effort that would otherwise be required if ad-hoc equation derivation and code writing and debugging.

### 5.10.6 Four-Body Mechanism with Translational Joints in Ground

The *four-body mechanism* shown in Fig. 5.10.20 is comprised of four bodies that translate along coordinate axes, connected by distance constraints and springs. The model employs 28 generalized coordinates and a total of 26 constraint equations, for two *degrees of*

*freedom.* Body reference frames in each of the four bodies are parallel to the global reference frame, leading to the constraint data  $\mathbf{u}'^i = \mathbf{u}_z$  and  $\mathbf{w}'^i = \mathbf{u}_x$ ,  $i = 1, 0$  and  $4, 0$ ,  $\mathbf{u}'^i = \mathbf{u}_z$  and  $\mathbf{w}'^i = \mathbf{u}_y$ ,  $i = 2, 0$ , and  $\mathbf{u}'^i = \mathbf{u}_y$  and  $\mathbf{w}'^i = \mathbf{u}_z$ ,  $i = 3, 0$ . Distance constraint lengths are  $\ell_1 = 5$  and  $\ell_2 = 7$  and initial positions of the masses on their associated axes are  $x_1^0 = 4$ ,  $y_2^0 = 3$ ,  $z_3^0 = 6.32$ , and  $x_4^0 = 5$ . The *AppData Function* is given in Fig. 5.10.21.

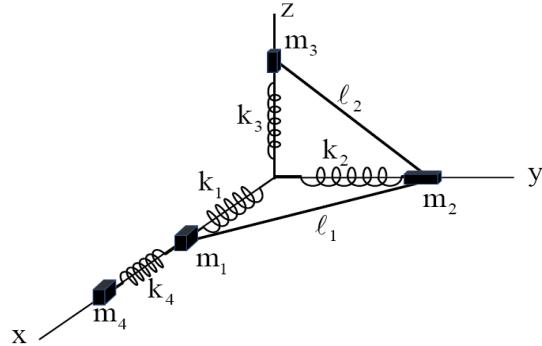


Figure 5.10.20 Four Body Mechanism with Translational Joints in Ground

```

412 if app==7 %4-Translating Mass Model
413 nb=4; %Number of bodies
414 ngc=7*nb; %number of generalized coordinates
415 nh=6; %Number of holonomic constraints
416 nhc=22; %Number of holonomic constraint equations
417 nd=0; %number of driving constraints
418 nc=nhc+nb; %Number of constraint equations
419 nv=ngc-nc;
420 nu=nc;
421 NTSDA=4; %Number of TSDA force elements
422 NRSDA=0; %Number of RSDA force elements
423 %SJDT(22,nh): Spatial Joint Data Table
424 %SJDT(:,k)=[T;i;j;sipr;sjpr;d;uipr;wipr;ujpr;wjpr];
425 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
426 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,i>0;
427 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=joint vectors
428 SJDT(:,1)=[5;1;0;zer;zer;0;uz;ux;uz;ux]; %Tran-Body1 to Ground
429 SJDT(:,2)=[5;2;0;zer;zer;0;uz;uy;uz;uy]; %Tran-Body2 to Ground
430 SJDT(:,3)=[5;3;0;zer;zer;0;uy;uz;uy;uz]; %Tran-Body3 to Ground
431 SJDT(:,4)=[5;4;0;zer;zer;0;uz;ux;uz;ux]; %Tran-Body4 to Ground
432 SJDT(:,5)=[1;1;2;zer;zer;5;zer;zer;zer]; %Dist-Body 1 to 2
433 SJDT(:,6)=[1;2;3;zer;zer;7;zer;zer;zer]; %Dist-Body 2 to 3
434 %SMDT(4,nb): Mass Data Table (With full inertia matrix in centroidal frame
435 %parallel to body reference frame)
436 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...,
437 %[mnb;Jnbx;Jnby;Jnbz;Jnbxy;Jnbxz;Jnbyz;snbc]];
438 %mi=mass of body i,Jix,Jiy,Jiz are diagonal of inertia matrix,
439 %Jixy, Jixz, and Jixz are products of inertia, sibc is body fixed
440 %vector from origin of body fixed frame to centroid
441 SMDT=[[2;0.2;0.2;0.2;0;0;zer],[2;0.2;0.2;0.2;0;0;zer],...
442 [6;0.2;0.2;0.2;0;0;0;zer],[6;0.2;0.2;0.2;0;0;0;zer]];
443 %STSDAT(12,1): TSDA Data Table
444 if NTSDA==0

```

```

445 STSDAT=zeros(12,NTSDA);
446 end
447 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
448 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
449 %C=damping coefficient; el0=spring free length; F=const. force
450 STSDAT(:,1)=[1;0;10*ux;zer;2;0;14;0]; %mass1 to ground
451 STSDAT(:,2)=[2;0;10*uy;zer;2;0;13;0]; %mass2 to ground
452 STSDAT(:,3)=[3;0;10*uz;zer;2;0;16.32;0]; %mass3 to ground
453 STSDAT(:,4)=[1;4;zer;10*ux;10;0;11;0]; %mass4 to mass1
454 %SRSDAT(15,1): RSDA Data Table
455 %SRSDAT(:,R)=[i;j;uipr;wipr;ujpr;K;C;thet0;T];
456 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
457 %C=damping coefficient; thet0=spring free angle; T=const. torque
458 if NRSDA==0
459 SRSDAT=zeros(15,NRSDA);
460 end
461 %Body Initial Condition Data Table
462 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
463 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
464 %rQ=vector in x-y plane (as close to y axis as possible),
465 %rOd=velocity of origin, Omeg=angular velocity in global frame
466 BPDDT(:,1)=[4*ux;5*ux;4*ux+uy;zer;zer];
467 BPDDT(:,2)=[3*uy;3*uy+ux;4*uy;zer;zer];
468 BPDDT(:,3)=[6.32*uz;6.32*uz+ux;6.32*uz+uy;zer;zer];
469 BPDDT(:,4)=[5*ux;6*ux;5*ux+uy;-ux;zer];
470 %Initial generalized coordinates
471 q0=zeros(7*nb,1);
472 qd0=zeros(7*nb,1);
473 j=1;
474 while j<=nb
475 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
476 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
477 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
478 [q,p,A]=InitConfig(rO,rP,rQ);
479 q0=Add(q0,q,7*(j-1),0);
480 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
481 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
482 pd=0.5*GEval(p)*Omeg;
483 qd=[rOd;pd];
484 qd0=Add(qd0,qd,7*(j-1),0);
485 j=j+1;
486 end
487 end

```

Fig. 5.10.21 AppData Function, Spatial Four Body Mechanism

Simulations with *error control parameters*  $intol = \text{e-}6$  and  $Atol = \text{e-}5$ , using the variable step size SDIRK54 integrator with an upper bound on step size of  $h = 0.01$ , yielded position and velocity plots presented in Fig. 5.10.22. Since this application has a smooth response, the excellent step size control feature of the integrator quickly raised the initial e-3 step size to e-2, consistent with the *integration error tolerance*. Even in this simulation with 28 generalized coordinates and 26 constraints, it provided maximum norms of position, velocity, and acceleration constraint error of e-10, 3e-9, and 2e-8, respectively. While this model is of higher dimension than previous examples, all eight formulations and integration algorithms supported in

Code 5.9 perform comparably. Time steps taken were at the maximum allowable hmax and 2 to 5 *reparameterizations* were sufficient in all cases.

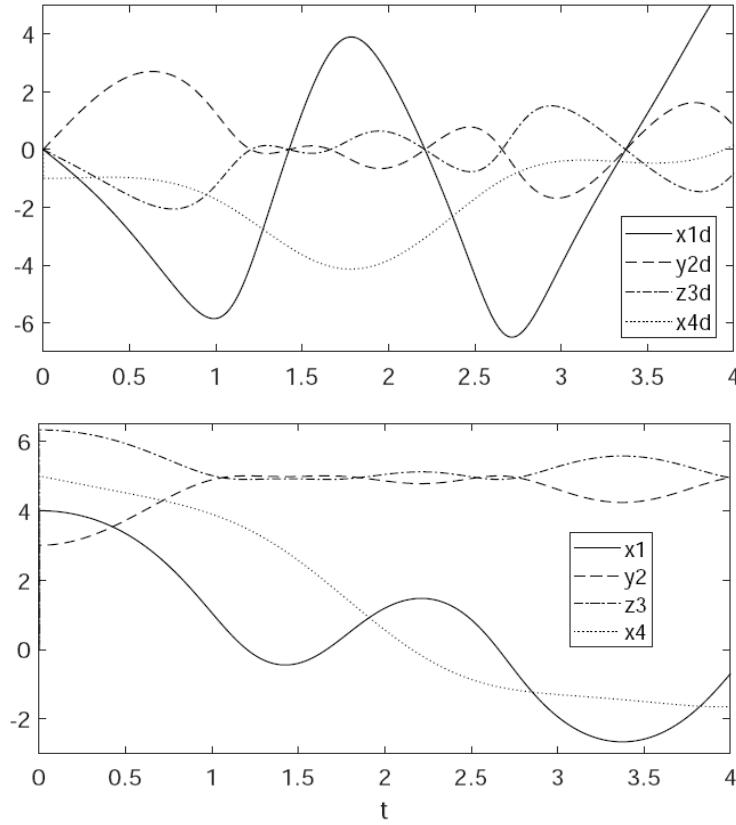


Figure 5.10.22 Positions and Velocities of the Four Bodies

### 5.10.7 Fly-Ball Governor

The *fly-ball governor* of Fig. 5.10.23, defined in Section 3.3.7.4 and kinematically analyzed in Section 3.12.4, is comprised of four moving bodies and ground. Body 1 is the rotor in a revolute joint with ground. Body 2 is the *collar* that translates on and rotates with the rotor. Bodies 3 and 4 contain heavy balls at their ends, which are taken as their centroids, and rotate in revolute joints at the top of the rotor ( $z = 0.2$ ). Two distance constraints connect bodies 3 and 4 with the collar, to *control fuel feed*. The revolute joints with bodies 3 and 4 on body 1 are offset 0.02 m from its centerline, as are the distance constraint attachment points on body 2. With three revolute and one translational joint, two distance constraints and four Euler parameter normalization constraints, there are a total of 26 constraint equations in 28 generalized coordinates. If the constraint Jacobian has full rank, the system has two degrees of freedom.

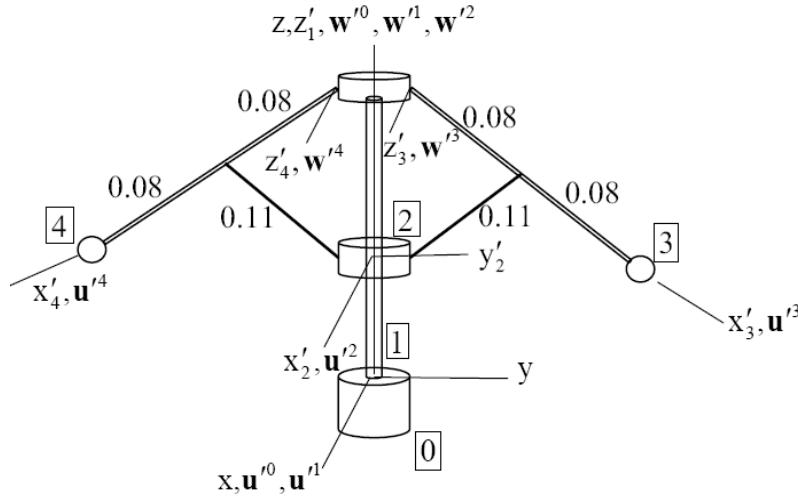


Figure 5.10.23 Fly-Ball Governor

Gravity acts in the negative  $z$ -direction. The rotor is geared to rotate with the engine, so that as engine speed increases, centrifugal force throws the balls outward and upward. A linkage connects the collar to engine fuel feed, so as the engine slows and the balls and collar drop, fuel feed is increased, with the goal of maintaining a constant engine speed. The initial angular velocity of the engine is zero. The *torque control algorithm* is presented as the torque tau that acts in the revolute joint between the shaft and ground in Fig. 5.10.23. An engine torque of  $-300$  n.m is applied if the  $z$  coordinate of the collar,  $q(10)$  is less than  $0.1$  m, and an extreme external load  $\tau_{\text{load}} = 500 \sin((tn - 6)\pi)^2$  n.m acts between  $tn = 3$  and  $tn = 6$  sec. These torques are implemented with the code of Fig. 5.10.24 in the QAEval function of Code 5.9 for dynamic simulation.

```

77 % For the fly-bball governor,
78 if app==8
79 tauload=0;
80 if q(10)<0.1
81 tau=-300;
82 else
83 tau=0;
84 end
85 if tn>3
86 tauload=500*sin((tn-3)*pi)^2;
87 end
88 if tn>6
89 tauload=0;
90 end
91 tau=tau+tauload;
92 end

```

Figure 5.10.24 Torques Acting in Fly-Ball Governor

Clearly, carrying out an ad-hoc derivation of equations of motion and code generation would be oppressive. This is an excellent example of a system that requires automated equation generation and computation. Data for simulation are given in the AppDataFunction of Fig. 5.10.25

490 if app==8 %Fly-Ball Governor

```

491 nb=4; %Number of bodies
492 ngc=7*nb; %number of generalized coordinates
493 nh=6; %Number of holonomic constraints
494 nhc=22; %Number of holonomic constraint equations
495 nd=0; %number of driving constraints
496 nc=nhc+nb; %Number of constraint equations
497 nv=ngc-nc;
498 nu=nc;
499 NTSDA=1; %Number of TSDA force elements
500 NRSDA=1; %Number of RSDA force elements
501 %SJDT(22,nh): Spatial Joint Data Table
502 %SJDT(:,k)=[T;ij;jipr;sjpr;d;uipr;wipr;ujpr;wjpr];
503 %k=joint No.; T=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
504 %6=Univ, 7=Strut, 8=Rev-Sph, 9=DistDr); i&j=bodies conn.,i>0;
505 %si&jpr=vectors to Pi&j; d=dist.; uipr, wipr, ujpr, wjpr=joint vectors
506 SJDT(:,1)=[4;1;0;zer;zer;0;ux;uz;ux;uz]; %Rev-Body1 to Ground
507 SJDT(:,2)=[5;1;2;zer;zer;0;ux;uz;ux;uz]; %Tran-Body1 to body 2
508 SJDT(:,3)=[4;1;3;0.2*uz+0.02*uy;zer;0;uz;ux;ux;uz]; %Rev-Body1 to body 3
509 SJDT(:,4)=[4;1;4;0.2*uz-0.02*uy;zer;0;uz;ux;ux;uz]; %Rev-Body1 to body 4
510 SJDT(:,5)=[1;2;3;0.02*uy;0.08*ux;0.11;zer;zer;zer]; %Dist-Body 2 to body3
511 SJDT(:,6)=[1;2;4;-0.02*uy;0.08*ux;0.11;zer;zer;zer]; %Dist-Body 2 to body4
512 %SMDT(4,nb): Mass Data Table (With full inertia matrix in centroidal frame
513 %parallel to body reference frame)
514 %SMDT=[[m1;J1x;J1y;J1z;J1xy;J1xz;J1yz;s1c],...,
515 %[mnb;Jnbx;Jnby;Jnbz;Jnbxy;Jnbxz;Jnbyz;snbc]];
516 %mi=mass of body i,Jix,Jiy,Jiz are diagonal of inertia matrix,
517 %Jixy, Jixz, and Jixz are products of inertia, sibc is body fixed
518 %vector from origin of body fixed frame to centroid
519 SMDT=[[200;25;25;50;0;0;0;zer],[1;0.15;0.15;0.125;0;0;0;zer],...
520 [1;0.1;0.1256;0.1256;0;0;0;0.16*ux],[1;0.1;0.1256;0.1256;0;0;0;...
521 0.16*ux]];
522 %STSDAT(12,1): TSDA Data Table
523 if NTSDA==0
524 STSDAT=zeros(12,NTSDA);
525 end
526 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
527 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
528 %C=damping coefficient; el0=spring free length; F=const. force
529 STSDAT(:,1)=[1;2;zer;uz;1000;300;1.1;0]; %Body 1 to Body 2
530 %SRSDAT(15,1): RSDA Data Table
531 %SRSDAT(:,R)=[i;j;uipr;wipr;ujpr;K;C;thet0;T];
532 %R=RSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
533 %C=damping coefficient; thet0=spring free angle; T=const. torque
534 if NRSDA==0
535 SRSDAT=zeros(15,NRSDA);
536 end
537 T=0;
538 SRSDAT(:,1)=[1;0;ux;uz;ux;0;0;0;T];
539 %Body Initial Condition Data Table
540 %BPDDT(:,k)=[rO;rP;rQ;rOd;Omeg];
541 %k=body No., rO=vector to body fixed origin, rP=vector to point on x-axis,
542 %rQ=vector in x-y plane (as close to y axis as possible),
543 %rOd=velocity of origin, Omeg=angular velocity in global frame
544 omegaz0=0;

```

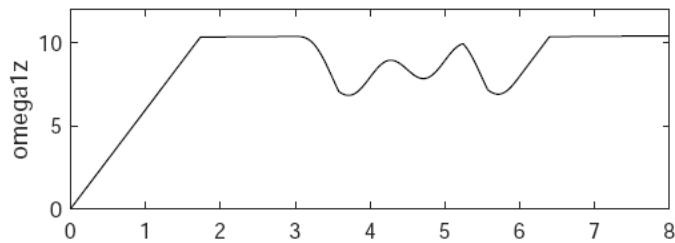
```

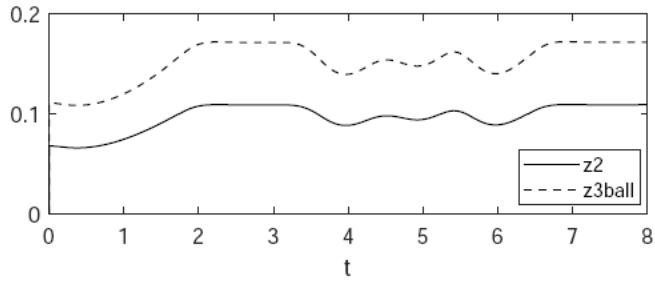
545 BPDDT(:,1)=[zer;ux;uy;zer;omegaz0*uz];
546 BPDDT(:,2)=[0.06*uz;0.06*uz+ux;0.06*uz+uy;zer;omegaz0*uz];
547 BPDDT(:,3)=[0.2*uz+0.02*uy;0.2*uz+0.02*uy+0.15*uy-0.1*uz;...
548 0.2*uz+0.02*uy+atil(ux)*(0.15*uy-0.1*uz);...
549 zer;omegaz0*uz];
550 BPDDT(:,4)=[0.2*uz-0.02*uy;0.2*uz-0.02*uy-0.15*uy-0.1*uz;...
551 0.2*uz-0.02*uy+atil(ux)*(-0.15*uy-0.1*uz);zer;omegaz0*uz];
552 %Initial generalized coordinates
553 q0=zeros(7*nb,1);
554 qd0=zeros(7*nb,1);
555 j=1;
556 while j<=nb
557 rO=[BPDDT(1,j);BPDDT(2,j);BPDDT(3,j)];
558 rP=[BPDDT(4,j);BPDDT(5,j);BPDDT(6,j)];
559 rQ=[BPDDT(7,j);BPDDT(8,j);BPDDT(9,j)];
560 [q,p,A]=InitConfig(rO,rP,rQ);
561 q0=Add(q0,q,7*(j-1),0);
562 rOd=[BPDDT(10,j);BPDDT(11,j);BPDDT(12,j)];
563 Omeg=[BPDDT(13,j);BPDDT(14,j);BPDDT(15,j)];
564 pd=0.5*GEval(p)*Omeg;
565 qd=[rOd;pd];
566 qd0=Add(qd0,qd,7*(j-1),0);
567 j=j+1;
568 end
569 end

```

Figure 5.10.25 AppData for Fly-Ball Governor

Results of the simulation carried out with the SDIRK54 implicit integrator with variable time step are plotted in Fig. 5.10.26. The governor brings the output shaft from rest to a steady angular velocity of 10 rad/sec. When the large load is applied, the governor works to maintain angular velocity and, after the load subsides, brings the angular velocity back to 10 rad sec. This simulation required 61 *reparameterizations* in 870 time steps (14 time steps per reparameterization). Maximum configuration, velocity, and acceleration constraint error norms were e-10, e-11, and e-11.





### 5.10.26 Fly-Ball Governor Dynamic Performance

Analysis of spatial multibody systems using Code 5.9 of Appendix 5.A confirms that properties of the tangent space ODE and Index 0 DAE formulations and four numerical integration methods in generating and solving the equations of motion are as demonstrated in planar applications with Code 5.7. The effectiveness of using a general-purpose code for spatial applications is amplified by the complexity that would otherwise be encountered in ad-hoc derivation and coding of spatial equations of motion, such as those presented in Sections 5.4 and 5.6.

## 5.11 Well Posed Formulations of Holonomic Mechanical System Dynamics

In preparation for treatment of the Full DAE of Section 4.10.2 in Chapter 7 and friction effects in Chapter 8, existence, uniqueness, and regularity of solutions of equations of motion must be established. The result presented in Section 5.2.4 that the tangent space ODE is *well posed* is extended in this section, to include the tangent space Index 0 DAE, Full DAE, and d'Alembert variational formulations. To realize this objective, dependence of system kinematic and dynamic equations on *design variables*, the primary form of *problem data* considered, is accounted for and an abbreviated derivation of the tangent space ODE, as a function of design variables, is presented. Theorems that establish well posedness of each of the formulations treated are presented and proved.

### 5.11.1 Introduction

As stated in the *ODE existence theorem* of Section 4.7.3, a *well posed problem* is defined as one for which *there exists a unique solution that depends continuously on problem data*. The existence and uniqueness components of the definition are broadly embraced in the mathematical literature, but the issue of *continuous dependence on problem data* receives less attention. In the field of mechanical system dynamics, *well posedness* is most often invoked for systems whose dynamics are represented by explicitly defined ODE initial-value problems. For holonomic systems that are modelled using the *Full DAE*, as in Section 4.10.2, existence and uniqueness results are available (Rheinboldt, 1991), but continuous dependence on data remains to be established. For systems whose dynamics are represented by d'Alembert's form of variational equations of motion in Section 4.6, the concept of well posedness is seldom addressed.

Recent work on *dynamic system design sensitivity analysis (DSA)* (Callejo and Garc,a de Jalon,2014; Dopico, Zhu, Sandu, and Sandu, 2015; Banerjee and McPhee, 2016), which relies on differentiability of dynamic response with respect to design variables, has used Maggi and Kane ODE formulations of system dynamics, assuming they are well posed. As shown in Appendix 5.D, these formulations are subject to errors due to failure to enforce configuration constraints (Haug, 2019d), and may not be well posed.

Design dependence in the *d'Alembert variational formulation* of the equations of mechanical system dynamics is introduced in Section 5.11.2 and associated *dependence of initial conditions on design variables* is analyzed in Section 5.11.3. Tangent space ODE of system dynamics that account for design dependence are derived in Section 5.11.4. It is shown in Section 5.11.5 that the tangent space ODE are well posed and equivalent to the Index 0 DAE formulation, which is also shown to be well posed. It is proved in Sections 5.11.6 and 5.11.7 that the variational and Full DAE formulations are well posed.

### 5.11.2 Design Dependence in Variational Equations of Motion

*Mechanical systems* treated here are defined as collections of rigid bodies whose position and orientation relative to an inertial reference frame are defined by generalized coordinates  $\mathbf{q} = [q_1 \quad \dots \quad q_n]^T \in \mathbb{R}^{n_{gc}}$  that satisfy *holonomic constraints* of the form

$$\Phi(\mathbf{q}, \mathbf{b}) = [\Phi_1(\mathbf{q}, \mathbf{b}) \quad \dots \quad \Phi_{n_{hc}}(\mathbf{q}, \mathbf{b})]^T = \mathbf{0} \quad (5.11.1)$$

where  $\mathbf{b} = [b_1 \ \dots \ b_k]^T \in \mathbb{R}^k$  is a *vector of problem data, or design variables*, that do not depend on time and define the geometry and kinetics of the system. *Time independent constraints* are treated here, but the formulation may be extended to time dependent constraints, as in Sections 5.2 and 5.5. Differentiating Eq. (5.11.1) with respect to time yields velocity and acceleration constraints,

$$\Phi_q(\mathbf{q}, \mathbf{b})\dot{\mathbf{q}} = \mathbf{0} \quad (5.11.2)$$

$$\Phi_q(\mathbf{q}, \mathbf{b})\ddot{\mathbf{q}} + \left( \Phi_q(\mathbf{q}, \mathbf{b})\ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}} \equiv \Phi_q(\mathbf{q}, \mathbf{b})\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}) = \mathbf{0} \quad (5.11.3)$$

It is important that the Jacobian  $\Phi_q(\mathbf{q}, \mathbf{b})$  of the holonomic constraints has full row rank, for all  $\mathbf{q}$  and  $\mathbf{b}$  in neighborhoods  $X_0 \subset \mathbb{R}^{ngc}$  of  $\mathbf{q}^0 = \mathbf{q}(t^0)$  and  $B_0 \subset \mathbb{R}^k$  of  $\mathbf{b}$ .

Dynamics of the system is governed by *d'Alembert's principle* of Section 4.6,

$$\delta \mathbf{q}^T (\mathbf{M}(\mathbf{q}, \mathbf{b})\ddot{\mathbf{q}} - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b})) = 0 \quad (5.11.4)$$

for  $\mathbf{b} \in B_0$ , which must hold for all  $\delta \mathbf{q}$  that satisfy

$$\Phi_q(\mathbf{q}, \mathbf{b})\delta \mathbf{q} = \mathbf{0} \quad (5.11.5)$$

It is critical to note that the noncentroidal form of Eq. (4.6.16) is used, since the vector  $\mathbf{s}_i^c(\mathbf{b})$  from the origin of the body reference frame to the centroid of body  $i$  will in general depend on design  $\mathbf{b}$ , for both planar and spatial systems.

Hypotheses for mechanical systems considered are that the mass matrix  $\mathbf{M}(\mathbf{q}, \mathbf{b})$  is positive definite on the null space of  $\Phi_q(\mathbf{q}, \mathbf{b})$ ,  $\Phi_q(\mathbf{q}, \mathbf{b})$  has full row rank,  $\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t)$  is a vector of generalized applied forces,  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b})$  is a vector of velocity coupling terms, and all functions that appear in Eqs. (5.11.1) through (5.11.4) are at least continuously differentiable with respect to  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\mathbf{b}$ . Consistent with d'Alembert's principle, no forces of constraint appear in Eq. (5.11.4).

### 5.11.3 Initial Conditions as Functions of Problem Data

At initial time  $t^0$ , Eq. (5.11.1) must hold; i.e.,  $\Phi(\mathbf{q}^0, \mathbf{b}) = \mathbf{0}$ . In addition to these nhc conditions,  $ngc - nhc$  *initial configuration conditions* on  $\mathbf{q}$  must be imposed, in the form

$$\Psi(\mathbf{q}^0, \mathbf{b}) = \mathbf{0} \quad (5.11.6)$$

where  $\Psi \in \mathbb{R}^{ngc-nhc}$ . The combined initial conditions are

$$\begin{bmatrix} \Phi(\mathbf{q}^0, \mathbf{b}) \\ \Psi(\mathbf{q}^0, \mathbf{b}) \end{bmatrix} = \mathbf{0} \quad (5.11.7)$$

and the associated Jacobian,

$$\begin{bmatrix} \Phi_q^0 \\ \Psi_{q^0} \end{bmatrix} = \begin{bmatrix} \Phi(q^0, \ddot{\mathbf{b}})_{q^0} \\ \Psi(q^0, \ddot{\mathbf{b}})_{q^0} \end{bmatrix} = \begin{bmatrix} \Phi_q(q^0, \mathbf{b}) \\ \Psi_{q^0}(q^0, \mathbf{b}) \end{bmatrix}$$

is required to have full rank. As long as  $\mathbf{q}^0$  satisfies Eq. (5.11.7), the implicit function theorem of Section 2.2.5 implies that Eq. (5.11.7) uniquely determines the initial configuration as a continuously differentiable function of  $\mathbf{b}$ ; i.e.,  $\mathbf{q}(t^0, \mathbf{b}) = \mathbf{q}^0 = \mathbf{q}^0(\mathbf{b})$ , for all  $\mathbf{b} \in B_0$ .

*Initial velocity conditions* on  $\dot{\mathbf{q}}$  are generally nge – nhc linear equations of the form

$$\Theta(\mathbf{q}^0(\mathbf{b}), \mathbf{b})\dot{\mathbf{q}}^0 = \mathbf{a}(\mathbf{b}) \quad (5.11.8)$$

where the coefficient matrix  $\Theta(\mathbf{q}^0(\mathbf{b}), \mathbf{b})$  and vector  $\mathbf{a}(\mathbf{b})$  are continuously differentiable functions of  $\mathbf{q}^0$  and  $\mathbf{b}$ . This includes initial conditions in spatial systems that are stated in terms of angular velocity. Combining Eq. (5.11.2), evaluated at  $t^0$ , with Eq. (5.11.8),

$$\begin{bmatrix} \Phi_q(\mathbf{q}^0(\mathbf{b}), \mathbf{b}) \\ \Theta(\mathbf{q}^0(\mathbf{b}), \mathbf{b}) \end{bmatrix} \dot{\mathbf{q}}^0(\mathbf{b}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{a}(\mathbf{b}) \end{bmatrix} \quad (5.11.9)$$

The coefficient matrix of Eq. (5.11.9) must be nonsingular to uniquely determine the initial velocity as a continuously differentiable function of  $\mathbf{b}$ ; i.e.,  $\dot{\mathbf{q}}(t^0, \mathbf{b}) = \dot{\mathbf{q}}^0 = \dot{\mathbf{q}}^0(\mathbf{b})$ , for all  $\mathbf{b} \in B_0$ .

Accounting for *design dependence of initial conditions* is a critical component of DSA. Neglecting design dependence of initial conditions leads to serious errors in results. In particular, even if initial configuration or velocity conditions of Eqs. (5.11.6) and (5.11.8) do not depend explicitly on  $\mathbf{b}$ , Eqs. (5.11.7) and (5.11.9) depend on  $\mathbf{b}$ , through the kinematic constraint equations and initial conditions depend on  $\mathbf{b}$ .

#### 5.11.4 ODE of Motion

Derivation of design dependent tangent space ODE is summarized in this section, as an extension of the development of Section 5.2. At initial time  $t^0$  and at a nominal design  $\mathbf{b}^0$ ,  $\mathbf{U}$  is defined as the nge  $\times$  nhc matrix

$$\mathbf{U} \equiv \Phi_q^T(\mathbf{q}^0(\mathbf{b}^0), \mathbf{b}^0) \quad (5.11.10)$$

which has full rank. A second matrix  $\mathbf{V}$  is defined at  $(t^0, \mathbf{b}^0)$  as the solution of

$$\begin{aligned} \Phi_q(\mathbf{q}^0(\mathbf{b}^0), \mathbf{b}^0)\mathbf{V} &= \mathbf{U}^T\mathbf{V} = \mathbf{0} \\ \mathbf{V}^T\mathbf{V} &= \mathbf{I} \end{aligned} \quad (5.11.11)$$

using singular value decomposition (Strang, 1980). The second of Eqs. (5.11.11) implies that  $\mathbf{V}$  has full rank. Further, since the rank of  $\Phi_q(\mathbf{q}^0(\mathbf{b}^0), \mathbf{b}^0)$  is nhc,  $\mathbf{V}$  is an nge  $\times$  (nge-nhc) matrix (Strang, 1980). Since its columns are orthogonal to the columns of  $\mathbf{U}$ , the columns of  $\mathbf{U}$  and  $\mathbf{V}$  span  $R^{nge}$ . As in Eq. (5.2.7), matrices  $\mathbf{V}$  and  $\mathbf{U}$  are further related by the identity

$$\mathbf{V}\mathbf{V}^T + \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T = \mathbf{I} \quad (5.11.12)$$

Since every vector in  $\mathbb{R}^{\text{ngc}}$  can be represented as a unique linear combination of columns of  $\mathbf{V}$  and  $\mathbf{U}$ , a *tangent space parameterization* of system kinematics that accounts for design dependence is

$$\mathbf{q} = \mathbf{q}^0(\mathbf{b}) + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u} \quad (5.11.13)$$

At  $t^0$ ,  $\mathbf{q}(t^0, \mathbf{b}) = \mathbf{q}^0(\mathbf{b})$ , so Eq. (5.11.13) reduces to  $\mathbf{V}\mathbf{v}^0 - \mathbf{U}\mathbf{u}^0 = \mathbf{0}$ . Multiplication on the left by  $\mathbf{V}^T$  and  $\mathbf{U}^T$ , using Eqs. (5.11.11) and the fact that  $\mathbf{U}^T\mathbf{U}$  is nonsingular, yields

$$\begin{aligned} \mathbf{v}^0 &= \mathbf{0} \\ \mathbf{u}^0 &= \mathbf{0} \end{aligned} \quad (5.11.14)$$

which do not depend on  $\mathbf{b}$ .

For  $\mathbf{q}$  of Eq. (5.11.13) to satisfy the constraints of Eq. (5.11.1), it is required that

$$\Phi\left(\left(\mathbf{q}^0(\mathbf{b}) + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}\right), \mathbf{b}\right) = \mathbf{0} \quad (5.11.15)$$

The partial derivative of the left side of this equation with respect to  $\mathbf{u}$  is

$$\left(\Phi\left(\left(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}\right), \mathbf{b}\right)\right)_u = -\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{U}. \text{ At } t^0, \text{ this is}$$

$$\Phi_u(\mathbf{q}^0(\mathbf{b}), \mathbf{b}) = -\Phi_q(\mathbf{q}^0(\mathbf{b}), \mathbf{b})\mathbf{U} \equiv -(\mathbf{B}^0(\mathbf{b}))^{-1} \quad (5.11.16)$$

which is nonsingular in a neighborhood  $B_0$  of  $\mathbf{b}^0$ . Since  $\Phi_q(\mathbf{q}, \mathbf{b})$  is a continuous function of  $\mathbf{q}$  and  $\mathbf{b}$ ,

$$\mathbf{B}(\mathbf{q}, \mathbf{b}) \equiv (\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{U})^{-1} \quad (5.11.17)$$

is nonsingular and continuously differentiable with respect to  $\mathbf{q}$  and  $\mathbf{b}$  in a neighborhood  $(X_0, B_0)$  of  $(\mathbf{q}^0(\mathbf{b}^0), \mathbf{b}^0)$ . Since  $\Phi_u(\mathbf{q}, \mathbf{b})$  is nonsingular in a neighborhood of  $(\mathbf{q}^0(\mathbf{b}), \mathbf{b})$  and  $\mathbf{u} = \mathbf{0}$  is a solution of Eq. (5.11.15) with  $\mathbf{v} = \mathbf{0}$ , the *implicit function theorem* of Section 2.2.5 guarantees that Eq. (5.11.15) has a unique solution,

$$\mathbf{u} = \mathbf{h}(\mathbf{v}, \mathbf{b}) \quad (5.11.18)$$

that is continuously differentiable with respect to  $\mathbf{v}$  and  $\mathbf{b}$  in a neighborhood  $(V_0, B_0)$  of  $(\mathbf{v}^0, \mathbf{b}^0)$ . Thus, Eq. (5.11.13) is

$$\mathbf{q}(\mathbf{v}, \mathbf{b}) = \mathbf{q}^0(\mathbf{b}) + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{b}) \quad (5.11.19)$$

which satisfies Eq. (5.11.1) for all  $\mathbf{v}$  and  $\mathbf{b}$  in a neighborhood  $(V_0, B_0)$  of  $(\mathbf{v}^0, \mathbf{b}^0)$ . As in Section 5.2,  $\mathbf{B}(\mathbf{q}, \mathbf{b})$  and  $\mathbf{h}(\mathbf{q}, \mathbf{b})$  can be efficiently evaluated numerically, for each  $\mathbf{b} \in B_0$ , but the magnitude of  $\mathbf{v}$  and condition number of  $\mathbf{B}(\mathbf{q}, \mathbf{b})$  must be monitored to assure the parameterization of Eq. (5.11.9) remains valid.

Differentiating Eq. (5.11.19) with respect to  $\mathbf{v}$ ,

$$\mathbf{q}_v = \mathbf{V} - \mathbf{U}\mathbf{h}_v(\mathbf{v}, \mathbf{b}) \quad (5.11.20)$$

To evaluate  $\mathbf{h}_v(\mathbf{v}, \mathbf{b})$ , note that Eq. (5.11.15) with  $\mathbf{b}$  constant and  $\mathbf{u} = \mathbf{h}(\mathbf{v}, \mathbf{b})$  is an identity in  $\mathbf{v} \in V_0$ . Its derivative with respect to  $\mathbf{v}$ , suppressing arguments, is  $\Phi_q(\mathbf{V} - \mathbf{U}\mathbf{h}_v) = \mathbf{0}$ . Using Eq. (5.11.17),

$$\mathbf{h}_v(\mathbf{v}, \mathbf{b}) = \mathbf{B}(\mathbf{q}, \mathbf{b})\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{V} \quad (5.11.21)$$

which is a continuously differentiable function of  $\mathbf{q}$  and  $\mathbf{b}$  and Eq. (5.11.20) reduces to

$$\dot{\mathbf{q}}_v = \mathbf{V} - \mathbf{U}\mathbf{B}(\mathbf{q}, \mathbf{b})\Phi_q(\mathbf{q}, \mathbf{b}) = \mathbf{D}(\mathbf{q}, \mathbf{b}) \quad (5.11.22)$$

where

$$\mathbf{D}(\mathbf{q}, \mathbf{b}) \equiv (\mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{q}, \mathbf{b})\Phi_q(\mathbf{q}, \mathbf{b}))\mathbf{V} \quad (5.11.23)$$

Thus,

$$\dot{\mathbf{q}} = \mathbf{q}_v \dot{\mathbf{v}} = \mathbf{D}(\mathbf{q}, \mathbf{b})\dot{\mathbf{v}} \quad (5.11.24)$$

At  $t^0$ ,  $\dot{\mathbf{q}}(t^0, \mathbf{b}) = (\mathbf{V} - \mathbf{U}\mathbf{B}(\mathbf{q}^0, \mathbf{b})\Phi_q(\mathbf{q}^0, \mathbf{b})\mathbf{V})\dot{\mathbf{v}}^0$ . Multiplying both sides of this equation on the left by  $\mathbf{V}^T$  and using the facts that  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T\mathbf{U} = \mathbf{0}$ ,

$$\dot{\mathbf{v}}^0(\mathbf{b}) = \mathbf{V}^T\dot{\mathbf{q}}(t^0, \mathbf{b}) = \mathbf{V}^T\dot{\mathbf{q}}^0(\mathbf{b}) \quad (5.11.25)$$

An important property of  $\mathbf{D}(\mathbf{q}, \mathbf{b})$  is obtained using Eq. (5.11.17), suppressing arguments,

$$\Phi_q \mathbf{D} = (\Phi_q - \Phi_q \mathbf{U}\mathbf{B}\Phi_q) \mathbf{V} = (\Phi_q - \Phi_q) \mathbf{V} = \mathbf{0} \quad (5.11.26)$$

This shows that  $\mathbf{D}(\mathbf{q}, \mathbf{b})$  is in the null space of the constraint Jacobian, for all  $\mathbf{b} \in B_0$  and  $\mathbf{q}$  in a neighborhood of  $\mathbf{q}^0(\mathbf{b}^0)$ . Multiplying Eq. (5.11.24) on the left by the Jacobian  $\Phi_q$ ,

$\Phi_q \dot{\mathbf{q}} = \Phi_q \mathbf{D} \dot{\mathbf{v}} = \mathbf{0}$ , so  $\dot{\mathbf{q}}$  of Eq. (5.11.24) satisfies the velocity constraint equation of Eq. (5.11.2), for all  $\mathbf{v} \in V_0$ ,  $\mathbf{b} \in B_0$ , and  $\dot{\mathbf{v}}$ .

Differentiating Eq. (5.11.24) with respect to  $\dot{\mathbf{v}}$  and  $\mathbf{v}$ ,

$$\dot{\mathbf{q}}_{\dot{\mathbf{v}}} = \mathbf{D}(\mathbf{q}, \mathbf{b}) \quad (5.11.27)$$

$$\dot{\mathbf{q}}_v = (\mathbf{D}(\mathbf{q}, \mathbf{b})\dot{\mathbf{v}})_q \mathbf{q}_v = (\mathbf{D}(\mathbf{q}, \mathbf{b})\dot{\mathbf{v}})_q \mathbf{D}(\mathbf{q}, \mathbf{b}) \quad (5.11.28)$$

From Eq. (5.11.23), for a constant vector  $\mathbf{a}$ ,

$$(\mathbf{D}(\mathbf{q}, \mathbf{b})\ddot{\mathbf{a}})_q = -\mathbf{U}(\mathbf{B}(\mathbf{q}, \mathbf{b})\ddot{\Phi}_q \mathbf{V} \ddot{\mathbf{a}})_q - \mathbf{U}\mathbf{B}(\Phi_q(\mathbf{q}, \mathbf{b})\ddot{\mathbf{V}} \ddot{\mathbf{a}})_q \quad (5.11.29)$$

To evaluate  $(\mathbf{B}(\mathbf{q}, \mathbf{b})\ddot{\Phi}_q \mathbf{V} \ddot{\mathbf{a}})_q$ , Eq. (5.11.17) is written in the form  $\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{U}\mathbf{B}(\mathbf{q}, \mathbf{b})\ddot{\mathbf{c}} = \ddot{\mathbf{c}}$ , which is an identity in  $\mathbf{q}$  for  $\mathbf{b} \in B_0$ . Differentiating with respect to  $\mathbf{q}$ ,

$(\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{U}\mathbf{B}\ddot{\mathbf{c}})_q + \Phi_q \mathbf{U}(\mathbf{B}(\mathbf{q}, \mathbf{b})\ddot{\mathbf{c}})_q = \mathbf{0}$ . Using Eq. (5.11.17), this is

$$\left( \mathbf{B}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{c}} \right)_{\mathbf{q}} = -\mathbf{B} \left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{B}} \ddot{\mathbf{c}} \right)_{\mathbf{q}} \quad (5.11.30)$$

Substituting this result, with  $\mathbf{c} = \Phi_{\mathbf{q}} \mathbf{V} \mathbf{a}$ , into Eq. (5.11.29),

$$\begin{aligned} \left( \mathbf{D}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{a}} \right)_{\mathbf{q}} &= \mathbf{U} \mathbf{B} \left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{B}} \ddot{\mathbf{B}} \Phi_{\mathbf{q}} \ddot{\mathbf{V}} \ddot{\mathbf{a}} \right)_{\mathbf{q}} - \mathbf{U} \mathbf{B} \left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{V}} \ddot{\mathbf{a}} \right)_{\mathbf{q}} \\ &= -\mathbf{U} \mathbf{B} \left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) (\mathbf{I} - \ddot{\mathbf{B}} \ddot{\mathbf{B}} \Phi_{\mathbf{q}}) \ddot{\mathbf{V}} \ddot{\mathbf{a}} \right)_{\mathbf{q}} = -\mathbf{U} \mathbf{B} \left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{D}} \ddot{\mathbf{a}} \right)_{\mathbf{q}} \end{aligned} \quad (5.11.31)$$

Using Eqs. (5.11.24) and (5.11.31), Eq. (5.11.28) reduces to

$$\dot{\mathbf{q}}_v = -\mathbf{U} \mathbf{B} \left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{D}} \ddot{\mathbf{V}} \right)_{\mathbf{q}} \mathbf{D} = -\mathbf{U} \mathbf{B} \left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{q}} \right)_{\mathbf{q}} \mathbf{D} \quad (5.11.32)$$

Using Eqs. (5.11.27) and (5.11.32) with  $\ddot{\mathbf{q}} = \dot{\mathbf{q}}_v \ddot{\mathbf{v}} + \dot{\mathbf{q}}_v \dot{\mathbf{v}}$ ,

$$\ddot{\mathbf{q}} = \mathbf{D}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \gamma(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}) \quad (5.11.33)$$

where  $\left( \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{q}} \right)_{\mathbf{q}} \dot{\mathbf{q}} \equiv \gamma(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b})$  of Eq. (5.11.3) has been used.

Multiplying both sides of Eq. (5.11.33) on the left by  $\Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b})$ , suppressing arguments and using Eqs. (5.11.17) and (5.11.33),  $\Phi_{\mathbf{q}} \ddot{\mathbf{q}} = \Phi_{\mathbf{q}} \mathbf{D} \ddot{\mathbf{v}} - \Phi_{\mathbf{q}} \mathbf{U} \mathbf{B} \gamma = -\gamma$ . Thus, the acceleration constraint of Eq. (5.11.3) is satisfied for all  $\mathbf{v} \in V_0$ ,  $\mathbf{b} \in B_0$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ . In summary, all three forms of constraint of Eqs. (5.11.1) through (5.11.3) are satisfied by  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  of Eqs. (5.11.19), (5.11.24), and (5.11.33), for all  $\mathbf{v} \in V_0$ ,  $\mathbf{b} \in B_0$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ .

In differential form, Eq. (5.11.24) is  $\delta \mathbf{q} = \mathbf{D}(\mathbf{q}, \mathbf{b}) \delta \mathbf{v}$  and, since  $\Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \mathbf{D}(\mathbf{q}, \mathbf{b}) = \mathbf{0}$ ,

$$\Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \delta \mathbf{q} = \Phi_{\mathbf{q}}(\mathbf{q}, \mathbf{b}) \mathbf{D}(\mathbf{q}, \mathbf{b}) \delta \mathbf{v} = \mathbf{0} \quad (5.11.34)$$

for arbitrary  $\delta \mathbf{v}$ . Substituting this result and  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  of Eqs. (5.11.19), (5.11.24), and (5.11.33) into Eq. (5.11.4) yields

$$\delta \mathbf{v}^T \left\{ \begin{array}{l} \mathbf{D}^T(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{M}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{D}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \ddot{\mathbf{v}} \\ -\mathbf{D}^T(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{M}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{U} \mathbf{B}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \gamma(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}) \\ -\mathbf{D}^T(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{Q}^A(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}, t) - \mathbf{D}^T(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{S}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}) \end{array} \right\} = 0$$

Since  $\delta \mathbf{v}$  is arbitrary,

$$\begin{aligned} &\mathbf{D}^T(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{M}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{D}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \ddot{\mathbf{v}} \\ &- \mathbf{D}^T(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{M}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{U} \mathbf{B}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \gamma(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}) \\ &- \mathbf{D}^T(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \left( \mathbf{Q}^A(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}, t) + \mathbf{S}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}) \right) = \mathbf{0} \end{aligned} \quad (5.11.35)$$

which is a second order ODE in  $\mathbf{v}$  that depends on  $\mathbf{b}$ .

Evaluating Eq. (5.11.13) at  $t^0$  and multiplying by  $\mathbf{V}^T$ ,  $\mathbf{0} = \mathbf{V}^T(\mathbf{q}^0 - \mathbf{q}^0) = \mathbf{v}^0$ . This and Eq. (5.11.25) yield the *design dependent initial conditions*

$$\begin{aligned} \mathbf{v}(t^0) &= \mathbf{0} \\ \dot{\mathbf{v}}(t^0) &= \mathbf{V}^T \dot{\mathbf{q}}^0(\mathbf{b}) \end{aligned} \quad (5.11.36)$$

### 5.11.5 Well Posed ODE and Index 0 DAE of Motion

By the hypotheses of Section 5.11.2, since  $\mathbf{D}(\mathbf{q}, \mathbf{b})$  is in the null space of  $\Phi_q(\mathbf{q}, \mathbf{b})$ , the matrix  $\mathbf{D}^T(\mathbf{q}, \mathbf{b})\mathbf{M}(\mathbf{q}, \mathbf{b})\mathbf{D}(\mathbf{q}, \mathbf{b})$  is positive definite in a neighborhood of  $(\mathbf{q}^0, \mathbf{b}^0)$ . Since the coefficient matrix of  $\ddot{\mathbf{v}}$  in Eq. (5.11.35) is nonsingular, the *ODE existence theorem* of Section 4.7.3 shows that the initial-value problem of Eqs. (5.11.35) and (5.11.36) has a unique solution  $\mathbf{v}(t, \mathbf{b})$  in a neighborhood  $(V_0, B_0)$  of  $\mathbf{v} = \mathbf{0}$  and  $\mathbf{b}$  that is continuously differentiable with respect to  $t$  and  $\mathbf{b}$ . Further, as shown in Section 5.2, the solution can be continued on the constraint manifold. These results are summarized as follows:

**Theorem 5.11.1 Well Posed Holonomic ODE:** If

- (1) all functions that appear in Eqs. (5.11.1) through (5.11.4) are  $k \geq 1$  times continuously differentiable on a bounded domain  $D$  of  $\mathbf{q}\text{-}\dot{\mathbf{q}}\text{-}t\text{-}\mathbf{b}$  space,
- (2) the constraint Jacobian  $\Phi_q(\mathbf{q}, \mathbf{b})$  has full rank in  $D$ ,
- (3) the mass matrix  $\mathbf{M}(\mathbf{q}, \mathbf{b})$  is positive definite on the null space of  $\Phi_q((\mathbf{q}, \mathbf{b}))$  in  $D$ , and
- (4)  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  satisfy Eqs. (5.11.1) and (5.11.2),

then Eqs. (5.11.35) and (5.11.36) have a unique solution  $\mathbf{v}(t, \mathbf{b})$  that is  $k$  times continuously differentiable with respect to  $t$  and  $\mathbf{b}$  in a neighborhood of  $(t_0, \mathbf{b})$ .

Further, Eqs. (5.11.19), (5.11.24), and (5.11.33) define a unique solution, in a neighborhood of  $(\mathbf{q}^0, \dot{\mathbf{q}}^0, t^0, \mathbf{b}^0)$ , of the variational equations of motion of Section 5.11.2 that is  $k$  times continuously differentiable with respect to  $t$  and  $\mathbf{b}$ .

To show that Eqs. (5.11.19), (5.11.24), and (5.11.33) define a solution of the variational equation of motion, note first that  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  satisfy Eqs. (5.11.19), (5.11.24), and (5.11.33), for arbitrary  $\delta \mathbf{v}$ , equivalently Eq. (5.11.4) for all  $\delta \mathbf{q}$  such that  $\Phi_q \delta \mathbf{q} = \mathbf{0}$ . Since  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  of Eqs. (5.11.19), (5.11.24), and (5.11.33) satisfy the constraints of Eqs. (5.11.1) through (5.11.3), for any  $\mathbf{v} \in V_0$ ,  $\mathbf{b} \in B_0$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ , they satisfy all three forms of constraint. Thus,  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  is a twice continuously differentiable solution of the variational equations of motion of Section 5.11.2. This completes the proof.

It is important to note that this result and those that follow are *local in nature*. That is, they assure existence, uniqueness, and continuous dependence on data only in a neighborhood of each point in the domain  $D$  of regularity of functions that define the system equations of motion. The theorem does not guarantee existence, uniqueness, and continuous dependence on data for all possible time. While local solutions can be extended beyond the neighborhood assured, the continuations may become infinitesimal and the solution may not exist for all time. No comprehensive global theory exists for these nonlinear problems.

Algebraic manipulations of Eqs. (5.5.6) through (5.5.11) yield the *Index 0 DAE*

$$\begin{aligned}
& \mathbf{M}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{D}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \ddot{\mathbf{v}} + \Phi_q(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}^T) \boldsymbol{\lambda} \\
& - \mathbf{M}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \mathbf{U} \mathbf{B}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) \boldsymbol{\gamma}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}) \\
& - \mathbf{S}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}) - \mathbf{Q}^A(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b}, t) = \mathbf{0}
\end{aligned} \tag{5.11.37}$$

for each  $\mathbf{b} \in \mathcal{B}_0$ . Since the algebraic manipulations of Eqs (5.5.6) through (5.5.11) are reversible, Eq. (5.11.37) with initial conditions of Eq. (5.11.36) has the same solution as the ODE initial-value problem, hence the same regularity properties. This yields the following result:

**Theorem 5.11.2 Well Posed Index 0 Initial-Value Problem:** Under the hypotheses of Theorem 5.11.1, the initial-value problem of Eqs. (5.11.37) and (5.11.36) is well posed.

### 5.11.6 Well Posed Variational Equations of Motion

A little more work is required to show that the variational equations of motion are well posed, as follows:

**Theorem 5.11.3 Well Posed Holonomic Variational Equations:** Under the hypotheses of Theorem 5.11.1, the variational equations of motion of Eqs. (5.11.1) through (5.11.5), with initial conditions that satisfy Eqs. (5.11.7) and (5.11.9), are well posed.

To prove this result, Eqs. (5.11.1) through (5.11.5) and associated initial conditions must be shown to be equivalent to the ODE initial value problem of Eqs. (5.11.35) and (5.11.36) and the definitions of Eqs. (5.11.19), (5.11.24), and (5.11.33). The derivation of Section 5.11.4 shows that a solution of Eqs. (5.11.1) through (5.11.5) is a solution of Eqs. (5.11.35) and (5.11.36). It remains to show that the unique solution of Eqs. (5.11.35) and (5.11.36), with definitions of Eqs. (5.11.19), (5.11.24), and (5.11.33), is a unique solution of Eqs. (5.11.1) through (5.11.5). With a unique solution  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  of Eqs. (5.11.35) and (5.11.36), Eqs. (5.11.19), (5.11.24), and (5.11.33) define unique  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  that satisfy Eqs. (5.11.1) through (5.11.3). Equation (5.11.35) implies the variational equation holds, for arbitrary  $\delta \mathbf{v}$ , written in the form  $(\mathbf{D}\delta \mathbf{v})^T [\mathbf{M}(\mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\boldsymbol{\gamma}) - \mathbf{Q}^A - \mathbf{S}] = 0$ . Using Eqs. (5.11.33) and (5.11.34), this is  $\delta \mathbf{q}^T [\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}^A - \mathbf{S}] = 0$ , for all  $\delta \mathbf{q}$  such that  $\Phi_q \delta \mathbf{q} = \mathbf{0}$ , which is Eq. (5.11.4).

It remains only to show that the solution  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  of the variational equations is unique. For purposes of obtaining a contradiction, assume there is a solution  $\bar{\mathbf{q}} \neq \mathbf{q}$ . Then,  $\bar{\mathbf{q}}$  is defined by a solution  $\bar{\mathbf{v}}$  of Eqs. (5.11.35) and (5.11.36). Evaluating Eq. (5.11.19) for each solution and subtracting,  $\mathbf{q} - \bar{\mathbf{q}} = \mathbf{V}(\mathbf{v} - \bar{\mathbf{v}}) - \mathbf{U}(\mathbf{h}(\mathbf{v}) - \mathbf{h}(\bar{\mathbf{v}}))$ . Since the solution of Eqs. (5.11.35) and (5.11.36) is unique,  $\mathbf{v} - \bar{\mathbf{v}} = \mathbf{0}$ . This implies  $\mathbf{q} = \bar{\mathbf{q}}$ , which contradicts the assumption. Thus, the solution of the variational equations of Eqs. (5.11.1) through (5.11.4) is unique, completing the proof of the theorem.

### 5.11.7 Well Posed Full DAE of Motion

Applying the *Lagrange multiplier theorem* of Section 2.2.2 to the variational equations of motion of Eqs. (5.11.4) and (5.11.5), there exists a unique vector  $\boldsymbol{\lambda} \in \mathbb{R}^{n_{hc}}$  such that

$$\delta \mathbf{q}^T (\mathbf{M}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{q}} - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b})) + \delta \mathbf{q}^T \Phi_q^T(\mathbf{q}, \mathbf{b}) \lambda = 0$$

for arbitrary  $\delta \mathbf{q}$ . This yields the Lagrange multiplier form of the design dependent equations of motion,

$$\mathbf{M}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, \mathbf{b}) \lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}) = \mathbf{0} \quad (5.11.38)$$

which, with the constraint equations of Eqs. (5.11.1) through (5.11.3); i.e.,

$$\begin{aligned} \Phi(\mathbf{q}, \mathbf{b}) &= \mathbf{0} \\ \Phi_q(\mathbf{q}, \mathbf{b}) \dot{\mathbf{q}} &= \mathbf{0} \\ \Phi_q(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}) &= \mathbf{0} \end{aligned} \quad (5.11.39)$$

and initial conditions that satisfy the first two of Eqs. (5.11.39),

$$\begin{aligned} \mathbf{q}(t^0) &= \mathbf{q}^0(\mathbf{b}) \\ \dot{\mathbf{q}}(t^0) &= \dot{\mathbf{q}}^0(\mathbf{b}) \end{aligned} \quad (5.11.40)$$

comprise the *Full DAE of motion*.

The fact that Eqs. (5.11.38) through (5.11.40) were obtained so easily belies the fact that they are extraordinarily difficult to solve. While no attempt is made here to solve the Full DAE, it is shown to be well posed, as follows:

**Theorem 5.11.4 Well Posed Holonomic Full DAE:** If

- (1) all functions that appear in Eqs. (5.11.38) and (5.11.39) are  $k \geq 1$  times continuously differentiable on a bounded domain  $D$  of  $\mathbf{q}$ - $\dot{\mathbf{q}}$ - $t$ - $\mathbf{b}$  space;
- (2) the constraint Jacobian  $\Phi_q(\mathbf{q}, \mathbf{b})$  has full rank in  $D$ ;
- (3) the mass matrix  $\mathbf{M}(\mathbf{q}, \mathbf{b})$  is positive definite on the null space of  $\Phi_q(\mathbf{q}, \mathbf{b})$  in  $D$ ; and
- (4)  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  satisfy the first two of Eqs. (5.11.39);

then Eqs. (5.11.38) through (5.11.40) have a unique solution  $\mathbf{q}(t, \mathbf{b})$  in a neighborhood of  $(\mathbf{q}^0, \dot{\mathbf{q}}^0, t^0, \mathbf{b})$  that is  $k$  times continuously differentiable with respect to  $t$  and  $\mathbf{b}$ ; i.e., the *Full DAE are well posed*.

This theorem is proved by showing that the DAE initial-value problem of Eqs. (5.11.38) through (5.11.40) is equivalent to the ODE initial value problem of Eqs. (5.11.35) and (5.11.36) in  $\mathbf{v}$  and Eqs. (5.11.19), (5.11.24), and (5.11.33) that define  $\mathbf{q}$  and its derivatives, which is well posed. The proof is in two parts, as follows:

*Part 1: Solutions of Eqs. (5.11.38) Through (5.11.40) Define Solutions of Eqs. (5.11.35) and (5.11.36)*

Let  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  satisfy Eqs. (5.11.38) through (5.11.40). Matrices  $\mathbf{U}(\mathbf{b})$  and  $\mathbf{V}(\mathbf{b})$  of Eqs. (5.11.10) and (5.11.11) are defined using the given initial value  $\mathbf{q}^0 = \mathbf{q}(t^0)$  and  $\mathbf{B}(\mathbf{q}, \mathbf{b})$  of

Eq. (5.11.17) and  $\mathbf{D}(\mathbf{q}, \mathbf{b})$  of Eq. (5.11.23) are defined using the given  $\mathbf{q}$ . Since the columns of  $\mathbf{U}$  and  $\mathbf{V}$  span  $\mathbb{R}^{ngc}$ , there exist unique  $\mathbf{v}$  and  $\mathbf{u}$  such that the given function  $\mathbf{q}$  is written as  $\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}$ . More specifically, multiplying on the left by  $\mathbf{V}^T$ ,  $\mathbf{v} = \mathbf{V}^T(\mathbf{q} - \mathbf{q}^0)$ . Since the given  $\mathbf{q}$  satisfies the first of Eqs. (5.11.39),  $\Phi_q((\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}), \mathbf{b}) = \mathbf{0}$ . This is Eq. (5.11.15), which has a unique solution  $\mathbf{u} = \mathbf{h}(\mathbf{v}, \mathbf{b})$ . Thus,  $\mathbf{q} = \mathbf{q}(\mathbf{v}, \mathbf{b}) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{b})$ . Similarly, there exist unique  $\mathbf{w}$  and  $\mathbf{x}$  such that the given function  $\dot{\mathbf{q}}$  is written as  $\dot{\mathbf{q}} = \mathbf{V}\mathbf{w} - \mathbf{U}\mathbf{x}$ . Multiplying on the left by  $\mathbf{V}^T(\mathbf{b})$ ,  $\mathbf{w} = \mathbf{V}^T\dot{\mathbf{q}} = \dot{\mathbf{v}}$ . Since the given  $\dot{\mathbf{q}}$  satisfies the second of Eqs. (5.11.39),  $\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{V}\dot{\mathbf{v}} - \Phi_q(\mathbf{q}, \mathbf{b})\mathbf{U}\mathbf{x} = \mathbf{0}$ . Using Eq. (5.11.17),  $\mathbf{x} = \mathbf{B}(\mathbf{q}, \mathbf{b})\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{V}\dot{\mathbf{v}}$  and  $\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}) = \mathbf{D}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b})\dot{\mathbf{v}}$ . Representing the given  $\ddot{\mathbf{q}}$  as  $\ddot{\mathbf{q}} = \mathbf{V}\mathbf{y} - \mathbf{U}\mathbf{z}$ ,  $\mathbf{y} = \mathbf{V}^T\ddot{\mathbf{q}} = \ddot{\mathbf{v}}$  and the third of Eqs. (5.11.39) yields  $\mathbf{z} = \mathbf{B}(\mathbf{q}, \mathbf{b})\Phi_q(\mathbf{q}, \mathbf{b})\mathbf{V}\dot{\mathbf{v}} + \mathbf{B}(\mathbf{q}, \mathbf{b})\gamma(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b})$ . Thus,  $\ddot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \mathbf{b}) = \mathbf{D}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b}) - \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, \mathbf{b}), \mathbf{b})\gamma(\mathbf{q}(\mathbf{v}, \mathbf{b}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b}), \mathbf{b})$ .

With  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  written as functions of  $\mathbf{v}$  and its derivatives that satisfy Eqs. (5.11.36), (5.11.19), (5.11.24), and (5.11.33), it remains only to show that Eq. (5.11.35) is satisfied. Substituting  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  as functions of  $\mathbf{v}$  and its derivatives from Eqs. (5.11.19), (5.11.24), and (5.11.33) into Eq. (5.11.38) and suppressing arguments,

$$\mathbf{M}\mathbf{D}\ddot{\mathbf{v}} = \mathbf{M}\mathbf{U}\mathbf{B} + \Phi_q^T\lambda - \mathbf{Q}^A - \mathbf{S} = \mathbf{0}$$

Multiplying on the left by  $\mathbf{D}^T$  and noting that  $\mathbf{D}^T\Phi_q^T = (\Phi_q\mathbf{D})^T = \mathbf{0}$  yields

$$\mathbf{D}^T\mathbf{M}\mathbf{D}\ddot{\mathbf{v}} - \mathbf{D}^T\mathbf{M}\mathbf{U}\mathbf{B} - \mathbf{D}^T\mathbf{Q}^A - \mathbf{D}^T\mathbf{S} = \mathbf{0}$$

which is Eq. (5.11.35). Thus, a solution of the DAE of Eqs. (5.11.38) through (5.11.40) defines a solution of Eqs. (5.11.35) and (5.11.36).

*Part 2: A Unique Solution of Eqs. (5.11.35) and (5.11.36) Defines a Unique Solution of Eqs. (5.11.38) through (5.11.40)*

The functions  $\mathbf{q}(\mathbf{v}, \mathbf{b})$ ,  $\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{b})$ , and  $\ddot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \mathbf{b})$  of Eqs. (5.11.19), (5.11.24), and (5.11.33) are uniquely defined by  $\mathbf{v}$  and its derivatives. First note that the right side of Eq. (5.11.19) satisfies Eqs. (5.11.15), so  $\mathbf{q}$  defined by Eq. (5.11.19) satisfies the first of Eqs. (5.11.39). Multiplying  $\dot{\mathbf{q}}$  that is defined by Eq. (5.11.24) by  $\Phi_q(\mathbf{q}, \mathbf{b})$ , suppressing arguments for notational clarity,

$$\Phi_q\dot{\mathbf{q}} = \Phi_q\mathbf{D}\dot{\mathbf{v}} = \mathbf{0}$$

so  $\dot{\mathbf{q}}$  satisfies the second of Eqs. (5.11.39). Similarly, multiplying  $\ddot{\mathbf{q}}$  of Eq. (5.11.33) by  $\Phi_q$ ,

$$\Phi_q\ddot{\mathbf{q}} = \Phi_q(\mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\gamma) = -\Phi_q\mathbf{U}\mathbf{B}\gamma = -\gamma$$

so  $\ddot{\mathbf{q}}$  satisfies the third of Eqs. (5.11.39).

Expanding  $\mathbf{D}$  of Eq. (5.11.23), Eq. (5.11.35) becomes

$$\begin{aligned}
& \mathbf{V}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} - \mathbf{V}^T \Phi_q^T \mathbf{B}^T \mathbf{U}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} + \mathbf{V}^T \Phi_q^T \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) - \mathbf{V}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) \\
& \mathbf{V}^T \mathbf{M} = \mathbf{D} \ddot{\mathbf{v}} + \mathbf{V}^T \Phi_q^T \{ \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) - \mathbf{B}^T \mathbf{U}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} \} - \mathbf{V}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0}
\end{aligned} \tag{5.11.41}$$

Defining

$$\lambda \equiv \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) - \mathbf{B}^T \mathbf{U}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} \tag{5.11.42}$$

Eq. (5.11.41) reduces to

$$\mathbf{V}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} + \mathbf{V}^T \Phi_q^T \lambda - \mathbf{V}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0} \tag{5.11.43}$$

Multiplying both sides of Eq. (5.11.42) on the left by  $\mathbf{U}^T \Phi_q^T$  yields

$$\mathbf{U}^T \Phi_q^T \lambda = \mathbf{U}^T \Phi_q^T \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) - \mathbf{U}^T \Phi_q^T \mathbf{B}^T \mathbf{U}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} \tag{5.11.44}$$

Taking the transpose of Eq. (5.11.17) yields  $\mathbf{B}^T = (\mathbf{U}^T \Phi_q^T)^{-1}$ . Thus, Eq. (5.11.44) reduces to

$$\mathbf{U}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} + \mathbf{U}^T \Phi_q^T \lambda - \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0} \tag{5.11.45}$$

Combining Eqs. (5.11.43) and (5.11.45),

$$[\mathbf{V} \quad \mathbf{U}]^T (\mathbf{M} \mathbf{D} \ddot{\mathbf{v}} + \Phi_q^T \lambda - (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A)) = \mathbf{0} \tag{5.11.46}$$

Since the columns of  $\mathbf{V}$  and  $\mathbf{U}$  span  $\mathbf{R}^n$ , the matrix  $[\mathbf{V} \quad \mathbf{U}]^T$  is nonsingular. Thus,

$\mathbf{M} \mathbf{D} \ddot{\mathbf{v}} + \Phi_q^T \lambda - \mathbf{M} \mathbf{U} \mathbf{B} \gamma - \mathbf{S} - \mathbf{Q}^A = \mathbf{0}$  or,  $\mathbf{M}(\mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \gamma) + \Phi_q^T \lambda - \mathbf{S} - \mathbf{Q}^A = \mathbf{0}$ . From Eq. (5.11.33),  $\ddot{\mathbf{q}} = \mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B}$ , yielding

$$\mathbf{M}(\mathbf{q}, \mathbf{b}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, \mathbf{b}) \lambda - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{b}, t) = \mathbf{0} \tag{5.11.47}$$

Thus,  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$ , and  $\lambda$  is a solution of the DAE of Eqs. (5.11.38) through (5.11.40).

To see that this solution is unique, first recall that the Lagrange multiplier is unique. Next, assume there is a solution  $(\bar{\mathbf{q}}, \bar{\dot{\mathbf{q}}}, \bar{\ddot{\mathbf{q}}}, \lambda)$  of the DAE such that  $(\bar{\mathbf{q}}, \bar{\dot{\mathbf{q}}}, \bar{\ddot{\mathbf{q}}}, \lambda) \neq (\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \lambda)$ . The result of Part 1 of the proof shows that there is an associated solution  $\bar{\mathbf{v}}(t)$  of the ODE of Eqs. (5.11.35) and (5.11.36). Since  $\mathbf{v}$  and  $\bar{\mathbf{v}}$  satisfy Eq. (5.11.19),  $\bar{\mathbf{q}} - \mathbf{q} = (\mathbf{q}^0 + \mathbf{V} \bar{\mathbf{v}} - \mathbf{U} \mathbf{h}(\bar{\mathbf{v}})) - (\mathbf{q}^0 + \mathbf{V} \mathbf{v} - \mathbf{U} \mathbf{h}(\mathbf{v}))$ . Since the unique solution of Eqs. (5.11.35) and (5.11.36) is  $\mathbf{v} = \bar{\mathbf{v}}$ ,  $\bar{\mathbf{q}} - \mathbf{q} = \mathbf{0}$ . This is a contradiction of the assumption. Thus, the solution of the DAE is unique. This completes the proof of the theorem.

It is important to note that the proof of the theorem requires that all three forms of the constraint equations of Eq. (5.11.39) are satisfied. If only one of the three is enforced in methods that apply ODE integrators to the resulting equations, the results should be regarded with suspicion.

While tangent space generalized coordinates and the resulting ODE have been used in the proof of the foregoing theorems, validity of the theorems is not dependent on this specific ODE formulation.

Existence, uniqueness, and continuous dependence of solutions on problem data of the equations of mechanical system dynamics (well posed problems) are proved for variational, tangent space ODE, Index 0 DAE, and Full DAE formulations of the equations of dynamics. These results put the theory of mechanical system dynamics regarding well posedness on a footing comparable to that of the theory of ODE.

## **Appendix 5.A Holonomic Tangent Space Code**

- Code 5.1 Top-Analytical Tangent Space ODE*
- Code 5.4.1 Planar Double Pendulum (TSODE)*
- Code 5.4.2a Top Tip Fixed-Spin Stabilized (TSODE)*
- Code 5.4.2b Top Tip Fixed-Transient (TSODE)*
- Code 5.4.3 Top Tip on x-y Plane (TSODE)*
- Code 5.4.4 Spatial Double Pendulum (TSODE)*
- Code 5.6.1a Top Tip Fixed-Spin Stabilized (TSInd0)*
- Code 5.6.1b Top Tip Fixed-Transient (TSInd0)*
- Code 5.6.2 Spatial Double Pendulum (TSInd0)*
- Code 5.7 Planar Tangent Space Multibody Simulation*
- Code 5.8 Spatial Tangent Space Multibody Simulation*

## Appendix 5.B Kinematic and Kinetic Derivatives

Derivatives that define constraint Jacobians and operators such as  $\mathbf{P}2(\mathbf{q}, \chi)$  that are used in kinematic velocity and acceleration equations are defined in Sections 3.2 and 3.3. These quantities and kinetic expressions derived in Sections 4.2 and 4.3 are adequate for explicit numerical integration using formulations presented in this chapter. *Implicit numerical integration*, however, requires an additional derivative of both kinematic and kinetic quantities. *Derivatives of kinematic constraints* required for evaluation of  $\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}})$  and  $\mathbf{P}4(\mathbf{q}, \ddot{\mathbf{q}})$  in Sections 5.3 and 5.5 are presented in Section 5.B.1. *Derivatives of kinetic quantities* that are required for evaluation of  $\mathbf{M}2(\mathbf{q}, \mu)$ ,  $\mathbf{S}_q$ , and  $\mathbf{S}_{\dot{q}}$  in Section 5.3 and 5.5 are presented in Section 5.B.2. *Derivatives of generalized forces* are presented in Section 5.B.3.

### 5.B.1 Derivatives of Kinematic Constraints

#### 5.B.1.1 Derivatives of Planar Constraints

For the distance constraint of Eq. (3.2.2), Eqs. (3.2.3) through (3.2.5) define  $\mathbf{P}2^{\text{dist}}(\mathbf{q}_{ij}, \chi_{ij})$  of Eq. (3.1.9). From Eqs. (3.2.4), (5.3.30) and (5.3.31), terms required to evaluate  $\mathbf{P}3^{\text{dist}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij})$  are

$$\begin{aligned} \left( \left( \frac{\text{dist}}{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} &= -\mathbf{s}'^T \mathbf{A}_i^T \left[ \dot{\phi}_i^2 \mathbf{I}_2 \quad 2\dot{\phi}_i \bar{\mathbf{a}}_i + \dot{\phi}_i^2 \mathbf{P} \mathbf{d}_{ij}^T \right] \\ \left( \left( \frac{\text{dist}}{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} &= \dot{\phi}_i^2 \mathbf{s}'^T \mathbf{A}_i^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \right] \\ \left( \left( \frac{\text{dist}}{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} &= \left[ \mathbf{0} \quad \dot{\phi}_i \bar{\mathbf{a}}_j^T \mathbf{A}_i \mathbf{s}'_i \right] \\ \left( \left( \frac{\text{dist}}{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} &= \left[ \mathbf{0} \quad \dot{\phi}_j \bar{\mathbf{a}}_i^T \mathbf{A}_j \mathbf{s}'_j \right] \\ \left( \left( \frac{\text{dist}}{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} &= \dot{\phi}_j^2 \mathbf{s}'^T \mathbf{A}_j^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \right] \\ \left( \left( \frac{\text{dist}}{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} &= -\mathbf{s}'_j^T \mathbf{A}_j^T \left[ \dot{\phi}_j^2 \mathbf{I}_2 \quad 2\dot{\phi}_j \bar{\mathbf{a}}_j - \dot{\phi}_j^2 \mathbf{P} \mathbf{d}_{ij}^T \right] \end{aligned} \tag{5.B.1}$$

where

$$\begin{aligned} \bar{\mathbf{a}}_i &= \dot{\mathbf{r}}_i + \dot{\phi}_i \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \\ \bar{\mathbf{a}}_j &= \dot{\mathbf{r}}_j + \dot{\phi}_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \end{aligned}$$

Thus,

$$\mathbf{P}3^{\text{dist}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \left[ \mathbf{P}3_i^{\text{dist}} \quad \mathbf{P}3_j^{\text{dist}} \right] \tag{5.B.2}$$

where

$$\begin{aligned}\mathbf{P3}_i^{\text{dist}} &= -\mathbf{s}'_i^T \mathbf{A}_i^T \left[ \dot{\phi}_i^2 \mathbf{I}_2 \quad 2\dot{\phi}_i \bar{\mathbf{a}}_i + \dot{\phi}_i^2 \mathbf{P} \mathbf{d}_{ij} \right] + 2 \left[ \mathbf{0} \quad \dot{\phi}_i \bar{\mathbf{a}}_j^T \mathbf{A}_i \mathbf{s}'_i \right] + \dot{\phi}_i^2 \mathbf{s}'_j^T \mathbf{A}_j^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \right] \\ \mathbf{P3}_j^{\text{dist}} &= -\mathbf{s}'_j^T \mathbf{A}_j^T \left[ \dot{\phi}_j^2 \mathbf{I}_2 \quad 2\dot{\phi}_j \bar{\mathbf{a}}_j - \dot{\phi}_j^2 \mathbf{P} \mathbf{d}_{ij} \right] + 2 \left[ \mathbf{0} \quad \dot{\phi}_j \bar{\mathbf{a}}_i^T \mathbf{A}_j \mathbf{s}'_j \right] + \dot{\phi}_j^2 \mathbf{s}'_i^T \mathbf{A}_i^T \left[ \mathbf{I}_2 \quad \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \right]\end{aligned}$$

Terms in  $\mathbf{P4}^{\text{dist}}(\mathbf{q}_{ij}, \eta)$  of Eq. (5.3.35), using the subJacobians of Eq. (3.2.4) and the fact that  $\eta^{\text{dist}}$  is a scalar, are

$$\begin{aligned}\left( \Phi_{\mathbf{q}_i}^{\text{dist}T} \ddot{\eta}^{\text{dist}} \right)_{\mathbf{q}_i} &= \eta^{\text{dist}} \begin{bmatrix} \mathbf{I} & \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \\ -\mathbf{s}'_i^T \mathbf{A}_i^T \mathbf{P} & \mathbf{s}'_i^T \mathbf{s}'_i + \mathbf{s}'_i^T \mathbf{A}_i^T \mathbf{d}_{ij} \end{bmatrix} \\ \left( \Phi_{\mathbf{q}_i}^{\text{dist}T} \ddot{\eta}^{\text{dist}} \right)_{\mathbf{q}_j} &= \eta^{\text{dist}} \begin{bmatrix} -\mathbf{I} & -\mathbf{P} \mathbf{A}_j \mathbf{s}'_j \\ \mathbf{s}'_i^T \mathbf{A}_i^T \mathbf{P} & -\mathbf{s}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{s}'_j \end{bmatrix} \\ \left( \Phi_{\mathbf{q}_j}^{\text{dist}T} \ddot{\eta}^{\text{dist}} \right)_{\mathbf{q}_j} &= \eta^{\text{dist}} \begin{bmatrix} \mathbf{I} & \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \\ -\mathbf{s}'_j^T \mathbf{A}_j^T \mathbf{P} & \mathbf{s}'_j^T \mathbf{s}'_j - \mathbf{s}'_j^T \mathbf{A}_j^T \mathbf{d}_{ij} \end{bmatrix}\end{aligned}\tag{5.B.3}$$

For the revolute constraint of Eq. (3.2.7), Eqs. (3.2.8) through (3.2.10) define  $\mathbf{P2}^{\text{rev}}(\mathbf{q}_{ij}, \chi_{ij})$  of Eq. (3.1.9). From Eqs. (3.2.9), (5.3.30), and (5.3.31), terms required to evaluate  $\mathbf{P3}^{\text{rev}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_j)$  are

$$\begin{aligned}\left( \left( \Phi_{\mathbf{q}_i}^{\text{rev}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} &= \left[ \mathbf{0} \quad \dot{\phi}_i^2 \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \right] \\ \left( \left( \Phi_{\mathbf{q}_i}^{\text{rev}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} &= \mathbf{0} \\ \left( \left( \Phi_{\mathbf{q}_i}^{\text{rev}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} &= \mathbf{0} \\ \left( \left( \Phi_{\mathbf{q}_i}^{\text{rev}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} &= \mathbf{0} \\ \left( \left( \Phi_{\mathbf{q}_j}^{\text{rev}} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} &= \mathbf{0} \\ \left( \left( \Phi_{\mathbf{q}_j}^{\text{rev}} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} &= \left[ \mathbf{0} \quad -\dot{\phi}_j^2 \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \right]\end{aligned}\tag{5.B.4}$$

Thus,

$$\mathbf{P3}^{\text{rev}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \left[ \left[ \mathbf{0} \quad \dot{\phi}_i^2 \mathbf{P} \mathbf{A}_i \mathbf{s}'_i \right] \quad \left[ \mathbf{0} \quad -\dot{\phi}_j^2 \mathbf{P} \mathbf{A}_j \mathbf{s}'_j \right] \right]\tag{5.B.5}$$

Terms in  $\mathbf{P4}^{\text{rev}}(\mathbf{q}_{ij}, \eta)$  of Eq. (5.3.35), using the subJacobians of Eq. (3.2.9) and  $\eta^{\text{rev}}$ , are

$$\begin{aligned}
\left( \Phi_{q_i}^{\text{revT}} \ddot{\eta}^{\text{rev}} \right)_{q_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & s_i'^T A_i^T \eta^{\text{rev}} \end{bmatrix} \\
\left( \Phi_{q_i}^{\text{revT}} \ddot{\eta}^{\text{rev}} \right)_{q_j} &= \mathbf{0} \\
\left( \Phi_{q_j}^{\text{revT}} \ddot{\eta}^{\text{rev}} \right)_{q_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -s_j'^T A_j^T \eta^{\text{rev}} \end{bmatrix}
\end{aligned} \tag{5.B.6}$$

For the translational constraint of Eq. (3.2.15), Eqs. (3.2.17) through (3.2.19) define  $\mathbf{P}2^{\text{tran}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij})$  in Eq. (3.1.9). From Eqs. (3.2.15), (5.3.30) and (5.3.31), terms required to evaluate  $\mathbf{P}3^{\text{tran}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij})$  are

$$\begin{aligned}
\left( \left( \Phi_{q_i}^{\text{tran}} \ddot{\mathbf{q}}_i \right)_{q_i} \ddot{\mathbf{q}}_i \right)_{q_i} &= \begin{bmatrix} -\dot{\phi}_i^2 v_i'^T A_i^T \mathbf{P} & \dot{\phi}_i^2 v_i'^T s_i' - 2\dot{\phi}_i v_i'^T A_i^T \mathbf{P} \dot{r}_i + \dot{\phi}_i^2 v_i'^T A_i^T \mathbf{d}_{ij} \\ \mathbf{0} & \dot{\phi}_i^2 v_i'^T A_i^T A_j v_j' \end{bmatrix} \\
\left( \left( \Phi_{q_i}^{\text{tran}} \ddot{\mathbf{q}}_i \right)_{q_i} \ddot{\mathbf{q}}_i \right)_{q_j} &= \begin{bmatrix} \dot{\phi}_i^2 v_i'^T A_i^T \mathbf{P} & -\dot{\phi}_i^2 v_i'^T A_i^T A_j s_j' \\ \mathbf{0} & -\dot{\phi}_i^2 v_i'^T A_i^T A_j v_j' \end{bmatrix} \\
\left( \left( \Phi_{q_i}^{\text{tran}} \ddot{\mathbf{q}}_i \right)_{q_j} \ddot{\mathbf{q}}_j \right)_{q_i} &= \begin{bmatrix} \mathbf{0} & \dot{\phi}_i v_i'^T A_i^T \mathbf{P} \dot{r}_j - \dot{\phi}_i \dot{\phi}_j v_i'^T A_i^T A_j s_j' \\ \mathbf{0} & -\dot{\phi}_i \dot{\phi}_j v_i'^T A_i^T A_j v_j' \end{bmatrix} \\
\left( \left( \Phi_{q_i}^{\text{tran}} \ddot{\mathbf{q}}_i \right)_{q_j} \ddot{\mathbf{q}}_j \right)_{q_j} &= \begin{bmatrix} \mathbf{0} & \dot{\phi}_i \dot{\phi}_j v_i'^T A_i^T A_j s_j' \\ \mathbf{0} & \dot{\phi}_i \dot{\phi}_j v_i'^T A_i^T A_j v_j' \end{bmatrix} \\
\left( \left( \Phi_{q_j}^{\text{tran}} \ddot{\mathbf{q}}_j \right)_{q_i} \ddot{\mathbf{q}}_i \right)_{q_i} &= \begin{bmatrix} \mathbf{0} & \dot{\phi}_j^2 v_i'^T A_i^T A_j s_j' \\ \mathbf{0} & \dot{\phi}_j^2 v_i'^T A_i^T A_j v_j' \end{bmatrix} \\
\left( \left( \Phi_{q_j}^{\text{tran}} \ddot{\mathbf{q}}_j \right)_{q_i} \ddot{\mathbf{q}}_i \right)_{q_j} &= \begin{bmatrix} \mathbf{0} & -\dot{\phi}_j^2 v_i'^T A_i^T A_j s_j' \\ \mathbf{0} & -\dot{\phi}_j^2 v_i'^T A_i^T A_j v_j' \end{bmatrix}
\end{aligned} \tag{5.B.7}$$

Thus,

$$\mathbf{P}3^{\text{tran}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij}) = \begin{bmatrix} \mathbf{P}3_i^{\text{tran}} & \mathbf{P}3_j^{\text{tran}} \end{bmatrix} \tag{5.B.8}$$

where

$$\mathbf{P3}_i^{\text{tran}} = \begin{bmatrix} -\dot{\phi}_i^2 \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} & \mathbf{c}_i \\ \mathbf{0} & \mathbf{d}_i \end{bmatrix}$$

$$\mathbf{P3}_j^{\text{tran}} = \begin{bmatrix} \dot{\phi}_i^2 \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} & \mathbf{c}_j \\ \mathbf{0} & \mathbf{d}_j \end{bmatrix}$$

$$\mathbf{c}_i = \dot{\phi}_i^2 \mathbf{v}'^T \mathbf{s}'_i + 2\dot{\phi}_i \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} (\dot{\mathbf{r}}_i - \dot{\mathbf{r}}_i) + \dot{\phi}_i^2 \mathbf{v}'^T \mathbf{A}_i^T \mathbf{d}_{ij} + (\dot{\phi}_j^2 - 2\dot{\phi}_i \dot{\phi}_j) \mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{s}'_j$$

$$\mathbf{c}_j = -(\dot{\phi}_i^2 - 2\dot{\phi}_i \dot{\phi}_j + \dot{\phi}_j^2) \mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{s}'_j$$

$$\mathbf{d}_i = (\dot{\phi}_i^2 - 2\dot{\phi}_i \dot{\phi}_j + \dot{\phi}_j^2) \mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j$$

$$\mathbf{d}_j = -(\dot{\phi}_i^2 - 2\dot{\phi}_i \dot{\phi}_j + \dot{\phi}_j^2) \mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j$$

Terms in  $\mathbf{P4}^{\text{tran}}(\mathbf{q}_{ij}, \eta)$  of Eq. (5.3.35), using the subJacobians of Eq. (3.2.15) and

$$\eta^{\text{tran}} = [\eta_1^{\text{tran}} \quad \eta_2^{\text{tran}}]^T$$

$$\begin{aligned} \left( \Phi_{\mathbf{q}_i}^{\text{tran} T} \ddot{\eta}^{\text{tran}} \right)_{\mathbf{q}_i} &= \begin{bmatrix} \mathbf{0} & \eta_1^{\text{tran}} \mathbf{A}_i \mathbf{v}'_i \\ \eta_1^{\text{tran}} \mathbf{v}'^T \mathbf{A}_i^T & \eta_1^{\text{tran}} \mathbf{v}'^T \mathbf{P} \mathbf{s}'_i - \eta_1^{\text{tran}} \mathbf{d}_{ij}^T \mathbf{P} \mathbf{A}_i \mathbf{v}'_i + \eta_2^{\text{tran}} \mathbf{v}'^T \mathbf{A}_j^T \mathbf{P} \mathbf{A}_i \mathbf{v}'_i \end{bmatrix} \\ \left( \Phi_{\mathbf{q}_i}^{\text{tran} T} \ddot{\eta}^{\text{tran}} \right)_{\mathbf{q}_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\eta_1^{\text{tran}} \mathbf{v}'^T \mathbf{A}_i^T & -\mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j (\eta_1^{\text{tran}} \mathbf{s}'_j - \eta_2^{\text{tran}} \mathbf{v}'_j) \end{bmatrix} \\ \left( \Phi_{\mathbf{q}_j}^{\text{tran} T} \ddot{\eta}^{\text{tran}} \right)_{\mathbf{q}_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{A}_j (\eta_1^{\text{tran}} \mathbf{s}'_j + \eta_2^{\text{tran}} \mathbf{v}'_j) \end{bmatrix} \end{aligned} \tag{5.B.9}$$

### 5.B.1.2 Derivatives of Spatial Constraints

For the distance constraint of Eq. (3.1.3), Eqs. (3.3.2) through (3.3.4) define

$\mathbf{P2}^{\text{dist}}(\mathbf{q}_{ij}, \chi_{ij})$  of Eq. (3.1.9). From Eqs. (3.3.4), (5.3.30) and (5.3.31), terms required to evaluate

$\mathbf{P3}^{\text{dist}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_{ij})$  are

$$\begin{aligned}
\left( \left( \Phi_{q_i}^{\text{dist}} \ddot{q}_i \right)_{q_i} \ddot{q}_i \right)_{q_i} &= \left[ \dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i) \quad 2\bar{a}_i^T \mathbf{B}(\dot{p}_i, s'_i) + \dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i) \mathbf{B}(p_i, s'_i) \right] \\
\left( \left( \Phi_{q_i}^{\text{dist}} \ddot{q}_i \right)_{q_i} \ddot{q}_i \right)_{q_j} &= \left[ -\dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i) \quad -\dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i) \mathbf{B}(p_j, s'_j) \right] \\
\left( \left( \Phi_{q_i}^{\text{dist}} \ddot{q}_i \right)_{q_j} \ddot{q}_j \right)_{q_i} &= \left[ \mathbf{0} \quad -\bar{a}_j^T \mathbf{B}(\dot{p}_i, s'_i) \right] \\
\left( \left( \Phi_{q_i}^{\text{dist}} \ddot{q}_i \right)_{q_j} \ddot{q}_j \right)_{q_j} &= \left[ \mathbf{0} \quad -\bar{a}_i^T \mathbf{B}(\dot{p}_j, s'_j) \right] \\
\left( \left( \Phi_{q_j}^{\text{dist}} \ddot{q}_j \right)_{q_j} \ddot{q}_j \right)_{q_i} &= \left[ -\dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j) \quad -\dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j) \mathbf{B}(p_i, s'_i) \right] \\
\left( \left( \Phi_{q_j}^{\text{dist}} \ddot{q}_j \right)_{q_j} \ddot{q}_j \right)_{q_j} &= \left[ \dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j) \quad 2\bar{a}_j^T \mathbf{B}(\dot{p}_j, s'_j) + \dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j) \mathbf{B}(p_j, s'_j) \right]
\end{aligned} \tag{5.B.10}$$

where

$$\begin{aligned}
\bar{a}_i &= \dot{r}_i + \mathbf{B}(p_i, s'_i) \dot{p}_i \\
\bar{a}_j &= \dot{r}_j + \mathbf{B}(p_j, s'_j) \dot{p}_j
\end{aligned}$$

Thus,

$$\mathbf{P3}^{\text{dist}}(q_{ij}, \dot{q}_{ij}) = \left[ \mathbf{P3}_i^{\text{dist}} \quad \mathbf{P3}_j^{\text{dist}} \right] \tag{5.B.11}$$

where

$$\begin{aligned}
\mathbf{P3}_i^{\text{dist}} &= \left[ \dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i) - \dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j) \quad 2(\bar{a}_i - \bar{a}_j)^T \mathbf{B}(\dot{p}_i, s'_i) + (\dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i) - \dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j)) \mathbf{B}(p_i, s'_i) \right] \\
\mathbf{P3}_j^{\text{dist}} &= \left[ \dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j) - \dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i) \quad 2(\bar{a}_j - \bar{a}_i)^T \mathbf{B}(\dot{p}_j, s'_j) + (\dot{p}_j^T \mathbf{B}^T(\dot{p}_j, s'_j) - \dot{p}_i^T \mathbf{B}^T(\dot{p}_i, s'_i)) \mathbf{B}(p_j, s'_j) \right]
\end{aligned}$$

Terms in  $\mathbf{P4}^{\text{dist}}(q_{ij}, \eta^{\text{dist}})$  of Eq. (5.3.35), using the subJacobians of Eq. (3.3.4), the derivative identity of Eq. (2.6.38), and the fact that  $\eta^{\text{dist}}$  is a scalar, are

$$\begin{aligned}
\left( \Phi_{q_i}^{\text{distT}} \ddot{q}_i \right)_{q_i} &= \text{dist} \left[ \begin{array}{cc} \mathbf{I} & \mathbf{B}(p_i, s'_i) \\ \mathbf{B}^T(p_i, s'_i) & \mathbf{B}^T(p_i, s'_i) \mathbf{B}(p_i, s'_i) - \mathbf{K}(s'_i, d_{ij}) \end{array} \right] \\
\left( \Phi_{q_i}^{\text{distT}} \ddot{q}_i \right)_{q_j} &= - \text{dist} \left[ \begin{array}{cc} \mathbf{I} & \mathbf{B}(p_j, s'_j) \\ \mathbf{B}^T(p_i, s'_i) & \mathbf{B}^T(p_i, s'_i) \mathbf{B}(p_j, s'_j) \end{array} \right] \\
\left( \Phi_{q_j}^{\text{distT}} \ddot{q}_j \right)_{q_j} &= \text{dist} \left[ \begin{array}{cc} \mathbf{I} & \mathbf{B}(p_j, s'_j) \\ \mathbf{B}^T(p_j, s'_j) & \mathbf{B}^T(p_j, s'_j) \mathbf{B}(p_j, s'_j) + \mathbf{K}(s'_j, d_{ij}) \end{array} \right]
\end{aligned} \tag{5.B.12}$$

For the spherical constraint of Eq. (3.3.8), Eqs. (3.3.9) and (3.3.10) define  $\mathbf{P2}^{\text{dist}}(q_{ij}, \chi_{ij})$  of Eq. (3.1.9). From Eqs. (3.3.10), (5.3.30) and (5.3.31), terms required to evaluate  $\mathbf{P3}^{\text{dist}}(q_{ij}, \dot{q}_{ij})$  are all zero, so

$$\mathbf{P}3^{\text{sph}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_j) = \mathbf{0} \quad (5.B.13)$$

Terms in  $\mathbf{P}4^{\text{sph}}(\mathbf{q}_{ij}, \eta^{\text{sph}})$  of Eq. (5.3.35), using the subJacobians of Eq. (3.3.10) and the derivative identity of Eq. (2.6.38) are

$$\begin{aligned} \left( \Phi_{\mathbf{q}_i}^{\text{sph}^T} \dot{\mathbf{q}}^{\text{sph}} \right)_{\mathbf{q}_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{K}(\mathbf{s}'_i, \eta^{\text{sph}}) \end{bmatrix} \\ \left( \Phi_{\mathbf{q}_i}^{\text{sph}^T} \dot{\mathbf{q}}^{\text{sph}} \right)_{\mathbf{q}_j} &= \mathbf{0} \\ \left( \Phi_{\mathbf{q}_j}^{\text{sph}^T} \dot{\mathbf{q}}^{\text{sph}} \right)_{\mathbf{q}_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}(\mathbf{s}'_j, \eta^{\text{sph}}) \end{bmatrix} \end{aligned} \quad (5.B.14)$$

For the dot1 constraint of Eq. (3.3.13), Eqs. (3.3.15) and (3.3.16) define  $\mathbf{P}2^{\text{dot1}}(\mathbf{q}_{ij}, \chi_{ij})$  of Eq. (3.1.9). From Eq. (3.3.15), terms required to evaluate  $\mathbf{P}3^{\text{dot1}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_j)$  are

$$\begin{aligned} \left( \left( \Phi_{\mathbf{q}_i}^{\text{dot1}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} &= \mathbf{0} \\ \left( \left( \Phi_{\mathbf{q}_i}^{\text{dot1}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} &= \begin{bmatrix} \mathbf{0} & \dot{\mathbf{p}}_i^T \mathbf{B}^T(\dot{\mathbf{p}}_i, \mathbf{a}'_i) \mathbf{B}(\mathbf{p}_j, \mathbf{a}'_j) \end{bmatrix} \\ \left( \left( \Phi_{\mathbf{q}_i}^{\text{dot1}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} &= \begin{bmatrix} \mathbf{0} & \dot{\mathbf{p}}_j^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}'_j) \mathbf{B}(\dot{\mathbf{p}}_i, \mathbf{a}'_i) \end{bmatrix} \\ \left( \left( \Phi_{\mathbf{q}_i}^{\text{dot1}} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} &= \begin{bmatrix} \mathbf{0} & \dot{\mathbf{p}}_i^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{a}'_i) \mathbf{B}(\dot{\mathbf{p}}_j, \mathbf{a}'_j) \end{bmatrix} \\ \left( \left( \Phi_{\mathbf{q}_j}^{\text{dot1}} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_i} \ddot{\mathbf{q}}_i \right)_{\mathbf{q}_i} &= \begin{bmatrix} \mathbf{0} & \dot{\mathbf{p}}_j^T \mathbf{B}^T(\dot{\mathbf{p}}_j, \mathbf{a}'_j) \mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i) \end{bmatrix} \\ \left( \left( \Phi_{\mathbf{q}_j}^{\text{dot1}} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} \ddot{\mathbf{q}}_j \right)_{\mathbf{q}_j} &= \mathbf{0} \end{aligned} \quad (5.B.15)$$

Thus,

$$\mathbf{P}3^{\text{dot1}}(\mathbf{q}_{ij}, \dot{\mathbf{q}}_j) = \begin{bmatrix} \mathbf{P}3_i^{\text{dot1}} & \mathbf{P}3_j^{\text{dot1}} \end{bmatrix} \quad (5.B.16)$$

where

$$\begin{aligned} \mathbf{P}3_i^{\text{dot1}} &= \begin{bmatrix} \mathbf{0} & 2\dot{\mathbf{p}}_j^T \mathbf{B}^T(\mathbf{p}_j, \mathbf{a}'_j) \mathbf{B}(\dot{\mathbf{p}}_i, \mathbf{a}'_i) + \dot{\mathbf{p}}_j^T \mathbf{B}^T(\dot{\mathbf{p}}_j, \mathbf{a}'_j) \mathbf{B}(\mathbf{p}_i, \mathbf{a}'_i) \end{bmatrix} \\ \mathbf{P}3_j^{\text{dot1}} &= \begin{bmatrix} \mathbf{0} & 2\dot{\mathbf{p}}_i^T \mathbf{B}^T(\mathbf{p}_i, \mathbf{a}'_i) \mathbf{B}(\dot{\mathbf{p}}_j, \mathbf{a}'_j) + \dot{\mathbf{p}}_i^T \mathbf{B}^T(\dot{\mathbf{p}}_i, \mathbf{a}'_i) \mathbf{B}(\mathbf{p}_j, \mathbf{a}'_j) \end{bmatrix} \end{aligned}$$

Terms in  $\mathbf{P}4^{\text{dot1}}(\mathbf{q}_{ij}, \eta^{\text{dot1}})$  of Eq. (5.3.33), using the subJacobians of Eq. (3.3.14), are

$$\begin{aligned}
\left( \Phi_{q_i}^{\text{dot1}^T} \dot{\eta}^{\text{dot1}} \right)_{q_i} &= \overset{\text{dot1}}{\left[ \begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}(a'_i, A_j a'_j) \end{array} \right]} \\
\left( \Phi_{q_i}^{\text{dot1}^T} \dot{\eta}^{\text{dot1}} \right)_{q_j} &= \overset{\text{dot1}}{\left[ \begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B^T(p_i, a'_i) B(p_j, a'_j) \end{array} \right]} \\
\left( \Phi_{q_j}^{\text{dot1}^T} \dot{\eta}^{\text{dot1}} \right)_{q_j} &= \overset{\text{dot1}}{\left[ \begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}(a'_j, A_i a'_i) \end{array} \right]}
\end{aligned} \tag{5.B.17}$$

For the dot2 constraint of Eq. (3.3.18), Eqs. (3.3.20) and (3.3.21) define  $\mathbf{P}2^{\text{dot1}}(q_{ij}, \chi_{ij})$  of Eq. (3.1.9). From Eq. (3.3.20), terms required to evaluate  $\mathbf{P}3^{\text{dot1}}(q_{ij}, \dot{q}_{ij})$  are

$$\begin{aligned}
\left( \left( \Phi_{q_i}^{\text{dot2}} \ddot{q}_i \right)_{q_i} \ddot{q}_i \right)_{q_i} &= \mathbf{0} \\
\left( \left( \Phi_{q_i}^{\text{dot2}} \ddot{q}_i \right)_{q_i} \ddot{q}_i \right)_{q_j} &= - \left[ \mathbf{0} \quad \dot{p}_i^T B^T(p_i, s'_i) B(p_j, a'_j) \right] \\
\left( \left( \Phi_{q_i}^{\text{dot2}} \ddot{q}_i \right)_{q_j} \ddot{q}_i \right)_{q_i} &= - \left[ \mathbf{0} \quad \dot{p}_j^T B^T(p_j, a'_j) B(\dot{p}_i, s'_i) \right] \\
\left( \left( \Phi_{q_i}^{\text{dot2}} \ddot{q}_i \right)_{q_j} \ddot{q}_i \right)_{q_j} &= - \left[ \mathbf{0} \quad (\dot{r}_i^T + \dot{p}_i^T B^T(p_i, s'_i)) B(\dot{p}_j, a'_j) \right] \\
\left( \left( \Phi_{q_j}^{\text{dot2}} \ddot{q}_j \right)_{q_i} \ddot{q}_j \right)_{q_i} &= - \left[ \dot{p}_j^T B^T(p_j, a'_j) \quad \dot{p}_j^T B^T(p_j, a'_j) B(p_i, s'_i) \right] \\
\left( \left( \Phi_{q_j}^{\text{dot2}} \ddot{q}_j \right)_{q_i} \ddot{q}_j \right)_{q_j} &= \left[ \dot{p}_j^T B^T(p_j, a'_j) \quad \mathbf{d} \right]
\end{aligned} \tag{5.B.18}$$

where

$$\begin{aligned}
\mathbf{d} = & \dot{p}_j^T B^T(p_j, a'_j) B(p_j, s'_j) + \dot{p}_j^T B^T(p_j, s'_j) B(p_j, a'_j) + 2\dot{r}_j^T B(\dot{p}_j, a'_j) \\
& + 2\dot{p}_j^T B^T(p_j, s'_j) B(\dot{p}_j, a'_j) + 2\dot{p}_j^T B^T(p_j, a'_j) B(\dot{p}_j, s'_j)
\end{aligned}$$

Thus,

$$\mathbf{P}3^{\text{dot2}}(q_{ij}, \dot{q}_{ij}) = \left[ \mathbf{P}3_i^{\text{dot2}} \quad \mathbf{P}3_j^{\text{dot2}} \right] \tag{5.B.19}$$

where

$$\begin{aligned}
\mathbf{P}3_i^{\text{dot2}} &= - \left[ \dot{p}_j^T B^T(p_j, a'_j) \quad 2\dot{p}_j^T B^T(p_j, a'_j) B(\dot{p}_i, s'_i) + \dot{p}_j^T B^T(p_j, a'_j) B(p_i, s'_i) \right] \\
\mathbf{P}3_j^{\text{dot2}} &= \left[ \dot{p}_j^T B^T(p_j, a'_j) \quad -\dot{p}_i^T B^T(p_i, s'_i) B(p_j, a'_j) - 2(\dot{r}_i^T + \dot{p}_i^T B^T(p_i, s'_i)) B(\dot{p}_j, a'_j) + \mathbf{d} \right]
\end{aligned}$$

Terms in  $\mathbf{P}4^{\text{dot2}}(q_{ij}, \eta^{\text{dot2}})$  of Eq. (5.3.33), using the subJacobians of Eq. (3.3.19), are

$$\begin{aligned}
\left(\Phi_{q_i}^{\text{dot2}^T} \ddot{\eta}^{\text{dot2}}\right)_{q_i} &= -\ddot{\eta}^{\text{dot2}} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}(s'_i, A_j a'_j) \end{bmatrix} \\
\left(\Phi_{q_i}^{\text{dot1}^T} \ddot{\eta}^{\text{dot1}}\right)_{q_j} &= -\ddot{\eta}^{\text{dot1}} \begin{bmatrix} \mathbf{0} & \mathbf{B}(p_j, a'_j) \\ \mathbf{0} & B^T(p_i, s'_i) B(p_j, a'_j) \end{bmatrix} \\
\left(\Phi_{q_j}^{\text{dot1}^T} \ddot{\eta}^{\text{dot1}}\right)_{q_j} &= \ddot{\eta}^{\text{dot1}} \begin{bmatrix} \mathbf{0} & \mathbf{B}(p_j, a'_j) \\ B^T(p_j, a'_j) & \mathbf{e} \end{bmatrix}
\end{aligned} \tag{5.B.20}$$

where

$$\mathbf{e} = \mathbf{K}(s'_j, A_j a'_j) + \mathbf{K}(a'_j, d_{ij}) + \mathbf{B}^T(p_j, s'_j) \mathbf{B}(p_j, a'_j) + \mathbf{B}^T(p_j, a'_j) \mathbf{B}(p_j, s'_j)$$

For the Euler parameter normalization constraint, Eqs. (3.3.47) through (3.3.49) define  $\mathbf{P}2^{p_i}(p_i, \chi_{p_i})$ . From Eq. (3.3.48),  $\mathbf{P}3^{p_i}(p_i, \dot{p}_i)$  is

$$\mathbf{P}3^{p_i}(p_i, \dot{p}_i) = \left( \left( p_i (p_i)_{q_i} \ddot{q}_i \right)_{q_i} \ddot{q}_i \right)_{q_i} = \left( (p_i^T \dot{p}_i)_{q_i} \dot{q}_i \right)_{q_i} = (p_i^T \dot{p}_i)_{q_i} = \mathbf{0} \tag{5.B.21}$$

Finally,

$$\mathbf{P}4^{p_i}(p_i, \eta^{p_i}) = \left( \left( p_i (p_i)_{q_i} \ddot{\eta}^{p_i} \right)_{q_i} \ddot{\eta}^{p_i} \right)_{q_i} = \eta^{p_i} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ p_i & I_4 \end{bmatrix}_{q_i} = \eta^{p_i} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_4 \end{bmatrix} \tag{5.B.22}$$

## 5.B.2 Derivatives of Kinetic Quantities

Derivative expressions

$$\begin{aligned}
\mathbf{M}2(q, \mu) &= \text{diag}\left(\left(M_1(q_1)\ddot{\mu}_1\right)_{q_1} \cdots \left(M_{nb}(q_{nb})\ddot{\mu}_{nb}\right)_{q_{nb}}\right) \\
\mathbf{S}_q(q, \dot{q}) &= \left[ \left(S_{q_1}(q_1, \dot{q}_1)\right)^T \cdots \left(S_{q_{nb}}(q_{nb}, \dot{q}_{nb})\right)^T \right]^T \\
\mathbf{S}_{\dot{q}}(q, \dot{q}) &= \left[ \left(S_{\dot{q}_1}(q_1, \dot{q}_1)\right)^T \cdots \left(S_{\dot{q}_{nb}}(q_{nb}, \dot{q}_{nb})\right)^T \right]^T
\end{aligned} \tag{5.B.23}$$

involve terms for each planar and spatial body in a system. Derivatives required to evaluate the system level expressions of Eq. (5.B.23) are presented in the following subsections. Expressions are given for *noncentroidal reference frames*. Results for centroidal reference frames are obtained by setting the vector that locates the centroid to zero; i.e.,  $s_i^C = \mathbf{0}$ .

### 5.B.2.1 Derivatives of Planar Kinetics

For the planar mass matrix of Eq. (4.2.20),

$$\mathbf{M}_i(q_i) = \begin{bmatrix} m_i \mathbf{I} & m_i \mathbf{P} \mathbf{A}_i s_i'^C \\ m_i s_i'^{CT} \mathbf{A}_i^T \mathbf{P}^T & J_i' \end{bmatrix} \tag{5.B.24}$$

with  $\mu = \begin{bmatrix} \mu_r^T & \mu_\phi^T \end{bmatrix}^T$ ,

$$\mathbf{M}2(\mathbf{q}_i, \mu_i) = (\mathbf{M}_i(\mathbf{q}_i)\dot{\boldsymbol{\mu}}_i)_{\mathbf{q}_i} = \begin{bmatrix} \mathbf{0} & -\mu_\phi m_i \mathbf{A}_i \mathbf{s}'^C \\ \mathbf{0} & -m_i \mathbf{s}'^{CT} \mathbf{A}_i^T \mu_r \end{bmatrix} \quad (5.B.25)$$

For the velocity coupling term of Eq. (4.2.20),

$$\mathbf{S}_i(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} m_i \dot{\phi}_i^2 \mathbf{A}_i \mathbf{s}'^C \\ 0 \end{bmatrix} \quad (5.B.26)$$

and

$$\begin{aligned} \mathbf{S}_{i\dot{\mathbf{q}}_i} &= \begin{bmatrix} \mathbf{0} & m_i \dot{\phi}_i^2 \mathbf{P} \mathbf{A}_i \mathbf{s}'^C \\ \mathbf{0} & 0 \end{bmatrix} \\ \mathbf{S}_{i\ddot{\mathbf{q}}_i} &= \begin{bmatrix} \mathbf{0} & 2m_i \dot{\phi}_i \mathbf{A}_i \mathbf{s}'^C \\ \mathbf{0} & 0 \end{bmatrix} \end{aligned} \quad (5.B.27)$$

### 5.B.2.2 Derivatives of Spatial Kinetics

For the spatial mass matrix of Eq. (4.3.27),

$$\mathbf{M}_i(\mathbf{q}_i) = \begin{bmatrix} m_i \mathbf{I} & -2m_i \mathbf{A}_i \tilde{\mathbf{s}}_i^C \mathbf{G}_i \\ 2m_i \mathbf{G}_i^T \tilde{\mathbf{s}}_i^C \mathbf{A}_i^T & 4\mathbf{G}_i^T \mathbf{J}'_i \mathbf{G}_i \end{bmatrix} \quad (5.B.28)$$

where  $\mathbf{G}_i = \mathbf{G}(\mathbf{p}_i)$ . Defining  $\mu_i = [\mu_r^T \quad \mu_p^T]^T$  and using derivative identities of Eqs. (2.6.30), (2.6.44), and (2.6.50),

$$\begin{aligned} \mathbf{M}2(\mathbf{q}_i, \mu_i) &= (\mathbf{M}_i(\mathbf{q}_i)\dot{\boldsymbol{\mu}}_i)_{\mathbf{q}_i} \\ &= \begin{bmatrix} \mathbf{0} & -2m_i \mathbf{B}(\mathbf{p}_i, \tilde{\mathbf{s}}_i^C \mathbf{G}_i \mu_p) + 2m_i \mathbf{A}_i \tilde{\mathbf{s}}_i^C \mathbf{G}(\mu_p) \\ \mathbf{0} & \mathbf{T}(2m_i \tilde{\mathbf{s}}_i^C \mathbf{A}_i^T \mu_r + 4\mathbf{J}'_i \mathbf{G}_i \mu_p) + 2m_i \mathbf{G}_i^T \tilde{\mathbf{s}}_i^C \mathbf{C}(\mathbf{p}_i, \mu_r) - 4\mathbf{G}_i^T \mathbf{J}'_i \mathbf{G}(\mu_p) \end{bmatrix} \end{aligned} \quad (5.B.29)$$

For the velocity coupling term of Eq. (4.3.25),

$$\mathbf{S}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) = \begin{bmatrix} 4m_i \mathbf{E}_i \dot{\mathbf{G}}_i^T \dot{\mathbf{G}}_i \mathbf{G}_i^T \mathbf{s}'^C \\ 8\dot{\mathbf{G}}_i^T \mathbf{J}'_i \dot{\mathbf{G}}_i \mathbf{p}_i \end{bmatrix} \quad (5.B.30)$$

where  $\mathbf{E}_i = \mathbf{E}(\mathbf{p}_i)$ ,  $\dot{\mathbf{G}}_i = \mathbf{G}(\dot{\mathbf{p}}_i)$ , and using the above cited identities and that of Eq. (2.6.43),

$$\begin{aligned} \mathbf{S}_{i\dot{\mathbf{q}}_i} &= \begin{bmatrix} \mathbf{0} & -4m_i \mathbf{E}(\dot{\mathbf{G}}_i^T \dot{\mathbf{G}}_i \mathbf{G}_i^T \mathbf{s}'^C) + 4m_i \mathbf{E}_i \dot{\mathbf{G}}_i^T \dot{\mathbf{G}}_i \mathbf{T}(\mathbf{s}'^C) \\ \mathbf{0} & 8\dot{\mathbf{G}}_i^T \mathbf{J}'_i \dot{\mathbf{G}}_i \end{bmatrix} \\ \mathbf{S}_{i\ddot{\mathbf{q}}_i} &= \begin{bmatrix} \mathbf{0} & 4m_i \mathbf{E}_i \mathbf{T}(\dot{\mathbf{G}}_i \mathbf{G}_i^T \mathbf{s}'^C) - 4m_i \mathbf{E}_i \dot{\mathbf{G}}_i^T \mathbf{G}(\mathbf{G}_i^T \mathbf{s}'^C) \\ \mathbf{0} & 8\mathbf{T}(\mathbf{J}'_i \dot{\mathbf{G}}_i \mathbf{p}_i) - 8\dot{\mathbf{G}}_i^T \mathbf{J}'_i \mathbf{G}_i \end{bmatrix} \end{aligned} \quad (5.B.31)$$

### 5.B.3 Derivatives of Generalized Forces

It is not possible to anticipate all forms of force that may act on bodies in multibody systems, so ad-hoc calculation of generalized forces and their derivatives is often required. Two forms of force that can be systematically accounted for are due to gravity and translational spring-damper-actuators (TSDA) of Section 4.5.

#### 5.B.3.1 Derivatives of Planar Generalized Forces

For gravitational force in the negative y direction, Eqs. (4.2.15) and (4.2.16) are

$$\begin{aligned} \mathbf{F}_g &= -\int_A \mathbf{u}_y g \, dA = -g \mathbf{u}_y \int_A \, dA = -g m \mathbf{u}_y \\ n'_g &= -\int_A \mathbf{s}'^T \mathbf{A}^T(\phi) \mathbf{P}^T \mathbf{u}_y g \, dA = \int_A \mathbf{s}'^T \, dA \left( g \mathbf{A}^T(\phi) \mathbf{P} \mathbf{u}_y \right) = m g \mathbf{s}'^c T \mathbf{A}^T(\phi) \mathbf{P} \mathbf{u}_y \end{aligned} \quad (5.B.32)$$

where  $\mathbf{u}_y$  is the unit vector in the positive y direction and  $m$  is mass density. Thus, the gravitational generalized force on a body with a noncentroidal reference frame is

$$\mathbf{Q}^g = mg \begin{bmatrix} -\mathbf{u}_y \\ \mathbf{s}'^c T \mathbf{A}^T(\phi) \mathbf{P} \mathbf{u}_y \end{bmatrix} \quad (5.B.33)$$

For a body with generalized coordinates  $\mathbf{q} = [\mathbf{r}^T \ \phi]^T$ ,

$$\begin{aligned} \mathbf{Q}_{\mathbf{q}}^g &= mg \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{s}'^c T \mathbf{A}^T(\phi) \mathbf{P} \mathbf{u}_y \end{bmatrix} \\ \mathbf{Q}_{\dot{\mathbf{q}}}^g &= \mathbf{0} \end{aligned} \quad (5.B.34)$$

For the TSDA generalized forces of Eq. (4.5.16),

$$\begin{aligned} \mathbf{Q}_i &= (f/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ -\mathbf{s}_i'^T \mathbf{A}_i^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \\ \mathbf{Q}_j &= (f/\ell) \begin{bmatrix} -\mathbf{d}_{ij} \\ \mathbf{s}_j'^T \mathbf{A}_j^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \end{aligned} \quad (5.B.35)$$

With  $f$ ,  $\ell$ , and  $\dot{\ell}$  of Eqs. (4.5.1), (4.5.5), and (4.5.7),

$$\begin{aligned}
\mathbf{Q}_{iq_i} &= \begin{bmatrix} \mathbf{d}_{ij} \\ -\mathbf{s}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \left( (1/\ell)(K\ell_{q_i} + C\dot{\ell}_{q_i}) - (f/\ell^2)\ell_{q_i} \right) \\
&\quad + (f/\ell) \begin{bmatrix} [-\mathbf{I} \quad -\mathbf{P}\mathbf{A}_i \mathbf{s}'_i] \\ [\mathbf{s}'^T \mathbf{A}_i^T \mathbf{P} \quad -\mathbf{s}'^T \mathbf{s}'_i - \mathbf{s}'^T \mathbf{A}_i^T \mathbf{d}_{ij}] \end{bmatrix} \\
\mathbf{Q}_{iq_j} &= \begin{bmatrix} \mathbf{d}_{ij} \\ -\mathbf{s}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \left( (1/\ell)(K\ell_{q_j} + C\dot{\ell}_{q_j}) - (f/\ell^2)\ell_{q_j} \right) \\
&\quad + (f/\ell) \begin{bmatrix} [\mathbf{I} \quad \mathbf{P}\mathbf{A}_j \mathbf{s}'_j] \\ [-\mathbf{s}'^T \mathbf{A}_i^T \mathbf{P} [\mathbf{I} \quad \mathbf{P}\mathbf{A}_j \mathbf{s}'_j]] \end{bmatrix} \\
\mathbf{Q}_{jq_i} &= \begin{bmatrix} -\mathbf{d}_{ij} \\ \mathbf{s}'^T \mathbf{A}_j^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \left( (1/\ell)(K\ell_{q_i} + C\dot{\ell}_{q_i}) - (f/\ell^2)\ell_{q_i} \right) \\
&\quad + (f/\ell) \begin{bmatrix} [\mathbf{I} \quad \mathbf{P}\mathbf{A}_i \mathbf{s}'_i] \\ [-\mathbf{s}'^T \mathbf{A}_j^T \mathbf{P} [\mathbf{I} \quad \mathbf{P}\mathbf{A}_i \mathbf{s}'_i]] \end{bmatrix} \\
\mathbf{Q}_{jq_j} &= \begin{bmatrix} -\mathbf{d}_{ij} \\ \mathbf{s}'^T \mathbf{A}_j^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \left( (1/\ell)(K\ell_{q_j} + C\dot{\ell}_{q_j}) - (f/\ell^2)\ell_{q_j} \right) \\
&\quad + (f/\ell) \begin{bmatrix} -[\mathbf{I} \quad \mathbf{P}\mathbf{A}_j \mathbf{s}'_j] \\ [\mathbf{s}'^T \mathbf{A}_j^T \mathbf{P} \quad \mathbf{s}'^T \mathbf{s}'_j - \mathbf{s}'^T \mathbf{A}_j^T \mathbf{d}_{ij}] \end{bmatrix} \tag{5.B.36}
\end{aligned}$$

where

$$\begin{aligned}
\ell_{q_i} &= -(1/\ell) \mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{P}\mathbf{A}_i \mathbf{s}'_i] \\
\dot{\ell}_{q_i} &= -(1/\ell^2) \mathbf{d}_{ij}^T \mathbf{a} \ell_{q_i} - (1/\ell) \mathbf{a}^T [\mathbf{I} \quad \mathbf{P}\mathbf{A}_i \mathbf{s}'_i] + (1/\ell) \mathbf{d}_{ij}^T [\mathbf{0} \quad \dot{\phi}_i \mathbf{A}_i \mathbf{s}'_i] \\
\ell_{q_j} &= (1/\ell) \mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{P}\mathbf{A}_j \mathbf{s}'_j] \\
\dot{\ell}_{q_j} &= -(1/\ell^2) \mathbf{d}_{ij}^T \mathbf{a} \ell_{q_j} - (1/\ell) \mathbf{a}^T [\mathbf{I} \quad \mathbf{P}\mathbf{A}_j \mathbf{s}'_j] + (1/\ell) \mathbf{d}_{ij}^T [\mathbf{0} \quad \dot{\phi}_j \mathbf{A}_j \mathbf{s}'_j] \\
\mathbf{a} &= \dot{\mathbf{r}}_j + \dot{\phi}_j \mathbf{P}\mathbf{A}_j \mathbf{s}'_j - \dot{\mathbf{r}}_i - \dot{\phi}_i \mathbf{P}\mathbf{A}_i \mathbf{s}'_i
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathbf{Q}_{i\dot{q}_i} &= C(1/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ -\mathbf{s}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_i} \\
\mathbf{Q}_{i\dot{q}_j} &= C(1/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ -\mathbf{s}'^T \mathbf{A}_i^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_j} \\
\mathbf{Q}_{j\dot{q}_i} &= C(1/\ell) \begin{bmatrix} -\mathbf{d}_{ij} \\ \mathbf{s}'^T \mathbf{A}_j^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_i} \\
\mathbf{Q}_{j\dot{q}_j} &= C(1/\ell) \begin{bmatrix} -\mathbf{d}_{ij} \\ \mathbf{s}'^T \mathbf{A}_j^T \mathbf{P} \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_j}
\end{aligned} \tag{5.B.37}$$

where

$$\begin{aligned}
\dot{\ell}_{\dot{q}_i} &= (1/\ell) \mathbf{d}_{ij}^T [-\mathbf{I} \quad -\mathbf{P} \mathbf{A}_i \mathbf{s}'_i] \\
\dot{\ell}_{\dot{q}_j} &= (1/\ell) \mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{P} \mathbf{A}_j \mathbf{s}'_j]
\end{aligned}$$

Finally, for the RSDA with generalized forces in Eqs. (4.5.17) and (4.5.19), derivatives with respect to  $\mathbf{q}_i$  and  $\mathbf{q}_j$  are

$$\begin{aligned}
\mathbf{Q}_{i\mathbf{q}_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{K} \end{bmatrix} \\
\mathbf{Q}_{i\mathbf{q}_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K} \end{bmatrix} \\
\mathbf{Q}_{j\mathbf{q}_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K} \end{bmatrix} \\
\mathbf{Q}_{j\mathbf{q}_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{K} \end{bmatrix}
\end{aligned} \tag{5.B.38}$$

and derivatives with respect to  $\dot{\mathbf{q}}_i$  and  $\dot{\mathbf{q}}_j$  are

$$\begin{aligned}
\mathbf{Q}_{i\dot{\mathbf{q}}_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{C} \end{bmatrix} \\
\mathbf{Q}_{i\dot{\mathbf{q}}_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \\
\mathbf{Q}_{j\dot{\mathbf{q}}_i} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \\
\mathbf{Q}_{j\dot{\mathbf{q}}_j} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{C} \end{bmatrix}
\end{aligned} \tag{5.B.39}$$

### 5.B.3.2 Derivatives of Spatial Generalized Forces

For gravitational force in the negative z direction, Eq. (4.3.14) is

$$\begin{aligned} \mathbf{F}_g &= -\int_V \mathbf{u}_z g \, dV = -g \mathbf{u}_z \int_V \, dV = -g m \mathbf{u}_z \\ \mathbf{n}'_g &= -\int_V \tilde{\mathbf{s}}' \mathbf{A}^T \mathbf{u}_z g \, dV = -\int_V \tilde{\mathbf{s}}' \, dV (g \mathbf{A}^T \mathbf{u}_z) = -\widetilde{\int_V \tilde{\mathbf{s}}' \, dV} (g \mathbf{A}^T \mathbf{u}_z) = -mg \tilde{\mathbf{s}}'^c \mathbf{A}^T \mathbf{u}_z \end{aligned} \quad (5.B.40)$$

where  $\mathbf{u}_z$  is the unit vector in the positive z direction and  $\tilde{\mathbf{s}}$  is mass density. Thus, the gravitational generalized force on a body with a noncentroidal reference frame in angular velocity generalized coordinates is

$$\mathbf{Q}^g = mg \begin{bmatrix} -\mathbf{u}_z \\ -\tilde{\mathbf{s}}'^c \mathbf{A}^T \mathbf{u}_z \end{bmatrix} \quad (5.B.41)$$

For a body with generalized coordinates  $\mathbf{q} = [\mathbf{r}^T \quad \mathbf{p}^T]^T$  and a noncentroidal reference frame, from Eq. (4.3.25),

$$\mathbf{Q}^g = \begin{bmatrix} \mathbf{F}_g \\ 2\mathbf{G}^T \mathbf{n}'_g \end{bmatrix} = mg \begin{bmatrix} -\mathbf{u}_z \\ -2\mathbf{G}^T \tilde{\mathbf{s}}'^c \mathbf{A}^T \mathbf{u}_z \end{bmatrix} \quad (5.B.42)$$

and, from Eqs. (2.6.30) and (2.6.50),

$$\begin{aligned} \mathbf{Q}_{\mathbf{q}}^g &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -2mg (\mathbf{G}^T \tilde{\mathbf{s}}'^c \mathbf{C}(\mathbf{p}, \mathbf{u}_z) + \mathbf{T}(\tilde{\mathbf{s}}'^c \mathbf{A}^T \mathbf{u}_z)) \end{bmatrix} \\ \mathbf{Q}_{\dot{\mathbf{q}}}^g &= \mathbf{0} \end{aligned} \quad (5.B.43)$$

For the TSDA generalized forces of Eq. (4.5.10), in terms of generalized coordinates  $\mathbf{q} = [\mathbf{r}^T \quad \mathbf{p}^T]^T$ ,

$$\begin{aligned} \mathbf{Q}_i &= (f/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{d}_{ij} \end{bmatrix} \\ \mathbf{Q}_j &= -(f/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{d}_{ij} \end{bmatrix} \end{aligned} \quad (5.B.44)$$

With  $f$ ,  $\ell$ , and  $\dot{\ell}$  of Eqs. (4.5.1), (4.5.5), and (4.5.7),

$$\begin{aligned}
\mathbf{Q}_{iq_i} &= \left[ \begin{array}{c} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{d}_{ij} \end{array} \right] \left( (1/\ell)(K\ell_{q_i} + C\dot{\ell}_{q_i}) - (f/\ell^2)\ell_{q_i} \right) \\
&\quad + (f/\ell) \left[ \begin{array}{c} [-\mathbf{I} \quad -\mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] \\ [\mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \quad \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) + \mathbf{K}(\mathbf{s}'_i, \mathbf{d}_{ij})] \end{array} \right] \\
\mathbf{Q}_{iq_j} &= \left[ \begin{array}{c} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{d}_{ij} \end{array} \right] \left( (1/\ell)(K\ell_{q_j} + C\dot{\ell}_{q_j}) - (f/\ell^2)\ell_{q_j} \right) \\
&\quad + (f/\ell) \left[ \begin{array}{c} [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)] \\ [\mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)]] \end{array} \right] \\
\mathbf{Q}_{jq_i} &= - \left[ \begin{array}{c} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{d}_{ij} \end{array} \right] \left( (1/\ell)(K\ell_{q_i} + C\dot{\ell}_{q_i}) - (f/\ell^2)\ell_{q_i} \right) \\
&\quad - (f/\ell) \left[ \begin{array}{c} [-\mathbf{I} \quad -\mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] \\ [\mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) [-\mathbf{I} \quad -\mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)]] \end{array} \right] \\
\mathbf{Q}_{jq_j} &= - \left[ \begin{array}{c} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{d}_{ij} \end{array} \right] \left( (1/\ell)(K\ell_{q_j} + C\dot{\ell}_{q_j}) - (f/\ell^2)\ell_{q_j} \right) \\
&\quad - (f/\ell) \left[ \begin{array}{c} [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)] \\ [\mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \quad \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) + \mathbf{K}(\mathbf{s}'_j, \mathbf{d}_{ij})] \end{array} \right]
\end{aligned} \tag{5.B.45}$$

where

$$\begin{aligned}
\ell_{q_i} &= -(1/\ell) \mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] \\
\dot{\ell}_{q_i} &= -(1/\ell^2) \mathbf{d}_{ij}^T \mathbf{a} \ell_{q_i} - (1/\ell) \mathbf{a}^T [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] + (1/\ell) \mathbf{d}_{ij}^T [\mathbf{0} \quad \mathbf{B}(\dot{\mathbf{p}}_i, \mathbf{s}'_i)] \\
\ell_{q_j} &= (1/\ell) \mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)] \\
\dot{\ell}_{q_j} &= -(1/\ell^2) \mathbf{d}_{ij}^T \mathbf{a} \ell_{q_j} + (1/\ell) \mathbf{a}^T [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)] + (1/\ell) \mathbf{d}_{ij}^T [\mathbf{0} \quad \mathbf{B}(\dot{\mathbf{p}}_j, \mathbf{s}'_j)] \\
\mathbf{a} &= \dot{\mathbf{r}}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j) \dot{\mathbf{p}}_j - \dot{\mathbf{r}}_i - \mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i) \dot{\mathbf{p}}_i
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathbf{Q}_{i\dot{q}_i} &= C(1/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_i} \\
\mathbf{Q}_{i\dot{q}_j} &= C(1/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}'_i) \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_j} \\
\mathbf{Q}_{j\dot{q}_i} &= -C(1/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_i} \\
\mathbf{Q}_{j\dot{q}_j} &= -C(1/\ell) \begin{bmatrix} \mathbf{d}_{ij} \\ \mathbf{B}^T(\mathbf{p}_j, \mathbf{s}'_j) \mathbf{d}_{ij} \end{bmatrix} \dot{\ell}_{\dot{q}_j}
\end{aligned} \tag{5.B.46}$$

where

$$\begin{aligned}
\dot{\ell}_{\dot{q}_i} &= (1/\ell) \mathbf{d}_{ij}^T [-\mathbf{I} \quad -\mathbf{B}(\mathbf{p}_i, \mathbf{s}'_i)] \\
\dot{\ell}_{\dot{q}_j} &= (1/\ell) \mathbf{d}_{ij}^T [\mathbf{I} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}'_j)]
\end{aligned}$$

For the spatial RSDA of Fig. 4.5.3, differentiating  $c \equiv \cos\theta_{ij} = \mathbf{u}'^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}'^j$  and  $s \equiv \sin\theta_{ij} = \mathbf{v}'^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{u}'^j$  associated with Eq. (4.5.13) with respect to  $\mathbf{p}_j$ ,

$$\begin{aligned}
-s_{ij\dot{p}_j} &= \mathbf{u}'^{iT} \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{u}'^j) \\
c_{ij\dot{p}_j} &= \mathbf{v}'^{iT} \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{u}'^j)
\end{aligned}$$

Multiplying the first equation by  $s$ , the second by  $c$ , subtracting, and using  $s^2 + c^2 = 1$ ,

$$s_{ij\dot{p}_j} = c\mathbf{v}'^{iT} \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{u}'^j) - s\mathbf{u}'^{iT} \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{u}'^j) = (c\mathbf{v}'^{iT} - s\mathbf{u}'^{iT}) \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{u}'^j) \tag{5.B.47}$$

Similarly, differentiating with respect to  $\mathbf{p}_i$ ,

$$\begin{aligned}
-s_{ij\dot{p}_i} &= \mathbf{u}'^{jT} \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{u}'^i) \\
c_{ij\dot{p}_i} &= \mathbf{u}'^{jT} \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}'^i)
\end{aligned}$$

and manipulating as above,

$$s_{ij\dot{p}_i} = c\mathbf{u}'^{jT} \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}'^i) - s\mathbf{u}'^{jT} \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{u}'^i) \tag{5.B.48}$$

Using the identity  $\mathbf{G}(\mathbf{p}_i)\mathbf{p}_j = -\mathbf{G}(\mathbf{p}_j)\mathbf{p}_i$ , Eq. (4.5.14) may be written in the form

$$\begin{aligned}
\dot{s}_{ij\dot{p}_i} &= 2\mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\mathbf{p}_j) \dot{\mathbf{p}}_j - \mathbf{G}(\mathbf{p}_i) \dot{\mathbf{p}}_i) \\
&= 2\mathbf{w}'^{iT} (-\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\dot{\mathbf{p}}_j) \mathbf{p}_j + \mathbf{G}(\dot{\mathbf{p}}_i) \mathbf{p}_i)
\end{aligned} \tag{5.B.49}$$

Differentiating the first form of Eq. (5.B.49) with respect to  $\dot{\mathbf{p}}_i$  and  $\dot{\mathbf{p}}_j$ ,

$$\begin{aligned}
\dot{s}_{ij\dot{p}_i} &= -2\mathbf{w}'^{iT} \mathbf{G}(\mathbf{p}_i) \\
\dot{s}_{ij\dot{p}_j} &= -2\mathbf{w}'^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\mathbf{p}_j)
\end{aligned} \tag{5.B.50}$$

Differentiating the second form of Eq. (5.B.49) with respect to  $\mathbf{p}_i$  and  $\mathbf{p}_j$ ,

$$\begin{aligned}\dot{\mathbf{p}}_{ij\mathbf{p}_i} &= -2\mathbf{p}_j^T \mathbf{G}^T(\dot{\mathbf{p}}_j) \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{w}'^i) + 2\mathbf{w}'^{iT} \mathbf{G}(\dot{\mathbf{p}}_i) \\ \dot{\mathbf{p}}_{ij\mathbf{p}_j} &= -2\mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\dot{\mathbf{p}}_j) + \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{G}(\dot{\mathbf{p}}_j) \mathbf{p}_j))\end{aligned}\quad (5.B.51)$$

Using Eqs. (5.B.47) through (5.B.51) with the torque of Eq. (4.5.11),

$$\begin{aligned}\tau_{\mathbf{p}_i} &= K \left( c\mathbf{u}'^{iT} \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}'^i) - s\mathbf{u}'^{iT} \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{u}'^i) \right) \\ &\quad + C \left( -2\mathbf{p}_j^T \mathbf{G}^T(\dot{\mathbf{p}}_j) \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{w}'^i) + 2\mathbf{w}'^{iT} \mathbf{G}(\dot{\mathbf{p}}_i) \right) \\ \tau_{\mathbf{p}_j} &= K \left( (c\mathbf{v}'^{iT} - s\mathbf{u}'^{iT}) \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{u}'^j) \right) \\ &\quad + C \left( -2\mathbf{w}'^{iT} (\mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\dot{\mathbf{p}}_j) + \mathbf{A}_i^T \mathbf{B}(\mathbf{p}_j, \mathbf{G}(\dot{\mathbf{p}}_j) \mathbf{p}_j)) \right) \\ \tau_{\dot{\mathbf{p}}_i} &= C (-2\mathbf{w}'^{iT} \mathbf{G}(\dot{\mathbf{p}}_i)) \\ \tau_{\dot{\mathbf{p}}_j} &= C (-2\mathbf{w}'^{iT} \mathbf{A}_i^T \mathbf{A}_j \mathbf{G}(\mathbf{p}_j))\end{aligned}\quad (5.B.52)$$

Using Eqs. (2.6.26) and (2.6.50), derivatives of RSDA generalized forces of Eq. (4.5.18) with respect to  $\mathbf{p}_i$  and  $\mathbf{p}_j$  are

$$\begin{aligned}\mathbf{Q}_{i\mathbf{p}_i} &= \begin{bmatrix} \mathbf{0} \\ 2\tau \mathbf{T}(\mathbf{w}'^i) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 2\mathbf{G}^T(\mathbf{p}_i) \mathbf{w}'^i \tau_{\mathbf{p}_i} \end{bmatrix} \\ \mathbf{Q}_{i\mathbf{p}_j} &= \begin{bmatrix} \mathbf{0} \\ 2\mathbf{G}^T(\mathbf{p}_i) \mathbf{w}'^i \tau_{\mathbf{p}_j} \end{bmatrix} \\ \mathbf{Q}_{j\mathbf{p}_i} &= \begin{bmatrix} \mathbf{0} \\ -2\tau \mathbf{G}^T(\mathbf{p}_j) \mathbf{A}_j^T \mathbf{B}(\mathbf{p}_i, \mathbf{w}'^i) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -2\mathbf{G}^T(\mathbf{p}_j) \mathbf{A}_j^T \mathbf{A}_i \mathbf{w}'^i \tau_{\mathbf{p}_i} \end{bmatrix} \\ \mathbf{Q}_{j\mathbf{p}_j} &= \begin{bmatrix} \mathbf{0} \\ -2\tau (\mathbf{T}(\mathbf{A}_j^T \mathbf{A}_i \mathbf{w}'^i) + \mathbf{G}^T(\mathbf{p}_j) \mathbf{C}(\mathbf{p}_j, \mathbf{A}_i \mathbf{w}'^i)) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -2\mathbf{G}^T(\mathbf{p}_j) \mathbf{A}_j^T \mathbf{A}_i \mathbf{w}'^i \tau_{\mathbf{p}_j} \end{bmatrix}\end{aligned}\quad (5.B.53)$$

More directly, derivatives of RSDA generalized forces with respect to  $\dot{\mathbf{p}}_i$  and  $\dot{\mathbf{p}}_j$  are

$$\begin{aligned}\mathbf{Q}_{i\dot{\mathbf{p}}_i} &= \begin{bmatrix} \mathbf{0} \\ 2\mathbf{G}^T(\mathbf{p}_i) \mathbf{w}'^i \tau_{\dot{\mathbf{p}}_i} \end{bmatrix} \\ \mathbf{Q}_{i\dot{\mathbf{p}}_j} &= \begin{bmatrix} \mathbf{0} \\ 2\mathbf{G}^T(\mathbf{p}_i) \mathbf{w}'^i \tau_{\dot{\mathbf{p}}_j} \end{bmatrix} \\ \mathbf{Q}_{j\dot{\mathbf{p}}_i} &= \begin{bmatrix} \mathbf{0} \\ -2\mathbf{G}^T(\mathbf{p}_j) \mathbf{A}_j^T \mathbf{A}_i \mathbf{w}'^i \tau_{\dot{\mathbf{p}}_i} \end{bmatrix} \\ \mathbf{Q}_{j\dot{\mathbf{p}}_j} &= \begin{bmatrix} \mathbf{0} \\ -2\mathbf{G}^T(\mathbf{p}_j) \mathbf{A}_j^T \mathbf{A}_i \mathbf{w}'^i \tau_{\dot{\mathbf{p}}_j} \end{bmatrix}\end{aligned}\quad (5.B.54)$$

## Appendix 5.C Key Formulas, Chapter 5

$$\Phi(\mathbf{q}, t) = 0 \quad \dot{\Phi}_q(\mathbf{q}, t)\dot{\mathbf{q}} = -\Phi_t(\mathbf{q}, t) \quad \ddot{\Phi}_q(\mathbf{q}, t)\ddot{\mathbf{q}} = -\gamma \quad (5.2.1) \quad (5.2.2)$$

$$\mathbf{U} \equiv \Phi_q^T(\mathbf{q}^0, t^0) \quad \Phi_q(\mathbf{q}^0, t^0)\mathbf{V} = \mathbf{0} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I} \quad \mathbf{U}^T\mathbf{V} = \mathbf{0} \quad (5.2.5) \quad (5.2.6)$$

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u} \quad (5.2.8)$$

$$\begin{aligned} \mathbf{u} &= \mathbf{h}(\mathbf{v}, t) & \Delta \mathbf{u}^i &= \mathbf{B}\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i, t) & i &= 1, 2, \dots \\ & & \mathbf{u}^{i+1} &= \mathbf{u}^i + \Delta \mathbf{u}^i & & \end{aligned} \quad (5.2.14) \quad (5.2.17)$$

$$\begin{aligned} \mathbf{B} &\equiv (\Phi_q \mathbf{U})^{-1} & \Delta \mathbf{B}^i &= -\mathbf{B}^i \Phi_q \mathbf{U} \mathbf{B}^i + \mathbf{B}^i & i &= 1, 2, \dots \\ & & \mathbf{B}^{i+1} &= \mathbf{B}^i + \Delta \mathbf{B}^i & & \end{aligned} \quad (5.2.13) \quad (5.2.20)$$

$$\mathbf{q}(\mathbf{v}, t) = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) \quad \mathbf{D}(\mathbf{q}, t) \equiv (\mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{q}, t)\Phi_q(\mathbf{q}, t))\mathbf{V} \quad (5.2.33) \quad (5.2.24)$$

$$\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}, t), t)\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, t), t)\Phi_t(\mathbf{q}(\mathbf{v}, t), t) \quad (5.2.34)$$

$$\ddot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, t) = \mathbf{D}(\mathbf{q}(\mathbf{v}, t), t)\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{q}(\mathbf{v}, t), t)\gamma(\mathbf{q}(\mathbf{v}, t), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}, t), t) \quad (5.2.35)$$

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} = \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) \quad (5.2.48)$$

$$\mathbf{v}(t^0) = \mathbf{0} \quad \dot{\mathbf{v}}(t^0) = \mathbf{V}^T \dot{\mathbf{q}}(t^0) \quad (5.2.51)$$

$$\mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} = \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) \quad \mathbf{v}(t^0) = \mathbf{0} \quad \dot{\mathbf{v}}(t^0) = \mathbf{V}^T \dot{\mathbf{q}}^0 \quad (5.3.1) \quad (5.3.2)$$

$$\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) \equiv (\Phi_q \hat{\boldsymbol{\chi}})_q \quad \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) \equiv \left( (\Phi_q(\mathbf{q}, t) \hat{\dot{\mathbf{q}}})_q \right)_q \quad (5.3.10) \quad (5.3.17)$$

$$\mathbf{P}4(\mathbf{q}, \boldsymbol{\eta}) \equiv (\Phi_q^T \hat{\boldsymbol{\eta}})_q \quad \mathbf{M}2(\mathbf{q}, \boldsymbol{\mu}) \equiv (\mathbf{M} \hat{\boldsymbol{\mu}})_q \quad (5.3.20) \quad (5.3.25)$$

$$\gamma = \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + 2\Phi_{\mathbf{q}}\dot{\mathbf{q}} + \Phi_{\mathbf{q}} \quad (5.3.18)$$

$$\gamma_q = \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) + 2(\Phi_{\mathbf{q}} \hat{\dot{\mathbf{q}}})_q + \Phi_{\mathbf{q}} \quad (5.3.18)$$

$$\gamma_q = 2\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) + 2\Phi_{\mathbf{q}}$$

$$\mathbf{R}(\ddot{\mathbf{v}}, \dot{\mathbf{v}}, \mathbf{v}) \equiv \mathbf{D}^T \mathbf{M} \mathbf{D} \ddot{\mathbf{v}} - \mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0} \quad (5.3.23)$$

$$\begin{aligned} \mathbf{R}_{\ddot{\mathbf{v}}} &= \mathbf{D}^T \mathbf{M} \mathbf{D} \quad \mathbf{R}_{\dot{\mathbf{v}}} = -\mathbf{D}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma_{\dot{\mathbf{q}}} + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}) \mathbf{D} \\ \mathbf{R}_{\mathbf{v}} &= \mathbf{D}^T \left\{ \begin{array}{l} -\mathbf{M} \mathbf{U} \mathbf{B} \mathbf{P}2(\mathbf{q}, 2\mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \gamma) + \mathbf{P}4(\mathbf{q}, \mathbf{B}^T \mathbf{U}^T (\mathbf{M} \mathbf{U} \mathbf{B} \gamma + \mathbf{S} + \mathbf{Q}^A)) \\ + \mathbf{M}2(\mathbf{q}, \mathbf{D} \ddot{\mathbf{v}} - \mathbf{U} \mathbf{B} \gamma) - \mathbf{M} \mathbf{U} \mathbf{B} \gamma_q - \mathbf{S}_q - \mathbf{Q}_q \\ - (\mathbf{M} \mathbf{U} \mathbf{B} \gamma_q + \mathbf{S}_q + \mathbf{Q}_q) \mathbf{U} \mathbf{B} (\mathbf{P}2(\mathbf{q}, \mathbf{U} \mathbf{B} \Phi_t - \mathbf{D} \ddot{\mathbf{v}}) - \Phi_{\mathbf{q}}) \end{array} \right\} \mathbf{D} \end{aligned} \quad (5.3.29)$$

$$\mathbf{J}^{\text{trap}} = \frac{d\mathbf{R}}{d\ddot{\mathbf{v}}_n} = \mathbf{R}_v + \mathbf{R}_v \frac{\partial \dot{\mathbf{v}}_n}{\partial \ddot{\mathbf{v}}_n} + \mathbf{R}_v \frac{\partial \mathbf{v}_n}{\partial \ddot{\mathbf{v}}_n} = \mathbf{R}_v + (h/2)\mathbf{R}_v + (h^2/4)\mathbf{R}_v \quad (5.3.43)$$

$$\mathbf{J}^{\text{RK}} = \frac{d\mathbf{R}}{dk_i} = \mathbf{R}_v + h\gamma\mathbf{R}_v + h^2\gamma^2\mathbf{R}_v \quad (5.3.52)$$

$$\begin{aligned} \mathbf{J}^{\text{trap}}(\dot{\mathbf{v}}_{n-1})\Delta\dot{\mathbf{v}}_n^i &= -\mathbf{R}(\dot{\mathbf{v}}_n^i) \quad i = 0, 1, \dots \\ \dot{\mathbf{v}}_n^{i+1} &= \dot{\mathbf{v}}_n^i + \Delta\dot{\mathbf{v}}_n^i \end{aligned} \quad \begin{aligned} \mathbf{J}^{\text{RK}}(\mathbf{v}_{n-1})\Delta k_i^j &= -\mathbf{R}(k_i^j) \quad i = 0, 1, \dots \\ k_i^{j+1} &= k_i^j + \Delta k_i^j \end{aligned} \quad (5.3.44) \quad (5.3.53)$$

$$\mathbf{MD}\dot{\mathbf{v}} + \Phi_q^T \lambda - \mathbf{MUB}\gamma - \mathbf{S} - \mathbf{Q}^A = \mathbf{0} \quad \mathbf{v}(t^0) = \mathbf{0}, \quad \dot{\mathbf{v}}(t^0) = \mathbf{V}^T \dot{\mathbf{q}}^0 \quad (5.5.11)$$

$$\begin{aligned} \mathbf{R}_v &= \mathbf{MD} \quad \mathbf{R}_v = -(\mathbf{MUB}\gamma_q + \mathbf{S}_q + \mathbf{Q}_q^A)\mathbf{D} \\ \mathbf{R}_v &= \left\{ \begin{array}{l} \mathbf{M2}(\mathbf{q}, \mathbf{D}\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}\gamma) - \mathbf{MUBP2}(\mathbf{q}, \mathbf{D}\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}\gamma) - \mathbf{MUB}\gamma_q + \mathbf{P4}(\mathbf{q}, \lambda) \\ -\mathbf{S}_q - \mathbf{Q}_q^A + (\mathbf{MUB}\gamma_{\dot{\mathbf{q}}} + \mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A)\mathbf{U}\mathbf{B}(\mathbf{P2}(\mathbf{q}, \dot{\mathbf{q}}) + \Phi_{\dot{\mathbf{q}}}) \end{array} \right\} \mathbf{D} \end{aligned} \quad (5.5.18)$$

$$\partial \mathbf{R} / \partial \lambda = \Phi_q^T \quad \mathbf{J}_n = \begin{bmatrix} \partial \mathbf{R}_n / \partial \dot{\mathbf{v}} & \Phi_q^T \end{bmatrix} \quad (5.5.19) \quad (5.5.22)$$

## Appendix 5.D The Trouble with Maggi and Kane Equations

Maggi published landmark papers at the end of the 19<sup>th</sup> century (1896, 1901) that establish a system of equations of motion for *nonholonomically constrained mechanical systems*. His equations of motion are ordinary differential equations (ODE) that avoid the need for Lagrange multipliers and the associated differential-algebraic equations (DAE) of motion.

With the goal of obtaining ODE of dynamics, the Maggi equations were rediscovered in the 1960s in the setting of *nonholonomic systems* (Neimark and Fufaev, 1972), for which they were intended. They were again rediscovered or reinvented by Kane and applied to both nonholonomic and holonomic systems (Kane and Levinson, 1985). *Maggi's equations*, equivalently *Kane's equations* (Borri, Bottasso, and Mantegazza, 1990), have been applied to holonomic systems by enforcing constraints only at the velocity level (Tseng, Ma, and Hulbert, 2003; Bauchau and Laulusa, 2008; Laulusa and Bauchau, 2008; Garcia de Jalon, Callejo, and Hidalgo, 2012). This raises both theoretical and practical issues with the resulting ODE. Since configuration kinematic constraints are not included in the formulation, the numerical solution of the Maggi/Kane equations leads to *drift* from the holonomic constraints; i.e., error that accumulates over time.

### 5.D.1 Kinematics Formulation

For purposes of this analysis, a mechanical system is defined as a collection of rigid bodies whose position and orientation in an inertial reference frame are defined by generalized coordinates  $\mathbf{q} = [q_1 \quad \dots \quad q_n]^T \in \mathbb{R}^n$  that must satisfy holonomic constraints of the form

$$\Phi(\mathbf{q}) = [\Phi_1(\mathbf{q}) \quad \dots \quad \Phi_m(\mathbf{q})]^T = \mathbf{0} \in \mathbb{R}^m \quad (5.D.1)$$

Time independent constraints are treated here, but the formulation can be extended to time dependent constraints. It is assumed that the vector function  $\Phi(\mathbf{q})$  of Eq. (5.D.1) has two continuous derivatives with respect to its arguments. With  $\dot{\mathbf{q}} = \mathbf{q}(t)$ , differentiating Eq. (5.D.1) with respect to time yields velocity and acceleration constraints,

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (5.D.2)$$

$$\Phi_{qq}(\mathbf{q})\ddot{\mathbf{q}} + (\Phi_q(\mathbf{q})\ddot{\mathbf{q}})_q \dot{\mathbf{q}} \equiv \Phi_q(\mathbf{q})\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0} \quad (5.D.3)$$

An implementation of the Maggi/Kane equations using tangent space independent generalized coordinates and enforcing holonomic velocity and acceleration constraints is presented in Section 5.D.2. It is shown in Section 5.D.3 that solutions of the resulting ODE need not satisfy Eq. (5.D.1), hence they may not satisfy all equations of system dynamics.

### 5.D.2 Maggi/Kane Equations with Only Velocity and Acceleration Constraints

At initial time  $t^0$ , with  $\mathbf{q}(t^0) = \mathbf{q}^0$  that satisfies Eq. (5.D.1), define  $\mathbf{U} \equiv \Phi_q^T(\mathbf{q}^0)$  and  $\mathbf{V}$  as a solution of  $\Phi_q(\mathbf{q}^0)\mathbf{V} = \mathbf{0}$  and  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ , using *singular value decomposition* (Strang, 1980). Note that  $\mathbf{V}^T\mathbf{U} = \mathbf{0}$  and  $\mathbf{U}^T\mathbf{V} = \mathbf{0}$ . Since the columns of  $\mathbf{U}$  and  $\mathbf{V}$  span  $\mathbb{R}^n$ , any vector  $\dot{\mathbf{q}}$  can be uniquely represented as

$$\dot{\mathbf{q}} = \mathbf{V}\mathbf{w} - \mathbf{U}\mathbf{z} \quad (5.D.4)$$

In order that this  $\dot{\mathbf{q}}$  satisfies Eq. (5.D.2), it is required that

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = \Phi_q(\mathbf{q})\mathbf{V}\mathbf{w} - \Phi_q(\mathbf{q})\mathbf{U}\mathbf{z} = \mathbf{0} \quad (5.D.5)$$

At  $t^0$ ,  $\Phi_q(\mathbf{q}^0)\mathbf{U} = \mathbf{U}^T\mathbf{U}$  is positive definite, since  $\mathbf{U}$  has full column rank. Hence  $\mathbf{U}^T\mathbf{U}$  is nonsingular. Since  $\Phi_q(\mathbf{q})$  is a continuous function of  $\mathbf{q}$ , the matrix  $\Phi_q(\mathbf{q})\mathbf{U}$  is nonsingular in a neighborhood  $W_1$  of  $\mathbf{q}^0$ ; i.e., in a nonempty open set, possibly large in extent, that contains  $\mathbf{q}^0$ . This justifies the definition

$$\mathbf{B}(\mathbf{q}) \equiv (\Phi_q(\mathbf{q})\mathbf{U})^{-1} \quad (5.D.6)$$

where  $\mathbf{B}(\mathbf{q})$  is a continuously differentiable, nonsingular, matrix function of  $\mathbf{q}$  in  $W_1$ .

Thus, from Eq. (5.D.5),  $\mathbf{z} = \mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q})\mathbf{V}\mathbf{w}$  and Eq. (5.D.4) reduces to

$$\dot{\mathbf{q}} = \mathbf{V}\mathbf{w} - \mathbf{U}\mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q})\mathbf{V}\mathbf{w} = (\mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q}))\mathbf{V}\mathbf{w} = \mathbf{D}(\mathbf{q})\mathbf{w} \quad (5.D.7)$$

where

$$\mathbf{D}(\mathbf{q}) \equiv (\mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q}))\mathbf{V} \quad (5.D.8)$$

The product

$$\Phi_q(\mathbf{q})\mathbf{D}(\mathbf{q}) = \Phi_q(\mathbf{q}) - \Phi_q(\mathbf{q})\mathbf{U}\mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q}) = \Phi_q(\mathbf{q}) - \Phi_q(\mathbf{q}) = \mathbf{0} \quad (5.D.9)$$

shows that the columns of the matrix function  $\mathbf{D}(\mathbf{q})$  are in the null space of the constraint Jacobian. Thus,  $\dot{\mathbf{q}}$  of Eq. (5.D.7) satisfies the constraint velocity equation of Eq. (5.D.2), for arbitrary values of  $\mathbf{w}$ . The vector  $\mathbf{w}$  comprises the *kinetic characteristics* in Maggiøs formulation and *generalized speeds* in Kaneøs formulation. Further, the columns of  $\mathbf{D}(\mathbf{q})$  are Kaneøs *partial velocities*, which in this formulation provide a differentiable basis for the null space of the constraint Jacobian.

Dynamics of the system is governed by the *variational equation of motion*

$$\delta\mathbf{q}^T [\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})] = 0 \quad (5.D.10)$$

which must hold for all  $\delta\mathbf{q}$  that satisfy  $\Phi_q(\mathbf{q})\delta\mathbf{q} = \mathbf{0}$ , where the mass matrix  $\mathbf{M}(\mathbf{q})$  is positive definite on the null space of  $\Phi_q(\mathbf{q})$ ,  $\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)$  is the vector of generalized applied forces, and  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$  is a vector of velocity coupling terms, all assumed to be twice

continuously differentiable functions of their arguments. Consistent with d'Alembert's principle, no constraint reaction forces or Lagrange multipliers appear in Eq. (5.D.10).

In variational form, Eq. (5.D.7) is the virtual displacement relation

$$\delta\mathbf{q} = \mathbf{D}(\mathbf{q})\delta\mathbf{w} \quad (5.D.11)$$

Using Eq. (5.D.9),

$$\Phi_q(\mathbf{q})\delta\mathbf{q} = \Phi_q(\mathbf{q})\mathbf{D}(\mathbf{q})\delta\mathbf{w} = \mathbf{0} \quad (5.D.12)$$

for arbitrary  $\delta\mathbf{w}$ ; i.e.,  $\delta\mathbf{q}$  of Eq. (5.D.11) satisfies Eq. (5.D.12) for arbitrary  $\delta\mathbf{w}$ .

Substituting  $\delta\mathbf{q} = \mathbf{D}(\mathbf{q})\delta\mathbf{w}$  into Eq. (5.D.10) yields

$$\delta\mathbf{w}^T [\mathbf{D}^T(\mathbf{q})\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{D}^T(\mathbf{q})\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{D}^T(\mathbf{q})\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})] = 0 \quad (5.D.13)$$

which holds for all  $\delta\mathbf{q}$  such that  $\Phi_q(\mathbf{q})\delta\mathbf{q} = \mathbf{0}$ , so by Eq. (5.D.12), it holds for arbitrary  $\delta\mathbf{w}$ . Using this fact and Eq. (5.D.7),

$$\mathbf{D}^T(\mathbf{q})\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{D}^T(\mathbf{q})\mathbf{Q}^A(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}, t) - \mathbf{D}^T(\mathbf{q})\mathbf{S}(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}) = \mathbf{0} \quad (5.D.14)$$

which are *Maggi's equations* (Neimark and Fufaev, 1972; Borri, Bottasso, and Mantegazza, 1990). Since Eq. (5.D.14) is comprised of  $n - m$  differential equations in  $2n + m$  variables  $\mathbf{q}$  and  $\mathbf{w}$ , there is an infinity of solutions. To obtain uniquely solvable equations,  $\ddot{\mathbf{q}}$  must be written in terms of  $\mathbf{w}$  and its derivatives.

Differentiating Eq. (5.D.4) with respect to time,

$$\ddot{\mathbf{q}} = \mathbf{V}\dot{\mathbf{w}} - \mathbf{U}\dot{\mathbf{z}} \quad (5.D.15)$$

For  $\ddot{\mathbf{q}}$  to satisfy Eq. (5.D.3),  $\Phi_q(\mathbf{q})\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) = \Phi_q(\mathbf{q})\mathbf{V}\dot{\mathbf{w}} - \Phi_q(\mathbf{q})\mathbf{U}\dot{\mathbf{z}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ .

Using Eq. (5.D.6),  $\dot{\mathbf{z}} = \mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q})\mathbf{V}\dot{\mathbf{w}} + \mathbf{B}(\mathbf{q})\gamma(\mathbf{q}, \dot{\mathbf{q}})$ . Substituting this and Eq. (5.D.7) into Eq. (5.D.15),

$$\begin{aligned} \ddot{\mathbf{q}} &= \mathbf{V}\dot{\mathbf{w}} - \mathbf{U}\mathbf{B}(\mathbf{q})\Phi_q(\mathbf{q})\mathbf{V}\dot{\mathbf{w}} - \mathbf{U}\mathbf{B}(\mathbf{q})\gamma(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}) \\ &= \mathbf{D}(\mathbf{q})\dot{\mathbf{w}} - \mathbf{U}\mathbf{B}(\mathbf{q})\gamma(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}) \end{aligned} \quad (5.D.16)$$

From  $\Phi_q(\mathbf{q})\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) = \Phi_q(\mathbf{q})(\mathbf{D}(\mathbf{q})\dot{\mathbf{w}} - \mathbf{U}\mathbf{B}(\mathbf{q})\gamma(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w})) + \gamma(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}) = \mathbf{0}$ ,  $\ddot{\mathbf{q}}$  of Eq. (5.D.16) is seen to satisfy Eq. (5.D.3), for arbitrary values of  $\mathbf{w}$  and  $\dot{\mathbf{w}}$ .

Substituting Eq. (5.D.16) into Eq. (5.D.14) and combining the result with Eq. (5.D.7) yields

$$\begin{aligned} \mathbf{D}^T(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{D}(\mathbf{q})\dot{\mathbf{w}} &= \mathbf{D}^T(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{U}\mathbf{B}(\mathbf{q})\gamma(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}) \\ &\quad + \mathbf{D}^T(\mathbf{q})\mathbf{Q}^A(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}, t) + \mathbf{D}^T(\mathbf{q})\mathbf{S}(\mathbf{q}, \mathbf{D}(\mathbf{q})\mathbf{w}) \\ \dot{\mathbf{q}} &= \mathbf{D}(\mathbf{q})\mathbf{w} \end{aligned} \quad (5.D.17)$$

which is a system of  $(2n - m)$  first order ODE in  $(2n - m)$  variables  $\mathbf{w}$  and  $\mathbf{q}$ . This is the *modern form of Maggi's equations* of motion that appears in the literature (Tseng, Ma, and Hulbert, 2003; Bauchau and Laulusa, 2008; Laulusa and Bauchau, 2008; Garcia de

Jalon, Callejo, and Hidalgo, 2012). As shown in Section 4.6.3, the mass matrix is positive definite on the null space of the constraint Jacobian, so the matrix

$\mathbf{D}^T(\mathbf{q}^0)\mathbf{M}(\mathbf{q}^0)\mathbf{D}(\mathbf{q}^0) = \Phi_q(\mathbf{q}^0)\mathbf{M}(\mathbf{q}^0)\Phi_q^T(\mathbf{q}^0)$  is positive definite. Since  $\mathbf{D}^T(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{D}(\mathbf{q})$  is a continuous function of  $\mathbf{q}$ , it is nonsingular in an open set  $W_2$  that contains  $\mathbf{q}^0$ . From Eq. (5.D.4),  $\mathbf{w} = \mathbf{V}^T \dot{\mathbf{q}}$ , so *initial conditions for Maggi's equations* are

$$\begin{aligned}\mathbf{w}(t_0) &= \mathbf{V}^T \dot{\mathbf{q}}(t_0) \\ \mathbf{q}(t_0) &= \mathbf{q}^0\end{aligned}\tag{5.D.18}$$

The theory of ODE guarantees that the initial-value problem of Eqs. (5.D.17) and (5.D.18) has a unique solution  $(\mathbf{w}(t), \mathbf{q}(t))$  on the open set  $W_1 \cap W_2$ .

### 5.D.3 Inaccuracy of Maggi and Kane Equations for Holonomic Constraints

Even though the *Maggi/Kane ODE initial value problem* of Eqs. (5.D.17) and (5.D.18) has a unique solution, it is not necessarily the solution of the dynamics problem. The theoretical difficulty in applying the Maggi/Kane equations with holonomic constraints is that  $\mathbf{q}$  and  $\mathbf{w}$  are not Lagrangian generalized coordinates (Pars, 1965); i.e., they do not identically satisfy all three forms of the equations of constraint of Eqs. (5.D.1) through (5.D.3). More specifically, the unique solution of Eqs. (5.D.17) and (5.D.18) need not satisfy Eq. (5.D.1).

A more important issue is that in numerical solution of the initial value problem of Eqs. (5.D.17) and (5.D.18), Eq. (5.D.1) is satisfied only to within error tolerances of the numerical integration method. This is a source of *drift*; i.e., error in satisfying Eq. (5.D.1) over time. Since the equations of motion are valid only on the *regular constraint manifold*  $\tilde{C} = \{\mathbf{q} \in \mathbb{R}^n : \Phi(\mathbf{q}) = \mathbf{0}, \text{rank}(\Phi_q(\mathbf{q})) = m\}$ , not in a neighborhood of it, numerical integration error corrupts the equations of motion and leads to an additional source of error that is difficult to quantify. Since the constraint of Eq. (5.D.1) is an integral component of system kinematics, the fact that it is not satisfied in numerical solution of the Maggi/Kane equations is a serious problem. Some implementations of the Maggi/Kane equations monitor error in satisfying Eq. (5.D.1), stop the simulation when error exceeds a given threshold, modify the approximate solution to satisfy Eq. (5.D.1), and continue the simulation, which induces further error in the approximate solution.

A second practical difficulty with the Maggi/Kane equations is that ad-hoc selection of generalized coordinates may not yield differentiable vector functions that span the null space of the constraint Jacobian, leading to significant error. The derivation of Section 5.D.2 avoids this problem, but only in the neighborhood  $W_1$ .

## Appendix 5.E Generalized Coordinate Partitioning

An early approach to solution of the *Full DAE of multibody dynamics* that shares some of the characteristics of the tangent space differential geometric method presented in this chapter is the *generalized coordinate partitioning method*. It was the foundation for Vol. I of this series (Haug, 1989) and has been used extensively for *explicit numerical integration* of an independent subset of system generalized coordinates (Wehage and Haug, 1982). The approach has been adapted to support *implicit numerical integration* (Haug and Yen, 1992; Negrut, Haug, and German, 2003), but suffers from complexities that have to date precluded a practical implementation. Nevertheless, its role in early development of the field of multibody dynamics warrants a summary presentation.

### 5.E.1 Gaussian Elimination with the Constraint Jacobian Matrix

Since the  $m \times n$  Jacobian  $\Phi_q(\mathbf{q})$  for a system with  $m$  holonomic constraint equations

$$\Phi(\mathbf{q}) = [\Phi_1(\mathbf{q}) \quad \dots \quad \Phi_m(\mathbf{q})]^T = \mathbf{0} \quad (5.E.1)$$

in  $n$  generalized coordinates  $\mathbf{q} = [q_1 \quad \dots \quad q_n]^T$  is required to have rank  $m$ , there exist subvectors  $\mathbf{u} \in \mathbb{R}^m$  and  $\mathbf{v} \in \mathbb{R}^{n-m}$ , denoted

$$\mathbf{q} = [\mathbf{u}^T \quad \mathbf{v}^T]^T \quad (5.E.2)$$

with a *reordering of variables*, such that in a neighborhood  $X^0$  of  $\mathbf{q}^0$  that satisfies Eq. (5.E.1)

$$|\Phi_u(\mathbf{q})| \neq 0 \quad (5.E.3)$$

Thus, in a neighborhood  $V^0$  of  $\mathbf{v}^0$  there exists a unique solution of Eq. (5.E.1)

$$\mathbf{u} = \mathbf{h}(\mathbf{v}) \quad (5.E.4)$$

That is,  $\Phi([\mathbf{h}^T(\mathbf{v}) \quad \mathbf{v}^T]^T) = \mathbf{0}$ , for all  $\mathbf{v} \in V^0$ .

To simplify notation in writing the equation for kinematically admissible virtual displacements  $\mathbf{q} \equiv \mathbf{x}$  and  $\Phi_q \equiv [a_{ij}]_{m \times n}$ ; i.e.,  $\Phi_q \mathbf{q} = \mathbf{0}$ , in component form,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= 0 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= 0 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= 0 \end{aligned} \quad (5.E.5)$$

or, in matrix form,  $\mathbf{Ax} = \mathbf{0}$ . If *full pivoting* in *Gaussian elimination* is used and no zero pivots are encountered in *forward elimination* (Strang, 1980), after  $m$  steps the system reduces to the form

$$\begin{bmatrix} 1 & a_{12}^{(m)} & a_{13}^{(m)} & \cdots & a_{1m}^{(m)} & a_{1m+1}^{(m)} & \cdots & a_{1n}^{(m)} \\ 0 & 1 & a_{23}^{(m)} & \cdots & a_{2m}^{(m)} & a_{2m+1}^{(m)} & \cdots & a_{2n}^{(m)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & & 1 & a_{mm+1}^{(m)} & \cdots & a_{mn}^{(m)} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \\ v_1 \\ \vdots \\ v_{n-m} \end{bmatrix} = \mathbf{0} \quad (5.E.6)$$

where  $\mathbf{u} = [u_1 \cdots u_m]^T$  and  $\mathbf{v} = [v_1 \cdots v_{n-m}]^T$  contain reordered elements of  $\mathbf{x}$ , due to column pivoting.

Equation (5.E.6) may be written in partitioned form as

$$\mathbf{U}\mathbf{u} + \mathbf{R}\mathbf{v} = \mathbf{0} \quad (5.E.7)$$

where

$$\mathbf{U} = \begin{bmatrix} 1 & a_{12}^{(m)} & a_{13}^{(m)} & \cdots & a_{1m}^{(m)} \\ 0 & 1 & a_{23}^{(m)} & \cdots & a_{2m}^{(m)} \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & & 1 \end{bmatrix} \quad (5.E.8)$$

$$\mathbf{R} = \begin{bmatrix} a_{1m+1}^{(m)} & \cdots & a_{1n}^{(m)} \\ a_{2m+1}^{(m)} & \cdots & a_{2n}^{(m)} \\ \vdots & & \vdots \\ a_{mm+1}^{(m)} & \cdots & a_{mn}^{(m)} \end{bmatrix}$$

Note that for any value of  $\mathbf{v}$  in Eq. (5.E.7),  $\mathbf{u}$  may be determined through back substitution. Thus,  $\mathbf{v}$  may be treated as a vector of *independent coordinates* and  $\mathbf{u}$  as a vector of *dependent coordinates*. The vector of generalized coordinates  $\mathbf{q}$  is said to be partitioned into independent and dependent coordinates, according to column pivoting that occurs during factorization of  $\mathbf{A} = \Phi_q(\mathbf{q})$  that leads to Eq. (5.E.6).

The matrix  $\mathbf{U}$  is obtained by carrying out elementary row operations on  $\Phi_u$ , so both have the same rank. Since  $|\mathbf{U}|=1$ ,  $\mathbf{U}$  has *full rank* and so does  $\Phi_u$ . The foregoing *Gaussian elimination* process thus defines the desired partitioning of Eq. (5.E.2).

### 5.E.2 The Differential-Algebraic Equations of Motion

The Lagrange multiplier form of the equations of motion of Eq. (4.10.13) is

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q})\lambda = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (5.E.9)$$

Differentiating Eq. (5.E.1) yields the velocity and acceleration equations

$$\Phi_q(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (5.E.10)$$

$$\Phi_q(\mathbf{q})\ddot{\mathbf{q}} = -\left(\Phi_q(\mathbf{q})\dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} \equiv \gamma(\mathbf{q}, \dot{\mathbf{q}}) \quad (5.E.11)$$

With initial conditions

$$\begin{aligned} \mathbf{q}(t^0) &= \mathbf{q}^0 \\ \dot{\mathbf{q}}(t^0) &= \dot{\mathbf{q}}^0 \end{aligned} \quad (5.E.12)$$

where  $\Phi(\mathbf{q}^0) = \mathbf{0}$  and  $\Phi_q(\mathbf{q}^0)\dot{\mathbf{q}}^0 = \mathbf{0}$ , Eqs. (5.E.9), (5.E.1), (5.E.10), and (5.E.11) comprise the *Full DAE of motion* that is to be solved.

Combining Eqs. (5.E.9) and (5.E.11), in matrix form,

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \Phi_q^T(\mathbf{q}) \\ \Phi_q(\mathbf{q}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \gamma(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} \quad (5.E.13)$$

To show that the coefficient matrix on the left of Eq. (5.E.13) is nonsingular; i.e., it uniquely determines accelerations and Lagrange multipliers, suppressing arguments, define

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \mathbf{0} \quad (5.E.14)$$

The coefficient matrix is nonsingular if and only if Eq. (5.E.14) implies  $\alpha$  and  $\beta$  are zero.

Expanding Eq. (5.E.14),

$$\begin{aligned} \mathbf{M}\alpha + \Phi_q^T\beta &= \mathbf{0} \\ \Phi_q\alpha &= \mathbf{0} \end{aligned} \quad (5.E.15)$$

Multiplying the first equation on the left by  $\alpha^T$  yields

$$\alpha^T \mathbf{M}\alpha + \alpha^T \Phi_q^T \beta = \alpha^T \mathbf{M}\alpha + (\Phi_q \alpha)^T = \alpha^T \mathbf{M}\alpha = \mathbf{0} \quad (5.E.16)$$

where the second of Eqs. (5.E.15) was used. Since  $\mathbf{M}$  is *positive definite* on the null space of  $\Phi_q$  and  $\alpha$  is in the null space,  $\alpha = \mathbf{0}$ . The first of Eqs. (5.E.15) thus reduces to

$$\Phi_q^T \beta = \mathbf{0}$$

Since the kinematic constraints are independent, the columns of  $\Phi_q^T$  are independent. Thus  $\beta = \mathbf{0}$  and the coefficient matrix in Eq. (5.E.13) is nonsingular.

### 5.E.3 Partitioning the DAE of Dynamics

With the reordering  $\mathbf{q} = [\mathbf{u}^T \quad \mathbf{v}^T]^T$  defined by the foregoing process,  $|\Phi_u(\mathbf{q}^0, t^0)| \neq 0$ ,  $\mathbf{u}$  is a dependent set of generalized coordinates, and  $\mathbf{v}$  is an independent set. By the *implicit function theorem* of Section 2.2, if  $\Phi(\mathbf{q})$  has  $k$  continuous derivatives with respect to all variables, there is a  $k$ -times continuously differentiable function  $\mathbf{h}(\mathbf{v})$  of Eq. (5.E.4) such that

$$\Phi(\mathbf{h}(\mathbf{v}), \mathbf{v}) = \mathbf{0} \quad (5.E.17)$$

for all  $\mathbf{v}$  in a neighborhood  $V^0$  of  $\mathbf{v}^0$ .

In terms of this partitioning, the kinematic velocity and acceleration equations are

$$\Phi_u \dot{\mathbf{u}} + \Phi_v \dot{\mathbf{v}} = \mathbf{0} \quad (5.E.18)$$

$$\Phi_u \ddot{\mathbf{u}} + \Phi_v \ddot{\mathbf{v}} = -\left[ (\Phi_u \dot{\mathbf{u}})_u \dot{\mathbf{u}} + (\Phi_u \dot{\mathbf{u}})_v \dot{\mathbf{v}} + (\Phi_v \dot{\mathbf{v}})_u \dot{\mathbf{u}} + (\Phi_v \dot{\mathbf{v}})_v \dot{\mathbf{v}} \right] \equiv \gamma(\mathbf{v}, \dot{\mathbf{v}}, t) \quad (5.E.19)$$

Since  $\Phi_u$  is nonsingular, Eq. (5.E.18) is solved for  $\dot{\mathbf{u}}$  to obtain

$$\dot{\mathbf{u}} = -\Phi_u^{-1} \Phi_v \dot{\mathbf{v}} + \Phi_u^{-1} \gamma \quad (5.E.20)$$

and Eq. (5.E.19) is solved for  $\ddot{\mathbf{u}}$  to obtain

$$\ddot{\mathbf{u}} = -\Phi_u^{-1} \Phi_v \ddot{\mathbf{v}} + \Phi_u^{-1} \gamma \quad (5.E.21)$$

Partitioning the equations of motion of Eq. (5.E.9),

$$\begin{bmatrix} \mathbf{M}^{uu} & \mathbf{M}^{uv} \\ \mathbf{M}^{vu} & \mathbf{M}^{vv} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{v}} \end{bmatrix} + \begin{bmatrix} \Phi_u^T \\ \Phi_v^T \end{bmatrix} \lambda = \begin{bmatrix} \mathbf{Q}^u \\ \mathbf{Q}^v \end{bmatrix} \quad (5.E.22)$$

where all functions involved are functions of only  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $t$ , as a result of using Eqs. (5.E.4) and (5.E.20) to eliminate dependence on  $\mathbf{u}$  and  $\dot{\mathbf{u}}$ . Expanding this partition yields two equations,

$$\mathbf{M}^{uu} \ddot{\mathbf{u}} + \mathbf{M}^{uv} \ddot{\mathbf{v}} + \Phi_u^T \lambda = \mathbf{Q}^u \quad (5.E.23)$$

$$\mathbf{M}^{vu} \ddot{\mathbf{u}} + \mathbf{M}^{vv} \ddot{\mathbf{v}} + \Phi_v^T \lambda = \mathbf{Q}^v \quad (5.E.24)$$

#### 5.E.4 Second Order Partitioned ODE of Dynamics

Since  $\Phi_u$  is nonsingular, so is its transpose, and the notation  $(\Phi_u^T)^{-1} = (\Phi_u^{-1})^T \equiv \Phi_u^{-T}$  is used. Solving Eq. (5.E.23) for  $\lambda$ ,

$$\lambda = \Phi_u^{-T} \left[ \mathbf{Q}^u - \mathbf{M}^{uu} \left( -\Phi_u^{-1} \Phi_v \dot{\mathbf{v}} + \Phi_u^{-1} \gamma \right) - \mathbf{M}^{uv} \dot{\mathbf{v}} \right] \quad (5.E.25)$$

Substituting this result into Eq. (5.E.24),

$$\begin{aligned} & \mathbf{M}^{vu} \left( -\Phi_u^{-1} \Phi_v \dot{\mathbf{v}} + \Phi_u^{-1} \gamma \right) + \mathbf{M}^{vv} \dot{\mathbf{v}} \\ & + \Phi_v^T \Phi_u^{-T} \left[ \mathbf{Q}^v - \mathbf{M}^{vu} \left( -\Phi_u^{-1} \Phi_v \dot{\mathbf{v}} + \Phi_u^{-1} \gamma \right) - \mathbf{M}^{uv} \dot{\mathbf{v}} \right] = \mathbf{Q}^v \end{aligned} \quad (5.E.26)$$

Collecting terms yields the *second order ODE* in partitioned independent coordinates is

$$\ddot{\mathbf{M}}(\mathbf{v}) \dot{\mathbf{v}} = \ddot{\mathbf{Q}}(\mathbf{v}, \dot{\mathbf{v}}, t) \quad (5.E.27)$$

where

$$\begin{aligned} \ddot{\mathbf{M}}(\mathbf{v}) & \equiv \mathbf{M}^{vv} - \mathbf{M}^{vu} \Phi_u^{-1} \Phi_v + \Phi_v^T \Phi_u^{-T} \mathbf{M}^{uu} \Phi_u^{-1} \Phi_v - \Phi_v^T \Phi_u^{-T} \mathbf{M}^{uv} \\ \ddot{\mathbf{Q}}(\mathbf{v}, \dot{\mathbf{v}}, t) & \equiv \mathbf{Q}^v - \mathbf{M}^{vu} \Phi_u^{-1} \gamma - \Phi_v^T \Phi_u^{-T} \left( \mathbf{Q}^u - \mathbf{M}^{uu} \Phi_u^{-1} \gamma \right) \end{aligned} \quad (5.E.28)$$

To investigate properties of the matrix  $\ddot{\mathbf{M}}$ , for any given  $\beta \neq \mathbf{0}$  and the associated  $\alpha \equiv -\Phi_u^{-1}\Phi_v\beta$ , consider the quadratic form

$$\begin{aligned}\beta^T \ddot{\mathbf{M}} \beta &= \beta^T \left[ \mathbf{M}^{vv} - \mathbf{M}^{vu} \Phi_u^{-1} \Phi_v + \Phi_v^T \Phi_u^{-T} \mathbf{M}^{uu} \Phi_u^{-1} \Phi_v - \Phi_v^T \Phi_u^{-T} \mathbf{M}^{uv} \right] \beta \\ &= \beta^T \mathbf{M}^{vv} \beta + \beta^T \mathbf{M}^{vu} \alpha + \alpha^T \mathbf{M}^{uu} \alpha + \alpha^T \mathbf{M}^{uv} \beta \\ &= \begin{bmatrix} \alpha^T & \beta^T \end{bmatrix} \begin{bmatrix} \mathbf{M}^{uu} & \mathbf{M}^{uv} \\ \mathbf{M}^{vu} & \mathbf{M}^{vv} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \begin{bmatrix} \alpha^T & \beta^T \end{bmatrix} \mathbf{M} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} > 0\end{aligned}\tag{5.E.29}$$

since the vector  $\begin{bmatrix} \alpha^T & \beta^T \end{bmatrix}$  is nonzero and kinematically admissible; i.e.,  $\Phi_u \alpha + \Phi_v \beta = \mathbf{0}$  and  $\mathbf{M}$  is positive definite on the space of kinematically admissible velocities. The result for  $\ddot{\mathbf{M}}$  is even stronger. Since  $\beta$  is an arbitrary nonzero vector,  $\ddot{\mathbf{M}}$  is *positive definite*, hence nonsingular.

Since  $\ddot{\mathbf{M}}$  in Eq. (5.E.27) is nonsingular,

$$\ddot{\mathbf{v}} = \ddot{\mathbf{M}}^{-1}(\mathbf{v}) \ddot{\mathbf{Q}}(\mathbf{v}, \dot{\mathbf{v}}, t)\tag{5.E.30}$$

is a classical second order ODE in the independent variable  $\mathbf{v}$ . With *initial conditions*

$$\begin{aligned}\mathbf{v}(t^0) &= \mathbf{v}^0 \\ \dot{\mathbf{v}}(t^0) &= \dot{\mathbf{v}}^0\end{aligned}\tag{5.E.31}$$

Eq. (5.E.30) has a unique solution in some neighborhood of  $t^0$ . Since the right side of Eq. (5.E.30) is  $k$ -times continuously differentiable in its arguments, the solution is  $k$ -times continuously differentiable. The solution can be continued in time until the constraint Jacobian becomes row rank deficient; i.e., until a singular configuration such as lock-up or bifurcation occurs (a bad design), or until a constraint fails to imply the geometry of a well designed system (a bad model of a good design).

Since Eq. (5.E.30) is equivalent to the Full DAE, with  $\mathbf{u} = \mathbf{h}(\mathbf{v})$ ,  $\dot{\mathbf{u}} = -\Phi_u^{-1}\Phi_v\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{u}} = -\Phi_u^{-1}\Phi_v\ddot{\mathbf{v}} + \Phi_u^{-1}\gamma$ , they have the same solutions. To solve the Full DAE, it thus remains only to solve Eqs. (5.E.30) and (5.E.31).

### 5.E.5 Explicit Integration of Partitioned Equations of Motion

Rather than carrying out the extensive computations required to evaluate terms in Eq. (5.E.27) for evaluation on  $\ddot{\mathbf{v}}$ , a practical approach based on *sparse matrix methods* applied to Eq. (5.E.13) has been used (Wehage and Haug, 1982). The matrix equation of Eq. (5.E.13), repeated here for clarity,

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \Phi_q^T(\mathbf{q}) \\ \Phi_q(\mathbf{q}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \gamma(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}\tag{5.E.32}$$

whose coefficient matrix has been shown to be nonsingular, may be formed and efficiently solved for  $\ddot{\mathbf{q}}$  and  $\lambda$  using sparse matrix solution techniques. The solution  $\ddot{\mathbf{q}} = [\ddot{\mathbf{u}}^T \quad \ddot{\mathbf{v}}^T]^T$  yields  $\ddot{\mathbf{v}}$  and  $\ddot{\mathbf{u}}$  at each time step in the solution process. This is equivalent to forming and solving Eq. (5.E.27) for  $\ddot{\mathbf{v}}$ . Applying an explicit integrator advances the solution for  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  on the time grid. Equations (5.E.4) and (5.E.20) then yield  $\mathbf{u}$  and  $\dot{\mathbf{u}}$ , hence  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  on the time grid. The process is continued until the *condition number* of  $\Phi_u$  exceeds an assigned tolerance, at which time a new partitioning is defined and integration is continued. This process is presented in detail in Vol. I of this series (Haug, 1989), as it formed the basis for the large scale commercial multibody dynamic simulation software system called DADS.

To illustrate a key difference between *generalized coordinate partitioning* and the *tangent space parameterization* approach used in this chapter, consider a particle on a unit circle shown in Fig. 5.E.1. The generalized coordinates  $\mathbf{q} = [q_1 \quad q_2]^T$  are subject to the constraint

$$\Phi(\mathbf{q}) = (1/2)(q_1^2 + q_2^2 - 1) = 0 \quad (5.E.33)$$

with Jacobian  $\Phi_q(\mathbf{q}) = \mathbf{q}^T$ . The tangent space at  $\mathbf{q}^0$  is thus spanned by  $\mathbf{V} = \mathbf{P}\mathbf{q}^0$  and the normal is  $\mathbf{U} = \mathbf{q}^0$ , as shown on the left of Fig. 5.E.1. Generalized coordinates are partitioned as

$\mathbf{q} = [\mathbf{u} \quad \mathbf{v}]^T$ , as shown on the right of Fig. 5.E.1. Perturbations in  $\mathbf{v}$  with the two parameterizations yield very different changes in  $\mathbf{v}$  and  $\mathbf{u}$ , as shown in Fig. 5.E.1. The result is a more stable and efficient computation with the tangent space parameterization, compared with that with generalized coordinates. This difference has been verified in numerical computation.

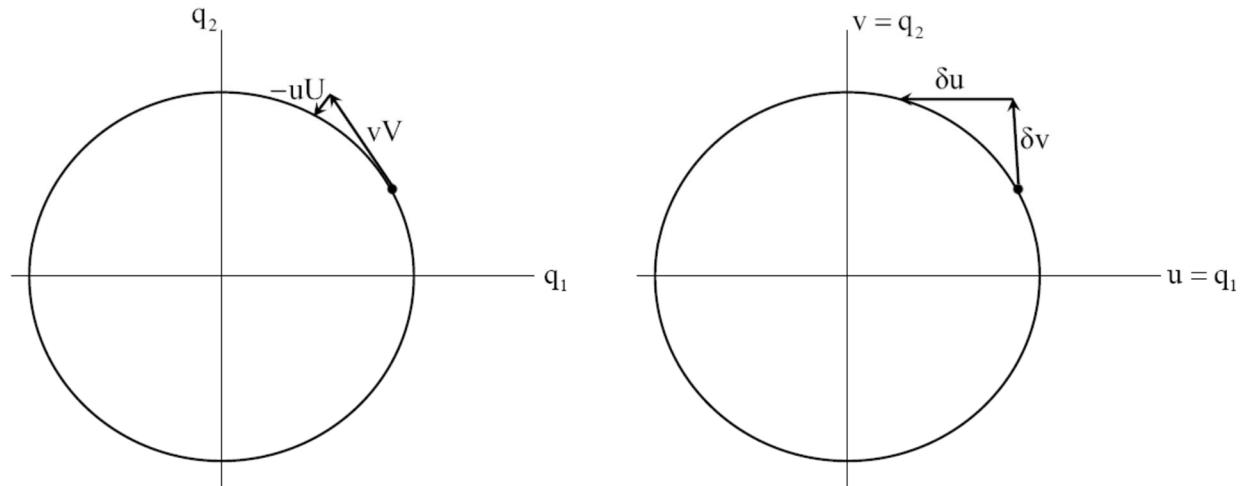


Figure 5.E.1 Tangent Space and Generalized Coordinate Parameterization of Unit Circle

### 5.E.6 Implicit Integration of Second Order Partitioned ODE of Dynamics

The second order partitioned ODE of dynamics that is to be integrated using implicit numerical integration algorithms is repeated from Eqs. (5.E.27) and (5.E.28) as

$$\ddot{\mathbf{M}}(\mathbf{v})\ddot{\mathbf{v}} = \ddot{\mathbf{Q}}(\mathbf{v}, \dot{\mathbf{v}}, t) \quad (5.E.34)$$

where

$$\begin{aligned}\ddot{\mathbf{M}}(\mathbf{v}) &\equiv \mathbf{M}^{vv} - \mathbf{M}^{vu}\Phi_u^{-1}\Phi_v + \Phi_v^T\Phi_u^{-T}\mathbf{M}^{uu}\Phi_u^{-1}\Phi_v - \Phi_v^T\Phi_u^{-T}\mathbf{M}^{uv} \\ \ddot{\mathbf{Q}}(\mathbf{v}, \dot{\mathbf{v}}, t) &\equiv \mathbf{Q}^v - \mathbf{M}^{vu}\Phi_u^{-1}\gamma - \Phi_v^T\Phi_u^{-T}(\mathbf{Q}^u - \mathbf{M}^{uu}\Phi_u^{-1}\gamma)\end{aligned}\quad (5.E.35)$$

Application of implicit Runge-Kutta or trapezoidal methods defined in Sections 4.8.3 and 4.8.4 requires evaluation of the following derivatives:

$$\begin{aligned}& \left( \ddot{\mathbf{M}}(\mathbf{v}) \ddot{\mathbf{u}} \right)_v \\ & \ddot{\mathbf{Q}}_v(\mathbf{v}, \dot{\mathbf{v}}, t) \\ & \ddot{\mathbf{Q}}_{\dot{\mathbf{v}}}(\mathbf{v}, \dot{\mathbf{v}}, t)\end{aligned}\quad (5.E.36)$$

Considering the complexity of derivative expressions for  $\ddot{\mathbf{M}}(\mathbf{v}) = \ddot{\mathbf{M}}(\mathbf{q}(\mathbf{v}))$  and

$\ddot{\mathbf{Q}}(\mathbf{v}, \dot{\mathbf{v}}, t) = \ddot{\mathbf{Q}}(\mathbf{q}(\mathbf{v}), \dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}}), t)$ , derivation of expressions required is clearly a daunting task.

While such derivations have been carried out and implemented in (Haug, and Yen, 1992; Haug, Negrut, and Engstler, 1999; Negrut, Haug, and German, 2003; Negrut, Sandu, Haug, Potra, and Sandu, , 2003; Wang, Haug, and Pan, 2005), the extreme complexity of derivative formulas for general mechanical systems has rendered the approach impractical.

# CHAPTER 6

## Tangent Space ODE for Nonholonomic Systems

### 6.0 Introduction

A broadly applicable tangent space parameterization of nonholonomic multibody system kinematics is presented in Section 6.1. As in the holonomic case, a basis for the tangent space and its orthogonal complement are used to represent system configuration. Due to the appearance of *constraints at the velocity level in nonholonomic systems* that have no counterpart at the configuration level, a separate *parameterization of velocity space* is required. A basis for the *velocity tangent space* and its orthogonal complement is used to represent system velocities and accelerations. A *kinematic differential equation* couples the configuration and velocity parameterizations.

A first order ODE initial-value problem is derived that represents *nonholonomic multibody system dynamics* in Section 6.2, using tangent space generalized coordinates. As a result of the separate parameterizations of position and velocity coordinates, a pair of first order ODE of motion is obtained. Computational algorithms for integrating the equations of motion with explicit and implicit first order numerical integration methods are presented. Four examples are presented in Section 6.3. The first is a relatively simple planar three-wheel transporter, the second and third a moderately complex planar articulated vehicle and a spatial coin rolling on a plane. The last rather intricate spatial example is an ellipsoid rolling without slip on a moving surface. Performance of the method in satisfying all three forms of kinematic constraint, based on error tolerances embedded in the formulation, is verified for each example.

An Index 0 differential-algebraic equation (DAE) formulation of the equations of motion for nonholonomic systems is derived in Section 6.4, using tangent space generalized coordinates. First order DAE are obtained that include Lagrange multipliers. Explicit and implicit numerical integration algorithms are presented for solution of the Index 0 DAE. Two examples are presented in Section 6.5 using the Index 0 DAE approach, a planar two chassis articulated vehicle and a classical spatial problem of a disk that rolls without slip on a plane. Performance of the *Index 0 DAE formulation* is shown to be comparable to that of the ODE formulation.

Theorems that show four formulations of nonholonomic multibody dynamics are well posed are proved in Section 6.6.

MATLAB computer codes that implement the formulations presented are provided in Appendix 6.A and used to obtain numerical solutions for examples treated in the chapter. A summary of key formulas appearing in the chapter is presented in Appendix 6.B.

## 6.1 Tangent Space Kinematics for Nonholonomic Systems

A nonholonomic system is defined as one that includes at least one *nonholonomic constraint*, or *differential constraint*, and possibly several *holonomic constraints*. Technically, if all differential constraints are integrable, the system is holonomic. From a practical point of view, however, if it is not known whether differential constraints are integrable, the system must be treated as nonholonomic. The tangent space formulation introduced in Chapter 5 for holonomic systems is extended to nonholonomic systems, for which configuration and velocity spaces are independently parameterized. Conditions that assure accuracy of constraint satisfaction at configuration, velocity, and acceleration levels are embedded in the formulation.

### 6.1.1 Differential Constraints

Occasionally, constraints on motion of a system are specified in terms of velocities; e.g., in the form of a scalar *first order differential equation*

$$\mathbf{a}^T(\mathbf{q}, t)\dot{\mathbf{q}} + c(\mathbf{q}, t) = 0 \quad (6.1.1)$$

where  $\mathbf{a}(\mathbf{q}, t)$  and  $\mathbf{q}$  are an n-vector of functions and an n-vector of variables, respectively.

Equivalently, Eq. (6.1.1) can be written as the *differential constraint*

$$\mathbf{a}^T(\mathbf{q}, t)d\mathbf{q} + c(\mathbf{q}, t)dt = 0 \quad (6.1.2)$$

whose left side is called a *differential form* in the variables  $\mathbf{q}$  and  $t$ .

A natural question arises in consideration of an equation of the form of Eq. (6.1.2), namely is it an *exact differential form*; i.e., is it the differential of some scalar function  $f(\mathbf{q}, t)$  such that  $df \equiv f_q d\mathbf{q} + f_t dt = \mathbf{a}^T d\mathbf{q} + c dt$ ? If it is, then Eq. (6.1.2) is equivalent to the algebraic equation  $f(\mathbf{q}, t) = \text{constant}$ .

To simplify the statement of conditions for exactness of differential forms, consider the time independent differential form

$$\mathbf{a}^T(\mathbf{q})d\mathbf{q} = \sum_i a_i(\mathbf{q})dq_i = 0 \quad (6.1.3)$$

The following condition from multivariable calculus determines whether a differential form with two or more variables is exact (Lovelock and Rund, 1975):

**Exact Differential Theorem:** The differential form on the left side of Eq. (6.1.3), with  $\mathbf{q} \in \mathbb{R}^n$ ,  $n \geq 2$ , and continuously differentiable coefficients in an open set  $U \subset \mathbb{R}^n$  is exact if and only if

$$\frac{\partial a_i(\mathbf{q})}{\partial q_j} = \frac{\partial a_j(\mathbf{q})}{\partial q_i} \quad (6.1.4)$$

for all  $i$  and  $j$  and all values of  $\mathbf{q} \in U$ .

---

**Example 6.1.1:** The differential form in  $\mathbf{q} \in \mathbb{R}^3$ ,

$$a(\mathbf{q})d\mathbf{q} = 2q_2q_3dq_1 + q_1q_3dq_2 + q_1q_2dq_3$$

is not exact, since  $\frac{\partial a_1}{\partial q_2} = 2q_3 \neq q_3 = \frac{\partial a_2}{\partial q_1}$ . There is no need to test the remaining conditions of Eq. (6.1.4), since if a condition fails for any  $i$  and  $j$ , the differential form is not exact.

---

Even if the differential form of Eq. (6.1.3) is not exact, there may be a scalar function  $b(\mathbf{q})$  such that the differential form  $b(\mathbf{q})\mathbf{a}^T(\mathbf{q})d\mathbf{q}$  is exact. Such a function is called an *integrating factor* and the differential form  $\mathbf{a}^T(\mathbf{q})d\mathbf{q}$  is said to be an *integrable differential form*. A condition for the differential form of Eq. (6.1.3) to be integrable is as follows (Lovelock and Rund, 1975):

**Integrable Differential Theorem:** The differential form of Eq. (6.1.3), with  $\mathbf{q} \in \mathbb{R}^n$ ,  $n \geq 3$ , and continuously differentiable coefficients  $a_i(\mathbf{q})$  in an open set  $U \subset \mathbb{R}^n$  is *integrable* if and only if

$$a_\gamma \left( \frac{\partial a_\beta}{\partial q_\alpha} - \frac{\partial a_\alpha}{\partial q_\beta} \right) + a_\beta \left( \frac{\partial a_\alpha}{\partial q_\gamma} - \frac{\partial a_\gamma}{\partial q_\alpha} \right) + a_\alpha \left( \frac{\partial a_\gamma}{\partial q_\beta} - \frac{\partial a_\beta}{\partial q_\gamma} \right) = 0 \quad (6.1.5)$$

for all  $1 \leq \alpha < \beta < \gamma \leq n$  and for all values of  $\mathbf{q} \in U$ . If the differential form of Eq. (6.1.3) satisfies Eq. (6.1.5), there exists an *integrating factor*  $b(\mathbf{q}) \neq 0$  such that the differential form  $b(\mathbf{q})\mathbf{a}(\mathbf{q})^T d\mathbf{q}$  is exact.

The reason for the condition "in three or more variables" is that any differential form in two variables is integrable (Lovelock and Rund, 1975). For  $k = 3$ , there is only one condition to evaluate in Eq. (6.1.5). For  $k > 3$ , however, there are  $k(k-1)(k-2)/6$  conditions, which becomes oppressive for  $k$  large. Even for  $k = 4$ , there are four conditions. The *integrable differential theorem* may assure existence of an integrating factor, but it does not show how to construct one.

---

**Example 6.1.2:** Applying the test of Eq. (6.1.5) to the differential form of Example 6.1.1 yields

$$q_1q_2(q_3 - 2q_3) + q_1q_3(2q_2 - q_2) + 2q_3q_2(q_1 - q_1) = -q_1q_2q_3 + q_1q_2q_3 = 0 \quad (6.1.6)$$

Thus, the differential form of Example 6.1.1 is *integrable*, even though it is not exact.

---

### 6.1.2 Examples of Nonholonomic Constraints

A kinematic constraint that is stated in terms of velocity is called a *differential constraint*. If such a constraint is the derivative of an algebraic constraint; i.e., it is integrable, then it behaves as a holonomic constraint. If it is not integrable, it is called a *nonholonomic constraint*. No comprehensive approach has been defined for modeling nonholonomic systems, in contrast

to that presented in Sections 3.1 through 3.3 for holonomic systems. Thus, the focus here is on specific examples and an applicable mathematical formulation. Three examples are presented with nonholonomic constraints, two with both holonomic and nonholonomic constraints.

**Example 6.1.3.** The *three-wheel vehicle* shown in Fig. 6.1.1 is used to transport material in a manufacturing facility. The radius of the rear wheels and the distance from the floor to point a on the front wheel assembly are R, so the bed of the transporter, the  $x'y'$  plane, is horizontal. Point b at the center of the rear axle is the origin of the  $x'y'z'$  frame. Generalized coordinates are

$\mathbf{q} = [\mathbf{r}^T \quad \phi]^T = [x \quad y \quad \phi]^T \in \mathbb{R}^3$ . The front wheel is on a trailing arm that pivots about a vertical guide at point a that is parallel to the  $z'$  axis, so it has no influence on the direction of motion. The rear wheels must *roll without slip*, so the velocity of point b must be orthogonal to the axle; i.e., to the  $x'$  axis. This leads to the *velocity constraint*

$$(\mathbf{A}(\phi)\mathbf{u}'_x)^T \dot{\mathbf{r}} = [\cos\phi \quad \sin\phi] \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = 0 \quad (6.1.7)$$

In differential form, this is the *differential constraint*

$$\cos\phi dx + \sin\phi dy + 0d\phi = 0 \quad (6.1.8)$$

Applying the integrability test of Eq. (6.1.4)

$$\sin\phi(-\sin\phi) + \cos\phi(-\cos\phi) = -1 \neq 0 \quad (6.1.9)$$

Thus, the constraint of Eq. (6.1.7) is not integrable; i.e., it is nonholonomic.

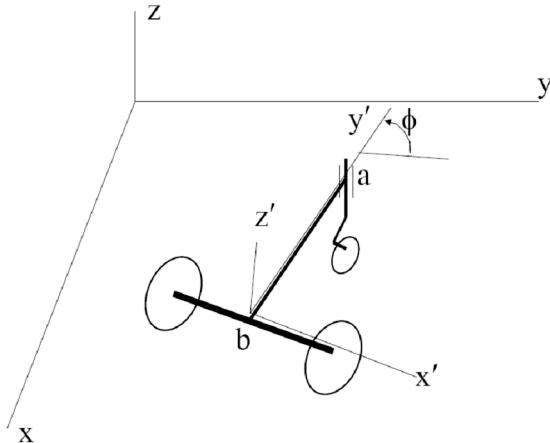


Figure 6.1.1 Three Wheel Material Transporter

It is clear on physical grounds that the transporter can maneuver, using control of rotation of each of the rear wheels, to any point in the  $x'y'$  plane, so the nonholonomic constraint does not impose restrictions on the configuration of the system. If a holonomic constraint were added that requires both rear wheels roll together, as attached to a rigid axle, then the transporter could move only along a straight line specified by the initial value of  $\phi$ .

**Example 6.1.4** A mosquito tracking prey flies near the ground so that its velocity vector is aimed at prey that moves along the x axis with x-coordinate  $\xi(t)$ , as shown in Fig. 6.1.2. The mosquito is modeled as a particle in the x-y plane, with a velocity vector that points toward its prey; i.e.,

$$\dot{\mathbf{r}} - k \left( \mathbf{r} - \begin{bmatrix} \xi(t) \\ 0 \\ 0 \end{bmatrix} \right) = \mathbf{0} \quad (6.1.10)$$

where  $k$  is a negative constant. Multiplying the first component of Eq. (6.1.10) by  $y$ , the second by  $x - \xi(t)$ , and subtracting the results yields the *velocity constraint*

$$y\dot{x} - (x - \xi(t))\dot{y} = k(y(x - \xi(t)) - (x - \xi(t))y) = 0$$

or, in differential form, the *differential constraint* in coordinates  $\mathbf{q} = [x \ y \ t]^T$

$$ydx + (\xi(t) - x)dy + 0dt = 0 \quad (6.1.11)$$

Unlike previous examples, this constraint is explicitly time dependent. Applying the integrability test of Eq. (6.1.4) with  $\alpha = 1$ ,  $\beta = 2$ , and  $\gamma = 3$ ,

$$0 - (\xi(t) - x)(0 - 0) + y(0 + \dot{\xi}(t)) = y\dot{\xi}(t) \neq 0 \quad (6.1.12)$$

so the constraint of Eq. (6.1.11) is not integrable; i.e., it is *nonholonomic*.

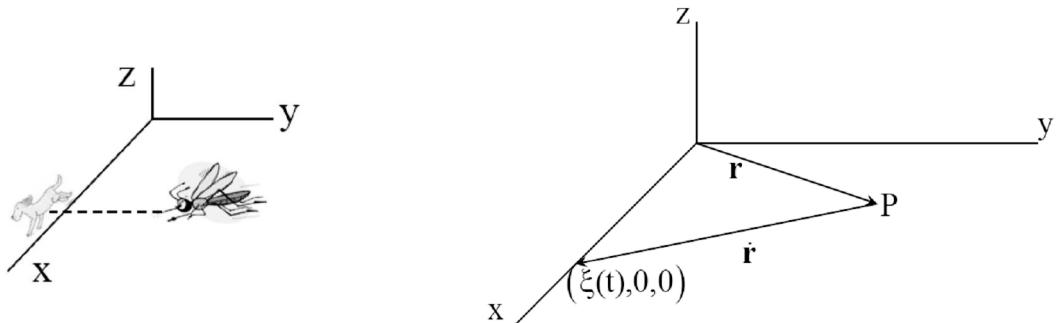


Figure 6.1.2 Mosquito in x-y Plane Tracking Prey

This example has a single nonholonomic constraint and no holonomic constraints. To illustrate a simple system with both holonomic and nonholonomic constraints, consider a more sophisticated mosquito shown in Fig. 6.1.3 that flies in a plane that is defined by the *holonomic constraint*

$$(\mathbf{r}) \equiv z - y = 0 \quad (6.1.13)$$

which contains the x-axis. The tracking constraint of Eq. (6.1.10) is extended to space as

$$\dot{\mathbf{r}} - k \left( \mathbf{r} - \begin{bmatrix} \xi(t) \\ 0 \\ 0 \end{bmatrix} \right) = \mathbf{0} \quad (6.1.14)$$

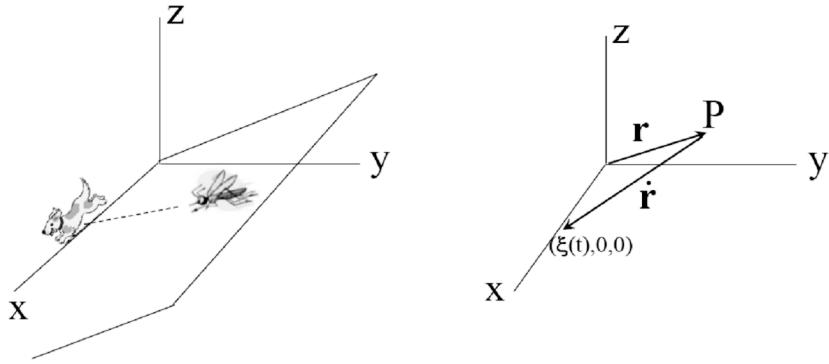


Figure 6.1.3 Mosquito in Plane  $z = y$  Tracking Prey

Multiplying the first component of Eq. (6.1.14) by  $y$  and the second by  $x - \xi(t)$ , subtracting results, and converting to a differential form in  $\mathbf{q} = [x \ y \ z \ t]^T$  yields

$$ydx - (x - \xi(t))dy + 0dz + 0dt = 0 \quad (6.1.15)$$

Applying the integrability test of Eq. (6.1.5) with  $\alpha = 1$ ,  $\beta = 2$ , and  $\gamma = 4$ ,  $y\dot{\xi}(t) \neq 0$ , so Eq. (6.1.15) is nonholonomic. Similarly, multiplying the first component of Eq. (6.1.14) by  $z$  and the third by  $x - \xi(t)$ , subtracting the results, and converting to differential form yields

$$zdx + 0dy - (x - \xi(t))dz + 0dt = 0 \quad (6.1.16)$$

Applying the integrability test of Eq. (6.1.5) with  $\alpha = 1$ ,  $\beta = 3$ , and  $\gamma = 4$ ,  $z\dot{\xi}(t) \neq 0$ , so Eq. (6.1.16) is also nonholonomic.

Converting Eqs. (6.1.15) and (6.1.16) to derivative form yields the *nonholonomic velocity constraints*

$$\mathbf{E}(\mathbf{r}, t)\dot{\mathbf{r}} = \begin{bmatrix} y & -(x - \xi(t)) & 0 \\ z & 0 & -(x - \xi(t)) \end{bmatrix} \dot{\mathbf{r}} = \mathbf{0} \quad (6.1.17)$$

**Example 6.1.5.** The disk rolling without slip on the  $x$ - $y$  plane shown in Fig. 6.1.4 has unit radius. The plane of the disk is defined by body fixed  $y'$ - $z'$  axes and the body fixed  $x'$  axis is normal to the plane of the disk. Unit vector  $\mathbf{a}'$  in the  $x'$ - $y'$ - $z'$  frame from the center of the disk to contact point C on the periphery with the  $x$ - $y$  plane is

$$\mathbf{a}' = \begin{bmatrix} 0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{a} \\ \mathbf{I} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{a}' \\ \mathbf{I} \end{bmatrix} \mathbf{a} = \mathbf{a}' \mathbf{a} \quad (6.1.18)$$

and the vector  $\mathbf{a}$  satisfies the condition

$$(\mathbf{a}^T \mathbf{a} - 1)/2 = 0 \quad (6.1.19)$$

The normal to the disk periphery at contact point C, in the plane of the disk, is the vector  $\mathbf{a}'$ , so the tangent to the disk periphery at point C in the plane of the disk is

$$\mathbf{b}' = \begin{bmatrix} 0 \\ -a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{P}\mathbf{a} \\ \mathbf{P}\mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{P} \\ \mathbf{P} \end{bmatrix} \mathbf{a} \equiv \mathbf{b}'_a \mathbf{a} \quad (6.1.20)$$

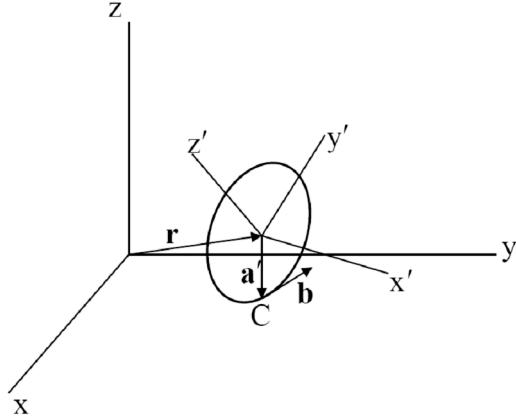


Figure 6.1.4. Disk Rolling Without Slip on x-y Plane

*Euler parameters* are used as generalized coordinates for orientation of the disk. Contact conditions are that point C is in the x-y plane; i.e.,  $\mathbf{u}_z^T (\mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}'_a \mathbf{a}) = 0$ , and the vector  $\mathbf{b} = \mathbf{A}(\mathbf{p})\mathbf{b}'$  is in the x-y plane; i.e.,  $\mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a \mathbf{a} = 0$ . Combining these conditions, Eq. (6.1.19), and the Euler parameter normalization condition yields the *holonomic constraints*

$$\Phi(\mathbf{q}) = \begin{bmatrix} \mathbf{u}_z^T (\mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}'_a \mathbf{a}) \\ \mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a \mathbf{a} \\ (\mathbf{p}^T \mathbf{p} - 1)/2 \\ (\mathbf{a}^T \mathbf{a} - 1)/2 \end{bmatrix} = \mathbf{0} \quad (6.1.21)$$

where generalized coordinates are  $\mathbf{q} = [\mathbf{r}^T \quad \mathbf{p}^T \quad \mathbf{a}^T]^T \in \mathbb{R}^9$ .

The velocity of point C on the periphery of the disk is

$$\begin{aligned} \mathbf{v}^C &= \dot{\mathbf{r}} + \mathbf{A}(\mathbf{p})\tilde{\omega}'\mathbf{a}'_a \mathbf{a} = \dot{\mathbf{r}} + \mathbf{A}(\mathbf{p})\mathbf{A}^T(\mathbf{p})\dot{\mathbf{A}}(\mathbf{p})\mathbf{a}'_a \mathbf{a} \\ &= \dot{\mathbf{r}} + \dot{\mathbf{A}}(\mathbf{p})\mathbf{a}'_a \mathbf{a} = \dot{\mathbf{r}} + (\mathbf{A}(\mathbf{p})\mathbf{a}'_a \hat{\mathbf{a}})_p \dot{\mathbf{p}} = \dot{\mathbf{r}} + \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \mathbf{a})\dot{\mathbf{p}} \end{aligned} \quad (6.1.22)$$

Conditions that no slip occurs between the disk and the x-y plane are that the horizontal components of  $\mathbf{v}^C$  are zero; i.e.,  $\mathbf{u}_x^T \mathbf{v}^C = 0$  and  $\mathbf{u}_y^T \mathbf{v}^C = 0$ . With Eq. (6.1.22), this is the *velocity constraint*

$$\mathbf{E}(\mathbf{q})\dot{\mathbf{q}} \equiv \begin{bmatrix} \mathbf{u}_x^T & \mathbf{u}_x^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \mathbf{a}) & 0 \\ \mathbf{u}_y^T & \mathbf{u}_y^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \mathbf{a}) & 0 \end{bmatrix} \dot{\mathbf{q}} \equiv \mathbf{e}(\mathbf{q}, t) = \mathbf{0} \quad (6.1.23)$$

With four holonomic and two differential constraints on nine generalized coordinates, the system has five configuration and three velocity degrees of freedom. As in this example, most mechanical systems that are subjected to differential constraints are also subjected to holonomic constraints. In such situations, it is impractical to determine whether the differential constraint is integrable, so it is treated as if it is nonholonomic.

### 6.1.3 Nonholonomic System Kinematics

*Nonholonomic multibody systems* are modeled using a set of n<sub>gc</sub> generalized coordinates  $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_{n_{gc}}]^T$  that are subject to n<sub>hc</sub> < n<sub>gc</sub> *holonomic constraints*,

$$\Phi(\mathbf{q}, t) = [\varphi_1(\mathbf{q}, t), \varphi_2(\mathbf{q}, t), \dots, \varphi_{n_{hc}}(\mathbf{q}, t)]^T = \mathbf{0} \quad (6.1.24)$$

which include Euler parameter normalization conditions for spatial bodies and n<sub>dc</sub>, 0 < n<sub>dc</sub> < n<sub>gc</sub> – n<sub>hc</sub>, *nonholonomic constraints*, or *differential constraints*,

$$\mathbf{E}(\mathbf{q}, t)\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{E}_1(\mathbf{q}, t) \\ \vdots \\ \mathbf{E}_{n_{dc}}(\mathbf{q}, t) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{e}(\mathbf{q}, t) \quad (6.1.25)$$

Functions that appear in the constraints of Eqs. (6.1.24) and (6.1.25) are assumed to have as many continuous derivatives as are needed in formulating equations of kinematics and dynamics.

Velocity and acceleration equations associated with Eq. (6.1.24) are

$$\begin{aligned} \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t) \equiv \mathbf{v}_h(\mathbf{q}, t) \\ \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} &= -\left(\Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}}\right)_q - 2\Phi_{tq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \\ &= -\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - 2\Phi_{tq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \equiv -\gamma_h(\mathbf{q}, \dot{\mathbf{q}}, t) \end{aligned} \quad (6.1.26)$$

where, with  $\chi \in \mathbb{R}^{n_{gc}}$ ,

$$\mathbf{P}2(\mathbf{q}, \chi) \equiv \left(\Phi_q(\mathbf{q}, t)\ddot{\chi}\right)_q \quad (6.1.27)$$

and the acceleration equation associated with Eq. (6.1.25) is

$$\begin{aligned} \mathbf{E}(\mathbf{q}, t)\ddot{\mathbf{q}} &= \mathbf{e}_q(\mathbf{q}, t)\dot{\mathbf{q}} + \mathbf{e}_t(\mathbf{q}, t) - \left(\mathbf{E}(\mathbf{q}, t)\hat{\mathbf{q}}\right)_q \dot{\mathbf{q}} - \mathbf{E}_t(\mathbf{q}, t)\dot{\mathbf{q}} \\ &= \mathbf{e}_q(\mathbf{q}, t)\dot{\mathbf{q}} + \mathbf{e}_t(\mathbf{q}, t) - \mathbf{E}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{E}_t(\mathbf{q}, t)\dot{\mathbf{q}} \equiv -\gamma_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{aligned} \quad (6.1.28)$$

where, with  $\kappa \in \mathbb{R}^{n_{gc}}$ ,

$$\mathbf{E}2(\mathbf{q}, \kappa) \equiv \left(\mathbf{E}(\mathbf{q}, t)\ddot{\kappa}\right)_q \quad (6.1.29)$$

Combined, the first of Eqs. (6.1.26) and Eq. (6.1.25) comprise *system velocity constraints*,

$$\mathbf{C}(\mathbf{q}, t)\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{v}_h(\mathbf{q}, t) \\ \mathbf{e}(\mathbf{q}, t) \end{bmatrix} \equiv \mathbf{v}(\mathbf{q}, t) \quad (6.1.30)$$

where

$$\mathbf{C}(\mathbf{q}, t) \equiv \begin{bmatrix} \Phi_q(\mathbf{q}, t) \\ \mathbf{E}(\mathbf{q}, t) \end{bmatrix} \quad (6.1.31)$$

The  $n_{hc} \times n_{gc}$  holonomic constraint Jacobian  $\Phi_q(\mathbf{q}, t)$  and the  $n_{dc} \times n_{gc}$  coefficient matrix  $\mathbf{E}(\mathbf{q}, t)$  of Eq. (6.1.25) are each required to be of full row rank. Furthermore, their rows must be mutually independent; i.e., the  $(n_{hc} + n_{dc}) \times n_{gc}$  matrix  $\mathbf{C}(\mathbf{q}, t)$  must have full rank for  $(\mathbf{q}, t)$  that satisfy Eqs. (6.1.24) and (6.1.25). Similarly, the second of Eqs. (6.1.26) and Eq. (6.1.28) comprise *system acceleration constraints*,

$$\mathbf{C}(\mathbf{q}, t)\ddot{\mathbf{q}} = \begin{bmatrix} -\gamma_h(\mathbf{q}, \dot{\mathbf{q}}, t) \\ -\gamma_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} \equiv -\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (6.1.32)$$

*Admissible displacements*  $d\mathbf{q}$  over a time increment  $dt$  satisfy Eq. (6.1.30), in differential form; i.e.,

$$\mathbf{C}(\mathbf{q}, t)d\mathbf{q} = \mathbf{v}(\mathbf{q}, t)dt$$

As in holonomic systems, variational equations of motion cannot be stated in terms of admissible displacements that occur over a specified time increment. As in the treatment of holonomic constraints, kinematically admissible *virtual displacements*  $\mathbf{q}$ , with time held fixed, are defined to satisfy

$$\mathbf{C}(\mathbf{q}, t)\mathbf{q} = \mathbf{0} \quad (6.1.33)$$

and are essential in formulation of variational equations of motion.

As in Section 5.2, a *configuration tangent space* formulation is defined that uses columns of  $\Phi_q^T(\mathbf{q}^0, t^0)$  and *null vectors* of  $\Phi_q(\mathbf{q}^0, t^0)$ , where  $\Phi(\mathbf{q}^0, t^0) = \mathbf{0}$ , to span the *regular configuration space*

$$\tilde{\mathbf{C}}(t) = \left\{ \mathbf{q} : \Phi(\mathbf{q}, t) = \mathbf{0} \text{ and } \text{rank}(\Phi_q(\mathbf{q}, t)) = n_{hc} \right\} \quad (6.1.34)$$

In the nonholonomically constrained case, with differential constraints of Eq. (6.1.25), the parameterization of configuration space cannot represent system velocities. To represent velocities, a *velocity tangent space* is defined that uses columns of  $\mathbf{C}^T(\mathbf{q}^0, t^0)$  and null vectors of  $\mathbf{C}(\mathbf{q}^0, t^0)$  to span the *regular velocity space*

$$\tilde{\mathbf{D}}(\mathbf{q}, t) = \left\{ \dot{\mathbf{q}} : \mathbf{q} \in \tilde{\mathbf{C}}(t), \mathbf{C}(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q}, t), \text{and } \text{rank}(\mathbf{C}(\mathbf{q}, t)) = n_{hc} + n_{dc} \right\} \quad (6.1.35)$$

#### 6.1.4 Configuration Tangent Space Parameterization

Parameterization of the regular configuration space of Eq. (6.1.34) is identical to that presented in Section 5.2 and is only summarized here. As in Section 5.2, a basis for the regular configuration tangent space is provided by the columns of matrices  $\mathbf{V}$  and  $\mathbf{U}$  that are defined by

$$\begin{aligned}
\mathbf{U} &= \Phi_q(\mathbf{q}^0, t^0)^T \\
\Phi_q(\mathbf{q}^0, t^0) \mathbf{V} &= \mathbf{0} \\
\mathbf{V}^T \mathbf{V} &= \mathbf{I} \\
\mathbf{V}^T \mathbf{U} &= \mathbf{0}
\end{aligned} \tag{6.1.36}$$

That is, any generalized coordinate vector  $\mathbf{q} \in \mathbb{R}^{n_{gc}}$  can be uniquely represented as

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u} \tag{6.1.37}$$

The vectors  $\mathbf{v}$  and  $\mathbf{u}$  are *configuration space generalized coordinates* that are equivalent to  $\mathbf{q}$  in a neighborhood of  $\mathbf{q}^0$ . Multiplying Eq. (6.1.37) by  $\mathbf{V}$  and  $\mathbf{U}$ ,

$$\begin{aligned}
\mathbf{v} &= \mathbf{V}^T(\mathbf{q} - \mathbf{q}^0) \\
\mathbf{u} &= -(\mathbf{U}^T \mathbf{U})^{-1}(\mathbf{q} - \mathbf{q}^0)
\end{aligned} \tag{6.1.38}$$

which are the inverse of Eq. (6.1.37). Thus,

$$\begin{aligned}
\mathbf{v}^0 &= 0 \\
\mathbf{u}^0 &= 0
\end{aligned} \tag{6.1.39}$$

As in Section 5.2, for a given  $\mathbf{v}$ ,  $\mathbf{u}$  is uniquely determined in a neighborhood  $V_0$  of  $\mathbf{v}^0 = \mathbf{0}$  as the solution of

$$\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}, t) = \mathbf{0} \tag{6.1.40}$$

denoted

$$\mathbf{u} = \mathbf{h}(\mathbf{v}, t) \tag{6.1.41}$$

and evaluated using Newton-Raphson iteration for  $\mathbf{u}$ ,

$$\begin{aligned}
\Delta \mathbf{u}^i &= \mathbf{B}\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i, t) & i=1, 2, \dots \text{ until } \|\Phi\| \leq \text{utol} \\
\mathbf{u}^{i+1} &= \mathbf{u}^i + \Delta \mathbf{u}^i
\end{aligned} \tag{6.1.42}$$

where the matrix  $\mathbf{B}$  is defined as

$$\mathbf{B}(\mathbf{q}, t) = (\Phi_q(\mathbf{q}, t) \mathbf{U})^{-1} \tag{6.1.43}$$

and evaluated using Newton-Raphson iteration,

$$\begin{aligned}
\Delta \mathbf{B}^i &= -\mathbf{B}^i \Phi_q(\mathbf{q}, t) \mathbf{U} \mathbf{B}^i - \mathbf{B}^i & i=1, 2, \dots \text{ until } \|\Phi_q(\mathbf{q}, t) \mathbf{U} \mathbf{B}^i - \mathbf{I}\| \leq \text{Btol} \\
\mathbf{B}^{i+1} &= \mathbf{B}^i + \Delta \mathbf{B}^i
\end{aligned} \tag{6.1.44}$$

with  $\mathbf{B}^0 = (\Phi_q(\mathbf{q}^0, t^0) \Phi_q^T(\mathbf{q}^0, t^0))^{-1}$ . The result is

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) \tag{6.1.45}$$

which satisfies the holonomic constraint equation of Eq. (6.1.24), for arbitrary  $\mathbf{v} \in V_0$ .

Since  $\mathbf{V}$  and  $\mathbf{U}$  are constant,

$$\dot{\mathbf{q}} = \mathbf{V}\dot{\mathbf{v}} - \mathbf{U}\dot{\mathbf{h}} \quad (6.1.46)$$

and using the fact that  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T\mathbf{U} = \mathbf{0}$ ,

$$\dot{\mathbf{v}} = \mathbf{V}^T\dot{\mathbf{q}} \quad (6.1.47)$$

### 6.1.5 Velocity Tangent Space Parameterization

The *regular velocity tangent space* for the constraint of Eq. (6.1.30) at  $(\mathbf{q}^0, t^0)$  is spanned by the orthonormal columns of a matrix  $\mathbf{W}$  that defines the null space of  $\mathbf{C}^0 \equiv \mathbf{C}(\mathbf{q}^0, t^0)$ ,

$$\begin{aligned} \mathbf{C}^0\mathbf{W} &= \mathbf{0} \\ \mathbf{W}^T\mathbf{W} &= \mathbf{I} \end{aligned} \quad (6.1.48)$$

which may be computed using *singular value decomposition* of  $\mathbf{C}^{0T}$  (Strang, 1980). Expanding Eq. (6.1.48), using the definition of Eq. **Error! Reference source not found.**,  $\Phi_q^0\mathbf{W} = \mathbf{0}$  and  $\mathbf{E}^0\mathbf{W} = \mathbf{0}$ . Thus, the columns of  $\mathbf{W}$  are a subset of the space spanned by columns of the matrix  $\mathbf{V}$ ; i.e.,  $\mathbf{W} = \mathbf{VR}$ , where  $\mathbf{R}$  is an  $(ngc - nh) \times (ngc - nh - nd)$  matrix such that  $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ . Thus,  $\mathbf{E}^0\mathbf{VR} = \mathbf{0}$  and  $\mathbf{R}$  may be determined using singular value decomposition of  $(\mathbf{E}^0\mathbf{V})^T$ , which is more efficient than singular value decomposition of  $\mathbf{C}^{0T}$ . To confirm that columns of  $\mathbf{W}$  are orthogonal, note that  $\mathbf{W}^T\mathbf{W} = \mathbf{R}^T\mathbf{V}^T\mathbf{VR} = \mathbf{R}^T\mathbf{R} = \mathbf{I}$ .

Since the columns of  $\mathbf{W}$  are orthogonal to the linearly independent columns of  $\mathbf{C}^{0T} = \mathbf{C}(\mathbf{q}^0, t^0)^T$ , the columns of  $\mathbf{W}$  and  $\mathbf{C}^{0T}$  span the regular velocity space  $\tilde{D} \subset \mathbb{R}^{ngc}$  of Eq. (6.1.35). Defining

$$\mathbf{X} \equiv \mathbf{C}^{0T} \quad (6.1.49)$$

by Eq. (6.1.48),

$$\mathbf{X}^T\mathbf{W} = \mathbf{0} \quad (6.1.50)$$

so every vector  $\dot{\mathbf{q}}$  in  $\mathbb{R}^{ngc}$  may be uniquely written in the form

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}^0 + \mathbf{Ww} - \mathbf{Xx} \quad (6.1.51)$$

Where  $\mathbf{C}^0\dot{\mathbf{q}}^0 = \mathbf{v}(\mathbf{q}^0, t^0)$ ,  $\mathbf{w}$  and  $\mathbf{x}$  are *velocity space generalized coordinates*, and the minus sign is chosen to indicate projection onto the regular velocity space of Eq. (6.1.35). Multiplying Eq. (6.1.51) by  $\mathbf{W}^T$  and  $\mathbf{X}^T$  and using Eqs. (6.1.48) and (6.1.50),

$$\begin{aligned} \mathbf{w} &= \mathbf{W}^T(\dot{\mathbf{q}} - \dot{\mathbf{q}}^0) \\ \mathbf{x} &= -(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}(\dot{\mathbf{q}} - \dot{\mathbf{q}}^0) \end{aligned} \quad (6.1.52)$$

Thus,

$$\begin{aligned} \mathbf{w}^0 &= \mathbf{0} \\ \mathbf{x}^0 &= \mathbf{0} \end{aligned} \quad (6.1.53)$$

Since  $\mathbf{C}^0$  has full rank,  $\mathbf{C}^0\mathbf{C}^{0T} = (\mathbf{H}^0)^{-1}$  is nonsingular. Since  $\mathbf{C}(\mathbf{q}, t)$  is continuous,  $\mathbf{C}(\mathbf{q}, t)\mathbf{C}^{0T} = \mathbf{C}(\mathbf{q}, t)\mathbf{X}$  is nonsingular in a neighborhood of  $(\mathbf{q}^0, t^0)$ . Thus, define

$$\mathbf{H}(\mathbf{q}, t) = (\mathbf{C}(\mathbf{q}, t)\mathbf{X})^{-1} \quad (6.1.54)$$

in a neighborhood of  $(\mathbf{q}^0, t^0)$ . Since  $\mathbf{H}(\mathbf{q}, t)$  satisfies

$$\mathbf{C}(\mathbf{q}, t)\mathbf{X}\mathbf{H}(\mathbf{q}, t) - \mathbf{I} = \mathbf{0} \quad (6.1.55)$$

it can be evaluated using the iterative process of Eq. (6.1.44); i.e.,

$$\begin{aligned} \Delta\mathbf{H}^i &= -\mathbf{H}^i\mathbf{C}(\mathbf{q}, t)\mathbf{X}\mathbf{H}^i - \mathbf{H}^i & i = 1, 2, \dots \text{ until } \|\mathbf{C}\mathbf{X}^T\mathbf{H} - \mathbf{I}\| \leq \text{Htol} \\ \mathbf{H}^{i+1} &= \mathbf{H}^i + \Delta\mathbf{H}^i \end{aligned} \quad (6.1.56)$$

where  $\mathbf{H}(\mathbf{q}^0, t^0) = (\mathbf{C}(\mathbf{q}^0, t^0)\mathbf{X})^{-1}$ .

Substituting Eq. (6.1.51) into Eq. (6.1.30),  $\mathbf{C}\dot{\mathbf{q}}^0 + \mathbf{C}\mathbf{W}\mathbf{w} - \mathbf{C}\mathbf{X}\mathbf{x} = \mathbf{v}$ . Using Eq. (6.1.54),

$$\mathbf{x} = \mathbf{H}\mathbf{C}\mathbf{W}\mathbf{w} + \mathbf{H}\mathbf{C}\dot{\mathbf{q}}^0 - \mathbf{H}\mathbf{v} \quad (6.1.57)$$

Substituting this result into Eq. (6.1.51),

$$\begin{aligned} \dot{\mathbf{q}} &= \dot{\mathbf{q}}^0 + \mathbf{W}\mathbf{w} - \mathbf{X}(\mathbf{H}\mathbf{C}\mathbf{W}\mathbf{w} + \mathbf{H}\mathbf{C}\dot{\mathbf{q}}^0 - \mathbf{H}\mathbf{v}) \\ &= \mathbf{K}\mathbf{w} + (\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{C})\dot{\mathbf{q}}^0 + \mathbf{X}\mathbf{H}\mathbf{v} \end{aligned} \quad (6.1.58)$$

where

$$\mathbf{K} \equiv (\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{C})\mathbf{W} \quad (6.1.59)$$

Differentiating Eq. (6.1.51) with respect to time, since  $\mathbf{W}$  and  $\mathbf{X}$  are constant,

$$\ddot{\mathbf{q}} = \mathbf{W}\dot{\mathbf{w}} - \mathbf{X}\dot{\mathbf{x}} \quad (6.1.60)$$

Substituting this into Eq. (6.1.32),  $\mathbf{C}\mathbf{W}\dot{\mathbf{w}} - \mathbf{C}\mathbf{X}\dot{\mathbf{x}} = -\gamma$ . Using Eq. (6.1.54),

$$\dot{\mathbf{x}} = \mathbf{H}\mathbf{C}\mathbf{W}\dot{\mathbf{w}} + \mathbf{H}\gamma \quad (6.1.61)$$

Substituting this result into Eq. (6.1.60),

$$\ddot{\mathbf{q}} = \mathbf{W}\dot{\mathbf{w}} - \mathbf{X}(\mathbf{H}\mathbf{C}\mathbf{W}\dot{\mathbf{w}} + \mathbf{H}\gamma) = \mathbf{K}\dot{\mathbf{w}} - \mathbf{X}\mathbf{H}\gamma \quad (6.1.62)$$

By construction,  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  of Eqs. (6.1.58) and (6.1.62) satisfy the system velocity and acceleration constraints of Eqs. (6.1.30) and (6.1.32), for arbitrary  $\mathbf{w}$  and  $\dot{\mathbf{w}}$ . As in the case of holonomically constrained systems, this assures satisfaction of velocity and acceleration constraints, without problems of *constraint drift*. The vector  $\mathbf{w}$  is thus comprised of *independent velocity coordinates*.

## 6.1.6 Kinematic Differential Equation

Equation (6.1.47), taken with Eq. (6.1.58), yields a *kinematic differential equation*

$$\dot{\mathbf{v}} = \mathbf{V}^T\dot{\mathbf{q}} = \mathbf{V}^T\mathbf{K}\mathbf{w} + \mathbf{V}^T(\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{C})\dot{\mathbf{q}}^0 + \mathbf{V}^T\mathbf{X}\mathbf{H}\mathbf{v} \quad (6.1.63)$$

that couples configuration and velocity spaces. Displaying arguments of functions involved makes clear that this is a first order differential equation in  $\mathbf{v}$  and  $\mathbf{w}$ ,

$$\dot{\mathbf{v}} = \mathbf{V}^T \mathbf{K}(\mathbf{v}, t) \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{XH}(\mathbf{q}(t), t)) \mathbf{C}(\mathbf{q}(t), t) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{XH}(\mathbf{q}(t), t) \mathbf{v}(\mathbf{q}(t), t) \quad (6.1.64)$$

### 6.1.7 Kinematically Admissible Virtual Displacements

Since columns of  $\mathbf{W}$  and  $\mathbf{X}$  span  $\mathbb{R}^{ngc}$ , any virtual displacement  $\mathbf{q}$  can be written as

$$\mathbf{q} = \mathbf{W}\alpha - \mathbf{X}\beta \quad (6.1.65)$$

From Eq. (6.1.33), a kinematically admissible  $\mathbf{q}$  must satisfy

$$\mathbf{C} \cdot \mathbf{q} = \mathbf{CW}\alpha - \mathbf{CX}\beta = \mathbf{0} \quad (6.1.66)$$

Using Eq. (6.1.54),  $\beta = \mathbf{HCW}\alpha$ . Thus, kinematically admissible virtual displacements are of the form

$$\mathbf{q} = \mathbf{W}\alpha - \mathbf{XHCW}\alpha = (\mathbf{I} - \mathbf{XHC})\mathbf{W}\alpha = \mathbf{K}\alpha \quad (6.1.67)$$

for arbitrary  $\alpha$ .

In contrast to holonomic constraints encountered in Sections 3.1 through 3.3, nonholonomic constraints on and between bodies that are written in terms of velocities of the bodies tend to be ad-hoc in nature. Three examples illustrate the form of nonholonomic constraint equations, which are linear in velocities and nonlinear in configuration generalized coordinates. Two of the examples include both holonomic and nonholonomic constraints.

Tangent space configuration coordinates of Section 5.2 are augmented with tangent space velocity coordinates, which is required since the space of kinematically admissible velocities is of reduced dimension, relative to the configuration space. The two spaces are coupled by a kinematic differential equation. Independent generalized coordinates at the configuration and velocity levels are defined, guaranteeing that all forms of constraint equations are satisfied.

## Key Formulas

$$\begin{aligned}
& \Phi(\mathbf{q}, t) = \mathbf{0} & \mathbf{E}(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{e}(\mathbf{q}, t) & (6.1.24) \quad (6.1.25) \\
& \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{v}_h(\mathbf{q}, t) & \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} = -\gamma_h(\mathbf{q}, \dot{\mathbf{q}}, t) & (6.1.26) \\
& -P2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - 2\Phi_{tq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \equiv -\gamma_h(\mathbf{q}, \dot{\mathbf{q}}, t) & & (6.1.26) \\
& \mathbf{E}(\mathbf{q}, t)\ddot{\mathbf{q}} = -\gamma_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) & \mathbf{C}(\mathbf{q}, t) \equiv \begin{bmatrix} \Phi_q(\mathbf{q}, t) \\ \mathbf{E}(\mathbf{q}, t) \end{bmatrix} & (6.1.28) \quad (6.1.31) \\
& \mathbf{e}_q(\mathbf{q}, t)\dot{\mathbf{q}} + \mathbf{e}_t(\mathbf{q}, t) - E2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - E_t(\mathbf{q}, t)\dot{\mathbf{q}} \equiv -\gamma_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) & & (6.1.28) \\
& P2(\mathbf{q}, \chi) = (\Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}})_q & E2(\mathbf{q}, \kappa) = (E(\mathbf{q}, t)\ddot{\mathbf{q}})_q & (6.1.27) \quad (6.1.29) \\
& C(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q}, t) & C(\mathbf{q}, t)\ddot{\mathbf{q}} = -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) & (6.1.30) \quad (6.1.32) \\
& \Phi_q(\mathbf{q}^0, t^0)\mathbf{V} = \mathbf{0} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I} \quad \mathbf{U} = \Phi_q^T(\mathbf{q}^0, t^0) & & (6.1.36) \\
& \mathbf{u} = \mathbf{h}(\mathbf{v}, t) & \Delta\mathbf{u}^i = \mathbf{B}\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i, t) & (6.1.41) \quad (6.1.42) \\
& & \mathbf{u}^{i+1} = \mathbf{u}^i + \Delta\mathbf{u}^i & \\
& \mathbf{B}(\mathbf{q}, t) = (\Phi_q(\mathbf{q}, t)\mathbf{U})^{-1} & \Delta\mathbf{B}^i = -\mathbf{B}^i\Phi_q(\mathbf{q}, t)\mathbf{U}\mathbf{B}^i - \mathbf{B}^i & (6.1.43) \quad (6.1.44) \\
& \mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) & & (6.1.45) \\
& \mathbf{C}^0\mathbf{W} = \mathbf{0} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I} \quad \mathbf{X} \equiv \mathbf{C}^{0T} & & (6.1.48) \quad (6.1.49) \\
& \mathbf{H}(\mathbf{q}, t) = (\mathbf{C}(\mathbf{q}, t)\mathbf{X})^{-1} & \Delta\mathbf{H}^i = -\mathbf{H}^i\mathbf{C}(\mathbf{q}, t)\mathbf{X}\mathbf{H}^i - \mathbf{H}^i & (6.1.54) \quad (6.1.56) \\
& & \mathbf{H}^{i+1} = \mathbf{H}^i + \Delta\mathbf{H}^i & \\
& \dot{\mathbf{q}} = \mathbf{K}\mathbf{w} + (\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{C})\dot{\mathbf{q}}^0 + \mathbf{X}\mathbf{H}\mathbf{v} & \mathbf{K} \equiv (\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{C})\mathbf{W} & (6.1.58) \quad (6.1.59) \\
& \ddot{\mathbf{q}} = \mathbf{K}\dot{\mathbf{w}} - \mathbf{X}\mathbf{H}\gamma & \dot{\mathbf{v}} = \mathbf{V}^T\mathbf{K}\mathbf{w} + \mathbf{V}^T(\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{C})\dot{\mathbf{q}}^0 + \mathbf{V}^T\mathbf{X}\mathbf{H}\mathbf{v} & (6.1.62) \quad (6.1.63)
\end{aligned}$$

## 6.2 Tangent Space ODE for Dynamics of Nonholonomic Systems

The tangent space nonholonomic kinematics formulation of Section 6.1 is next combined with the variational equations of dynamics of Section 4.6 to create ODE of dynamics.

### 6.2.1 ODE of Motion

For systems comprised of particles, planar bodies, and spatial bodies, holonomic constraints of Eq. (6.1.24) must hold, including Euler parameter normalization conditions for spatial bodies. The velocity form of holonomic constraints of the first of Eqs. (6.1.26) must also hold. In addition, *nonholonomic constraints*, or *differential constraints*, of Eq. (6.1.25), must hold. Combined, they form the system velocity constraints of Eq. (6.1.30) and the virtual displacement equation of Eq. (6.1.33). Finally, the *system acceleration constraints* of Eq. (6.1.32) must hold.

From Eq. (4.6.16), the variational equation of motion for the system is

$$\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})) = 0 \quad (6.2.1)$$

which must hold for all  $\mathbf{q}$  that satisfy Eq. (6.1.33). From Eq. (6.1.42), all virtual displacements that satisfy Eq. (6.1.67) are of the form

$$\mathbf{q} = \mathbf{K}\alpha \quad (6.2.2)$$

for arbitrary  $\alpha$ , where  $\mathbf{K} = (\mathbf{I} - \mathbf{XH}\mathbf{C})\mathbf{W}$ . Substituting from Eqs. (6.2.2) and (6.1.62) into Eq. (6.2.1),

$$\mathbf{a}^T \mathbf{K}(\mathbf{q}, t)^T (\mathbf{M}(\mathbf{q})\mathbf{K}(\mathbf{q}, t)\dot{\mathbf{w}} - \mathbf{M}(\mathbf{q})\mathbf{XH}(\mathbf{q}, t)\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})) = 0$$

which must hold for arbitrary  $\alpha$ . Thus,

$$\mathbf{K}(\mathbf{q}, t)^T \mathbf{M}(\mathbf{q})\mathbf{K}(\mathbf{q}, t)\dot{\mathbf{w}} = \mathbf{K}(\mathbf{q}, t)^T (\mathbf{M}(\mathbf{q})\mathbf{XH}(\mathbf{q}, t)\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})) \quad (6.2.3)$$

Displaying arguments in Eqs. (6.1.45) and (6.1.58),

$$\begin{aligned} \mathbf{q} &= \mathbf{q}(\mathbf{v}, t) = \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) \\ \dot{\mathbf{q}} &= \dot{\mathbf{q}}(\mathbf{w}, \mathbf{v}, t) = \mathbf{K}(\mathbf{v}, t)\mathbf{w} + (\mathbf{I} - \mathbf{XH}(\mathbf{v}, t)\mathbf{C}(\mathbf{v}, t))\dot{\mathbf{q}}^0 + \mathbf{XH}(\mathbf{v}, t)\mathbf{v}(\mathbf{v}, t) \end{aligned} \quad (6.2.4)$$

so Eq. (6.2.3) is a first ODE in  $\mathbf{w}$  and  $\mathbf{v}$ , called the *nonholonomic system kinetic ODE*.

Suppressing arguments of functions for simplicity of notation, the *first order kinematic ODE* of Eq. (6.1.63) and the first order kinetic ODE of Eq. (6.2.3) comprise the *system first order ODE of motion*, or *system ODE*, for the nonholonomic system,

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{V}^T \mathbf{K}\mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{XH}\mathbf{C})\dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{XH}\mathbf{v} \\ \mathbf{K}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} &= \mathbf{K}^T (\mathbf{M} \mathbf{XH} \gamma + \mathbf{Q}^A + \mathbf{S}) \end{aligned} \quad (6.2.5)$$

With initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  that satisfy Eqs. (6.1.24) and (6.1.30) at  $t^0$ , Eqs. (6.1.39) and (6.1.53) yield initial conditions for the ODE of Eq. (6.2.5),

$$\begin{aligned}\mathbf{v}^0 &= \mathbf{0} \\ \mathbf{w}^0 &= \mathbf{0}\end{aligned}\tag{6.2.6}$$

To see that the matrix  $\mathbf{K}^T \mathbf{M} \mathbf{K}$  is nonsingular, note from Eq. (6.1.59) at  $t^0$  that  $\mathbf{K}^0 = \mathbf{W}$ . Since columns of  $\mathbf{W}$  span the null space of the system velocity constraint Jacobian,  $\mathbf{K}^{0T} \mathbf{M}^0 \mathbf{K}^0 = \mathbf{W}^T \mathbf{M}^0 \mathbf{W}$  must be positive definite, hence nonsingular. Since all functions involved are continuous in their arguments,  $\mathbf{K}^T \mathbf{M} \mathbf{K}$  is *positive definite*, hence nonsingular, in a neighborhood of  $(\mathbf{v}^0, t^0)$ . Thus, the ODE of Eq. (6.2.5) can be written in the classical form

$$\begin{aligned}\dot{\mathbf{v}} &= \mathbf{V}^T \mathbf{K} \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \dot{\mathbf{w}} &= (\mathbf{K}^T \mathbf{M} \mathbf{K})^{-1} \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \boldsymbol{\gamma} + \mathbf{Q}^A + \mathbf{S})\end{aligned}\tag{6.2.7}$$

The *initial-value problem* of Eqs. (6.2.7) and (6.2.6) thus has a unique solution in a neighborhood of  $(\mathbf{v}^0, \mathbf{w}^0, t^0)$  that depends continuously on problem data (see Section 4.7.3); i.e., the initial-value problem is *well-posed*.

To evaluate terms in Eqs. (6.2.5) that are needed for *explicit numerical integration*, the following derivatives involving holonomic constraints are required:

$$\begin{aligned}\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) &= (\Phi_q(\mathbf{q}, t) \ddot{\boldsymbol{\chi}})_q \\ \boldsymbol{\gamma}_h(\mathbf{q}, \dot{\mathbf{q}}, t) &= \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + 2\Phi_{tq}(\mathbf{q}, t) \dot{\mathbf{q}} + \Phi_{tt}(\mathbf{q}, t) \\ \Phi_t &= \Phi_t(\mathbf{q}, t) \\ \Phi_{tt} &= \Phi_{tt}(\mathbf{q}, t) \\ \Phi_{tq} &= \Phi_{tq}(\mathbf{q}, t)\end{aligned}\tag{6.2.8}$$

Similarly, the following derivatives involving nonholonomic constraints are required:

$$\begin{aligned}\mathbf{E}2(\mathbf{q}, \boldsymbol{\kappa}) &= (\mathbf{E}(\mathbf{q}, t) \ddot{\boldsymbol{\kappa}})_q \\ \boldsymbol{\gamma}_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) &= -\mathbf{e}_q(\mathbf{q}, t) \dot{\mathbf{q}} - \mathbf{e}_t(\mathbf{q}, t) + \mathbf{E}2(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{E}_t(\mathbf{q}, t) \dot{\mathbf{q}} \\ \mathbf{E}_t &= \mathbf{E}_t(\mathbf{q}, t) \\ \mathbf{e}_q &= \mathbf{e}_q(\mathbf{q}, t) \\ \mathbf{e}_t &= \mathbf{e}_t(\mathbf{q}, t)\end{aligned}\tag{6.2.9}$$

With these terms,  $\mathbf{C}$ ,  $\mathbf{v}$ , and  $\boldsymbol{\gamma}$  are evaluated in Eqs. (6.1.31), (6.1.30), and (6.1.32) and  $\mathbf{V}$ ,  $\mathbf{U}$ ,  $\mathbf{W}$ ,  $\mathbf{X}$ ,  $\mathbf{B}$ ,  $\mathbf{H}$ , and  $\mathbf{K}$  are determined in Eqs. (6.1.36), (6.1.48), (6.1.49), (6.1.43), (6.1.54), and (6.1.59). Finally, problem dependent terms  $\mathbf{M}(\mathbf{q})$ ,  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$ , and  $\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)$  of Eq. (4.6.16) are evaluated.

## 6.2.2 Explicit Numerical Integration

**Explicit integration of the ODE initial-value problem** of Eqs. (6.2.5) and (6.2.6), using Runge-Kutta methods, is as follows:

- (1) Define initial configuration  $\mathbf{q}^0$  at  $t^0$  that satisfies Eq. (6.1.24). Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{U}$ ,  $\mathbf{V}$ , and  $\mathbf{B}$ . Similarly, evaluate  $\mathbf{C}(\mathbf{q}^0, t^0)$  of Eq. (6.1.31) and matrices  $\mathbf{W}$ ,  $\mathbf{X}$ , and  $\mathbf{K}$ . Define initial velocity  $\dot{\mathbf{q}}^0$  that satisfies Eq. (6.1.30) and initial conditions on  $\mathbf{v}$  and  $\mathbf{w}$  from Eq. (6.2.6) for integration of Eq. (6.2.5).
- (2) At time step  $t_i$ , solve Eq. (6.2.5) to determine  $\dot{\mathbf{v}}_i$  and  $\dot{\mathbf{w}}_i$ . Apply an explicit numerical integrator to proceed on a time grid with step size  $h$  to obtain  $\mathbf{v}_{i+1}$  and  $\mathbf{w}_{i+1}$ . Use Eqs. (6.1.45), (6.1.58), and (6.1.62) to evaluate  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  on the time grid.
- (3) Monitor the condition number of the reduced mass matrix  $\mathbf{K}^T \mathbf{M} \mathbf{K}$ , the norms of  $\mathbf{v}$  and  $\mathbf{w}$ , and the number of iterations required to evaluate  $\mathbf{u}$ ,  $\mathbf{B}$ , and  $\mathbf{H}$ . If tolerances are exceeded, define a new time  $\bar{t}^0$  and associated  $\bar{\mathbf{q}}^0$  and  $\bar{\dot{\mathbf{q}}}^0$ . Repeat calculations in Step (1) to define a new *parameterization* and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\mathbf{w}}^0$ . This process follows the trajectory shown in Fig. 5.2.2, moving smoothly across the constraint set.
- (4) Continue the process until the final time  $t_f$  is reached, or a singular configuration associated with a faulty design or model occurs.

### 6.2.3 Derivatives for Implicit Numerical Integration

To use an implicit numerical integration method in solving Eq. (6.2.5), derivatives of all terms appearing with respect to  $\mathbf{v}$  and  $\mathbf{w}$  are required. For the reader who is satisfied with the use of explicit numerical integration methods, the somewhat intricate manipulations in the remainder of this section can be bypassed.

Equation (6.2.5) may be written in residual form as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{v}} - \mathbf{V}^T \mathbf{K} \mathbf{w} - \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 - \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \mathbf{K}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} - \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S}) \end{bmatrix} = \mathbf{0} \quad (6.2.10)$$

Evaluation of the Jacobian of  $\mathbf{R}$  with respect to  $\mathbf{v}$ ,  $\mathbf{w}$ ,  $\dot{\mathbf{v}}$ , and  $\dot{\mathbf{w}}$  is required for implicit integration. The easy part is

$$\mathbf{R}_{\dot{\mathbf{v}}, \dot{\mathbf{w}}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^T \mathbf{M} \mathbf{K} \end{bmatrix} \quad (6.2.11)$$

Since all other terms depend on  $\mathbf{q}(\mathbf{v}, t)$  and  $\dot{\mathbf{q}}(\mathbf{w}, \mathbf{v}, t)$ , as defined in Eq. (6.2.4), the *chain rule of differentiation* is required. From Eqs. (6.1.45) and (6.1.58),

$$\begin{aligned} \mathbf{q}_v &= \mathbf{V} - \mathbf{U} \mathbf{h}_v \\ \mathbf{q}_w &= \mathbf{0} \\ \dot{\mathbf{q}}_v &= \left( (\mathbf{K} \mathbf{w})_q - \mathbf{X} (\mathbf{H} \ddot{\mathbf{C}} \dot{\mathbf{q}}^0)_q - \mathbf{X} \mathbf{H} (\mathbf{C} \dot{\mathbf{q}}^0)_q + \mathbf{X} (\mathbf{H} \dot{\mathbf{v}})_q + \mathbf{X} \mathbf{H} \mathbf{v}_q \right) \mathbf{q}_v \\ \dot{\mathbf{q}}_w &= \mathbf{K} \end{aligned} \quad (6.2.12)$$

For the residuals of Eq. (6.2.10),

$$\begin{aligned}
\mathbf{R}_{1v} &= \left( -\mathbf{V}^T (\mathbf{K}\ddot{\mathbf{w}})_q + \mathbf{V}^T \mathbf{X} \left( \mathbf{H} \ddot{\mathbf{C}} \dot{\mathbf{q}}^0 \right)_q + \mathbf{V}^T \mathbf{X} \mathbf{H} \left( \mathbf{C} \ddot{\mathbf{q}}^0 \right)_q - \mathbf{V}^T \mathbf{X} (\mathbf{H} \ddot{\mathbf{v}})_q - \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v}_q \right) \dot{\mathbf{q}}_v \\
\mathbf{R}_{1w} &= -\mathbf{V}^T \mathbf{K} \\
\mathbf{R}_{2v} &= \left( \begin{array}{l} 2\mathbf{K}^T \mathbf{M} (\mathbf{K}\ddot{\mathbf{w}})_q + \mathbf{K}^T (\mathbf{M} \ddot{\mathbf{K}} \dot{\mathbf{w}})_q - \left( \widehat{\mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S})} \right)_q \\ -\mathbf{K}^T \left( (\mathbf{M} \mathbf{X} \ddot{\mathbf{H}} \dot{\gamma})_q + \mathbf{M} \mathbf{X} (\mathbf{H} \dot{\gamma})_q + \mathbf{M} \mathbf{X} \mathbf{H} \gamma_q + \mathbf{Q}_q^A + \mathbf{S}_q \right) \end{array} \right) \dot{\mathbf{q}}_v \\
&\quad - \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma_q + \mathbf{Q}_q^A + \mathbf{S}_q) \ddot{\mathbf{q}}_v \\
\mathbf{R}_{2w} &= -\mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma_q + \mathbf{Q}_q^A + \mathbf{S}_q) \ddot{\mathbf{q}}_w
\end{aligned} \tag{6.2.13}$$

where the fact that  $\mathbf{M}$  is symmetric has been used.

Since  $\dot{\mathbf{q}}$  of Eq. (6.1.45) satisfies Eq. (6.1.24),  $\Phi(\mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t), t) = \mathbf{0}$ . Taking the derivative with respect to  $\mathbf{v}$ ,  $\Phi_q \mathbf{V} - \Phi_q \mathbf{U}\mathbf{h}_v = \mathbf{0}$ . Using Eq. (6.1.43),

$$\mathbf{h}_v = \mathbf{B}\Phi_q \mathbf{V} \tag{6.2.14}$$

From Eq. (6.1.31),

$$(\mathbf{C}\ddot{\mathbf{a}})_q = \begin{bmatrix} (\Phi_q \ddot{\mathbf{a}})_q \\ (\mathbf{E}\ddot{\mathbf{a}})_q \end{bmatrix} \equiv \begin{bmatrix} \mathbf{P2}(\mathbf{q}, \mathbf{a}) \\ \mathbf{E2}(\mathbf{q}, \mathbf{a}) \end{bmatrix} \tag{6.2.15}$$

From Eq. (6.1.54),  $\mathbf{C}\mathbf{X}\mathbf{H} = \mathbf{I}$  and with a constant vector  $\mathbf{b}$ ,  $\mathbf{C}\mathbf{X}(\mathbf{H}\ddot{\mathbf{b}})_q + (\mathbf{C}\ddot{\mathbf{X}}\mathbf{H}\ddot{\mathbf{b}})_q = \mathbf{0}$ . Using Eqs. (6.1.54) and (6.2.15), this yields

$$(\mathbf{H}\ddot{\mathbf{b}})_q = -\mathbf{H}(\mathbf{C}\ddot{\mathbf{X}}\mathbf{H}\ddot{\mathbf{b}})_q = -\mathbf{H} \begin{bmatrix} \mathbf{P2}(\mathbf{q}, \mathbf{X}\mathbf{H}\mathbf{b}) \\ \mathbf{E2}(\mathbf{q}, \mathbf{X}\mathbf{H}\mathbf{b}) \end{bmatrix} \tag{6.2.16}$$

Multiplying Eq. (6.1.59) on the right by an arbitrary vector  $\mathbf{c}$  and differentiating,

$$(\mathbf{K}\ddot{\mathbf{c}})_q \equiv -\mathbf{X}(\mathbf{H}\ddot{\mathbf{C}}\mathbf{W}\ddot{\mathbf{c}})_q - \mathbf{X}\mathbf{H}(\mathbf{C}\ddot{\mathbf{W}}\ddot{\mathbf{c}})_q \tag{6.2.17}$$

With the foregoing expressions and  $\mathbf{P3}(\mathbf{q}, \dot{\mathbf{q}})$  of Eq. (5.3.17),

$$\mathbf{P3}(\mathbf{q}, \dot{\mathbf{q}}) \equiv \left( \left( \Phi_q (\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} \right)_q \ddot{\mathbf{q}} \right)_q = \left( \left( \mathbf{P2}(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} \right)_q \right)_q \tag{6.2.18}$$

and Eqs. (5.3.18) for holonomic constraints are

$$\begin{aligned}
\gamma_h &= \mathbf{P2}(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} + 2\Phi_{tq} \dot{\mathbf{q}} + \Phi_{tt} \\
\gamma_{hq} &= \mathbf{P3}(\mathbf{q}, \dot{\mathbf{q}}) + 2(\Phi_{tq} \ddot{\mathbf{q}})_q + \Phi_{ttq} \\
\gamma_{hq} &= 2\mathbf{P2}(\mathbf{q}, \dot{\mathbf{q}}) + 2\Phi_{tq}
\end{aligned} \tag{6.2.19}$$

Define

$$E3(\mathbf{q}, \dot{\mathbf{q}}) \equiv \left( \left( E(\mathbf{q}, t) \ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}} \right) = \left( E2(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} \right)_q \quad (6.2.20)$$

where  $E2(\mathbf{q}, \kappa)$  is defined in Eq. (6.2.9), and establish the identity

$$\left( \left( E(\mathbf{q}, t) \ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}} \right)_{\dot{\mathbf{q}}} = 2 \left( E(\mathbf{q}, t) \ddot{\mathbf{q}} \right)_q = 2E2(\mathbf{q}, \dot{\mathbf{q}}) \quad (6.2.21)$$

This result is valid since, as shown in Section 5.3.3, the expression being differentiated with respect to  $\dot{\mathbf{q}}$  is symmetric in its  $\dot{\mathbf{q}}$  arguments; i.e.,  $\left( \left( E(\mathbf{q}, t) \ddot{\mathbf{q}}^1 \right)_q \dot{\mathbf{q}}^2 \right) = \left( \left( E(\mathbf{q}, t) \ddot{\mathbf{q}}^2 \right)_q \dot{\mathbf{q}}^1 \right)$ . Thus, for  $\gamma_{nh}$  of Eq. (6.1.29),

$$\begin{aligned} \gamma_{nh} &\equiv \left( E(\mathbf{q}, t) \hat{\mathbf{q}} \right)_q \dot{\mathbf{q}} + E_t(\mathbf{q}, t) \dot{\mathbf{q}} - e_q(\mathbf{q}, t) \dot{\mathbf{q}} - e_t(\mathbf{q}, t) \\ &= E2(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + E_t(\mathbf{q}, t) \dot{\mathbf{q}} - e_q(\mathbf{q}, t) \dot{\mathbf{q}} - e_t(\mathbf{q}, t) \\ \gamma_{nhq} &= \left( E2(\mathbf{q}, \hat{\mathbf{q}}) \hat{\mathbf{q}} \right)_q + \left( E_t(\mathbf{q}, t) \hat{\mathbf{q}} \right)_q - \left( e_q(\mathbf{q}, t) \hat{\mathbf{q}} \right)_q - e_{tq}(\mathbf{q}, t) \\ &= E3(\mathbf{q}, \dot{\mathbf{q}}) + \left( E_t(\mathbf{q}, t) \hat{\mathbf{q}} \right)_q - \left( e_q(\mathbf{q}, t) \hat{\mathbf{q}} \right)_q - e_{tq}(\mathbf{q}, t) \\ \gamma_{nh\dot{\mathbf{q}}} &= \left( \left( E(\hat{\mathbf{q}}, t) \hat{\mathbf{q}} \right)_q \dot{\mathbf{q}} \right)_{\dot{\mathbf{q}}} + E_t(\mathbf{q}, t) - e_q(\mathbf{q}, t) \\ &= 2E2(\mathbf{q}, \dot{\mathbf{q}}) + E_t(\mathbf{q}, t) - e_q(\mathbf{q}, t) \end{aligned} \quad (6.2.22)$$

Derivatives of  $\mathbf{v}$  and  $\gamma$  defined in Eqs. (6.164) and (6.2.22) are thus available. These relations and evaluation of the kinetic quantities  $(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}})$ ,  $\mathbf{S}_q$ ,  $\mathbf{S}_{\dot{q}}$ ,  $\mathbf{Q}_q^A$ , and  $\mathbf{Q}_{\dot{q}}^A$  presented in Appendix 5.B provide quantities needed to evaluate terms in Eq. (6.2.13), hence the Jacobian of the residual of Eq. (6.2.10) that is used in implicit numerical integration.

#### 6.2.4 Implicit Numerical Integration Algorithms

*Implicit numerical integration algorithms* of Section 4.8 for solution of first order ODE are applied for solution of Eqs. (6.2.5) and (6.2.6). Since *trapezoidal integration* formulas are stated in terms of generalized coordinates, their application is rather simple, as presented in Section 6.2.4.1. Implicit Runge-Kutta integration formulas, in contrast, are stated in terms of stage values that are not generalized coordinates. They are applied to the explicit ODE of Eq. (6.2.7) and expanded for solution of Eq. (6.2.5) in Section 6.2.4.2.

##### 6.2.4.1 Trapezoidal Integration

*Implicit trapezoidal formulas* for integration of  $\dot{\mathbf{v}}$  and  $\dot{\mathbf{w}}$  of Eq. (4.8.40) are

$$\begin{aligned} \mathbf{v}_n &= \mathbf{v}_{n-1} + (h/2)(\dot{\mathbf{v}}_{n-1} + \dot{\mathbf{v}}_n) \\ \mathbf{w}_n &= \mathbf{w}_{n-1} + (h/2)(\dot{\mathbf{w}}_{n-1} + \dot{\mathbf{w}}_n) \end{aligned} \quad (6.2.23)$$

These equations are used with Eqs. (6.1.22) and (6.1.35) to evaluate  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  in Eq. (6.2.10),

$$\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \equiv \begin{bmatrix} \dot{\mathbf{v}}_n - \mathbf{V}^T \mathbf{K} (\mathbf{w}_{n-1} + (h/2)(\dot{\mathbf{w}}_{n-1} + \dot{\mathbf{w}}_n)) - \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 - \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \mathbf{K}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}}_n - \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{S} + \mathbf{Q}^A) \end{bmatrix} = \mathbf{0} \quad (6.2.24)$$

Equations (6.2.23) allow all arguments of functions appearing in Eq. (6.2.24) to be written as functions of  $\dot{\mathbf{v}}_n$  and  $\dot{\mathbf{w}}_n$ . Using the *chain rule of differentiation* with relations derived in Section 6.2.3, the *Jacobian of the residual* of Eq. (6.2.24) with respect to  $\mathbf{z}_n \equiv [\dot{\mathbf{v}}_n^T \quad \dot{\mathbf{w}}_n^T]^T$  is

$$\mathbf{J}^{\text{trap}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^T \mathbf{M} \mathbf{K} \end{bmatrix} + (h/2) \begin{bmatrix} \mathbf{R1}_v & \mathbf{R1}_w \\ \mathbf{R2}_v & \mathbf{R2}_w \end{bmatrix} \equiv \mathbf{J}_1 + (h/2) \mathbf{J}_2 \quad (6.2.25)$$

where

$$\begin{aligned} \mathbf{R1}_v &= -\mathbf{V}^T \left\{ \left( \mathbf{K} (\widehat{\mathbf{w}_{n-1} + (h/2)(\dot{\mathbf{w}}_{n-1} + \dot{\mathbf{w}}_n)}) \right)_q + \mathbf{X} (\mathbf{H} \ddot{\mathbf{q}}^0)_q + \mathbf{X} \mathbf{H} (\mathbf{C} \ddot{\mathbf{q}}^0)_q - \mathbf{X} (\mathbf{H} \ddot{\mathbf{v}})_q - \mathbf{X} \mathbf{H} \mathbf{v}_q \right\} \mathbf{q}_v \\ \mathbf{R1}_w &= \mathbf{0} \\ \mathbf{R2}_v &= \left\{ \begin{aligned} &\left( \mathbf{K}^T (\mathbf{M} \mathbf{K} \dot{\mathbf{w}}_n - \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{S} + \mathbf{Q}^A)) \right)_q + \mathbf{K}^T (\mathbf{M} (\widehat{\mathbf{K} \dot{\mathbf{w}}_n - \mathbf{X} \mathbf{H} \gamma})_q \\ &+ \mathbf{K}^T \mathbf{M} (\mathbf{K} \ddot{\mathbf{w}}_n) - \mathbf{K}^T \mathbf{M} \mathbf{X} (\mathbf{H} \ddot{\mathbf{v}})_q - \mathbf{K}^T \mathbf{M} \mathbf{X} \mathbf{H} \gamma_q - \mathbf{K}^T (\mathbf{S}_q + \mathbf{Q}_q^A) \\ &- \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \dot{\mathbf{q}}_v \end{aligned} \right\} \mathbf{q}_v \\ \mathbf{R2}_w &= -\mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \dot{\mathbf{q}}_w \end{aligned} \quad (6.2.26)$$

and  $\mathbf{q}_v$ ,  $\dot{\mathbf{q}}_v$ , and  $\dot{\mathbf{q}}_w$  are given by Eq. (6.2.12).

At  $t^n$ , with  $h$  small, the first matrix dominates. As shown in Section 6.2.1, it is nonsingular, so the Jacobian is nonsingular in a neighborhood of  $\mathbf{q}^0$ , for  $h$  sufficiently small.

*Newton-Raphson iteration* for solution of Eq. (6.2.24) is

$$\begin{aligned} \mathbf{J}^{\text{trap}} \Delta \mathbf{z}^i &= -\mathbf{R}^i & i = 0, 1, \dots \text{ until } \|\mathbf{R}^i\| < \text{intol} \\ \mathbf{z}^{i+1} &= \mathbf{z}^i + \Delta \mathbf{z}^i \end{aligned} \quad (6.2.27)$$

At  $t^n$ , with initial conditions of Eq. (6.2.6) and  $h = 0$ , Eq. (6.2.24) is solved to obtain an estimate  $\mathbf{z}^0 \equiv [\dot{\mathbf{v}}_n^{0T} \quad \dot{\mathbf{w}}_n^{0T}]^T$  to start the numerical integration process. At subsequent time steps, good estimates for  $\mathbf{z}^0$  are available from prior time steps.

Bounds on the norms of  $\mathbf{v}$  and  $\mathbf{w}$ , the number of iterations in evaluating  $\mathbf{u}$ ,  $\mathbf{B}$ , and  $\mathbf{H}$ , and the *condition number* of the coefficient matrix  $\mathbf{J}^{\text{trap}}$  of Eq. (6.2.27) are used to determine whether a *reparameterization* is required. If not, the integration process is continued. If so, the current time is set as  $\bar{t}^0$  and the current value of  $\mathbf{q}$  is used as  $\bar{\mathbf{q}}^0$  to redefine  $\bar{\mathbf{V}}$ ,  $\bar{\mathbf{U}}$ ,  $\bar{\mathbf{W}}$ ,  $\bar{\mathbf{X}}$ ,  $\bar{\mathbf{B}}$ ,  $\bar{\mathbf{H}}$ , and  $\bar{\mathbf{K}}$ . The process is restarted with initial conditions  $\bar{\mathbf{v}}^0 = \mathbf{0}$  and  $\bar{\mathbf{w}}^0 = \mathbf{0}$  and estimates for their time derivatives  $\bar{\mathbf{v}}^0 = \bar{\mathbf{V}}^T \bar{\mathbf{q}}^0$  and  $\bar{\mathbf{w}}^0 = \bar{\mathbf{W}}^T \bar{\mathbf{q}}^0$ . The estimate  $\bar{\mathbf{z}}^0 \equiv [\bar{\mathbf{v}}^{0T} \quad \bar{\mathbf{w}}^{0T}]^T$  is used to evaluate the Jacobian of Eq. (6.2.25) and restart the numerical integration process.

#### 6.2.4.2 SDIRK Integration

Runge-Kutta numerical integration methods are presented in Section 4.8 for the solution of first order ODE of the form

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}) \quad (6.2.28)$$

where  $\mathbf{y}$  is an  $n$ -vector variable. A variety of *RK integrators* can be written in the form

$$\mathbf{k}_i = \mathbf{f} \left( t_n + c_i h, \mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}_j \right), \quad i=1, \dots, s \quad (6.2.29)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{k}_i \quad (6.2.30)$$

where  $t_n$  is the current time step;  $\mathbf{y}_n$  is the approximate solution at  $t_n$ ;  $a_{ij}$ ,  $b_i$ , and  $c_i = \sum_{j=1}^i a_{ij}$

are constants;  $\sum_{i=1}^s b_i = 1$ ;  $s$  is the *number of stages* in integration;  $\mathbf{k}_i$  are stage variables; and  $h$  is the *step-size*. If diagonal terms of the matrix  $\mathbf{a} = [a_{ij}]$  are  $a_{ii} = \alpha \neq 0$ ,  $i=1, 2, \dots, s$ , the method is called singly diagonal, or SDIRK. The stiffly-accurate, L-stable, 5 stage, order 4 *SDIRK54* formula for first order ODE is defined in Table 4.8.6.

To integrate the ODE of Eq. (6.2.7) using a RK method, *stage variables* are  $\mathbf{k}_i = [\mathbf{kv}_i^T \quad \mathbf{kw}_i^T]^T$  and arguments of functions in the *stage equations* are  $\mathbf{q}(\mathbf{v}, t)$  and  $\dot{\mathbf{q}}(\mathbf{v}, \mathbf{w}, t)$  of Eqs. (6.2.4). Evaluated as functions of stage variables, they are

$$\begin{aligned} \mathbf{q}_i &= \mathbf{q} \left( \mathbf{v}_n + h \sum_{j=1}^i a_{ij} \mathbf{kv}_j, t_n \right) \\ \dot{\mathbf{q}}_i &= \dot{\mathbf{q}} \left( \mathbf{v}_n + h \sum_{j=1}^i a_{ij} \mathbf{kv}_j, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{kw}_j, t_n \right) \end{aligned} \quad (6.2.31)$$

Denoting a general function  $\mathbf{g}(t, \mathbf{v}, \mathbf{w})$ , evaluated as a function of stage variables,

$$\check{\mathbf{g}}_i = \mathbf{g} \left( t_n + c_i h, \mathbf{v}_n + h \sum_{j=1}^i a_{ij} \mathbf{kv}_j, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{kw}_j \right) \quad (6.2.32)$$

Stage equations for Eqs. (6.2.7), written in residual form, are

$$\check{\mathbf{R}}1 \equiv \mathbf{kv}_i - \mathbf{V}^T \left( \check{\mathbf{K}}(\mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{kw}_j) + (\mathbf{I} - \mathbf{X} \check{\mathbf{H}} \check{\mathbf{C}}) \dot{\mathbf{q}}^0 + \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}} \right) = \mathbf{0} \quad (6.2.33)$$

$$\check{\mathbf{R}}(X, \tau)2 \equiv \mathbf{kw}_i - \left( \check{\mathbf{K}}^T \check{\mathbf{M}} \check{\mathbf{K}} \right)^{-1} \check{\mathbf{K}}^T \left( \check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}} + \check{\mathbf{Q}}^A + \check{\mathbf{S}} \right) = \mathbf{0} \quad (6.2.34)$$

Multiplying Eq. (6.2.34) by  $(\check{\mathbf{K}}_i^T \check{\mathbf{M}}_i \check{\mathbf{K}}_i)$  yields

$$\check{\mathbf{R}}2 \equiv (\check{\mathbf{K}}^T \check{\mathbf{M}} \check{\mathbf{K}}) \mathbf{kw}_i - \check{\mathbf{K}}^T \left( \check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}} + \check{\mathbf{Q}}^A + \check{\mathbf{S}} \right) = \mathbf{0} \quad (6.2.35)$$

Equations (6.2.33) and (6.2.35) comprise the residual form of *Runge-Kutta discretization* of Eq. (6.2.24). Using derivatives of residual terms in Eqs. (6.2.12) and (6.2.13) and the chain rule of differentiation with Eq. (6.2.32), the Jacobian of Eqs. (6.2.33) and (6.2.35) for an SDIRK integrator with  $a_{ii} = \alpha$ , in variables  $\mathbf{z}^i = [\mathbf{kv}_i^T \quad \mathbf{kw}_i^T]^T$ , is

$$\mathbf{J}^{RK} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \check{\mathbf{K}}^T \check{\mathbf{M}} \check{\mathbf{K}} \end{bmatrix} + h \begin{bmatrix} \check{\mathbf{R}}1_v & \check{\mathbf{R}}1_w \\ \check{\mathbf{R}}2_v & \check{\mathbf{R}}2_w \end{bmatrix} \equiv \check{\mathbf{J}}_1 + h \check{\mathbf{J}}_2 \quad (6.2.36)$$

i.e.,

$$\mathbf{J}^{RK} \Delta \mathbf{z}^i = \begin{bmatrix} \mathbf{V}^T \left( \check{\mathbf{K}}(\mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{kw}_j) + (\mathbf{I} - \check{\mathbf{X}} \check{\mathbf{H}} \check{\mathbf{C}}) \dot{\mathbf{q}}^0 + \check{\mathbf{X}} \check{\mathbf{H}} \check{\mathbf{v}} \right) \\ \check{\mathbf{K}}^T (\check{\mathbf{M}} \check{\mathbf{X}} \check{\mathbf{H}} \check{\mathbf{v}} + \check{\mathbf{Q}}^A + \check{\mathbf{S}}) \end{bmatrix} \quad (6.2.37)$$

where

$$\begin{aligned} \check{\mathbf{R}}1_v &= -\mathbf{V}^T \left\{ \left( \check{\mathbf{K}}(\mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{kw}_j) \right)_q - \mathbf{X} \left( \check{\mathbf{H}} \check{\mathbf{C}} \ddot{\mathbf{q}}^0 \right)_q - \mathbf{X} \check{\mathbf{H}} \left( \check{\mathbf{C}} \ddot{\mathbf{q}}^0 \right)_q - \mathbf{X} \check{\mathbf{H}} \check{\mathbf{C}} \dot{\mathbf{q}}_q + \mathbf{X} \left( \check{\mathbf{H}} \ddot{\mathbf{v}} \right)_q + \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}}_q \right\} \check{\mathbf{q}}_v \\ \check{\mathbf{R}}1_w &= \mathbf{0} \\ \check{\mathbf{R}}2_v &= \left\{ \begin{array}{l} \left( \check{\mathbf{K}}^T (\check{\mathbf{M}} \check{\mathbf{K}} \mathbf{kw}_i - \check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}} - \check{\mathbf{Q}}^A - \check{\mathbf{S}}) \right)_q + \check{\mathbf{K}}^T \left( \check{\mathbf{M}} (\check{\mathbf{K}} \mathbf{kw}_i - \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}}) \right)_q \\ + \check{\mathbf{K}}^T \check{\mathbf{M}} (\check{\mathbf{K}} \mathbf{kw})_q - \check{\mathbf{K}}^T \left( \check{\mathbf{M}} (\mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}}) \right)_q - \check{\mathbf{K}}^T (\check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}}_q + \check{\mathbf{Q}}^A + \check{\mathbf{S}}_q) \\ - \check{\mathbf{K}}^T (\check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}}_q + \check{\mathbf{Q}}^A + \check{\mathbf{S}}_q) \dot{\mathbf{q}}_v \end{array} \right\} \check{\mathbf{q}}_v \\ \check{\mathbf{R}}2_w &= -\check{\mathbf{K}}^T (\check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\mathbf{v}}_q + \check{\mathbf{Q}}^A + \check{\mathbf{S}}_q) \dot{\mathbf{q}}_w \end{aligned} \quad (6.2.38)$$

and derivatives of terms with an over score,  $\check{\bullet}$ , are evaluated as defined in Eq. (6.2.32).

The Newton-Raphson algorithm of Eq. (6.2.27) is used with the same approach outlined for the trapezoidal method to obtain initial estimates to start the iterative process and to redefine the parameterization and restart integration, as required.

#### 6.2.4.3 Implicit Integration Algorithm

**Implicit integration** of Eqs. (6.2.5) and (6.2.6), using trapezoidal and Runge-Kutta methods is as follows:

(1) Define  $\mathbf{q}^0$  at  $t^0$  that satisfies Eq. (6.1.24). Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{U}$  and  $\mathbf{V}$  in Eqs. (6.1.36) and the system velocity coefficient matrix  $\mathbf{C}(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{W}$  and  $\mathbf{X}$  in Eqs. (6.1.48) and (6.1.49). Define  $\dot{\mathbf{q}}^0$  that satisfies Eq. (6.1.30) and evaluate initial conditions  $\mathbf{v}^0 = \mathbf{0}$  and  $\mathbf{w}^0 = \mathbf{0}$  in Eq. (6.2.6).

(2) Apply an implicit numerical integrator to proceed stepwise on a time grid with step size  $h$ , using a factored form of the integration Jacobian of Eq. (6.2.25) or (6.2.36) to iteratively

determine  $\Delta\mathbf{z}^i$  of Eq. (6.2.27) or (6.2.37). Use Eq. (6.2.23) or (6.2.30) to determine  $\mathbf{v}_n$  and  $\mathbf{w}_n$ . Use Eqs. (6.1.45), (6.1.58), and (6.1.62) to evaluate  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ , and  $\ddot{\mathbf{q}}_n$  on the time grid.

(3) Monitor the *condition number* of the integration Jacobian, the number of Newton-Raphson iterations required in Step 2, the norms of  $\mathbf{v}$  and  $\mathbf{w}$ , and the number of iterations required to evaluate  $\mathbf{u}$ ,  $\mathbf{B}$ , and  $\mathbf{H}$ . If tolerances are exceeded, define a new  $\bar{\mathbf{t}}^0$  and associated  $\bar{\mathbf{q}}^0$  and  $\bar{\dot{\mathbf{q}}}^0$ . Repeat calculations in Step (1) to define a new *parameterization* and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\mathbf{w}}^0$ .

(4) Continue the process until the final time  $tf$  is reached, or a singularity is encountered due to a faulty design or model.

Beginning with the variational equation of motion for a nonholonomic system and tangent space parameterizations presented in Section 6.1, a first order ODE initial-value problem is obtained and shown to have a unique solution. Derivatives that are required for implicit numerical integration are derived and explicit and implicit numerical integration algorithms based on methods of Section 4.8 are presented.

## Key Formulas

$$\begin{aligned}\dot{\mathbf{v}} &= \mathbf{V}^T \mathbf{K} \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \mathbf{K}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} &= \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S})\end{aligned}$$

$$\begin{aligned}\mathbf{v}^0 &= \mathbf{0} \\ \mathbf{w}^0 &= \mathbf{0}\end{aligned}\quad (6.2.5) \quad (6.2.6)$$

### 6.3 Numerical Examples with Nonholonomic Tangent Space ODE

Four examples of varying degrees of complexity are studied in this section, using explicit numerical integration of the tangent space ODE formulation. A simple planar vehicle example, two spatial models of intermediate complexity, and a rather intricate spatial example of mixed holonomic-nonholonomic systems are presented. A still more complex example of a three-wheel motorcycle may be found in (Haug, 2017a). For illustrations using implicit numerical integration, the reader is referred to Section 6.5, where examples are treated using the Index 0 formulation. Extensive use of examples is adopted, because there is not yet a broadly applicable library of joints and differential constraints for a spectrum of practical systems.

#### 6.3.1 Planar Three Wheel Transporter

The *planar three-wheel transporter* shown in Fig. 6.3.1 is used to transport material in a manufacturing facility. The radius  $R$  of the rear wheels and the distance from the floor to point  $a$  are the same, so the bed of the transporter is the  $x'$ - $y'$  plane and is horizontal. Clockwise torques  $T_L$  and  $T_R$  are applied to the wheels, relative to the  $x'$  axis that is along the rear axle, leading to forces  $F_L = T_L/R$  and  $F_R = T_R/R$  applied at the respective wheel bearings in the  $y'$  direction to control motion of the transporter. Point  $b$  at the center of the axle is the origin of the  $x'$ - $y'$ - $z'$  body reference frame. Generalized coordinates are  $\mathbf{q} = [\mathbf{r}^T \quad \phi]^T = [x \quad y \quad \phi]^T$ . Since the front wheel is on a trailing arm that rotates freely about a vertical pivot in the chassis, it has no influence on motion in the  $x'$ - $y'$  plane. The rear wheels must *roll without slip*, so the velocity of point  $b$  must be orthogonal to the axle; i.e., to the  $x'$  axis. This leads to the *differential constraint*

$$(\mathbf{A}(\phi)\mathbf{u}'_x)^T \dot{\mathbf{r}} = [\cos\phi \quad \sin\phi]^T \dot{\mathbf{r}} = [\cos\phi \quad \sin\phi \quad 0]^T \dot{\mathbf{q}} \equiv \mathbf{E}(\mathbf{q})\dot{\mathbf{q}} = 0 \quad (6.3.1)$$

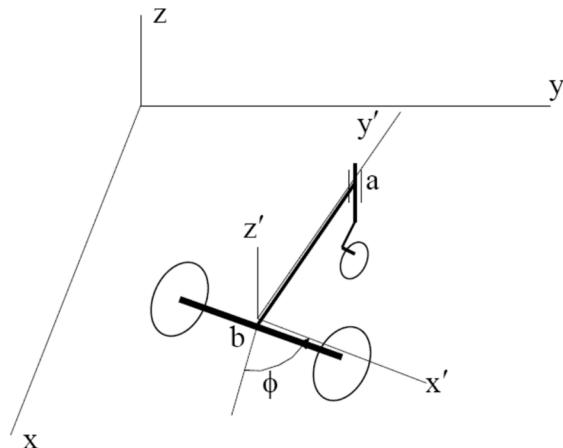


Figure 6.3.1 Three-Wheel Transporter

Since there are no holonomic constraints, the configuration tangent space is all of  $\mathbb{R}^3$  and the system has three *configuration degrees of freedom*. In the formulation of Section 6.2, there is no  $\mathbf{U}$ ,  $\mathbf{u}$ , or  $\mathbf{B}$  and  $\mathbf{V} = \mathbf{I}$ , so  $\mathbf{v} = \mathbf{q}$ . The tangent space of the velocity constraint of Eq. (6.3.1) at initial configuration  $\mathbf{q}^0$  is spanned by orthonormal vectors  $\mathbf{W}^i$  such that  $\mathbf{E}(\mathbf{q}^0)\mathbf{W}^i = 0$ ,  $i=1,2$ . A direct calculation yields orthonormal solutions

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^1 & \mathbf{W}^2 \end{bmatrix} = \begin{bmatrix} -\sin\phi^0 & 0 \\ \cos\phi^0 & 0 \\ 0 & 1 \end{bmatrix} \quad (6.3.2)$$

and, since  $\mathbf{C}(\mathbf{q}) = \mathbf{E}(\mathbf{q})$ ,

$$\mathbf{X} = \mathbf{E}(\mathbf{q}^0)^T = \begin{bmatrix} \cos\phi^0 & \sin\phi^0 & 0 \end{bmatrix}^T \quad (6.3.3)$$

Equation (6.1.54) is thus

$$\mathbf{H}(\mathbf{q}) = (\mathbf{E}(\mathbf{q})\mathbf{X})^{-1} = 1/(\cos\phi\cos\phi^0 + \sin\phi\sin\phi^0) \equiv 1/a(\mathbf{q}) \quad (6.3.4)$$

and since  $a(\mathbf{q}) = (\cos\phi\cos\phi^0 + \sin\phi\sin\phi^0) \neq 0$  in a neighborhood of  $\mathbf{q}^0$ , no iteration is required to evaluate  $\mathbf{H}(\mathbf{q})$ . Equations (6.1.58) and (6.1.59) thus simplify as

$$\dot{\mathbf{q}} = \mathbf{K}\mathbf{w} + \left( \mathbf{I} - \frac{1}{a} \mathbf{X}\mathbf{E} \right) \dot{\mathbf{q}}^0 \quad (6.3.5)$$

$$\mathbf{K} \equiv \left( \mathbf{I} - \frac{1}{a} \mathbf{X}\mathbf{E} \right) \mathbf{W} \quad (6.3.6)$$

Finally, Eq. (6.1.29) is

$$\mathbf{E}(\mathbf{q})\ddot{\mathbf{q}} = -\left( \mathbf{E}(\mathbf{q}, t)\hat{\mathbf{q}} \right)_q \dot{\mathbf{q}} = -\dot{x}\dot{\phi}\sin\phi + \dot{y}\dot{\phi}\cos\phi \equiv - \quad (6.3.7)$$

With the centroid 1 m forward of the rear axle on the  $y'$  axis,  $\mathbf{s}'^c = [0 \ 1]^T$ , and with a rear axle length of 1 m, terms required in *noncentroidal equations of motion* are

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} m\mathbf{I}_2 & m\mathbf{P}\mathbf{A}(\phi)\mathbf{s}'^c \\ ms'^{ctT}\mathbf{A}^T(\phi)\mathbf{P}^T & \mathbf{J}' \end{bmatrix} \\ \mathbf{S} &= \begin{bmatrix} m\dot{\phi}^2\mathbf{A}(\phi)\mathbf{s}'^c \\ 0 \end{bmatrix} \\ \mathbf{Q}^A &= \begin{bmatrix} 0 \\ F_L + F_R \\ R(F_R - F_L) \end{bmatrix} \\ \mathbf{A}(\phi) &= \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}; \quad \mathbf{P} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \end{aligned} \quad (6.3.8)$$

The equations of motion of Eqs. (6.2.5) and initial conditions of Eq. (6.2.6) are thus

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{Kw} + \left( \mathbf{I} - \frac{1}{a} \mathbf{XE} \right) \dot{\mathbf{q}}^0 \\ \mathbf{K}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} &= \mathbf{K}^T \mathbf{Q}^A + \frac{1}{a} \mathbf{K}^T \mathbf{M} \mathbf{X} \gamma + \mathbf{K}^T \mathbf{S} \\ \mathbf{q}(t^0) &= \mathbf{q}^0 \\ \mathbf{w}(t^0) &= \mathbf{0}\end{aligned}\quad (6.3.9)$$

Since there is no  $\mathbf{u}$  or  $\mathbf{B}$  to be iteratively determined and  $\mathbf{H}$  is given by the expression of Eq. (6.3.4), this is an explicitly defined *ODE initial-value problem*.

The mass and polar moment of inertia of the transporter, relative to the  $x'y'$  frame, are

$m = 10 \text{ kg}$  and  $J' = 10 \text{ kg} \cdot \text{m}^2$ ,  $g = 9.8 \text{ m/sec}^2$ , and  $R = 1 \text{ m}$ . The initial configuration is with point b on the z axis and the  $x'y'$  and x-y frames aligned, so  $\mathbf{q}^0 = [0 \ 0 \ 0]^T$ . The initial velocity is 5 m/sec along the y axis, so  $\dot{\mathbf{q}}^0 = [0 \ 5 \ 0]^T$ . Applied forces  $FL = 20 \text{ N}$ ,  $0 \leq t \leq 2$ ;  $FL = -20 \text{ N}$ ,  $2 \leq t \leq 4$ ;  $FL = 0$ ,  $t > 4$ ;  $FR = -20 \text{ N}$ ,  $0 \leq t \leq 2$ ;  $FR = 20 \text{ N}$ ,  $2 \leq t \leq 4$ ;  $FR = 0$ ,  $t > 4$  in the  $y'$  direction on the left and right rear wheel spindles yield a clockwise torque about the z-axis for two seconds, followed by a counterclockwise torque about the z-axis for two seconds, and no applied torque after four seconds.

The x-y trajectory of point b for a 10 sec simulation with the *explicit RK4* integrator of Section 4.8.1.1 is shown in Fig. 6.3.2. Simulations are carried out with MATLAB Code 6.3.1 of Appendix 6.A. Velocity and acceleration constraint errors are less than  $10^{-15}$ . Even though a bound of unity on the norm of  $\mathbf{w}$  was set as a criterion for *reparameterization*, it was never reached, so only one parameterization was used for the entire simulation. The *condition number* of the reduced mass matrix was almost exactly one, throughout the simulation.

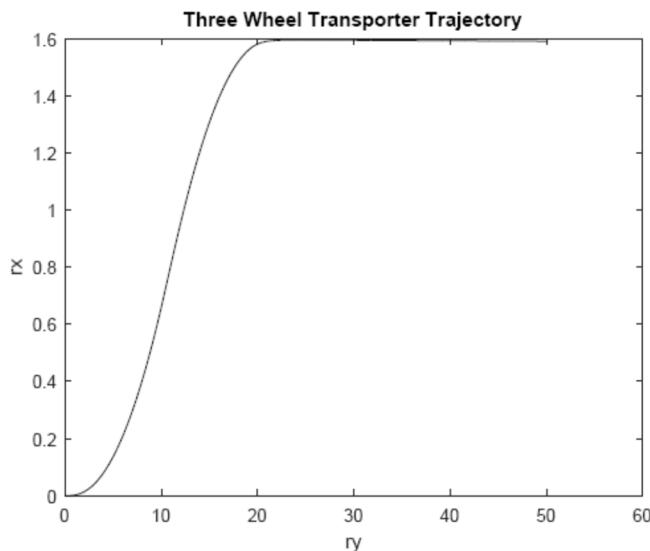


Figure 6.3.2 ry-rx Trajectory of Three Wheel Transporter

### 6.3.2 Planar Articulated Vehicle

A two chassis articulated vehicle is modeled in the x-y plane, as shown in Fig. 6.3.3. The front axle of chassis one is steerable, with angle  $\theta(t)$ . Since *roll without slip* conditions for both front wheels would be redundant, a single wheel model is placed at the center of the front axle at point  $P_1$  and the *roll without slip* condition is applied to this wheel model. The chassis are coupled by a revolute joint at point  $P_3$  that is orthogonal to their common plane of motion, 3 m from the origins of body-fixed reference frames.

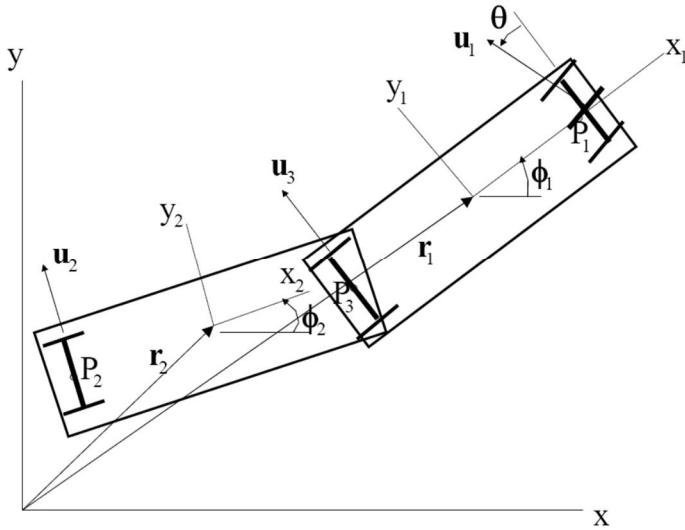


Figure 6.3.3 Planar Articulated Vehicle

Six *generalized coordinates* for the system are

$$\mathbf{q} = \begin{bmatrix} \mathbf{r}_1^T & \phi_1 & \mathbf{r}_2^T & \phi_2 \end{bmatrix}^T = [x_1 \ y_1 \ \phi_1 \ x_2 \ y_2 \ \phi_2]^T$$

The revolute joint between the chassis is characterized by the *holonomic constraint*

$$\Phi(\mathbf{q}) = (\mathbf{r}_1 - 3\mathbf{A}(\phi_1)\mathbf{i}'_1) - (\mathbf{r}_2 + 3\mathbf{A}(\phi_2)\mathbf{i}'_2) = \mathbf{0} \quad (6.3.10)$$

whose Jacobian is

$$\Phi_q(\mathbf{q}) = [\mathbf{I}_2 \ -3\mathbf{PA}(\phi_1)\mathbf{i}'_1 \ -\mathbf{I}_2 \ -3\mathbf{PA}(\phi_2)\mathbf{i}'_2] \quad (6.3.11)$$

where  $\mathbf{A}(\phi_i)$  is the rotation transformation matrix from the  $x'_i - y'_i$  frame to the x-y frame and  $\mathbf{i}'_i$  are unit vectors along the  $x'_i$  axes.

Vectors from origins of body-fixed reference frames to centers of axles are

$\mathbf{r}_1^{P_1} = \mathbf{r}_1 + 3\mathbf{A}(\phi_1)\mathbf{i}'_1$ ,  $\mathbf{r}_1^{P_3} = \mathbf{r}_1 - 3\mathbf{A}(\phi_1)\mathbf{i}'_1$ , and  $\mathbf{r}_2^{P_2} = \mathbf{r}_2 - 3\mathbf{A}(\phi_2)\mathbf{i}'_2$ . Velocities of these points are

$$\begin{aligned}\dot{\mathbf{r}}_1^{P_1} &= \dot{\mathbf{r}}_1 + 3\dot{\phi}_1 \mathbf{PA}(\phi_1) \mathbf{i}'_1 \\ \dot{\mathbf{r}}_1^{P_3} &= \dot{\mathbf{r}}_1 - 3\dot{\phi}_1 \mathbf{PA}(\phi_1) \mathbf{i}'_1 \\ \dot{\mathbf{r}}_2^{P_2} &= \dot{\mathbf{r}}_2 - 3\dot{\phi}_2 \mathbf{PA}(\phi_2) \mathbf{i}'_2\end{aligned}\quad (6.3.12)$$

where  $\mathbf{P} = \mathbf{A}(\pi/2)$  and  $\mathbf{A}(\phi)_\phi = \mathbf{PA}(\phi)$ . The velocities of Eqs. (6.3.12) must be orthogonal to unit vectors  $\mathbf{u}_1$ ,  $\mathbf{u}_3$ , and  $\mathbf{u}_2$ , respectively, as conditions that the *wheels do not slip laterally*; i.e.,  $\mathbf{u}_1^T \dot{\mathbf{r}}_1^{P_1} = [-\sin \theta \quad \cos \theta] (\mathbf{A}(\phi_1)^T \dot{\mathbf{r}}_1 + 3\dot{\phi}_1 \mathbf{Pi}') = 0$ ,  $\mathbf{u}_3^T \dot{\mathbf{r}}_1^{P_3} = \mathbf{j}'_1^T \mathbf{A}(\phi_1)^T \dot{\mathbf{r}}_1 - 3\dot{\phi}_1 = 0$ , and

$\mathbf{u}_2^T \dot{\mathbf{r}}_2^{P_2} = \mathbf{j}'_2^T \mathbf{A}(\phi_2)^T \dot{\mathbf{r}}_2 - 3\dot{\phi}_2 = 0$ , where  $\mathbf{A}^T \mathbf{PA} = \mathbf{P}$  has been used. *Differential constraints* that act on the system are thus

$$\mathbf{E}(\mathbf{q}, t)\dot{\mathbf{q}} = \begin{bmatrix} [-\sin(\theta) \quad \cos(\theta)] \mathbf{A}(\phi_1)^T & 3[-\sin(\theta) \quad \cos(\theta)] \mathbf{j}'_1 & \mathbf{0} & 0 \\ \mathbf{j}'_1^T \mathbf{A}(\phi_1)^T & -3 & \mathbf{0} & 0 \\ \mathbf{0} & 0 & \mathbf{j}'_2^T \mathbf{A}(\phi_2)^T & -3 \end{bmatrix} \dot{\mathbf{q}} = \mathbf{0} \quad (6.3.13)$$

so  $\mathbf{e}(\mathbf{q}, t) = \mathbf{0}$ . The composite constraint coefficient matrix of Eqs. (6.3.11) and (6.3.13) is

$$\mathbf{C}(\mathbf{q}, t) = \begin{bmatrix} \Phi_q(\mathbf{q}) \\ \mathbf{E}(\mathbf{q}, t) \end{bmatrix} \quad (6.3.14)$$

As defined in Section 6.1, and in turn in Section 6.2, derivative terms needed for *explicit numerical integration* of the equations of motion, with  $\boldsymbol{\chi} = [\boldsymbol{\chi}_{r_1}^T \quad \boldsymbol{\chi}_{\phi_1}^T \quad \boldsymbol{\chi}_{r_2}^T \quad \boldsymbol{\chi}_{\phi_2}^T]^T$ , are  $\Phi_t = -\mathbf{v} = \mathbf{0}$ ,  $\Phi_{tq} = \Phi_{ttq} = \mathbf{0}$ , and

$$\begin{aligned}\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) &= (\Phi_q(\mathbf{q}, \ddot{\mathbf{t}}) \ddot{\boldsymbol{\chi}})_q = [\mathbf{0} \quad 3\boldsymbol{\chi}_{\phi_1} \mathbf{A}(\phi_1) \mathbf{i}'_1 \quad \mathbf{0} \quad 3\boldsymbol{\chi}_{\phi_2} \mathbf{A}(\phi_2) \mathbf{i}'_2] \\ \mathbf{E}_t &= -\dot{\theta} \begin{bmatrix} [\cos \theta \quad \sin \theta] \mathbf{A}(\phi_1)^T & 3[\cos \theta \quad \sin \theta] \mathbf{j}'_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{2 \times 6} & \end{bmatrix} \\ (\mathbf{E}_t(\mathbf{q}, \ddot{\mathbf{t}}) \ddot{\mathbf{q}})_q &= \dot{\theta} \begin{bmatrix} \mathbf{0} & [\cos \theta \quad \sin \theta] \mathbf{A}(\phi_1)^T \mathbf{P} \dot{\mathbf{r}}_1 & \mathbf{0} & 0 \\ \mathbf{0}_{2 \times 6} & \end{bmatrix} \\ \mathbf{E}2(\mathbf{q}, \boldsymbol{\chi}) &= (\mathbf{E}(\mathbf{q}, \ddot{\mathbf{t}}) \ddot{\boldsymbol{\chi}})_q = \begin{bmatrix} \mathbf{0} & -[-\sin(\theta) \quad \cos(\theta)] \mathbf{A}(\phi_1)^T \mathbf{P} \boldsymbol{\chi}_{r_1} & \mathbf{0} & 0 \\ \mathbf{0} & -\mathbf{j}'_1^T \mathbf{A}(\phi_1)^T \mathbf{P} \boldsymbol{\chi}_{r_1} & \mathbf{0} & 0 \\ \mathbf{0} & 0 & \mathbf{0} & -\mathbf{j}'_2^T \mathbf{A}(\phi_2)^T \mathbf{P} \boldsymbol{\chi}_{r_2} \end{bmatrix}\end{aligned}$$

With these results,  $\gamma$  of Eqs. (6.2.19) may be evaluated. In the planar equations of motion for this system,  $\mathbf{M} = m\mathbf{I}_{6 \times 6}$ ,  $\mathbf{S} = \mathbf{0}$ , and  $\mathbf{Q}^A = \mathbf{0}$ .

With two holonomic and three differential constraints on six generalized coordinates and velocities, this system has four *kinematic degrees of freedom* and one *kinetic degree of freedom*.

Accordingly, there are four first order kinematic equations for  $\mathbf{v}$  and one first order kinetic equation for  $\mathbf{w}$  in Eqs. (6.2.5).

As a numerical example, each of the two chassis has a mass of 1,500 kg and the same polar moment of inertia. Acceleration due to gravity is  $9.8 \text{ m/sec}^2$ . The  $x'$ -axes of the chassis are initially coincident with the global  $x$ -axis, with the centroid of the rear chassis at the origin of the  $x$ - $y$  frame, so  $\mathbf{q}^0 = [6, 0, 0, 0, 0, 0]^T$ . The initial velocity of both chassis, consistent with kinematic and differential constraints, is 15 m/sec (35 mph) along the global  $x$  axis, so  $\dot{\mathbf{q}}^0 = [15, 0, 0, 15, 0, 0]^T$ . A steer angle  $\theta = 0.017\sin(t)\text{rad}$ , for  $0 \leq t \leq 2$ , and  $\theta = 0$  thereafter is imposed to simulate a *lane change maneuver* to the left. Simulations are carried out with Code 6.5.1 of Appendix 6.A that supports both the tangent space ODE and Index 0 DAE formulations, using the *explicit RK4* integration algorithm.

Results shown in Fig. 6.3.4 for  $tf = 10$  sec,  $h = 0.001$  sec,  $utol = Btol = Htol = 10^{-10}$ , and  $intol = 0.0001$  confirm the lane change trajectory for the first chassis in the left plot and the orientation of the second chassis relative to the first in the right plot, subsequently trailing with both chassis aligned. For the 10,000 time steps of the simulation, the algorithm required 22 *reparameterizations* (455 time steps per reparameterization), all due to a limit of 10 imposed on the norm of  $\mathbf{v}$ . *Kinetic energy* for all three simulations is constant to five significant figures for this *conservative system*.

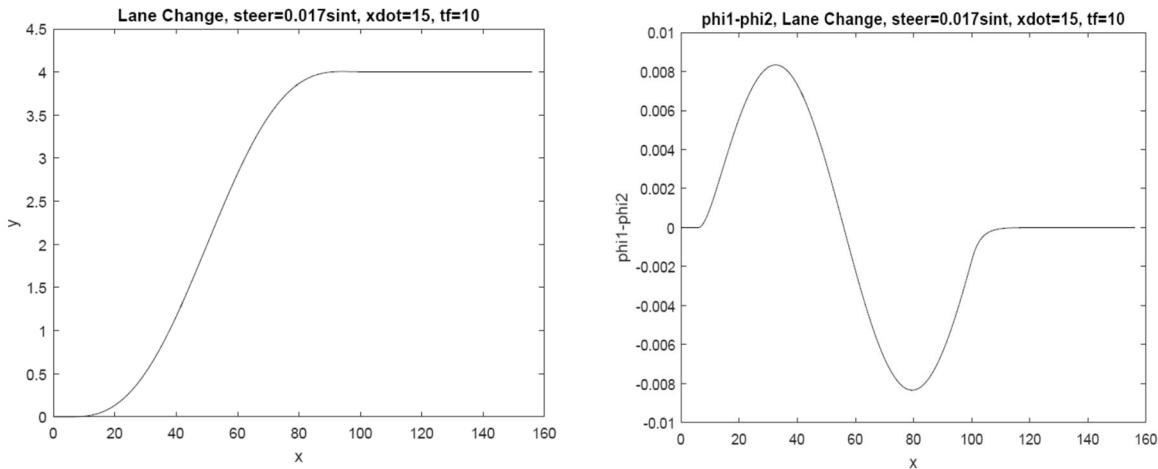


Figure 6.3.4 Lane Change Maneuver

As a check on constraint error control,  $Tol = utol = Btol = Htol$  in Eqs. (6.1.42), (6.1.44), and (6.1.56) is varied. Results presented in Table 6.3.1 show that maximum norms of position, velocity, and acceleration constraint errors over the simulation interval are driven toward zero to computer precision as tolerances are tightened, with little increase in compute time.

Table 6.3.1 Maximum Constraint Error for Planar Articulated Vehicle

Tol	Position Err.	Velocity Err.	Acceleration Err.
e-6	7e-13	e-10	e-10
e-9	5e-14	e-10	e-10
e-12	5e-14	6e-15	e-14

### 6.3.3 Spatial Disk Rolling Without Slip on x-y Plane

The disk with unit radius shown in Fig. 6.3.5 rolls without slip on the x-y plane. The plane of the disk is defined by body fixed  $y'$ - $z'$  axes, where the body fixed  $x'$  axis is normal to the plane of the disk. Unit vector  $\mathbf{a}'$  in the  $x'$ - $y'$ - $z'$  frame from the center of the disk to contact point C on the periphery with the x-y plane is

$$\mathbf{a}' = \begin{bmatrix} 0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{a} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \mathbf{a} \equiv \mathbf{a}'_a \mathbf{a} \quad (6.3.15)$$

and the vector  $\mathbf{a} = [a_1 \ a_2]^T$  satisfies the condition

$$(\mathbf{a}^T \mathbf{a} - 1)/2 = 0 \quad (6.3.16)$$

The normal to the disk periphery at contact point C in the plane of the disk is the vector  $\mathbf{a}'$ , so the tangent to the disk periphery at point C in the plane of the disk is

$$\mathbf{b}' = \begin{bmatrix} 0 \\ -a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathbf{a} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathbf{P} \mathbf{a} \\ \mathbf{P} \mathbf{a} \end{bmatrix} \mathbf{a} \equiv \mathbf{b}'_a \mathbf{a} \quad (6.3.17)$$

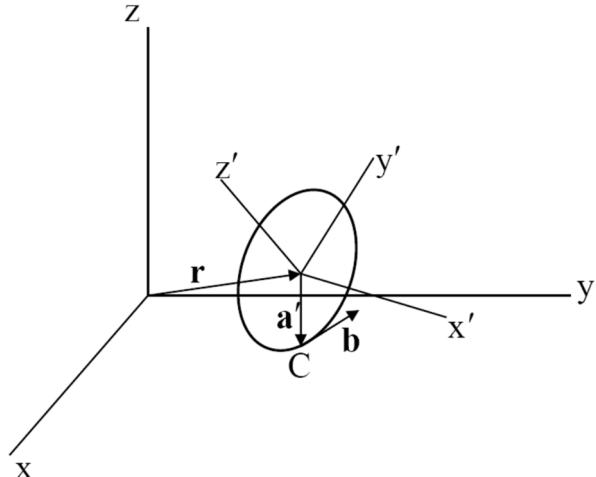


Figure 6.3.5 Disk Rolling Without Slip on the x-y Plane

*Euler parameters* are used as *orientation generalized coordinates*. Contact conditions are that point C on the disk is in the x-y plane; i.e.,  $\mathbf{u}_z^T(\mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}'_a\mathbf{a}) = 0$ , and that the vector  $\mathbf{b} = \mathbf{A}(\mathbf{p})\mathbf{b}'$  is in the x-y plane; i.e.,  $\mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a\mathbf{a} = 0$ . Combining these conditions, Eq. (6.3.16), and the Euler parameter normalization condition yield the *holonomic constraints*

$$\Phi(\mathbf{q}) = \begin{bmatrix} \mathbf{u}_z^T(\mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}'_a\mathbf{a}) \\ \mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a\mathbf{a} \\ (\mathbf{p}^T \mathbf{p} - 1)/2 \\ (\mathbf{a}^T \mathbf{a} - 1)/2 \end{bmatrix} = \mathbf{0} \quad (6.3.18)$$

where generalized coordinates are  $\mathbf{q} = [\mathbf{r}^T \quad \mathbf{p}^T \quad \mathbf{a}^T]^T \in \mathbb{R}^9$ . Since  $\Phi(\mathbf{q})$  does not depend explicitly on  $t$ ,  $\Phi_t = -\mathbf{v}_h = \mathbf{0}$ ,  $\Phi_{tt} = \mathbf{0}$ , and  $\Phi_{qt} = \mathbf{0}$ . The holonomic constraint Jacobian is

$$\Phi_q = \begin{bmatrix} \mathbf{u}_z^T & \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a\mathbf{a}) & \mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{b}'_a\mathbf{a}) & \mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a \\ \mathbf{0} & \mathbf{p}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{a}^T \end{bmatrix} \quad (6.3.19)$$

The velocity of point C on the periphery of the disk is

$$\begin{aligned} \mathbf{v}^C &= \dot{\mathbf{r}} + \mathbf{A}(\mathbf{p})\tilde{\mathbf{a}}'\mathbf{a} = \dot{\mathbf{r}} + \mathbf{A}(\mathbf{p})\mathbf{A}^T(\mathbf{p})\dot{\mathbf{A}}(\mathbf{p})\mathbf{a}'_a\mathbf{a} \\ &= \dot{\mathbf{r}} + \dot{\mathbf{A}}(\mathbf{p})\mathbf{a}'_a\mathbf{a} = \dot{\mathbf{r}} + (\mathbf{A}(\mathbf{p})\mathbf{a}'_a\hat{\mathbf{a}})_p \dot{\mathbf{p}} = \dot{\mathbf{r}} + \mathbf{B}(\mathbf{p}, \mathbf{a}'_a\mathbf{a})\dot{\mathbf{p}} \end{aligned} \quad (6.3.20)$$

Conditions that *no slip* occurs between the disk and the x-y plane are that the horizontal components of  $\mathbf{v}^C$  are zero; i.e.  $\mathbf{u}_x^T \mathbf{v}^C = 0$  and  $\mathbf{u}_y^T \mathbf{v}^C = 0$ . With Eq. (6.3.20), this is

$$\mathbf{E}(\mathbf{q})\dot{\mathbf{q}} \equiv \begin{bmatrix} \mathbf{u}_x^T & \mathbf{u}_x^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a\mathbf{a}) & 0 \\ \mathbf{u}_y^T & \mathbf{u}_y^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a\mathbf{a}) & 0 \end{bmatrix} \dot{\mathbf{q}} \equiv \mathbf{e}(\mathbf{q}, t) = \mathbf{0} \quad (6.3.21)$$

Since  $\mathbf{E}$  does not depend explicitly on  $t$  and  $\mathbf{e} = \mathbf{0}$ ;  $\mathbf{E}_t = \mathbf{0}$ ,  $\mathbf{e}_q = \mathbf{0}$ , and  $\mathbf{e}_t = \mathbf{0}$ . With four holonomic and two differential constraints on nine generalized coordinates, the system has five *kinematic degrees of freedom* and three *kinetic degrees of freedom*.

Kinetic and force terms in the equations of motion are

$$\begin{aligned}
\mathbf{M}(\mathbf{q}) &= \begin{bmatrix} m\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}^T(\mathbf{p})\mathbf{J}'\mathbf{G}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} \mathbf{0} \\ 8\mathbf{G}^T(\dot{\mathbf{q}})\mathbf{J}'\mathbf{G}(\dot{\mathbf{q}})\mathbf{p} \\ \mathbf{0} \end{bmatrix} \\
\mathbf{Q}^A &= \begin{bmatrix} -mg\mathbf{u}_z \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}
\end{aligned} \tag{6.3.22}$$

Derivatives of terms that are required for explicit integration of the equations of motion, with  $\boldsymbol{\chi} = [\boldsymbol{\chi}_r^T \ \boldsymbol{\chi}_p^T \ \boldsymbol{\chi}_a^T]^T$ , are

$$\begin{aligned}
\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) &= (\Phi_q(\mathbf{q})\ddot{\boldsymbol{\chi}})_q = \begin{bmatrix} \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{a}'_a) + \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \boldsymbol{\chi}_a) & \mathbf{u}_z^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{b}'_a) + \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{b}'_a \boldsymbol{\chi}_a) & \mathbf{u}_z^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{b}'_a \\ \mathbf{0} & \boldsymbol{\chi}_p^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\chi}_a^T \end{bmatrix} \\
\mathbf{E}2(\mathbf{q}, \boldsymbol{\chi}) &= (\mathbf{E}(\mathbf{q})\ddot{\boldsymbol{\chi}})_q = \begin{bmatrix} \mathbf{0} & \mathbf{u}_x^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{a}'_a) & \mathbf{u}_x^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_y^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{a}'_a) & \mathbf{u}_y^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{a}'_a \end{bmatrix}
\end{aligned}$$

For a disk with thickness 0.01 m, radius 1 m, mass  $m = 10$  kg, and  $g = 9.6$  m/sec<sup>2</sup>, the density is  $1000/\pi$  kg/m<sup>3</sup>, so  $J_{x'x'} = m/2 = 5$  and  $J_{y'y'} = J_{z'z'} = m(3+10^{-4})/12 = 2.5001$ . The disk is initially in the y-z plane, with  $x'$  and x and  $y'$  and y axes parallel, so the initial configuration is  $\mathbf{r}^0 = \mathbf{u}_z$ ,  $\mathbf{p}^0 = [1 \ 0 \ 0]^T$ , and  $\mathbf{a}^0 = [0 \ -1]^T$ . The disk initially rolls along the y axis, with a small rotation perturbation about the y axis, so the initial angular velocity-Euler parameter derivative relation is  $[-\omega_{x0} \ 0.01 \ 0]^T = 2\mathbf{G}(\mathbf{p}^0)\dot{\mathbf{p}}^0$ . Appending this relation to Eq. (6.1.6) yields 9 equations in 9 unknowns that are solved for  $\dot{\mathbf{q}}^0$ .

The formulation is implemented in Code 6.5.2 of Appendix 6.A that supports both the tangent space ODE and Index 0 DAE formulations, using the explicit RK4 integration method. Plots of centroid ry-rx trajectory and rz-coordinate vs time for initial condition  $\omega_{x0} = -1$  rad/sec are shown in Fig. 6.3.6. The nearly straight segments of the ry-rx trajectory on the left are with  $r_z \approx 1$  and define the direction of roll, which changes after each loop and associated dip of the centroid shown in the right plot of Fig. 6.3.6. Numerical integration is carried out using  $utol = Btol = Htol = intol = 10^{-7}$  and  $h = 0.001$  sec with the RK4 algorithm. In 40,000 time steps, 127 reparameterizations were required (315 time steps per reparameterization), all due to a limit of 0.7 on the norm of  $\mathbf{v}$ . Kinetic energy in each simulation is constant to seven significant figures for this conservative system.

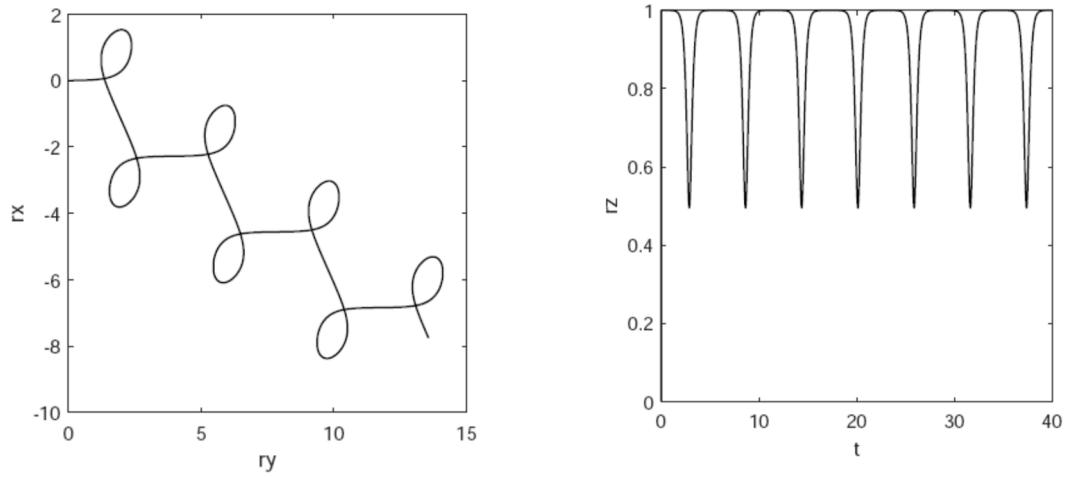


Figure 6.3.6 Disk Centroid rx-ry Trajectory and z-Coordinate vs Time

As a check on constraint error control, Tol = utol = Btol = Htol in Eqs. (6.1.42), (6.1.44), and (6.1.56) is varied in simulations. Data in Table 6.3.2 show that maximum norms of position, velocity, and acceleration constraint norm errors over the simulation interval are driven toward zero, to computer precision, as tolerances are tightened. There is little increase in compute time associated with the tighter tolerances.

Table 6.3.2 Maximum Constraint Error for Rolling Disk

Tol	Position Err.	Velocity Err.	Acceleration Err.
e-6	2e-8	3e-7	e-6
e-9	2e-11	2e-9	2e-8
e-12	3e-14	2e-13	e-12

#### 6.3.4 Ellipsoid Rolling Without Slip on a Moving Surface

A more complex example is the *ellipsoid* of Fig. 6.3.7 that *rolls without slip* on a moving surface, whose geometry is defined by the equation

$$z = d(x_1, y_1) + g(t) \equiv (x_1^2 + 2y_1^2) + \text{amp}(1 - \cos(\epsilon t)) \quad (6.3.23)$$

where variables  $x_1$  and  $y_1$  are coordinates in the x-y plane that locate points on the moving surface. They are not to be confused with the x and y coordinates of the centroid of the ellipsoid. With  $\epsilon$  and amp small, the surface deviates only moderately from the x-y plane.

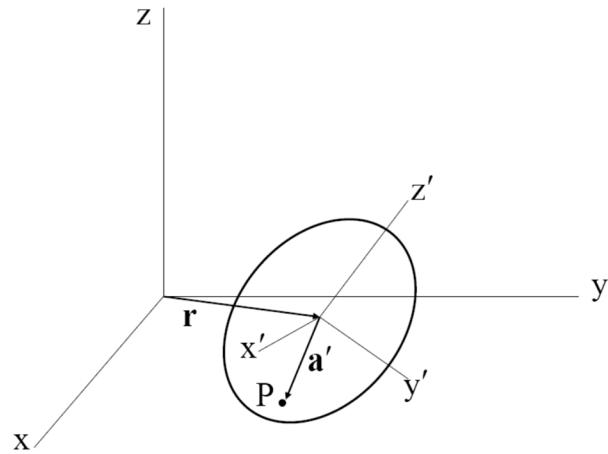


Figure 6.3.7 Ellipsoid Rolling on Moving Surface

The velocity of point  $P_e$  on the ellipsoid that is in contact with the surface, at point  $P_s$  in the surface, is zero relative to the tangent plane of the surface; i.e., it cannot slip on the surface. In terms of body fixed coordinates  $\mathbf{a}' = [a'_x \ a'_y \ a'_z]^T$  in a centroidal  $x'$ - $y'$ - $z'$  body fixed frame in the ellipsoid, the equation that defines the boundary of the ellipsoid is

$$\Psi(\mathbf{a}') = \frac{1}{2} \left( \left( a'_x / \alpha \right)^2 + \left( a'_y / \beta \right)^2 + \left( a'_z / \delta \right)^2 - 1 \right) = \frac{1}{2} (\mathbf{a}'^T \mathbf{L} \mathbf{a}' - 1) = 0 \quad (6.3.24)$$

where  $\mathbf{L} = \text{diag}(1/\alpha^2 \ 1/\beta^2 \ 1/\delta^2)$ .

The vector that locates point  $P_e$  on the boundary of the ellipsoid that is in contact with the surface at time  $t$  is

$$\mathbf{r}^{P_e} = \mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}' \quad (6.3.25)$$

where  $\mathbf{p} = [e_0 \ e^T]^T$  is the vector of *Euler parameters* that defines orientation of the  $x'$ - $y'$ - $z'$  frame relative to the  $x$ - $y$ - $z$  frame, and  $\mathbf{A}(\mathbf{p})$  is the associated orientation transformation matrix. The vector that locates point  $P_s$  on the surface that is in contact with the ellipsoid at time  $t$  is

$$\mathbf{s}^{P_s} = \begin{bmatrix} x_1 \\ y_1 \\ d(x_1, y_1) + g(t) \end{bmatrix} \quad (6.3.26)$$

The condition that points  $P_e$  and  $P_s$  coincide at time  $t$  is the *time dependent holonomic constraint*

$$\mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}' - \mathbf{s}^{P_s} = \mathbf{0} \quad (6.3.27)$$

It is important to note that points  $P_e$  and  $P_s$  are not fixed in the ellipsoid or surface, respectively. Vectors  $\mathbf{r}^{P_e}$  and  $\mathbf{s}^{P_s}$  locate points on trajectories of contact, not points fixed in their

respective surfaces. To emphasize this point, note that if  $g(t)$  is constant, the surface on which the ellipsoid rolls is fixed in space and the velocity of each point on the surface is zero. The time derivative of vector  $\mathbf{s}^{P_s}$  in Eq. (6.3.26), namely  $\dot{\mathbf{s}}^{P_s} = \begin{bmatrix} \dot{x}_1 & \dot{y}_1 & d_{x_1}\dot{x}_1 + d_{y_1}\dot{y}_1 \end{bmatrix}^T$ , is the time derivative of a point on the trajectory of contact, not the velocity of any point fixed in the surface, and likewise for the trajectory of contact points on the ellipse.

The normal to the ellipsoid at point  $P_e$ , represented in the  $x'$ - $y'$ - $z'$  frame, is the transpose of the gradient of the function in Eq. (6.3.24),

$$\mathbf{n}'_e = \Psi_{\mathbf{a}'}^T = \begin{bmatrix} a'_x / \alpha^2 & a'_y / \beta^2 & a'_z / \delta^2 \end{bmatrix}^T \equiv \mathbf{L}\mathbf{a}' \quad (6.3.28)$$

and the normal to the surface is the transpose of the gradient of  $z - d(x_1, y_1) - g(t) = 0$  from Eq. (6.3.23),

$$\mathbf{n}_s = \begin{bmatrix} -d_{x_1} & -d_{y_1} & 1 \end{bmatrix}^T \quad (6.3.29)$$

The vector  $\mathbf{A}(\mathbf{p})\mathbf{n}'_e$  must be collinear with  $\mathbf{n}_s$ , which is enforced by the condition that  $\mathbf{A}(\mathbf{p})\mathbf{n}'_e$  is orthogonal to a basis for the tangent plane to the surface at the point of contact. By inspection, a nonzero linearly independent pair of vectors that are orthogonal to  $\mathbf{n}_s$  is

$$\begin{aligned} \mathbf{w}_1 &= \begin{bmatrix} 1 & 0 & d_{x_1} \end{bmatrix}^T \\ \mathbf{w}_2 &= \begin{bmatrix} 0 & 1 & d_{y_1} \end{bmatrix}^T \end{aligned} \quad (6.3.30)$$

and the required orthogonality conditions are

$$\begin{aligned} \mathbf{w}_1^T \mathbf{A}(\mathbf{p}) \mathbf{L} \mathbf{a}' &= 0 \\ \mathbf{w}_2^T \mathbf{A}(\mathbf{p}) \mathbf{L} \mathbf{a}' &= 0 \end{aligned} \quad (6.3.31)$$

Generalized coordinates for this system are

$$\mathbf{q} = \begin{bmatrix} \mathbf{r}^T & \mathbf{p}^T & \mathbf{a}'^T & x_1 & y_1 \end{bmatrix}^T \in \mathbb{R}^{12}$$

Holonomic constraints on  $\mathbf{q}$ , from Eqs. (6.3.27), (6.3.31), (6.3.24), and the Euler parameter normalization condition, are

$$\Phi(\mathbf{q}) = \begin{bmatrix} \mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}' - \mathbf{s}^P \\ \mathbf{w}_1^T \mathbf{A}(\mathbf{p}) \mathbf{L} \mathbf{a}' \\ \mathbf{w}_2^T \mathbf{A}(\mathbf{p}) \mathbf{L} \mathbf{a}' \\ \frac{1}{2}(\mathbf{p}^T \mathbf{p} - 1) \\ \frac{1}{2}(\mathbf{a}'^T \mathbf{L} \mathbf{a}' - 1) \end{bmatrix} = \mathbf{0} \in \mathbb{R}^7 \quad (6.3.32)$$

The Jacobian of these constraints is

$$\Phi_q = \begin{bmatrix} I_3 & B(p, a') & A(p) & -s_{x_1}^p & -s_{y_1}^p \\ \mathbf{0} & w_1^T B(p, La') & w_1^T A(p)L & a'^T L A^T(p) w_{1x_1} & a'^T L A^T(p) w_{1y_1} \\ \mathbf{0} & w_2^T B(p, La') & w_2^T A(p)L & a'^T L A^T(p) w_{2x_1} & a'^T L A^T(p) w_{2y_1} \\ \mathbf{0} & p^T & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & a'^T L & 0 & 0 \end{bmatrix} \quad (6.3.33)$$

For the paraboloid  $d(x_1, y_1) = (x_1^2 + 2y_1^2)$ , from Eqs. (6.3.26) and (6.3.30),

$$\begin{aligned} s_{x_1}^p &= [1 \ 0 \ 2 \ x_1]^T \\ s_{y_1}^p &= [0 \ 1 \ 4 \ y_1]^T \\ w_{1x_1} &= 2 \ u_z, \quad w_{1y_1} = \mathbf{0}, \quad w_{2x_1} = \mathbf{0}, \quad w_{2y_1} = 4 \ u_z \end{aligned} \quad (6.3.34)$$

The Jacobian of Eq. (6.3.33) thus reduces to

$$\Phi_q = \begin{bmatrix} I_3 & B(p, a') & A(p) & -s_{x_1}^p & -s_{y_1}^p \\ \mathbf{0} & w_1^T B(p, La') & w_1^T A(p)L & 2 u_z^T A(p)La' & \mathbf{0} \\ \mathbf{0} & w_2^T B(p, La') & w_2^T A(p)L & \mathbf{0} & 4 u_z^T A(p)La' \\ \mathbf{0} & p^T & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & a'^T L & 0 & 0 \end{bmatrix} \quad (6.3.35)$$

The velocity of contact point  $P_e$  in the ellipse, for the value of  $a'$  that defines the contact point, is

$$\begin{aligned} v^{P_e} &= \dot{r} + A(p)\tilde{\omega}'a' = \dot{r} + A(p)A^T(p)\dot{A}(p)a' \\ &= \dot{r} + \dot{A}(p)a' = \dot{r} + (A(p)\ddot{a}')_p \dot{p} = \dot{r} + B(p, a')\dot{p} \end{aligned} \quad (6.3.36)$$

The velocity of contact point  $P_s$  in the surface, for values of  $x_1$  and  $y_1$  that define the contact point in Eq. (6.3.26), is

$$v^{P_s} = \dot{g}(t)u_z \quad (6.3.37)$$

where  $u_z$  is the unit vector along the positive z axis. The relative velocity of contact points  $P_s$  in the ellipsoid and  $P_e$  in the surface is thus

$$v^{P_e} - v^{P_s} = \dot{r} + B(p, a')\dot{p} - \dot{g}u_z \quad (6.3.38)$$

In order that the projection of this relative velocity on the tangent plane to the surface at the point of contact is zero; i.e., there is no relative slip between the ellipsoid and surface, it is necessary that

$$\begin{aligned} w_1^T (\dot{r} + B(p, a')\dot{p} - \dot{g}u_z) &= 0 \\ w_2^T (\dot{r} + B(p, a')\dot{p} - \dot{g}u_z) &= 0 \end{aligned} \quad (6.3.39)$$

In matrix form, these *differential constraints* are

$$\mathbf{E}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{w}_1^T & \mathbf{w}_1^T \mathbf{B}(\mathbf{p}, \mathbf{a}') & \mathbf{0} \\ \mathbf{w}_2^T & \mathbf{w}_2^T \mathbf{B}(\mathbf{p}, \mathbf{a}') & \mathbf{0} \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} 2 & \dot{x}_1 \\ 4 & \dot{y}_1 \end{bmatrix} = \mathbf{e} \quad (6.3.40)$$

Derivative expressions required for formulation of the equations of motion are

$$(\mathbf{E}\ddot{\mathbf{q}})_q \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{w}_1^T \left( \mathbf{B}(\dot{\mathbf{p}}, \mathbf{a}') \dot{\mathbf{p}} + (\mathbf{B}(\mathbf{p}, \mathbf{a}'_i) \dot{\mathbf{p}})_{\mathbf{a}'_i} \dot{\mathbf{a}}' \right) + 2 \dot{x}_1 \mathbf{u}_z^T (\dot{\mathbf{r}} + \mathbf{B}(\mathbf{p}, \mathbf{a}') \dot{\mathbf{p}}) \\ \mathbf{w}_2^T \left( \mathbf{B}(\dot{\mathbf{p}}, \mathbf{a}') \dot{\mathbf{p}} + (\mathbf{B}(\mathbf{p}, \mathbf{a}'_i) \dot{\mathbf{p}})_{\mathbf{a}'_i} \dot{\mathbf{a}}' \right) + 4 \dot{y}_1 \mathbf{u}_z^T (\dot{\mathbf{r}} + \mathbf{B}(\mathbf{p}, \mathbf{a}') \dot{\mathbf{p}}) \end{bmatrix} \quad (6.3.41)$$

$$(\Phi_q \ddot{\mathbf{q}})_q \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{B}(\dot{\mathbf{p}}, \mathbf{a}') \dot{\mathbf{p}} + (\mathbf{B}(\mathbf{p}, \mathbf{a}'_i) \dot{\mathbf{p}})_{\mathbf{a}'_i} \dot{\mathbf{a}}' + \mathbf{B}(\dot{\mathbf{p}}, \dot{\mathbf{a}}') \mathbf{p} - 2 (\dot{x}_1)^2 \mathbf{u}_z - 4 (\dot{y}_1)^2 \mathbf{u}_z \\ \mathbf{w}_1^T \left( \mathbf{B}(\dot{\mathbf{p}}, \mathbf{L}\mathbf{a}') \dot{\mathbf{p}} + \mathbf{B}(\dot{\mathbf{p}}, \mathbf{L}\dot{\mathbf{a}}') \mathbf{p} + (\mathbf{B}(\mathbf{p}, \mathbf{a}'_i) \dot{\mathbf{p}})_{\mathbf{a}'_i} \mathbf{L}\dot{\mathbf{a}}' \right) + \mathbf{c}_1 \\ \mathbf{w}_2^T \left( \mathbf{B}(\dot{\mathbf{p}}, \mathbf{L}\mathbf{a}') \dot{\mathbf{p}} + \mathbf{B}(\dot{\mathbf{p}}, \mathbf{L}\dot{\mathbf{a}}') \mathbf{p} + (\mathbf{B}(\mathbf{p}, \mathbf{a}'_i) \dot{\mathbf{p}})_{\mathbf{a}'_i} \mathbf{L}\dot{\mathbf{a}}' \right) + \mathbf{c}_2 \\ \dot{\mathbf{p}}^T \dot{\mathbf{p}} \\ \dot{\mathbf{a}}'^T \mathbf{L} \dot{\mathbf{a}}' \end{bmatrix} \quad (6.3.42)$$

where, from Eq. (2.6.81),

$$\begin{aligned} (\mathbf{B}(\mathbf{p}, \mathbf{a}'_i) \dot{\mathbf{p}})_{\mathbf{a}'_i} &= 2(\mathbf{E}(\mathbf{p}) \mathbf{G}^T(\dot{\mathbf{p}})) \\ \mathbf{c}_1 &= 4 \dot{x}_1 \mathbf{u}_z^T (\mathbf{B}(\dot{\mathbf{p}}, \mathbf{L}\mathbf{a}') \mathbf{p} + \mathbf{A}(\mathbf{p}) \mathbf{L}\dot{\mathbf{a}}') \\ \mathbf{c}_2 &= 8 \dot{y}_1 \mathbf{u}_z^T (\mathbf{B}(\dot{\mathbf{p}}, \mathbf{L}\mathbf{a}') \mathbf{p} + \mathbf{A}(\mathbf{p}) \mathbf{L}\dot{\mathbf{a}}') \end{aligned} \quad (6.3.43)$$

Time derivatives of holonomic constraints of Eq. (6.3.32) are  $\Phi_{qt} = \mathbf{0}$  and

$$\begin{aligned} \Phi_t &= \begin{bmatrix} -\dot{\mathbf{g}} \mathbf{u}_z \\ \mathbf{0}_{4 \times 1} \end{bmatrix} \\ \Phi_{tt} &= \begin{bmatrix} -\ddot{\mathbf{g}} \mathbf{u}_z \\ \mathbf{0}_{4 \times 1} \end{bmatrix} \end{aligned} \quad (6.3.44)$$

Derivatives of the right side of the differential constraints of Eq. (6.3.40) are

$$\begin{aligned} \mathbf{e}_t &= \begin{bmatrix} 2 & \dot{x}_1 \\ 4 & \dot{y}_1 \end{bmatrix} \ddot{\mathbf{g}} \\ \mathbf{e}_q \dot{\mathbf{q}} &= \begin{bmatrix} 2 & \dot{x}_1 \\ 4 & \dot{y}_1 \end{bmatrix} \dot{\mathbf{g}} \end{aligned} \quad (6.3.45)$$

and  $\mathbf{E}_t = \mathbf{0}$ . Finally, right side vectors in Eqs. (6.1.31) and (6.1.32) are

$$\begin{aligned} v &= \begin{bmatrix} -\Phi_t \\ e \end{bmatrix} \\ \gamma &= \begin{bmatrix} (\Phi_q \ddot{q})_q + \Phi_{tt} \\ (E \ddot{q})_q - e_q \dot{q} - e_t \end{bmatrix} \end{aligned} \quad (6.3.46)$$

Initial conditions  $\mathbf{q}^0$  on  $\mathbf{q}$  are readily defined to satisfy Eq. (6.3.32), using physical reasoning. Similarly, initial velocity and angular velocity  $\omega'^0$  are readily defined to be consistent with the no-slip condition and  $\dot{\mathbf{p}}^0 = 0.5\mathbf{G}(\mathbf{p}^0)^T \omega'^0$  of Eq. (2.6.64) defines associated Euler parameter time derivatives. Initial values for  $\dot{\mathbf{a}}'$ ,  $\dot{x}_1$ , and  $\dot{y}_1$  are not easily defined on physical grounds. Solving the velocity equations of Eq. (6.1.31), augmented with specified initial conditions at  $t^0$ , resolves this issue.

With density  $\rho$ , the mass of a homogeneous ellipsoid is  $m = 4\rho\pi\alpha\beta\delta/3$  and its moments of inertia  $J_{xx'} = m(\beta^2 + \delta^2)/5$ ,  $J_{yy'} = m(\alpha^2 + \delta^2)/5$ , and  $J_{zz'} = m(\beta^2 + \alpha^2)/5$  define the inertia matrix as  $\mathbf{J}' = \text{diag}(J_{xx'}, J_{yy'}, J_{zz'})$ . As a numerical example, the density is selected as  $\rho = 140 \text{ kg/m}^3$ . With the centroidal body-fixed reference frame used, terms required in the equations of motion are

$$\begin{aligned} \mathbf{M} &= \text{blockdiag}\left(m\mathbf{I}_3, 4\bar{\mathbf{G}}^T(\mathbf{p})\mathbf{J}'\bar{\mathbf{G}}(\mathbf{p}), \mathbf{I}_5\right) \\ \mathbf{S} &= \left[0, 0, 0, \left(8\bar{\mathbf{G}}^T(\dot{\mathbf{p}})\mathbf{J}'\bar{\mathbf{G}}(\dot{\mathbf{p}})\mathbf{p}\right)^T, 0, 0, 0, 0, 0\right]^T \\ \mathbf{Q}^A &= \left[\left(-mg\mathbf{u}_z - \mathbf{K}\mathbf{r}\right)^T, 0, 0, 0, 0, 0, 0, 0, 0\right]^T \end{aligned} \quad (6.3.47)$$

where the generalized force is due to gravity,  $g = 9.8 \text{ m/sec}^2$ , that acts in the negative z-direction and a spring with stiffness  $K \text{ N/m}$  that acts between the centroid of the ellipsoid and the origin of the x-y-z frame.

Simulations are carried out with Code 6.3.2 of Appendix 6.A, using constant step size *RK4* and variable step size *RKF45*, with error tolerance  $e-5$ , to study the effects of the geometry of the ellipsoid, initial angular velocity of the ellipsoid, motion of the surface, and reducing the surface to the x-y plane. A body of literature on conservation properties of dynamics of a body rolling without slip on a surface (Garcia-Naranjo and Marrero, 2013; Bloch., 2003) suggests that conservation of momentum and other measures of dynamic response depend on the ellipsoid having an axis of symmetry, which is not the case if  $\alpha \neq \beta$ . Checks are made with simulation to evaluate such conservative properties.

Simulations are first carried out with  $\alpha = 0.5$ ,  $\beta = 0.4$ ,  $\delta = 0.3$ ,  $\rho = 140 \text{ kg/m}^3$ , and  $K = 150 \text{ N/m}$ ; motion parameters  $\text{amp} = 0.1 \text{ m}$  and  $\omega = \omega_m = 3 \text{ rad/sec}$ ; surface parameter  $\gamma = 0.1$ ; and the initial configuration

$$\mathbf{q}^0 = \left[\mathbf{r}^{0T} \quad \mathbf{p}^{0T} \quad \mathbf{a}'^{0T} \quad x_1^0 \quad y_1^0\right]^T = [0, 0, 0.3, 1, 0, 0, 0, 0, 0, -0.3, 0, 0]^T$$

Initial velocity and angular velocity components are specified as  $\dot{x}^0 = \dot{y}^0 = 0.3$  m/sec and  $\omega_{z'}^0 = 10$  rad/sec. These conditions are appended to Eq. (6.1.8), to determine the complete set of initial velocities,  $\dot{\mathbf{q}}^0 = [0.3, 0.3, 0, 0, -0.5, -0.5, 5, 1, 6.78, 0, 1, 6.78]^T$ . The initial integration step size is selected as  $h = 0.005$  sec, subsequently adjusted by the algorithm with an upper limit of 0.01 sec, and the simulation final time is  $t_f = 10$  sec. Results are shown in Fig. 6.3.8.

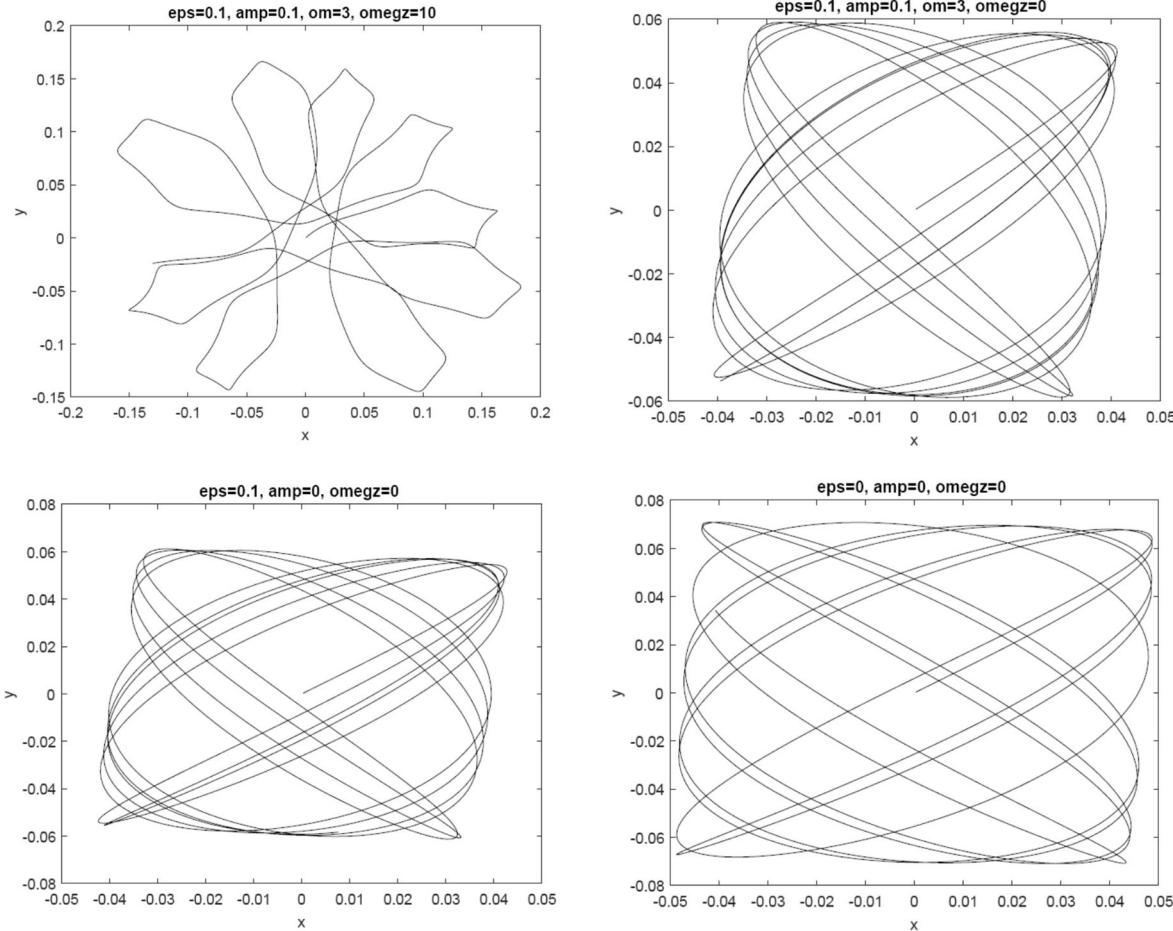


Figure 6.3.8 Simulations with Asymmetric Ellipsoid;  $\epsilon = 0.5$ ,  $a = 0.4$ ,  $b = 0.3$  m

Plots in Fig. 6.3.8 show x-y trajectories of the centroid of the ellipsoid. Captions on the four plots indicate a range from motion of the surface with nonzero initial vertical angular velocity of the ellipsoid at the upper left to rolling on the x-y plane with zero initial angular velocity of the ellipsoid at the lower right. In the upper left simulation, both *total energy* and vertical momentum vary, the former as a result of energy being introduced due to motion imposed by the surface. In the upper right simulation, even though the initial vertical momentum is zero, its value and that of total energy vary over time, due to motion of the surface. When motion of the surface is turned off, as in the simulation at the lower left, total energy is conserved but vertical momentum varies. With rolling on the x-y plane shown at the lower right, total energy is constant, but vertical momentum varies. The variation of vertical momentum in all four simulations is consistent with theoretical results (Garcia-Naranjo and Marrero, 2013; Bloch., 2003) associated with an ellipsoid that has no axis of symmetry. As is clear from trajectories in

all four simulations, the effects of shape and motion of the surface and initial vertical angular velocity of the ellipsoid lead to rather distinct motions.

The preceding four simulations are repeated for a sphere of radius  $R = h = 0.4$  m, with results shown in Fig. 6.3.9. For motion of the surface and nonzero initial vertical angular velocity in the simulation shown at the upper left, both total energy and vertical momentum vary with time. Even with zero initial vertical angular velocity for the simulation shown at the upper right, total energy and vertical momentum vary with time, due to shape and motion of the surface. With motion of the surface turned off, as shown at the lower left, total energy is constant but vertical momentum varies with time, likely due to the shape of the surface. Finally, as shown at the lower right, with motion on the plane, a simple straight line trajectory occurs and both total energy and vertical momentum are constant.

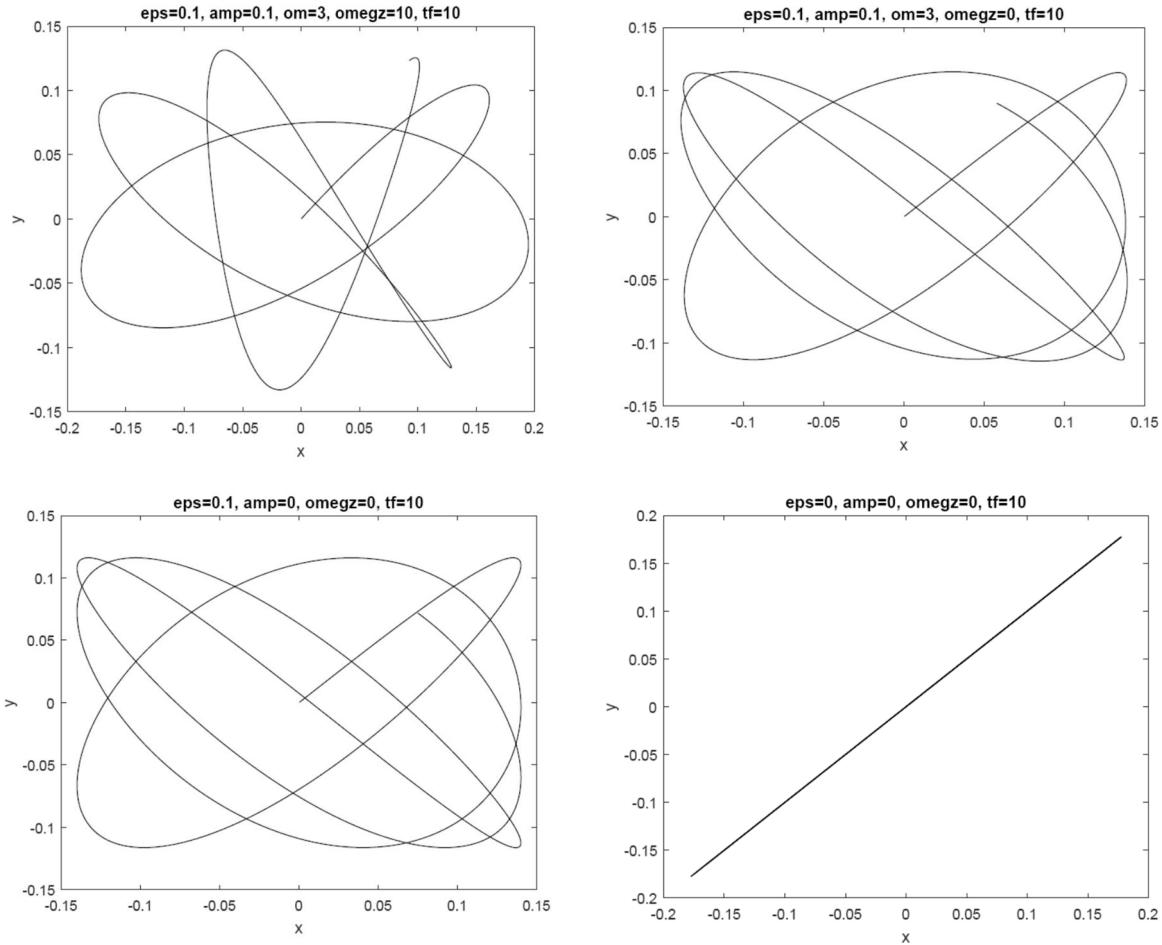


Figure 6.3.9 Simulations with Sphere;  $R = h = 0.4$  m

In the most severe simulation at the upper left of Fig. 6.3.8, 39 reparameterizations were required in 1970 time steps (50 time steps per reparameterization), all due to a unit limit on norm of  $\mathbf{v}$ . In most other simulations, only one parameterization was required for 2,000 time steps. In all cases, computing cost for reparameterization was negligible.

As a check on *constraint error control*, the tolerance  $Tol = utol = Btol = Htol$  in Eqs. (6.1.42), (6.1.44), and (6.1.56) is varied for the most severe simulation at the upper left of Fig.

6.3.9, carried out with the RK4 constant step size method. Results are shown in Table 6.3.3. Data in the table show that position, velocity, and acceleration constraint errors are driven toward zero, to computer precision, as tolerances are tightened. There was little increase in computing times associated with the tighter tolerances.

Table 6.3.3 Constraint Error for Ellipsoid on Moving Surface

Tol	Position Err.	Velocity Err.	Acceleration Err.
e-6	e-6	e-8	e-6
e-9	e-8	e-14	e-13
e-12	e-11	e-14	e-13

Two planar vehicle examples and two spatial models of substantial complexity are presented, the latter three involving mixed holonomic-nonholonomic systems, using expressions derived for derivatives that are required for implementation of explicit integration algorithms. Solutions obtained using MATLAB codes of Appendix 6.A are shown to satisfy all three forms of kinematic constraint and yield practical solutions. While the explicitly derived spatial examples involve intricate expressions, it is clear that a computer implementation with standard modeling elements is possible.

## 6.4 Index 0 DAE Formulation for Nonholonomic Systems

As with holonomic systems in Chapter 5, the *nonholonomic ODE* of Section 6.2 are converted to an equivalent *Index 0 DAE*, through introduction of Lagrange multipliers that are available for computation of *constraint reaction forces*. Explicit and implicit numerical integration methods of Section 4.8 are adapted for solution of the Index 0 DAE.

### 6.4.1 Index 0 DAE for Nonholonomic Systems

*D'Alembert's equations of motion* of Eq. (4.6.17) is

$$\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0 \quad (6.4.1)$$

which must hold for all  $\delta\mathbf{q}$  that satisfy Eq. (6.1.33),

$$\mathbf{C}(\mathbf{q}, t)\delta\mathbf{q} = \mathbf{0} \quad (6.4.2)$$

The *Lagrange multiplier theorem* implies existence of a vector  $\lambda$  such that

$$\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}^T(\mathbf{q}, t)\lambda - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0$$

for arbitrary  $\delta\mathbf{q}$ . Thus,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}^T(\mathbf{q}, t)\lambda - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) = 0 \quad (6.4.3)$$

Substituting  $\ddot{\mathbf{q}}$  of Eq. (6.1.62),

$$\ddot{\mathbf{q}} = \mathbf{K}\dot{\mathbf{w}} - \mathbf{X}\mathbf{H}\gamma \quad (6.4.4)$$

this is

$$\mathbf{M}\mathbf{K}\dot{\mathbf{w}} + \mathbf{C}^T\lambda - (\mathbf{M}\mathbf{X}\mathbf{H}\gamma + \mathbf{Q}^A + \mathbf{S}) = \mathbf{0} \quad (6.4.5)$$

Equation (6.4.5) is a DAE, since it involves the algebraic variable  $\lambda$ .

Using Eq. (6.1.54),

$$\mathbf{C}\mathbf{K} = \mathbf{C}(\mathbf{I} - \mathbf{X}\mathbf{H}\mathbf{C})\mathbf{W} = (\mathbf{C} - \mathbf{C}\mathbf{X}\mathbf{H}\mathbf{C})\mathbf{W} = (\mathbf{C} - \mathbf{C})\mathbf{W} = \mathbf{0} \quad (6.4.6)$$

Multiplying Eq. (6.4.5) on the left by  $\mathbf{K}^T$  yields

$$\begin{aligned} & \mathbf{K}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{K}^T \mathbf{C}^T \lambda - \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S}) \\ &= \mathbf{K}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} - \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S}) = \mathbf{0} \end{aligned} \quad (6.4.7)$$

Which is the ODE of the second of Eqs. (6.2.5).

Conversely, substituting  $\mathbf{K}^T = \mathbf{W}^T (\mathbf{I} - \mathbf{C}^T \mathbf{H}^T \mathbf{X}^T)$  from Eq. (6.1.59) into the second of Eqs. (6.2.5) and using  $\mathbf{Q} \equiv \mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S}$  to simplify notation,

$$\mathbf{W}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} - \mathbf{W}^T \mathbf{C}^T \mathbf{H}^T \mathbf{X}^T \mathbf{M} \mathbf{K} \dot{\mathbf{w}} = \mathbf{W}^T \mathbf{Q} - \mathbf{W}^T \mathbf{C}^T \mathbf{H}^T \mathbf{X}^T \mathbf{Q} \quad (6.4.8)$$

Defining

$$\lambda \equiv -\mathbf{H}^T \mathbf{X}^T (\mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{Q}) \quad (6.4.9)$$

Eq. (6.4.8) becomes

$$\mathbf{W}^T (\mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C} \lambda - \mathbf{Q}) = \mathbf{0} \quad (6.4.10)$$

Multiplying Eq. (6.4.9) on the left by  $\mathbf{X}^T \mathbf{C}^T$  yields

$$\mathbf{X}^T \mathbf{C}^T \lambda = -\mathbf{X}^T \mathbf{C}^T \mathbf{H}^T \mathbf{X}^T (\mathbf{M} \mathbf{K} \dot{\mathbf{w}} - \mathbf{Q}) \quad (6.4.11)$$

Taking the transpose of  $\mathbf{H} \mathbf{C} \mathbf{X} = \mathbf{I}$  from Eq. (6.1.31),  $\mathbf{X}^T \mathbf{C}^T \mathbf{H}^T = \mathbf{I}$  and Eq. (6.4.11) reduces to

$$\mathbf{X}^T (\mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C}^T \lambda - \mathbf{Q}) \quad (6.4.12)$$

Combining Eqs. (6.4.10) and (6.4.12)

$$[\mathbf{W} \quad \mathbf{X}]^T (\mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C}^T \lambda - \mathbf{Q}) = \mathbf{0}$$

Since the columns of  $\mathbf{W}$  and  $\mathbf{X}$  span  $\mathbb{R}^{ngc}$ ,  $[\mathbf{W} \quad \mathbf{X}]$  is nonsingular and

$$\mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C}^T \lambda - \mathbf{Q} = \mathbf{0}$$

Substituting the above definition of  $\mathbf{Q}$ , this is

$$\mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C}^T \lambda - (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S}) = \mathbf{0} \quad (6.4.13)$$

which is Eq. (6.4.5).

By definition of *differentiation index* (Ascher and Petzold, 1998), Eq. (6.4.5) is an *Index 0 DAE*, as is the first order system obtained by combining it with the kinematic differential equation of Eq. (6.1.63); i.e.,

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{V}^T \mathbf{K} \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C}^T \lambda &= \mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S} \end{aligned} \quad (6.4.14)$$

Writing Eqs. (6.4.14) in matrix form,

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \mathbf{K} & \mathbf{C}^T \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{w}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{V}^T \mathbf{K} \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{S} + \mathbf{Q}^A \end{bmatrix} \quad (6.4.15)$$

For existence of a unique solution, the coefficient matrix on the left must be nonsingular. At a parameterization point  $(\mathbf{q}^0, t^0)$  that satisfies Eqs. (6.1.24), Eqs. (6.1.48) and (6.1.59) show that  $\mathbf{K}^0 = \mathbf{W}$ . To show that the coefficient matrix on the left of Eq. (6.4.15) at  $(\mathbf{q}^0, t^0)$  is nonsingular with  $\mathbf{K}^0 = \mathbf{W}$  and  $\mathbf{C}^{0T} = \mathbf{X}$ , set

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \mathbf{W} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{d} \end{bmatrix} = \mathbf{0} \quad (6.4.16)$$

If this equation holds only for  $\alpha = \mathbf{0}$ ,  $\beta = \mathbf{0}$ , and  $\delta = \mathbf{0}$ , then the coefficient matrix is nonsingular. The first row of Eq. (6.4.16) is  $\alpha = \mathbf{0}$  and the second is  $\mathbf{M}\mathbf{W}\beta + \mathbf{X}\delta = \mathbf{0}$ . Multiplying both sides on the left by  $\mathbf{W}^T$ ,  $\mathbf{W}^T\mathbf{M}\mathbf{W}\beta + \mathbf{W}^T\mathbf{X}\delta = \mathbf{W}^T\mathbf{M}\mathbf{W}\beta = \mathbf{0}$ . It is shown in Section 4.6.2 that  $\mathbf{M}$  is positive definite on the null space of  $\mathbf{C}$ , so the matrix  $\mathbf{W}^T\mathbf{M}\mathbf{W}$  is *positive definite*, hence nonsingular, and  $\beta = \mathbf{0}$ . The second of Eqs. (6.4.16) thus reduces to  $\mathbf{X}\delta = \mathbf{0}$  and, since  $\mathbf{X}$  has full column rank,  $\delta = \mathbf{0}$ . This shows that the matrix on the left of Eq. (6.4.16) is nonsingular. The coefficient matrix of Eq. (6.4.15) is thus nonsingular at  $(\mathbf{q}^0, t^0)$  and, since functions in the matrix are continuous in  $\mathbf{q}$  and  $t$ , the coefficient matrix on the left of Eq. (6.4.15) is nonsingular in a neighborhood of  $(\mathbf{q}^0, t^0)$ . This result is the basis for applying explicit and implicit ODE integrators for numerical solution of the Index 0 DAE of Eqs. (6.4.14).

Define  $\mathbf{y} = [\mathbf{v}^T \quad \mathbf{w}^T]^T$  and

$$\mathbf{f}(t, \mathbf{y}) = \begin{bmatrix} \mathbf{f}_1(t, \mathbf{y}) \\ \mathbf{f}_2(t, \mathbf{y}) \end{bmatrix} = \begin{bmatrix} \mathbf{V}^T \mathbf{K} \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ (\mathbf{K}^T \mathbf{M} \mathbf{K})^{-1} \mathbf{K}^T (\mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{S} + \mathbf{Q}^A) \end{bmatrix} \quad (6.4.17)$$

where Eqs. (6.1.45) and (6.1.58) define  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  as functions of  $\mathbf{v}$  and  $\mathbf{w}$ , hence  $\mathbf{y}$ . The first order ODE of Eq. (6.2.5) are thus  $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ , which is equivalent to the Index 0 DAE of Eq. (6.4.14). Initial conditions of Eqs. (6.2.6) are

$$\mathbf{y}^0 = \begin{bmatrix} \mathbf{v}^0 \\ \mathbf{w}^0 \end{bmatrix} = \mathbf{0} \quad (6.4.18)$$

The first order initial-value ODE problem is thus

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{f}(t, \mathbf{y}) \\ \mathbf{y}^0 &= \mathbf{0} \end{aligned} \quad (6.4.19)$$

#### 6.4.2 Explicit Integration of Nonholonomic Index 0 DAE

**Explicit integration of the Index 0 DAE initial-value problem** of Eqs. (6.4.14) and (6.4.18), using explicit Runge-Kutta methods, is as follows:

- (1) With initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t^0$  that satisfy Eqs. (6.1.24) and (6.1.31), use *singular value decomposition* to determine  $\mathbf{V}$  and  $\mathbf{W}$  that satisfy Eqs. (6.1.36) and (6.1.48). Evaluate  $\mathbf{B}$  and  $\mathbf{H}$  of Eqs. (6.1.43) and (6.1.54) at  $t^0$  and evaluate  $\mathbf{U}$  and  $\mathbf{X}$  of Eqs. (6.1.36) and (6.1.49).
- (2) At time step  $t_i$ , solve Eq. (6.4.15) for  $\dot{\mathbf{w}}_i$ ,  $\dot{\mathbf{v}}_i$ , and  $\lambda_i$  and apply an explicit Runge-Kutta numerical integrator to obtain  $\mathbf{w}_{i+1}$  and  $\mathbf{v}_{i+1}$ . Evaluate  $\mathbf{B}$  using Eq. (6.1.44) and  $\mathbf{H}$  using Eq. (6.1.56). Use Eqs. (6.1.45), (6.1.58), and (6.1.62) to evaluate  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  on the time grid.
- (3) Monitor the *condition number* of the coefficient matrix on the left of Eq. (6.4.15), the norms of  $\mathbf{v}$  and  $\mathbf{w}$ , and the number of iterations required to evaluate  $\mathbf{u}$ ,  $\mathbf{B}$ , and  $\mathbf{H}$ . If tolerances are exceeded, define a new time  $\bar{t}^0$  and associated  $\bar{\mathbf{q}}^0$  and  $\bar{\dot{\mathbf{q}}}^0$ . Repeat Step (1) to

define a new *parameterization* and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\mathbf{w}}^0$ . This process follows the trajectory shown in Fig. 5.2.2, moving smoothly across the constraint set.

(4) Continue the process until the final time  $t_f$  is reached, or a singular configuration associated with a faulty design or model occurs.

A key aspect of this algorithm is *control of error in satisfying constraints* of Eqs. (6.1.24), (6.1.31), and (6.1.32). These errors can be controlled with adequate tolerances  $utol$ ,  $Btol$ , and  $Htol$  in iterative solution for  $\mathbf{u}$ ,  $\mathbf{B}$ , and  $\mathbf{H}$ . While the value of  $\lambda_i$  determined in Step (2) is not used in integration, it is available for computation of constraint reaction forces.

### 6.4.3 Derivatives for Implicit Numerical Integration

To use an implicit numerical integration method in solving Eqs. (6.4.14) and (6.4.18), derivatives of all terms appearing in Eq. (6.4.14) with respect to  $\mathbf{v}$ ,  $\mathbf{w}$ , and  $\lambda$  are required. For the reader who is satisfied with the use of explicit numerical integration methods, the somewhat intricate remainder of this section can be bypassed.

If Eqs. (6.4.14) are to be solved numerically with an implicit numerical integration method, a convenient form is the *residual equation*

$$\mathbf{R} \equiv \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{v}} - \mathbf{V}^T \mathbf{K} \mathbf{w} - \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 - \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C}^T \lambda - \mathbf{M} \mathbf{X} \mathbf{H} \gamma - \mathbf{S} - \mathbf{Q}^A \end{bmatrix} = \mathbf{0} \quad (6.4.20)$$

This form of the equations of motion in terms of  $\dot{\mathbf{v}}$ ,  $\dot{\mathbf{w}}$ ,  $\lambda$ ,  $\mathbf{q}$ , and  $\dot{\mathbf{q}}$  may be used to iteratively solve discretized equations that are based on implicit numerical integration formulas. To implement this approach, expressions for derivatives of terms in Eq. (6.4.20) with respect to  $\dot{\mathbf{v}}$ ,  $\dot{\mathbf{w}}$ ,  $\mathbf{v}$ ,  $\mathbf{w}$ , and  $\lambda$  must be derived. Fortunately, most terms in Eqs. (6.4.20) appear in Eq. (6.2.5), so derivative terms presented in Section 6.2 are applicable here. The only term in Eqs. (6.4.20) that does not appear in Eqs. (6.2.5) is  $\mathbf{C}^T \lambda$ . Fortunately, its derivative with respect to  $\mathbf{q}$  is the term

$$\mathbf{P}_4(\mathbf{q}, \lambda) \equiv \left( \mathbf{C}^T(\mathbf{q}, t) \lambda \right)_q \quad (6.4.21)$$

that was defined in Eq. (5.3.19) and expanded for computation in Eq. (5.3.35).

The easy part of the Jacobian of the residual in Eq. (6.4.20) is

$$\mathbf{R}_{\dot{\mathbf{v}}, \dot{\mathbf{w}}, \lambda} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \mathbf{K} & \mathbf{C}^T \end{bmatrix} \quad (6.4.22)$$

Kinematic derivatives  $\mathbf{q}_v$ ,  $\mathbf{q}_w$ ,  $\dot{\mathbf{q}}_v$ , and  $\dot{\mathbf{q}}_w$  are identical to Eq. (6.2.12), repeated here as

$$\begin{aligned} \mathbf{q}_v &= \mathbf{V} - \mathbf{U} \mathbf{h}_v \\ \mathbf{q}_w &= \mathbf{0} \\ \dot{\mathbf{q}}_v &= \left( (\mathbf{K} \ddot{\mathbf{w}})_q - \mathbf{X} \left( \mathbf{H} \ddot{\mathbf{C}} \dot{\mathbf{q}}^0 \right)_q - \mathbf{X} \mathbf{H} \left( \mathbf{C} \dot{\mathbf{q}}^0 \right)_q + \mathbf{X} \left( \mathbf{H} \ddot{\mathbf{v}} \right)_q + \mathbf{X} \mathbf{H} \mathbf{v}_q \right) \mathbf{q}_v \\ \dot{\mathbf{q}}_w &= \mathbf{K} \end{aligned} \quad (6.4.23)$$

Using these results, derivatives of the residual of Eq. (6.4.20) are

$$\begin{aligned}
\mathbf{R}_{1v} &= \left( -\mathbf{V}^T (\mathbf{K}\ddot{\mathbf{w}})_q + \mathbf{V}^T \mathbf{X} \left( \mathbf{H} \ddot{\mathbf{C}} \dot{\mathbf{q}}^0 \right)_q + \mathbf{V}^T \mathbf{X} \mathbf{H} \left( \mathbf{C} \dot{\mathbf{q}}^0 \right)_q - \mathbf{V}^T \mathbf{X} (\mathbf{H} \ddot{\mathbf{v}})_q - \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v}_q \right) \dot{\mathbf{q}}_v \\
\mathbf{R}_{1w} &= -\mathbf{V}^T \mathbf{K} \\
\mathbf{R}_{2v} &= \left( \mathbf{M} \left( \mathbf{K} \ddot{\mathbf{w}} \right)_q + \left( \mathbf{M} \mathbf{K} \ddot{\mathbf{w}} \right)_q - \left( \mathbf{M} \ddot{\mathbf{X}} \ddot{\mathbf{H}} \dot{\gamma} \right)_q - \mathbf{M} \mathbf{X} \left( \mathbf{H} \dot{\gamma} \right)_q - \mathbf{M} \mathbf{X} \mathbf{H} \gamma_q - \mathbf{Q}_q^A - \mathbf{S}_q \right) \dot{\mathbf{q}}_v \quad (6.4.24) \\
&\quad + \mathbf{P}4(\mathbf{q}, \lambda) - \left( \mathbf{M} \mathbf{X} \mathbf{H} \gamma_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A + \mathbf{S}_{\dot{q}} \right) \dot{\mathbf{q}}_v \\
\mathbf{R}_{2w} &= -\left( \mathbf{M} \mathbf{X} \mathbf{H} \gamma_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A + \mathbf{S}_{\dot{q}} \right) \dot{\mathbf{q}}_w
\end{aligned}$$

Derivative expressions for  $(\mathbf{C}\ddot{\mathbf{a}})_q$ ,  $(\mathbf{H}\ddot{\mathbf{b}})_q$ , and  $(\mathbf{K}\ddot{\mathbf{c}})_q$  are given in Eqs. (6.2.15), (6.2.16), and (6.2.17);  $\gamma$ ,  $\gamma_q$ , and  $\gamma_{\dot{q}}$  are given in Eqs. (6.2.19) and (6.2.22); and kinetic derivatives  $(\mathbf{M}(\mathbf{q})\ddot{\mu})_q$ ,  $\mathbf{S}_q$ ,  $\mathbf{S}_{\dot{q}}$ ,  $\mathbf{Q}_q^A$ , and  $\mathbf{Q}_{\dot{q}}^A$  are presented in Appendix 5.B. These results enable evaluation of all terms in Eq. (6.4.24).

#### 6.4.4 Implicit Numerical Integration Algorithms

Implicit numerical integration algorithms of Section 4.8.1 for solution of first order ODE are applied for solution of Eqs. (6.4.14) and (6.4.18). Since *trapezoidal integration* formulas are stated in terms of generalized coordinates, their application is rather simple, as presented in Section 6.4.4.1. Implicit Runge-Kutta integration formulas, in contrast, are stated in terms of *stage variables* that are not generalized coordinates. They are applied to the explicit ODE of Eq. (6.4.19) and transformed for solution of Eq. (6.4.14) in Section 6.4.4.2.

##### 6.4.4.1 Trapezoidal Integration

Implicit *trapezoidal integration* formulas for  $\dot{\mathbf{v}}$  and  $\dot{\mathbf{w}}$  of Eq. (4.8.40) are

$$\begin{aligned}
\mathbf{v}_n &= \mathbf{v}_{n-1} + (h/2)(\dot{\mathbf{v}}_{n-1} + \dot{\mathbf{v}}_n) \\
\mathbf{w}_n &= \mathbf{w}_{n-1} + (h/2)(\dot{\mathbf{w}}_{n-1} + \dot{\mathbf{w}}_n)
\end{aligned} \quad (6.4.25)$$

These equations are used with Eqs. (6.1.45) and (6.1.58) to evaluate  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  as functions of  $\dot{\mathbf{v}}_n$  and  $\dot{\mathbf{w}}_n$ , for use in Eq. (6.4.20), yielding

$$\mathbf{R}(\dot{\mathbf{v}}_n, \dot{\mathbf{w}}_n) = \begin{bmatrix} \dot{\mathbf{v}}_n - \mathbf{V}^T \mathbf{K} (\mathbf{w}_{n-1} + (h/2)(\dot{\mathbf{w}}_{n-1} + \dot{\mathbf{w}}_n)) - \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 - \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} \\ \mathbf{M} \mathbf{K} \dot{\mathbf{w}}_n + \mathbf{C}^T \lambda - \mathbf{M} \mathbf{X} \mathbf{H} \gamma - \mathbf{S} - \mathbf{Q}^A \end{bmatrix} = \mathbf{0} \quad (6.4.26)$$

Equations (6.4.25) allow all arguments of functions that appear in Eq. (6.4.26) to be written as functions of  $\dot{\mathbf{v}}_n$  and  $\dot{\mathbf{w}}_n$ . Using the chain rule of differentiation, with relations derived in Section 6.4.2, the Jacobian of the residual of Eq. (6.4.26) with respect to  $\mathbf{z}_n = [\dot{\mathbf{v}}_n^T \quad \dot{\mathbf{w}}_n^T \quad \lambda_n^T]^T$  is

$$\mathbf{J}^{\text{trap}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \mathbf{K} & \mathbf{C}^T \end{bmatrix} + (h/2) \begin{bmatrix} \mathbf{R1}_v & \mathbf{R1}_w & \mathbf{0} \\ \mathbf{R2}_v & \mathbf{R2}_w & \mathbf{0} \end{bmatrix} \equiv \mathbf{J}_1 + (h/2) \mathbf{J}_2 \quad (6.4.27)$$

At  $t^0$ , with  $h$  small, the first matrix dominates. As shown in Section 6.4.1, it is nonsingular, so the Jacobian is nonsingular in a neighborhood of  $\mathbf{q}^0$ , for  $h$  sufficiently small.

Newton-Raphson iteration for solution of Eq. (6.4.26) is

$$\begin{aligned}\mathbf{J}^{\text{trap}} \Delta \mathbf{z}^i &= -\mathbf{R}^i \\ \mathbf{z}^{i+1} &= \mathbf{z}^i + \Delta \mathbf{z}^i\end{aligned}\quad i = 0, 1, \dots \text{ until } \|\mathbf{R}^i\| < \text{intol} \quad (6.4.28)$$

At  $t^0$ , with initial conditions of Eq. (6.4.18) and  $h = 0$ , Eq. (6.4.26) is solved to obtain an estimate  $\mathbf{z}^0 \equiv [\dot{\mathbf{v}}^{0T} \quad \dot{\mathbf{w}}^{0T}]^T$  to start the numerical integration process. At subsequent time steps, good estimates for  $\mathbf{z}_n^0$  are available from prior time steps. Bounds on the norms of  $\mathbf{v}$  and  $\mathbf{w}$ , the number of iterations in evaluating  $\mathbf{u}$ ,  $\mathbf{B}$ , and  $\mathbf{H}$ , and the *condition number* of the coefficient matrix of Eq. (6.4.28) are used to determine whether a *reparameterization* is required. If not, the integration process is continued. If so, the current time is set as  $\bar{t}^0$  and the current value of  $\mathbf{q}$  is used as  $\bar{\mathbf{q}}^0$  to redefine  $\bar{\mathbf{V}}$ ,  $\bar{\mathbf{U}}$ ,  $\bar{\mathbf{W}}$ ,  $\bar{\mathbf{X}}$ ,  $\bar{\mathbf{B}}$ , and  $\bar{\mathbf{H}}$ . The process is restarted with initial conditions  $\bar{\mathbf{v}}^0 = \mathbf{0}$  and  $\bar{\mathbf{w}}^0 = \mathbf{0}$  and estimates for their time derivatives  $\bar{\mathbf{v}}^0 = \bar{\mathbf{V}}^T \bar{\mathbf{q}}^0$  and  $\bar{\mathbf{w}}^0 = \bar{\mathbf{W}}^T \bar{\mathbf{q}}^0$ . The estimate  $\bar{\mathbf{z}}_n^0 \equiv [\bar{\mathbf{v}}_n^{0T} \quad \bar{\mathbf{w}}_n^{0T} \quad \bar{\lambda}_n^{0T}]^T$  is used to evaluate the Jacobian of Eq. (6.4.27) and restart the numerical integration process.

#### 6.4.4.2 SDIRK Integration

Runge-Kutta numerical integration methods are presented in Section 4.8.1.1 for solution of first order ODE of the form

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}) \quad (6.4.29)$$

where  $\mathbf{y}$  is an  $n$ -vector variable. A variety of RK integrators can be written in the form

$$\mathbf{k}_i = \mathbf{f}\left(t_n + c_i h, \mathbf{y}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}_j\right), \quad i=1, \dots, s \quad (6.4.30)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{k}_i \quad (6.4.31)$$

where  $t_n$  is the current time step;  $\mathbf{y}_n$  is the approximate solution at  $t_n$ ;  $a_{ij}$ ,  $b_i$ , and  $c_i = \sum_{j=1}^i a_{ij}$  are constants;  $\sum_{i=1}^s b_i = 1$ ;  $s$  is the number of stages;  $\mathbf{k}_i$  are stage variables; and  $h$  is the step-size. If diagonal terms of the matrix  $\mathbf{a} = [a_{ij}]$  are  $a_{ii} = \alpha \neq 0$ ,  $i=1, 2, \dots, s$ , the method is called singly diagonal, or SDIRK. The stiffly-accurate, L-stable, 5 stage, order 4 SDIRK54 formula for first order ODE is defined in Table 4.8.6.

To integrate the first order ODE of Eq. (6.4.19) using a Runge-Kutta method, *stage variables* are  $\mathbf{k}_i = [\mathbf{k}\mathbf{v}_i^T \quad \mathbf{k}\mathbf{w}_i^T]^T$  and arguments of functions in the *stage equations* are  $\mathbf{q}(\mathbf{v})$  and  $\dot{\mathbf{q}}(\mathbf{v}, \dot{\mathbf{v}})$ . Evaluated as functions of stage variables, they are

$$\begin{aligned}\mathbf{q}_i &= \mathbf{q} \left( \mathbf{v}_n + h \sum_{j=1}^i a_{ij} \mathbf{k} \mathbf{v}_j \right) \\ \dot{\mathbf{q}}_i &= \dot{\mathbf{q}} \left( \mathbf{v}_n + h \sum_{j=1}^i a_{ij} \mathbf{k} \mathbf{v}_j, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k} \mathbf{w}_j \right)\end{aligned}\quad (6.4.32)$$

Denote a general function  $\mathbf{g}(t, \mathbf{v}, \mathbf{w})$  that is evaluated as a function of stage variables as

$$\check{\mathbf{g}}_i = \mathbf{g} \left( t_n + c_i h, \mathbf{v}_n + h \sum_{j=1}^i a_{ij} \mathbf{k} \mathbf{v}_j, \mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k} \mathbf{w}_j \right) \quad (6.4.33)$$

With this notation, *stage equations* for the ODE of Eqs. (6.4.19), in residual form, are

$$\check{\mathbf{R}}_1 \equiv \mathbf{k} \mathbf{v}_i - \mathbf{V}^T \left( \check{\mathbf{D}}_i (\mathbf{w}_n + h \sum_{j=1}^i a_{ij} \mathbf{k} \mathbf{w}_j) + (\mathbf{I} - \mathbf{X} \check{\mathbf{H}}_i \check{\mathbf{C}}_i) \dot{\mathbf{q}}^0 + \mathbf{X} \check{\mathbf{H}}_i \check{\mathbf{v}}_i \right) = \mathbf{0} \quad (6.4.34)$$

$$\check{\mathbf{R}}_2 \equiv \mathbf{k} \mathbf{w}_i - \left( \check{\mathbf{K}}_i^T \check{\mathbf{M}}_i \check{\mathbf{K}}_i \right)^{-1} \check{\mathbf{K}}_i^T \left( \check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\gamma} + \check{\mathbf{Q}}^A + \check{\mathbf{S}} \right)_i = \mathbf{0} \quad (6.4.35)$$

Repeating the manipulations of Eqs. (6.4.8) through (6.4.14) reduces Eq. (6.4.35) to

$$\check{\mathbf{R}}_2 = \left( \check{\mathbf{M}}_i \check{\mathbf{K}}_i \right) \mathbf{k} \dot{\mathbf{w}}_i + \check{\mathbf{C}}^{iT} \lambda_i - \left( \check{\mathbf{M}} \mathbf{X} \check{\mathbf{H}} \check{\gamma} + \check{\mathbf{Q}}^A + \check{\mathbf{S}} \right)_i = \mathbf{0} \quad (6.4.36)$$

Equations (6.4.34) and (6.4.36) comprise the residual form of Runge-Kutta discretization of Eq. (6.4.26). Using derivatives of residual terms in Eqs. (6.4.23) and (6.4.24) and the chain rule of differentiation with Eq. (6.4.33), the Jacobian of Eqs. (6.4.34) and (6.4.36) for an SDIRK integrator with  $a_{ii} = \alpha$ , in variables  $\mathbf{z}^i = [\mathbf{k} \mathbf{v}_i^T \quad \mathbf{k} \mathbf{w}_i^T \quad \lambda_i]^T$ , is

$$\mathbf{J}^{RK} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \check{\mathbf{M}}_i \check{\mathbf{K}}_i & \check{\mathbf{C}}_i^T \end{bmatrix} + h \begin{bmatrix} \check{\mathbf{R}}_1_v & \check{\mathbf{R}}_1_w & \mathbf{0} \\ \check{\mathbf{R}}_2_v & \check{\mathbf{R}}_2_w & \mathbf{0} \end{bmatrix} \equiv \check{\mathbf{J}}_1 + h \check{\mathbf{J}}_2 \quad (6.4.37)$$

The Newton-Raphson algorithm is used with the same approach outlined for the trapezoidal method to obtain initial estimates to start the iterative process and to redefine the parameterization and restart integration, as required.

#### 6.4.4.3 Implicit Integration Algorithm

**Implicit Numerical integration** of Eqs. (6.4.14) and (6.4.18), using trapezoidal and Runge-Kutta methods is as follows:

- (1) Define initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t^0$  that satisfy kinematic configuration and velocity constraints. Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{U}$  and  $\mathbf{V}$  in Eqs. (6.1.36) and the system velocity coefficient matrix  $\mathbf{C}(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{W}$  and  $\mathbf{X}$  in Eqs. (6.1.48) and (6.1.49).
- (2) Apply an implicit numerical integrator to proceed stepwise on a time grid with step size  $h$ , using a factored form of the integration Jacobian of Eq. (6.4.27) or (6.4.37) to iteratively

determine  $\mathbf{z}_n$ . Use Eq. (6.4.25) or (6.4.31) to determine  $\mathbf{v}_n$  and  $\mathbf{w}_n$ . Use Eqs. (6.1.45), (6.1.58), and (6.1.62) to evaluate  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ , and  $\ddot{\mathbf{q}}_n$  on the time grid.

(3) Monitor the *condition number* of  $\mathbf{J}^{\text{RK}}$  or  $\mathbf{J}^{\text{trap}}$ , the number of Newton-Raphson iterations required in Step (2), the norms of  $\mathbf{v}$  and  $\mathbf{w}$ , and the number of iterations required to evaluate  $\mathbf{u}$ ,  $\mathbf{B}$ , and  $\mathbf{H}$ . If tolerances are exceeded, define a new time  $\bar{t}^0$  and associated  $\bar{\mathbf{q}}^0$  and  $\bar{\mathbf{q}}^0$ . Repeat Step (1) to define a new parameterization and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\mathbf{w}}^0$ .

(4) Continue the process until the final time  $tf$  is reached, or a singularity is encountered due to a faulty design or model.

With straightforward adaptation of the ODE formulation of Section 6.2, a first order Index 0 DAE involving Lagrange multipliers is obtained. Numerical integration formulas are applied to an underlying ODE and inflated for solution with Lagrange multipliers, which are available for computation of constraint reaction forces.

### Key Formulas

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{V}^T \mathbf{K} \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{X} \mathbf{H} \mathbf{C}) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{X} \mathbf{H} \mathbf{v} & \begin{bmatrix} \mathbf{v}^0 \\ \mathbf{w}^0 \end{bmatrix} &= \mathbf{0} & (6.4.14) & (6.4.18) \\ \mathbf{M} \mathbf{K} \dot{\mathbf{w}} + \mathbf{C}^T \lambda &= \mathbf{M} \mathbf{X} \mathbf{H} \gamma + \mathbf{Q}^A + \mathbf{S} \end{aligned}$$

## 6.5 Numerical Examples with Index 0 DAE

Two of the examples treated in Section 6.3 using the ODE formulation are analyzed using the Index 0 DAE formulation. Quantities derived for implicit numerical integration are only slightly more extensive than those for explicit integration in Section 6.3.

### 6.5.1 Planar Articulated Vehicle

The two chassis *articulated vehicle* treated with the tangent space ODE formulation in Section 6.3.2 is shown in Fig. 6.5.1. It is treated here using the *Index 0 DAE formulation*. The front axle of chassis one is steerable, with angle  $\theta(t)$ . Since *roll without slip* conditions for both front wheels would be redundant, a single wheel model is placed at the center of the front axle at point  $P_1$  and the roll without slip condition is applied to this wheel model. The chassis are coupled by a revolute joint at point  $P_3$  that is orthogonal to their common plane of motion, 3 m from the origins of body-fixed reference frames.

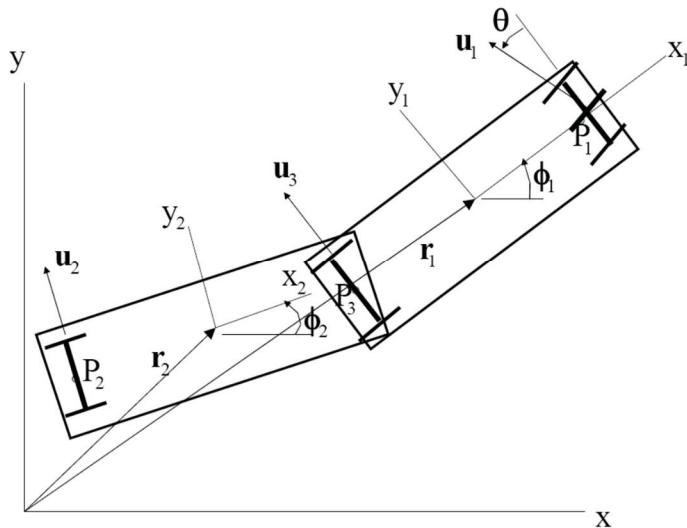


Figure 6.5.1 Planar Articulated Vehicle

Six *generalized coordinates* for the system are

$$\mathbf{q} = \begin{bmatrix} \mathbf{r}_1^T & \phi_1 & \mathbf{r}_2^T & \phi_2 \end{bmatrix}^T = \begin{bmatrix} x_1 & y_1 & \phi_1 & x_2 & y_2 & \phi_2 \end{bmatrix}^T$$

The revolute joint between the chassis is characterized by the *holonomic constraint*

$$\Phi(\mathbf{q}) = (\mathbf{r}_1 - 3\mathbf{A}(\phi_1)\mathbf{i}_1') - (\mathbf{r}_2 + 3\mathbf{A}(\phi_2)\mathbf{i}_2') = \mathbf{0} \quad (6.5.1)$$

whose Jacobian is

$$\Phi_q(\mathbf{q}) = [\mathbf{I}_2 \quad -3\mathbf{PA}(\phi_1)\mathbf{i}_1' \quad -\mathbf{I}_2 \quad -3\mathbf{PA}(\phi_2)\mathbf{i}_2'] \quad (6.5.2)$$

where  $\mathbf{A}(\phi_i)$  is the rotation transformation matrix from the  $x_i'y_i'$  frame to the  $x-y$  frame and  $\mathbf{i}_i'$  are unit vectors along the  $x_i'$ -axes.

Vectors from origins of body-fixed reference frames to centers of axles are  $\mathbf{r}_1^{P_1} = \mathbf{r}_1 + 3\mathbf{A}(\phi_1)\mathbf{i}'_1$ ,  $\mathbf{r}_1^{P_3} = \mathbf{r}_1 - 3\mathbf{A}(\phi_1)\mathbf{i}'_1$ , and  $\mathbf{r}_2^{P_2} = \mathbf{r}_2 - 3\mathbf{A}(\phi_2)\mathbf{i}'_2$ . The velocities of these points are

$$\begin{aligned}\dot{\mathbf{r}}_1^{P_1} &= \dot{\mathbf{r}}_1 + 3\dot{\phi}_1 \mathbf{PA}(\phi_1)\mathbf{i}'_1 \\ \dot{\mathbf{r}}_1^{P_3} &= \dot{\mathbf{r}}_1 - 3\dot{\phi}_1 \mathbf{PA}(\phi_1)\mathbf{i}'_1 \\ \dot{\mathbf{r}}_2^{P_2} &= \dot{\mathbf{r}}_2 - 3\dot{\phi}_2 \mathbf{PA}(\phi_2)\mathbf{i}'_2\end{aligned}\quad (6.5.3)$$

where  $\mathbf{P} = \mathbf{A}(\pi/2)$  and  $\mathbf{A}(\phi)_\phi = \mathbf{PA}(\phi)$ .

The velocities of Eqs. (6.5.3) must be orthogonal to unit vectors  $\mathbf{u}_1$ ,  $\mathbf{u}_3$ , and  $\mathbf{u}_2$ , respectively, as conditions that the wheels do not slip laterally; i.e.,  $\mathbf{u}_1^T \dot{\mathbf{r}}_1^{P_1} = [-\sin\theta \quad \cos\theta] (\mathbf{A}(\phi_1)^T \dot{\mathbf{r}}_1 + 3\dot{\phi}_1 \mathbf{Pi}') = 0$ ,  $\mathbf{u}_3^T \dot{\mathbf{r}}_1^{P_3} = \mathbf{j}_1^T \mathbf{A}(\phi_1)^T \dot{\mathbf{r}}_1 - 3\dot{\phi}_1 = 0$ , and  $\mathbf{u}_2^T \dot{\mathbf{r}}_2^{P_2} = \mathbf{j}_2^T \mathbf{A}(\phi_2)^T \dot{\mathbf{r}}_2 - 3\dot{\phi}_2 = 0$ , where  $\mathbf{A}^T \mathbf{PA} = \mathbf{P}$  has been used. *Differential constraints* that act on the system are thus

$$\mathbf{E}(\mathbf{q}, t)\dot{\mathbf{q}} = \begin{bmatrix} [-\sin(\theta) \quad \cos(\theta)]\mathbf{A}(\phi_1)^T & 3[-\sin(\theta) \quad \cos(\theta)]\mathbf{j}'_1 & \mathbf{0} & 0 \\ \mathbf{j}_1^T \mathbf{A}(\phi_1)^T & -3 & \mathbf{0} & 0 \\ \mathbf{0} & 0 & \mathbf{j}_2^T \mathbf{A}(\phi_2)^T & -3 \end{bmatrix} \dot{\mathbf{q}} = \mathbf{0} \quad (6.5.4)$$

so  $\mathbf{e}(\mathbf{q}, t) = \mathbf{0}$ . The composite constraint coefficient matrix of Eqs. (6.5.2) and (6.5.4) is

$$\mathbf{C}(\mathbf{q}, t) = \begin{bmatrix} \Phi_q(\mathbf{q}) \\ \mathbf{E}(\mathbf{q}, t) \end{bmatrix} \quad (6.5.5)$$

As defined in Section 6.4, and in turn in Section 6.2, derivative terms needed for numerical integration of the equations of motion, with  $\boldsymbol{\chi} = [\boldsymbol{\chi}_{\mathbf{r}_1}^T \quad \boldsymbol{\chi}_{\phi_1}^T \quad \boldsymbol{\chi}_{\mathbf{r}_2}^T \quad \boldsymbol{\chi}_{\phi_2}^T]^T$ , are  $\Phi_t = -\mathbf{v} = \mathbf{0}$ ,  $\Phi_{tq} = \Phi_{ttq} = \mathbf{0}$ , and

$$\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) = (\Phi_q(\mathbf{q}, \ddot{\mathbf{t}})\ddot{\boldsymbol{\chi}})_q = [\mathbf{0} \quad 3\boldsymbol{\chi}_{\phi_1} \mathbf{A}(\phi_1)\mathbf{i}'_1 \quad \mathbf{0} \quad 3\boldsymbol{\chi}_{\phi_2} \mathbf{A}(\phi_2)\mathbf{i}'_2]$$

$$\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) = (\mathbf{P}2(\mathbf{q}, \ddot{\mathbf{t}})\ddot{\boldsymbol{\chi}})_q = [\mathbf{0} \quad 3\dot{\phi}_1^2 \mathbf{PA}(\phi_1)\mathbf{i}'_1 \quad \mathbf{0} \quad 3\dot{\phi}_2^2 \mathbf{PA}(\phi_2)\mathbf{i}'_2]$$

$$\mathbf{E}_t = -\dot{\theta} \begin{bmatrix} [\cos\theta \quad \sin\theta]\mathbf{A}(\phi_1)^T & 3[\cos\theta \quad \sin\theta]\mathbf{j}'_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{2 \times 6} & & & \end{bmatrix}$$

$$(\mathbf{E}_t(\mathbf{q}, \ddot{\mathbf{t}})\ddot{\boldsymbol{\chi}})_q = \dot{\theta} \begin{bmatrix} \mathbf{0} & [\cos\theta \quad \sin\theta]\mathbf{A}(\phi_1)^T \mathbf{Pr}_1 & \mathbf{0} & 0 \\ \mathbf{0}_{2 \times 6} & & & \end{bmatrix}$$

$$\mathbf{E}2(\mathbf{q}, \boldsymbol{\chi}) = (\mathbf{E}(\mathbf{q}, \dot{\mathbf{t}})\ddot{\boldsymbol{\chi}})_{\mathbf{q}} = \begin{bmatrix} \mathbf{0} & -[-\sin(\theta) \cos(\theta)]\mathbf{A}(\phi_1)^T \mathbf{P}\boldsymbol{\chi}_{r_1} & \mathbf{0} & 0 \\ \mathbf{0} & -\mathbf{j}_1'^T \mathbf{A}(\phi_1)^T \mathbf{P}\boldsymbol{\chi}_{r_1} & \mathbf{0} & 0 \\ \mathbf{0} & 0 & \mathbf{0} & -\mathbf{j}_2'^T \mathbf{A}(\phi_2)^T \mathbf{P}\boldsymbol{\chi}_{r_2} \end{bmatrix}$$

$$\mathbf{E}3(\mathbf{q}, \dot{\mathbf{q}}) = (\mathbf{E}2(\mathbf{q}, \ddot{\mathbf{q}})\ddot{\mathbf{q}})_{\mathbf{q}} = \begin{bmatrix} \mathbf{0} & -\dot{\phi}_1[-\sin(\theta) \cos(\theta)]\mathbf{A}(\phi_1)^T \dot{\mathbf{r}}_1 & \mathbf{0} & 0 \\ \mathbf{0} & -\dot{\phi}_1 \mathbf{j}_1'^T \mathbf{A}(\phi_1)^T \dot{\mathbf{r}}_1 & \mathbf{0} & 0 \\ \mathbf{0} & 0 & \mathbf{0} & -\dot{\phi}_2 \mathbf{j}_2'^T \mathbf{A}(\phi_2)^T \dot{\mathbf{r}}_2 \end{bmatrix}$$

$$(\mathbf{C}^T(\mathbf{q}, \dot{\mathbf{t}})\ddot{\lambda})_{\mathbf{q}} = [\mathbf{0}_{6 \times 2} \quad \mathbf{C}_{\phi_1}^T \lambda \quad \mathbf{0}_{6 \times 2} \quad \mathbf{C}_{\phi_2}^T \lambda]$$

where

$$\mathbf{C}_{\phi_1} = \begin{bmatrix} \mathbf{0} & 3\mathbf{A}(\phi_1)\dot{\mathbf{i}}'_1 & \mathbf{0} & \mathbf{0} \\ -[-\sin(\theta) \cos(\theta)]\mathbf{A}^T(\phi_1)\mathbf{P} & 0 & \mathbf{0} & 0 \\ -\mathbf{j}_1'^T \mathbf{A}^T(\phi_1)\mathbf{P} & 0 & \mathbf{0} & 0 \\ \mathbf{0} & 0 & \mathbf{0} & 0 \end{bmatrix}$$

$$\mathbf{C}_{\phi_2} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & 3\mathbf{A}(\phi_2)\dot{\mathbf{i}}'_2 \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & -\mathbf{j}_2'^T \mathbf{A}^T(\phi_2)\mathbf{P} \end{bmatrix}$$

With these results,  $\gamma$ ,  $\gamma_q$ , and  $\gamma_{\dot{q}}$  of Eqs. (6.2.19) and (6.2.22) may be evaluated. In the planar equations of motion for this system,  $\mathbf{M} = m\mathbf{I}_{6 \times 6}$ ,  $\mathbf{S} = \mathbf{0}$ , and  $\mathbf{Q}^A = \mathbf{0}$ .

With two holonomic and three differential constraints on six generalized coordinates and velocities, this system has four *kinematic degrees of freedom* and one *kinetic degree of freedom*. Accordingly, there are four first order kinematic equations for  $\mathbf{v}$  and one first order kinetic equation for  $\mathbf{w}$  in Eqs. (6.4.6).

Numerical results obtained with the Index 0 formulation using Code 6.5.1 of Appendix 6.A, with explicit *RK4* and implicit *trapezoidal* and *SDIRK54* algorithms, yielded essentially identical results to those reported in Section 6.3.2 using the ODE formulation.

### 6.5.2 Spatial Disk Rolling Without Slip on x-y Plane

The spatial disk with unit radius shown in Fig. 6.5.3 that *rolls without slip* on the x-y plane is treated in section 6.3.3 with the tangent space ODE formulation. It is treated here using the Index 0 DAE formulation. The plane of the disk is defined by body fixed y'-z' axes, where the body fixed x' axis is normal to the plane of the disk. Unit vector  $\mathbf{a}'$  in the x'-y'-z' frame from the center of the disk to contact point C on the periphery with the x-y plane is

$$\mathbf{a}' = \begin{bmatrix} 0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{a} \\ \mathbf{I} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{a}'_a \\ \mathbf{a} \end{bmatrix} \quad (6.5.6)$$

and the vector  $\mathbf{a} = [a_1 \ a_2]^T$  satisfies the condition

$$(\mathbf{a}^T \mathbf{a} - 1)/2 = 0 \quad (6.5.7)$$

The normal to the disk periphery at contact point C in the plane of the disk is the vector  $\mathbf{a}'$ , so the tangent to the disk periphery at point C in the plane of the disk is

$$\mathbf{b}' = \begin{bmatrix} 0 \\ -a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{P}\mathbf{a} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{P} \end{bmatrix} \mathbf{a} \equiv \mathbf{b}'_a \mathbf{a} \quad (6.5.8)$$

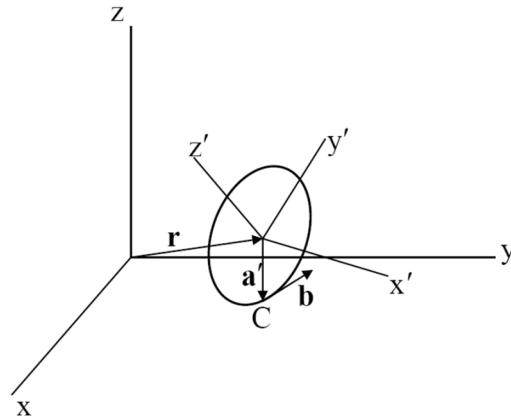


Figure 6.5.3 Disk Rolling Without Slip on the x-y Plane

Euler parameters are used as generalized coordinates for orientation. Contact conditions are that point C on the disk is in the x-y plane,  $\mathbf{u}_z^T (\mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}'_a \mathbf{a}) = 0$ , and that the vector  $\mathbf{b} = \mathbf{A}(\mathbf{p})\mathbf{b}'$  is in the x-y plane,  $\mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a \mathbf{a} = 0$ . Combining these conditions, Eq. (6.5.7), and the Euler parameter normalization condition yield the *holonomic constraints*

$$\Phi(\mathbf{q}) = \begin{bmatrix} \mathbf{u}_z^T (\mathbf{r} + \mathbf{A}(\mathbf{p})\mathbf{a}'_a \mathbf{a}) \\ \mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a \mathbf{a} \\ (\mathbf{p}^T \mathbf{p} - 1)/2 \\ (\mathbf{a}^T \mathbf{a} - 1)/2 \end{bmatrix} = \mathbf{0} \quad (6.5.9)$$

where generalized coordinates are  $\mathbf{q} = [\mathbf{r}^T \ \mathbf{p}^T \ \mathbf{a}^T]^T \in \mathbb{R}^9$ . Since  $\Phi(\mathbf{q})$  does not depend explicitly on t,  $\Phi_t = -\mathbf{v}_h = \mathbf{0}$ ,  $\Phi_{tt} = \mathbf{0}$ , and  $\Phi_{qt} = \mathbf{0}$ . The holonomic constraint Jacobian is

$$\Phi_q = \begin{bmatrix} \mathbf{u}_z^T & \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \mathbf{a}) & \mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{b}'_a \mathbf{a}) & \mathbf{u}_z^T \mathbf{A}(\mathbf{p})\mathbf{b}'_a \\ \mathbf{0} & \mathbf{p}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{a}^T \end{bmatrix} \quad (6.5.10)$$

The velocity of point C on the periphery of the disk is

$$\begin{aligned}\mathbf{v}^C &= \dot{\mathbf{r}} + \mathbf{A}(\mathbf{p})\tilde{\boldsymbol{\omega}}' \mathbf{a}' \mathbf{a} = \dot{\mathbf{r}} + \mathbf{A}(\mathbf{p})\mathbf{A}^T(\mathbf{p})\dot{\mathbf{A}}(\mathbf{p})\mathbf{a}'_a \mathbf{a} \\ &= \dot{\mathbf{r}} + \dot{\mathbf{A}}(\mathbf{p})\mathbf{a}'_a \mathbf{a} = \dot{\mathbf{r}} + (\mathbf{A}(\mathbf{p})\mathbf{a}'_a \hat{\mathbf{a}})_p \dot{\mathbf{p}} = \dot{\mathbf{r}} + \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \mathbf{a}) \dot{\mathbf{p}}\end{aligned}\quad (6.5.11)$$

Conditions that no slip occurs between the disk and the x-y plane are that the horizontal components of  $\mathbf{v}^C$  are zero; i.e.,  $\mathbf{u}_x^T \mathbf{v}^C = 0$  and  $\mathbf{u}_y^T \mathbf{v}^C = 0$ . With Eq. (6.5.11), this is

$$\mathbf{E}(\mathbf{q})\dot{\mathbf{q}} \equiv \begin{bmatrix} \mathbf{u}_x^T & \mathbf{u}_x^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \mathbf{a}) & 0 \\ \mathbf{u}_y^T & \mathbf{u}_y^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \mathbf{a}) & 0 \end{bmatrix} \dot{\mathbf{q}} \equiv \mathbf{e}(\mathbf{q}, t) = \mathbf{0} \quad (6.5.12)$$

Since  $\mathbf{E}$  does not depend explicitly on  $t$  and  $\mathbf{e} = \mathbf{0}$ ;  $\mathbf{E}_t = \mathbf{0}$ ,  $\mathbf{e}_q = \mathbf{0}$ , and  $\mathbf{e}_t = \mathbf{0}$ . With four holonomic and two *differential constraints* on nine generalized coordinates, the system has five *kinematic degrees of freedom* and three *kinetic degrees of freedom*.

Kinetic and force terms in the equations of motion are

$$\begin{aligned}\mathbf{M}(\mathbf{q}) &= \begin{bmatrix} m\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}^T(\mathbf{p})\mathbf{J}'\mathbf{G}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} \mathbf{0} \\ 8\mathbf{G}^T(\dot{\mathbf{q}})\mathbf{J}'\mathbf{G}(\dot{\mathbf{q}})\mathbf{p} \\ \mathbf{0} \end{bmatrix} \\ \mathbf{Q}^A &= \begin{bmatrix} -mg\mathbf{u}_z \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}\end{aligned}\quad (6.5.13)$$

Derivatives of terms that are required for integration of the equations of motion, with  $\boldsymbol{\chi} = [\boldsymbol{\chi}_r^T \quad \boldsymbol{\chi}_p^T \quad \boldsymbol{\chi}_a^T]^T$ , are

$$\begin{aligned}\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) &= (\Phi_q(\mathbf{q})\ddot{\boldsymbol{\chi}})_q = \begin{bmatrix} \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{a}'_a \mathbf{a}) + \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{a}'_a \boldsymbol{\chi}_a) & \mathbf{u}_z^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{b}'_a \mathbf{a}) + \mathbf{u}_z^T \mathbf{B}(\mathbf{p}, \mathbf{b}'_a \boldsymbol{\chi}_a) & \mathbf{u}_z^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{b}'_a \\ \mathbf{0} & \boldsymbol{\chi}_p^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\chi}_a^T \end{bmatrix} \\ \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) &= (\mathbf{P}2(\mathbf{q}, \ddot{\boldsymbol{\chi}})\dot{\boldsymbol{\chi}})_q = \begin{bmatrix} \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\dot{\mathbf{p}}, \mathbf{a}'_a \dot{\mathbf{a}}) + \mathbf{u}_z^T \mathbf{Z}(\dot{\mathbf{p}}, \mathbf{a}'_a \dot{\mathbf{a}}) & \mathbf{u}_z^T \mathbf{M}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) \mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_z^T \mathbf{B}(\dot{\mathbf{p}}, \mathbf{b}'_a \dot{\mathbf{a}}) + \mathbf{u}_z^T \mathbf{Z}(\dot{\mathbf{p}}, \mathbf{b}'_a \dot{\mathbf{a}}) & \mathbf{u}_z^T \mathbf{M}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) \mathbf{b}'_a \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\end{aligned}$$

where  $\mathbf{B}(\cdot, \cdot)$ ,  $\mathbf{M}(\cdot, \cdot)$ , and  $\mathbf{Z}(\cdot, \cdot)$  are defined in Eqs. (2.6.25), (2.6.81), and (2.6.83),

$$\mathbf{E}2(\mathbf{q}, \boldsymbol{\chi}) = (\mathbf{E}(\mathbf{q})\ddot{\boldsymbol{\chi}})_q = \begin{bmatrix} \mathbf{0} & \mathbf{u}_x^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{a}'_a \mathbf{a}) & \mathbf{u}_x^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_y^T \mathbf{B}(\boldsymbol{\chi}_p, \mathbf{a}'_a \mathbf{a}) & \mathbf{u}_y^T \mathbf{M}(\mathbf{p}, \boldsymbol{\chi}_p) \mathbf{a}'_a \end{bmatrix}$$

$$\mathbf{E}3(\mathbf{q}, \dot{\mathbf{q}}) = \left( \mathbf{E}2(\mathbf{q}, \ddot{\mathbf{q}}) \ddot{\mathbf{q}} \right)_\mathbf{q} = \begin{bmatrix} \mathbf{0} & \mathbf{u}_x^T \mathbf{Z}(\dot{\mathbf{p}}, \mathbf{a}'_a \dot{\mathbf{a}}) & \mathbf{u}_x^T \mathbf{M}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) \mathbf{a}'_a \\ \mathbf{0} & \mathbf{u}_y^T \mathbf{Z}(\dot{\mathbf{p}}, \mathbf{a}'_a \dot{\mathbf{a}}) & \mathbf{u}_y^T \mathbf{M}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) \mathbf{a}'_a \end{bmatrix}$$

$$\left( \mathbf{C}^T(\mathbf{q}) \ddot{\lambda} \right)_\mathbf{q} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{a}_1 & \mathbf{a}_2 \\ \mathbf{0} & \lambda_1 \mathbf{a}'_a^T \mathbf{C}(\mathbf{p}, \mathbf{u}_z) + \lambda_2 \mathbf{b}'_a^T \mathbf{C}(\mathbf{p}, \mathbf{u}_z) & \lambda_4 \mathbf{I}_2 \end{bmatrix}$$

$$\mathbf{a}_1 = \lambda_1 \mathbf{K}(\mathbf{a}'_a \mathbf{a}, \mathbf{u}_z) + \lambda_2 \mathbf{K}(\mathbf{b}'_a \mathbf{a}, \mathbf{u}_z) + \lambda_3 \mathbf{I}_4 + \lambda_5 \mathbf{K}(\mathbf{a}'_a \mathbf{a}, \mathbf{u}_x) + \lambda_6 \mathbf{K}(\mathbf{a}'_a \mathbf{a}, \mathbf{u}_y)$$

$$\mathbf{a}_2 = \lambda_1 \mathbf{C}^T(\mathbf{p}, \mathbf{u}_z) \mathbf{a}'_a + \lambda_2 \mathbf{C}^T(\mathbf{p}, \mathbf{u}_z) \mathbf{b}'_a + \lambda_5 \mathbf{C}^T(\mathbf{p}, \mathbf{u}_x) \mathbf{a}'_a + \lambda_6 \mathbf{C}^T(\mathbf{p}, \mathbf{u}_y) \mathbf{a}'_a$$

$$\mathbf{M}2(\mathbf{q}, \boldsymbol{\mu}) = \left( \mathbf{M}(\mathbf{q}) \ddot{\boldsymbol{\mu}} \right)_\mathbf{q} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 4 \left( \mathbf{T}(\mathbf{J}' \mathbf{G}(\mathbf{p}) \boldsymbol{\mu}_p) - \mathbf{G}^T(\mathbf{p}) \mathbf{J}' \mathbf{G}(\boldsymbol{\mu}_p) \right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{S}_\mathbf{q} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 8 \mathbf{G}^T(\dot{\mathbf{p}}) \mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{S}_{\dot{\mathbf{q}}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 8 \mathbf{T}(\mathbf{J}' \mathbf{G}(\dot{\mathbf{p}}) \mathbf{p}) - 8 \mathbf{G}^T(\dot{\mathbf{p}}) \mathbf{J}' \mathbf{G}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

and  $\mathbf{Q}_\mathbf{q}^A = \mathbf{Q}_{\dot{\mathbf{q}}}^A = \mathbf{0}$ , where  $\mathbf{K}(\cdot, \cdot)$  and  $\mathbf{T}(\cdot)$  are defined in Eqs. (2.6.37) and (2.6.20).

The formulation was implemented in MATLAB Code 6.5.2 of Appendix 6.A, using explicit RK4 and implicit SDIRK54 integration methods. Results obtained were essentially identical to those reported in Section 6.3.3 using the explicit ODE formulation.

A planar articulated vehicle with wheels that roll, but do not slip, and a disk that rolls without slip on a plane in three-dimensional space are successfully simulated using the Index 0 DAE formulation. Accurate solutions are obtained using MATLAB codes of Appendix 6.A. All three forms of kinematic constraint are satisfied to within specified tolerances. Quantities derived for implicit numerical integration are only slightly more extensive than those for explicit integration in Section 6.3.

## 6.6 Well Posed Formulations of Nonholonomic Mechanical System Dynamics

An abbreviated treatment of *well posed nonholonomic equations* of multibody dynamics is presented in this section. Details of dependence of the equations of motion on design variables is essentially identical to that of Section 5.11 for holonomic systems, so it is not repeated here. A more detailed development and proofs of results presented in this section may be found in (Haug, 2020b)

The tangent space ODE initial-value problem for nonholonomic systems of Sections 6.1 and 6.2 is as follows:

$$\dot{\mathbf{v}} = \mathbf{V}^T \mathbf{K}(\mathbf{q}, t) \mathbf{w} + \mathbf{V}^T (\mathbf{I} - \mathbf{X}\mathbf{H}(\mathbf{q}, t)\mathbf{C}(\mathbf{q}, t)) \dot{\mathbf{q}}^0 + \mathbf{V}^T \mathbf{X}\mathbf{H}(\mathbf{q}, t) \mathbf{v}(\mathbf{q}, t) \quad (6.6.1)$$

$$\begin{aligned} \dot{\mathbf{w}} &= (\mathbf{K}^T(\mathbf{q}, t) \mathbf{M}(\mathbf{q}) \mathbf{K}(\mathbf{q}, t))^{-1} \mathbf{K}^T(\mathbf{q}, t) (\mathbf{M}(\mathbf{q}) \mathbf{X}\mathbf{H}(\mathbf{q}, t) \gamma + \mathbf{Q}^A + \mathbf{S}) \\ \mathbf{v}^0 &= \mathbf{0} \\ \mathbf{w}^0 &= \mathbf{0} \end{aligned} \quad (6.6.2)$$

For simplicity of notation, explicit dependence on design variables  $\mathbf{b}$  that is detailed in Section 5.11 is suppressed.

Assuming that all functions involved have  $k \geq 1$  continuous derivatives with respect to  $\mathbf{q}$ ,  $\mathbf{b}$ , and  $t$ , Eqs. (6.6.1) and (6.6.2) comprise an ODE initial-value problem that is well posed (Teschl, 2012). As in the holonomic case, the tangent space ODE and Index 0 DAE are algebraically equivalent, so both have the same existence, uniqueness and regularity properties. This justifies the following result:

### Theorem 6.6.1: Well Posed Nonholonomic ODE If

(1) all functions that appear in Eq. (6.6.1) are  $k \geq 1$  times continuously differentiable on a bounded domain  $D$  of  $\mathbf{q}\text{-}\dot{\mathbf{q}}\text{-}\mathbf{b}\text{-}t$  space,

(2)  $\mathbf{C}(\mathbf{q}, \mathbf{b}, t)$  has full rank in  $D$ ,

(3)  $\mathbf{M}(\mathbf{q}, \mathbf{b})$  is positive definite on the null space of  $\mathbf{C}((\mathbf{q}, \mathbf{b}, t))$  in  $D$ , and

(4)  $\mathbf{q}^0(\mathbf{b})$  and  $\dot{\mathbf{q}}^0(\mathbf{b})$  satisfy Eqs. (6.1.24) and (6.1.30), respectively, at  $t^0$ ,

then Eqs. (6.6.1) and (6.6.2) have a unique solution  $(\mathbf{v}(t, \mathbf{b}), \mathbf{w}(t, \mathbf{b}))$  that is  $k$  times continuously differentiable with respect to  $t$  and  $\mathbf{b}$  in a neighborhood of  $(\mathbf{v}^0, \mathbf{w}^0, \mathbf{b}, t^0)$ .

Equations (6.1.45), (6.1.58), and (6.1.62) define a unique solution, in a neighborhood of  $(\mathbf{q}^0, \dot{\mathbf{q}}^0, \mathbf{b}, t^0)$ , of the variational equations of motion of Eqs. (6.2.1), and (6.1.24) through (6.1.31) that is  $k$  times continuously differentiable with respect to  $t$  and  $\mathbf{b}$ .

Since existence, uniqueness, and continuity of solution of Eqs. (6.6.1) and (6.6.2) follow from ODE theory of Section 4.7.3, it remains only to show that  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  determined by Eqs. (6.1.45), (6.1.58), and (6.1.62) satisfy the constraints of Eqs. (6.1.24), (6.1.31), and (6.1.32) and initial conditions  $\mathbf{q}(t^0) = \mathbf{q}^0$  and  $\dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0$  satisfy Eqs. (6.1.24) and (6.1.31).

By construction,  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  satisfy Eqs. (6.1.45), (6.1.58), and (6.1.62). At  $t^0$ , Eqs. (6.1.39) show that  $\mathbf{u}^0 = \mathbf{h}(\mathbf{v}^0, \mathbf{b}, t^0) = \mathbf{0}$ , so  $\mathbf{q}(\mathbf{v}^0, \mathbf{b}, t^0) = \mathbf{q}^0(\mathbf{b})$ , hence satisfying configuration

initial conditions, and Eq. (6.1.48) and  $\mathbf{K}(\mathbf{q}^0, \mathbf{b}, t^0) = \mathbf{0}$  show that  $\dot{\mathbf{q}}(\mathbf{v}^0, \mathbf{w}^0, \mathbf{b}, t^0) = \dot{\mathbf{q}}^0(\mathbf{b})$ . It remains to show that Eq. (6.2.1) is satisfied for all  $\mathbf{q}$  that satisfy Eq. (6.1.66). Since the equation following Eq. (6.2.2) holds for all  $\alpha$  and  $\mathbf{q} = \mathbf{K}\alpha$ , Eq. (6.2.1) holds for all  $\mathbf{q}$  such that  $\mathbf{C} \mathbf{q} = \mathbf{CD}\alpha = \mathbf{0}$ .

**Theorem 6.6.2 Well Posed Index 0 DAE** Under the hypotheses of Theorem 6.6.1, the Index 0 DAE formulation of Eqs. (6.4.6) with initial conditions of Eq. (6.6.2) is well posed.

The proof follows by simply noting that the ODE and Index 0 DAE formulations are equivalent.

Derivation of the well posed initial-value problem of Eqs. (6.6.1) and (6.6.2) shows that it is satisfied by  $\mathbf{v}(t, \mathbf{b})$  and  $\mathbf{w}(t, \mathbf{b})$  that are defined by  $\mathbf{q}(t, \mathbf{b})$  that satisfies Eq. (6.2.1), for all  $\mathbf{q}$  that satisfy Eq. (6.1.8), the constraints, and the initial conditions; i.e., by every solution of the d'Alembert variational equations. Conversely, Theorem 6.6.1 shows that  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  satisfy the d'Alembert variational equations. If a solution of the variational equations is unique, the ODE initial-value problem and d'Alembert variational equations are equivalent, hence both are well posed, leading to the following result.

**Theorem 6.6.3: Well Posed d'Alembert Nonholonomic Variational Equations** Under the hypotheses of Theorem 6.6.1, the nonholonomic d'Alembert variational equations are well posed.

To prove the theorem, it remains only to show that a solution of d'Alembert equations is unique. For purposes of obtaining a contradiction, assume there are solutions  $\mathbf{q} \neq \bar{\mathbf{q}}$  of the variational equations, constraints, and initial conditions. As shown, they define solutions  $(\mathbf{v}, \mathbf{w})$  and  $(\bar{\mathbf{v}}, \bar{\mathbf{w}})$  of the initial-value problem of Eqs. (6.6.1) and (6.6.2). From Eq. (6.1.22),

$$\mathbf{q} - \bar{\mathbf{q}} = \mathbf{V}(\mathbf{b})(\mathbf{v} - \bar{\mathbf{v}}) - \mathbf{U}(\mathbf{b})(\mathbf{h}(\mathbf{v}, \mathbf{b}, t) - \mathbf{h}(\bar{\mathbf{v}}, \mathbf{b}, t))$$

Since the solution of the ODE initial-value problem is unique,  $\mathbf{v} = \bar{\mathbf{v}}$ , hence,  $\mathbf{q} - \bar{\mathbf{q}} = \mathbf{0}$ . This yields the desired contradiction, which completes the proof.

The *Full DAE nonholonomic formulation* is

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}^T \boldsymbol{\lambda} - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathbf{0} \\ \mathbf{q}(t^0) &= \mathbf{q}^0 \\ \dot{\mathbf{q}}(t^0) &= \dot{\mathbf{q}}^0 \\ \Phi(\mathbf{q}, t) &= \mathbf{0} \\ \mathbf{C}(\mathbf{q}, t)\dot{\mathbf{q}} &= \begin{bmatrix} \mathbf{v}_h(\mathbf{q}, t) \\ \mathbf{e}(\mathbf{q}, t) \end{bmatrix} \equiv \mathbf{v}(\mathbf{q}, t) \\ \mathbf{C}(\mathbf{q}, t)\ddot{\mathbf{q}} &= \begin{bmatrix} -\gamma_h(\mathbf{q}, \dot{\mathbf{q}}, t) \\ -\gamma_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} \equiv -\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{aligned} \tag{6.6.3}$$

where  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  satisfy the fourth and fifth equations at  $t^0$ .

### Theorem 6.6.4 Well Posed Nonholonomic Full DAE If

- (1) all functions appearing in Eqs. (6.6.3) are  $k \geq 1$  times continuously differentiable on a bounded domain  $D$  of  $\mathbf{q}-\dot{\mathbf{q}}-\mathbf{t}-\mathbf{b}$  space, where  $\mathbf{b}$  is a vector of parameters appearing in the functions,
- (2) the matrix  $\mathbf{C}(\mathbf{q}, t)$  has full rank on  $D$ ,
- (3) the mass matrix  $\mathbf{M}(\mathbf{q})$  is positive definite on the null space of  $\mathbf{C}(\mathbf{q}, t)$ ; i.e.,  $\bar{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \bar{\mathbf{q}} > 0$  for all  $\bar{\mathbf{q}} \neq \mathbf{0}$  such that  $\mathbf{C}(\mathbf{q}, t) \bar{\mathbf{q}} = \mathbf{0}$ , and
- (4)  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  satisfy the fourth and fifth of Eqs.(6.6.3) at  $t^0$ ,

then Eqs. (6.6.3) have a unique solution  $\mathbf{q}(t, \mathbf{b})$  in a neighborhood of  $(\mathbf{q}^0, \dot{\mathbf{q}}^0, t^0, \mathbf{b})$  that is  $k$  times continuously differentiable with respect to  $t$  and  $\mathbf{b}$ .

Derivation of the Full DAE initial-value problem shows that a solution of the variational formulation is a solution of the Full DAE formulation. Since the variational and ODE formulations are equivalent, a solution of the ODE formulation generates, via Eqs. (6.1.45), (6.1.58), and (6.1.62), a solution of the Full DAE formulation. To show the converse is true, using the constraint parameterizations of Section 6.1 and suppressing arguments, the first of Eqs. (6.6.3) becomes

$$\mathbf{M}\mathbf{K}\dot{\mathbf{w}} + \mathbf{C}^T \boldsymbol{\lambda} - \mathbf{M}\mathbf{X}\mathbf{H}\boldsymbol{\gamma} - \mathbf{S} - \mathbf{Q}^A = \mathbf{0} \quad (6.6.4)$$

Multiplying by  $\mathbf{X}^T$ ,

$$\mathbf{X}^T \mathbf{M}\mathbf{K}\dot{\mathbf{w}} + \mathbf{X}^T \mathbf{C}^T \boldsymbol{\lambda} - \mathbf{X}^T (\mathbf{M}\mathbf{X}\mathbf{H}\boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0}$$

The transpose of Eq. (6.1.29) yields  $(\mathbf{H}^{-1})^T = \mathbf{X}^T \mathbf{C}^T$ , so  $(\mathbf{X}^T \mathbf{C}^T)^{-1} = \mathbf{H}^T$ . Thus,

$$\boldsymbol{\lambda} = -\mathbf{H}^T \mathbf{X}^T \mathbf{M}\mathbf{K}\dot{\mathbf{w}} + \mathbf{H}^T \mathbf{X}^T (\mathbf{M}\mathbf{X}\mathbf{H}\boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A)$$

Substituting this result into Eq. (6.6.4),

$$(\mathbf{I} - \mathbf{C}^T \mathbf{H}^T \mathbf{X}^T) \mathbf{M}\mathbf{K}\dot{\mathbf{w}} - (\mathbf{I} - \mathbf{C}^T \mathbf{H}^T \mathbf{X}^T) (\mathbf{M}\mathbf{X}\mathbf{H}\boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0}$$

Multiplying this relation on the left by  $\mathbf{W}^T$  and using the first of Eqs. (6.1.25), this is

$$\mathbf{K}^T \mathbf{M}\mathbf{K}\dot{\mathbf{w}} - \mathbf{K}^T (\mathbf{M}\mathbf{X}\mathbf{H}\boldsymbol{\gamma} + \mathbf{S} + \mathbf{Q}^A) = \mathbf{0} \quad (6.6.5)$$

Taken with the kinematic differential equation of Eq. (6.1.63), this is the first order ODE of Eqs. (6.4.6). Since the initial conditions of Eqs. (6.6.3) imply those of Eq. (6.6.2), a solution of the Full DAE initial-value problem is a solution of the ODE initial-value problem. To establish equivalence of the two formulations, it remains only to show that a solution of the Full DAE initial-value problem is unique. First, recall that the Lagrange multiplier

theorem implies that  $\lambda$  is unique. To obtain a contradiction that shows  $\mathbf{q}$  is unique, assume that two solutions  $\mathbf{q}$  and  $\bar{\mathbf{q}}$  of the Full DAE initial-value problem are distinct; i.e.,  $\mathbf{q} \neq \bar{\mathbf{q}}$ . They generate solutions  $\mathbf{v}$  and  $\bar{\mathbf{v}}$  of the ODE initial-value problem that satisfy Eq. (6.1.45), so  $\mathbf{q} - \bar{\mathbf{q}} = \mathbf{V}(\mathbf{v} - \bar{\mathbf{v}}) - \mathbf{U}(\mathbf{h}(\mathbf{v}, \mathbf{b}, t) - \mathbf{h}(\bar{\mathbf{v}}, \mathbf{b}, t))$ . Since the solution of the ODE initial-value problem is unique,  $\mathbf{v} = \bar{\mathbf{v}}$  and  $\mathbf{q} - \bar{\mathbf{q}} = \mathbf{0}$ . This establishes the desired contradiction and completes the proof.

Results presented show that four nonholonomic system formulations are equivalent and well posed, as in the case of holonomic systems. These results put the theory of nonholonomic system dynamics on a comparable status to the theory of ODE.

## **Appendix 6.A Tangent Space Nonholonomic Code**

*Code 6.3.1 Three Wheel Transporter-TSODE*

*Code 6.3.2 Ellipsoid Rolling on Moving Surface-TSODE*

*Code 6.5.1 Planar Articulated Vehicle-TSIndex 0*

*Code 6.5.2 Disk Rolling on x-y Plane-TSIndex 0*

## Appendix 6.B Key Formulas, Chapter 6

$$\begin{aligned}
& \Phi(\mathbf{q}, t) = \mathbf{0} & \mathbf{E}(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{e}(\mathbf{q}, t) & (6.1.24) \quad (6.1.25) \\
& \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{v}_h(\mathbf{q}, t) & \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} = -\boldsymbol{\gamma}_h(\mathbf{q}, \dot{\mathbf{q}}, t) & (6.1.27) \\
& \mathbf{E}(\mathbf{q}, t)\ddot{\mathbf{q}} = -\boldsymbol{\gamma}_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) & \mathbf{C}(\mathbf{q}, t) \equiv \begin{bmatrix} \Phi_q(\mathbf{q}, t) \\ \mathbf{E}(\mathbf{q}, t) \end{bmatrix} & (6.1.29) \quad (6.1.26) \\
& -P2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - 2\Phi_{eq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_n(\mathbf{q}, t) \equiv -\boldsymbol{\gamma}_h(\mathbf{q}, \dot{\mathbf{q}}, t) & & (6.1.27) \\
& \mathbf{e}_q(\mathbf{q}, t)\dot{\mathbf{q}} + \mathbf{e}_t(\mathbf{q}, t) - E2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - E_t(\mathbf{q}, t)\dot{\mathbf{q}} \equiv -\boldsymbol{\gamma}_{nh}(\mathbf{q}, \dot{\mathbf{q}}, t) & & (6.1.29) \\
& P2(\mathbf{q}, \boldsymbol{\chi}) \equiv (\Phi_q(\mathbf{q}, \hat{t})\hat{\boldsymbol{\chi}})_q & E2(\mathbf{q}, \boldsymbol{\kappa}) \equiv (E(\mathbf{q}, \hat{t})\hat{\boldsymbol{\kappa}})_q & (6.1.28) \quad (6.1.30) \\
& C(\mathbf{q}, t)\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q}, t) & C(\mathbf{q}, t)\ddot{\mathbf{q}} = -\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t) & (6.1.31) \quad (6.1.32) \\
& \Phi_q(\mathbf{q}^0, t^0)\mathbf{V} = \mathbf{0} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I} \quad \mathbf{U} = \Phi_q^T(\mathbf{q}^0, t^0) & & (6.1.36) \\
& \mathbf{u} = \mathbf{h}(\mathbf{v}, t) & \Delta \mathbf{u}^i = \mathbf{B}\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i, t) & (6.1.41) \quad (6.1.42) \\
& & \mathbf{u}^{i+1} = \mathbf{u}^i + \Delta \mathbf{u}^i \\
& \mathbf{B}(\mathbf{q}, t) = (\Phi_q(\mathbf{q}, t)\mathbf{U})^{-1} & \Delta \mathbf{B}^i = -\mathbf{B}^i \Phi_q(\mathbf{q}, t) \mathbf{U} \mathbf{B}^i - \mathbf{B}^i & (6.1.43) \quad (6.1.44) \\
& & \mathbf{B}^{i+1} = \mathbf{B}^i + \Delta \mathbf{B}^i \\
& \mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, t) & & (6.1.45) \\
& \mathbf{C}^0\mathbf{W} = \mathbf{0} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I} \quad \mathbf{X} \equiv \mathbf{C}^{0T} & & (6.1.48) \quad (6.1.49) \\
& \mathbf{H}(\mathbf{q}, t) = (\mathbf{C}(\mathbf{q}, t)\mathbf{X})^{-1} \quad \Delta \mathbf{H}^i = -\mathbf{H}^i \mathbf{C}(\mathbf{q}, t) \mathbf{X} \mathbf{H}^i - \mathbf{H}^i & & (6.1.54) \quad (6.1.56) \\
& & \mathbf{H}^{i+1} = \mathbf{H}^i + \Delta \mathbf{H}^i \\
& \dot{\mathbf{q}} = \mathbf{Kw} + (\mathbf{I} - \mathbf{XHC})\dot{\mathbf{q}}^0 + \mathbf{XHv} \quad \mathbf{K} \equiv (\mathbf{I} - \mathbf{XHC})\mathbf{W} & & (6.1.58) \quad (6.1.59) \\
& \ddot{\mathbf{q}} = \mathbf{Kw} - \mathbf{XH}\boldsymbol{\gamma} \quad \dot{\mathbf{v}} = \mathbf{V}^T\mathbf{Kw} + \mathbf{V}^T(\mathbf{I} - \mathbf{XHC})\dot{\mathbf{q}}^0 + \mathbf{V}^T\mathbf{XHv} & & (6.1.62) \quad (6.1.63)
\end{aligned}$$

$$\begin{aligned}
& \dot{\mathbf{v}} = \mathbf{V}^T\mathbf{Kw} + \mathbf{V}^T(\mathbf{I} - \mathbf{XHC})\dot{\mathbf{q}}^0 + \mathbf{V}^T\mathbf{XHv} & \mathbf{v}^0 = \mathbf{0} & (6.2.5) \quad (6.2.6) \\
& \mathbf{K}^T\mathbf{MKw} = \mathbf{K}^T(\mathbf{MXH}\boldsymbol{\gamma} + \mathbf{Q}^A + \mathbf{S}) & \mathbf{w}^0 = \mathbf{0} &
\end{aligned}$$

$$\begin{aligned}
& \dot{\mathbf{v}} = \mathbf{V}^T\mathbf{Kw} + \mathbf{V}^T(\mathbf{I} - \mathbf{XHC})\dot{\mathbf{q}}^0 + \mathbf{V}^T\mathbf{XHv} & \begin{bmatrix} \mathbf{v}^0 \\ \mathbf{w}^0 \end{bmatrix} = \mathbf{0} & (6.4.6) \quad (6.4.10) \\
& \mathbf{MKw} + \mathbf{C}^T\boldsymbol{\lambda} = \mathbf{MXH}\boldsymbol{\gamma} + \mathbf{Q}^A + \mathbf{S} &
\end{aligned}$$

## CHAPTER 7

### DAE Methods for Holonomic Systems

#### 7.0 Introduction

The *differential-algebraic equations (DAE)* of multibody dynamics that are introduced in Section 4.10 are shown in Section 5.11 to be equivalent to tangent space ODE of Chapter 5, hence to be well posed. Existence, uniqueness, and regularity of solutions as functions of problem data, or design, are thus established. Construction of solutions, however, remains a difficult task. The concept of *index of a DAE* is introduced in Section 7.1. A landmark paper (Petzold, 1982), *DAE are not ODE*, is cited as the basis for the *Petzold Imperative* that serves as a caution regarding unjustified direct application of ODE formulas for solution of the *DAE of multibody dynamics*.

Three formulations of DAE of multibody dynamics, based on the index of equations that enforce only one of the three forms of kinematic constraint, are presented in Section 7.2, with methods for correcting error that accumulates in ignored constraints. Numerical approaches for solution of Index 1, 2, and 3 DAE are outlined in Section 7.3, including implementation of numerical integration methods.

A general-purpose MATLAB Code 7.4 is presented in Section 7.4 that implements the methods of Sections 7.1 through 7.3 for simulation of planar multibody systems. Code 7.4 is used in Section 7.5 to simulate two planar systems that have been treated in preceding chapters and to study properties of the three *DAE formulations* and *DAE numerical integration algorithms* for their solution. This process is repeated in Sections 7.6 and 7.7 for spatial systems. Results suggest that both explicit and implicit integration methods applied to Index 1 and 2 formulations yield reasonable results. Direct application of even implicit integrators to the Index 3 formulation, however, leads to marginal results or outright failure. Codes 7.4 and 7.6 for simulation of planar and spatial multibody systems, respectively, are contained in Appendix 7.A.

## 7.1 The DAE of Holonomic Systems

A brief, but rigorous, derivation of the Lagrange multiplier form of the equations of multibody dynamics is presented, followed by definition of the *differentiation index*, or simply *index*. An alternative formulation with the Lagrange multiplier as the derivative of an unknown quantity is presented, yielding an index one lower than the classic formulation. Finally, fundamental results published by Petzold (1982) are discussed, providing the *Petzold Imperative* that *DAE are not ODE* and suggesting that care be taken in using ODE methods for solution of DAE.

### 7.1.1 Lagrange Multipliers

*Configurations of holonomic systems* are characterized by ngc *generalized coordinates*

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \dots & \mathbf{q}_{nb}^T \end{bmatrix}^T \in \mathbb{R}^{ngc} \quad (7.1.1)$$

that satisfy nhc < ngc *holonomic constraints*

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (7.1.2)$$

and their first and second derivatives

$$\begin{aligned} \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t) \\ \Phi_{qq}(\mathbf{q}, t)\ddot{\mathbf{q}} &= -\left(\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - 2\Phi_{tq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \equiv -\gamma \end{aligned} \quad (7.1.3)$$

d'Alembert (1743) and Lagrange (1788) provided the foundation for dynamics of holonomic systems, in the form of the *variational equation of motion* for systems of particles, called the *fundamental equation* by Pars (1965). The extension presented in Chapter 4 to account for constrained rigid bodies is

$$\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0 \quad (7.1.4)$$

for all *virtual displacements*  $\mathbf{q}$  such that

$$\Phi_q(\mathbf{q}, t) \mathbf{q} = \mathbf{0} \quad (7.1.5)$$

A crucial aspect of this formulation is that the *virtual work of constraint reaction forces is zero*, for all  $\mathbf{q}$  that satisfy Eq. (7.1.5). Thus, constraint reaction forces do not appear in Eq. (7.1.4).

At this point in the classical literature (Lanczos, 1962), a heuristic (nonrigorous) argument is made that one can multiply Eq. (7.1.5) by a vector  $\lambda^T$  to obtain  $\lambda^T \Phi_q \mathbf{q} = 0$  and add this form of zero to Eq. (7.1.4), obtaining  $\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) + \lambda^T \Phi_q(\mathbf{q}, t) = 0$ , or

$$\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0 \quad (7.1.6)$$

The folklore continues with the heuristic argument that  $\lambda$  may be selected so that the coefficient of  $\mathbf{q}^T$  in Eq. (7.1.6) is zero; i.e.,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0} \quad (7.1.7)$$

This heuristic argument can be corrected by invoking the *Lagrange Multiplier Theorem* of Section 2.2.2, as follows: provided  $\Phi_q$  has full rank, if Eq. (7.1.4) holds for all  $\mathbf{q}$  that satisfy Eq. (7.1.5), there exists a unique vector  $\lambda$  such that Eq. (7.1.6) holds for arbitrary  $\mathbf{q}$ , hence (7.1.7) is valid. The result is the same equation, but the latter argument rigorously assures existence of a unique Lagrange multiplier, whereas the former does not. In fact, the heuristic argument says nothing about the rank of  $\Phi_q$ .

Finally, in order to have any chance for a unique solution of Eq. (7.1.7), *initial conditions*

$$\begin{aligned}\mathbf{q}(t^0) &= \mathbf{q}^0 \\ \dot{\mathbf{q}}(t^0) &= \dot{\mathbf{q}}^0\end{aligned}\tag{7.1.8}$$

at time  $t^0$  must be defined that satisfy Eq. (7.1.2) and the first of Eqs. (7.1.3); i.e.,

$$\begin{aligned}\Phi(\mathbf{q}^0, t^0) &= \mathbf{0} \\ \Phi_q(\mathbf{q}^0, t^0)\dot{\mathbf{q}}^0 &= -\Phi_t(\mathbf{q}^0, t^0)\end{aligned}\tag{7.1.9}$$

Equations (7.1.7) and (7.1.8) are called *differential-algebraic equations (DAE)*, since  $\lambda$  appears algebraically and the algebraic constraints of Eq. (7.1.2) must be satisfied. As shown in Chapter 5, under reasonable hypotheses these equations have a unique solution that satisfies all three forms of constraint in Eqs. (7.1.2) and (7.1.3). That theory, however, says nothing about how to find the unique solution.

With the foregoing one-page rigorous derivation, the DAE formulation is easily obtained; i.e., it is ña cheap resultö. The moral of this chapter, however, is that ñyou get what you pay forö. In short, the foregoing *DAE are extraordinarily difficult to solve*. One of the complexities encountered is that, in addition to the constraint of Eq. (7.1.2), its first and second derivatives must be satisfied; i.e., the *kinematic velocity and acceleration equations* of Eqs. (7.1.3). Equations (7.1.3) have, unfortunately, inherited the name *hidden constraints* in the multibody dynamics literature. It is critical that they not be hidden, or more accurately ignored, in the theory and numerical methods of multibody dynamics.

## 7.1.2 Index of DAE

The *differentiation index* of a DAE is defined (Ascher and Petzold, 1998; Hairer and Wanner, 1996) as the minimum number of differentiations of equations in the DAE that is required to reduce it to an ODE in the solution variables. While there are other definitions of *index*, the differentiation index has become the most commonly used as a measure of the difficulty in obtaining numerical solutions. For the holonomic DAE of Eqs. (7.1.2) and (7.1.7), the constraint equation may be differentiated twice to obtain the constraints of Eq. (7.1.3). Combining Eq. (7.1.7) with the second of Eqs. (7.1.3), in matrix form, yields

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \Phi_q(\mathbf{q}, t)^T \\ \Phi_q(\mathbf{q}, t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\gamma \end{bmatrix} \tag{7.1.10}$$

Under the conditions that the constraint Jacobian  $\Phi_q(\mathbf{q}, t)$  has full row rank and  $\mathbf{M}(\mathbf{q})$  is positive definite on the null space of  $\Phi_q(\mathbf{q}, t)$ , it is to be shown that the coefficient matrix on the

left of Eq. (7.1.10) is nonsingular. Suppressing arguments of functions for notational convenience, consider the equation

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{0} \quad (7.1.11)$$

If the only solution of Eq. (7.1.11) is  $\boldsymbol{\alpha} = \mathbf{0}$  and  $\boldsymbol{\beta} = \mathbf{0}$ , the coefficient matrix is nonsingular. Multiplying the first row of Eq. (7.1.11) by  $\boldsymbol{\alpha}^T$ ,

$$\boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \Phi_q^T \boldsymbol{\beta} = \boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha} + \boldsymbol{\beta}^T \Phi_q \boldsymbol{\alpha} = 0$$

Since the second row of Eq. (7.1.11) is  $\Phi_q \boldsymbol{\alpha} = \mathbf{0}$ , this reduces to  $\boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha} = 0$ . As shown in Section 4.6.2,  $\mathbf{M}$  is positive definite on the null space of the constraint Jacobian, so  $\boldsymbol{\alpha} = \mathbf{0}$ . The first row of Eq. (7.1.11) reduces to  $\Phi_q^T \boldsymbol{\beta} = \mathbf{0}$  and, since  $\Phi_q^T$  has full column rank,  $\boldsymbol{\beta} = \mathbf{0}$ . Thus, the coefficient matrix of Eq. (7.1.10) is nonsingular.

The solution of Eq. (7.1.10) is

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\mathbf{q}) & \Phi_q(\mathbf{q}, t)^T \\ \Phi_q(\mathbf{q}, t) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\gamma \end{bmatrix} \equiv \begin{bmatrix} \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} \quad (7.1.12)$$

Differentiating the second row yields the system of equations

$$\begin{aligned} \ddot{\mathbf{q}} &= \mathbf{f} \\ \dot{\lambda} &= \mathbf{g}_q \dot{\mathbf{q}} + \mathbf{g}_q \ddot{\mathbf{q}} + \mathbf{g}_t = \mathbf{g}_q \dot{\mathbf{q}} + \mathbf{g}_q \mathbf{f} + \mathbf{g}_t \end{aligned} \quad (7.1.13)$$

which is an ODE in the solution variables  $\mathbf{q}$  and  $\lambda$ . Since only one derivative was required to obtain the second of Eqs. (7.1.3), the DAE of Eq. (7.1.10) is of *Index 1*.

Next, combine Eq. (7.1.7) and the first of Eq. (7.1.3) to obtain the DAE

$$\begin{aligned} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t) \lambda - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) &= \mathbf{0} \\ \Phi_q(\mathbf{q}, t) \dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t) \end{aligned} \quad (7.1.14)$$

Differentiating the second equation with respect to time yields Eq. (7.1.10), which is Index 1. Therefore, the DAE of Eq. (7.1.14) is of *Index 2*.

Finally, combine Eq. (7.1.7) with Eq. (7.1.2) to obtain the DAE

$$\begin{aligned} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t) \lambda - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) &= \mathbf{0} \\ \Phi(\mathbf{q}, t) &= \mathbf{0} \end{aligned} \quad (7.1.15)$$

Differentiating the second equation twice with respect to time yields Eq. (7.1.10), which is Index 1. Therefore, the DAE of Eq. (7.1.15) is of *Index 3*.

### 7.1.3 An Alternative DAE Formulation

In the DAE of Eqs. (7.1.7), the variable  $\lambda(t)$  may be integrated to define

$$\begin{aligned}\mu(t) &\equiv \int_{t^0}^t \lambda(\tau) d\tau \\ \mu(t^0) &= \mathbf{0}\end{aligned}\quad (7.1.16)$$

Differentiating yields

$$\dot{\mu} = \lambda \quad (7.1.17)$$

The DAE of Eq. (7.1.7) may thus be written as

$$\mathbf{M}(q)\ddot{q} + \Phi_q(q, t)^T \dot{\mu} - \mathbf{S}(q, \dot{q}, t) - \mathbf{Q}^A(q, \dot{q}, t) = \mathbf{0} \quad (7.1.18)$$

Combining the second of Eqs. (7.1.3) with Eq. (7.1.18) yields the ODE

$$\begin{bmatrix} \mathbf{M}(q) & \Phi_q^T(q, t) \\ \Phi_q^T(q, t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \dot{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{S}(q, \dot{q}, t) + \mathbf{Q}^A(q, \dot{q}, t) \\ -\gamma(q, \dot{q}, t) \end{bmatrix} \quad (7.1.19)$$

This result is perplexing. The DAE of Eq. (7.1.10) is clearly equivalent to that of Eq.(7.1.19), with only the name of the multiplier changed. Yet one is of Index 1 and the other is an ODE. It is inconceivable that a mere renaming of variable can change a meaningful problem characteristic, suggesting that the differentiation index may be of dubious value.

### 7.1.4 The Petzold Imperative; DAE are not ODE

In 1982, Petzold published a landmark paper entitled Differential/Algebraic Equations are not ODEs (Petzold, 1982); i.e., *DAE are not ODE*. The focus was on behavior of numerical integration methods that were being considered for solution of DAE at the time, which was the infancy of research on methods of solving DAE. Among other things, it was shown that direct application of ODE integration methods in an attempt to numerically solve DAE, especially those of Index 3, is problematic and potentially disastrous. This warning is denoted here as the *Petzold Imperative*. The author has taken the Petzold Imperative seriously, applying ODE integration formulas to only ODE; hence the origin of the initial focus of this text on formulations that create ODE of multibody dynamics and their associated numerical solution using ODE integrators.

As observed in previous subsections, the *Petzold Imperative* is not limited to numerical integration issues. The unfortunate concept of *hidden constraints* reflects a viewpoint that if it were not for these pesky hidden imperfections, the DAE of multibody dynamics would be ODEs. As seen in Chapter 5, the tangent space formulation yields ODE that may be integrated with numerical error control on all three forms of constraint. Numerical approaches to direct solution of the DAE of multibody dynamics using ODE integrators that have evolved in a massive research effort over the last half century are considered in the remainder of this chapter. Most, if not all, methods proposed and currently used in numerical solution of the DAE of mechanical system dynamics are based on applying or adapting ODE numerical formulas for solution of the DAE, without a near perfect solution in sight.

The concept of index of a DAE provides a basis for making distinctions between alternative formulations of the equations of multibody dynamics. Potential difficulties that are encountered in the Index 3 formulation identified by Petzold serve as a warning that severe problems may be encountered in direct integration of the Index 3 DAE of multibody dynamics using ODE methods.

## 7.2 Formulations for Numerical Solution of DAE

While it is shown in Section 5.11 that the *Full DAE* of properly modeled holonomic multibody dynamic systems have unique solutions that depend continuously on time and model parameters, there are severe difficulties in constructing these solutions. The *Petzold Imperative* of Section 7.1.4 emphasizes that *DAE are not ODE*, suggesting that direct application of ODE integrators to generalized coordinates in the DAE setting may lead to erroneous results. An extensive literature, far too large in scope to summarize here, confirms the risk in direct application of ODE methods to Index 3 DAE; e.g., (Brenan, Campbell, and Petzold, 1989; Ascher and Petzold, 1998; Hairer and Wanner, 1996; Rabier and Rheinboldt, 2002; Arnold, 2017).

In spite of the foregoing warnings, hundreds of papers have been published that attempt to integrate DAE through application of ODE integrators, some based on modifications of ODE integrators to correct errors introduced. Approaches presented include modifying ODE predictions to satisfy constraints that are unenforced in the formulation, or adding penalty terms or additional multiplier terms in the equations of motion in an attempt to compensate for errors introduced by direct application of ODE integrators to the DAE. This might be thought of as the *Larry Bird Paradox*, in honor of the legendary basketball player who is reported to have declared that he made a career in basketball öcompensating for a lack of speedö. The analogy in multibody dynamics is making a career of compensating for application of ODE methods for solution of DAE, contrary to the Petzold Imperative.

Some numerical approaches in the literature have been based on reduction of the DAE of motion to equivalent ODE (Wehage and Haug, 1982; Potra and Rheinboldt, 1991; Negrut, Haug, and German, 2003), called the *state space approach*. The tangent space ODE formulation of Chapter 5 is in this class. The generalized coordinate partitioning algorithm of Appendix 5.E is an early member of this family. While accepted as valid methods of solution, such approaches have often been dismissed as too complicated and computationally intensive. Proponents of the state space ODE approach have observed that öyou get what you pay for" and have suggested that öthe only good DAE is one with an equivalent ODE". Without further focus on this debate, the remainder of this chapter presents three approaches to adaptation of ODE methods for numerical solution of the DAE of multibody dynamics and their implementation in MATLAB computer code for numerical test and evaluation.

To be specific regarding formulations that use ODE integrators for solution of the DAE of multibody dynamics, three forms of holonomic DAE are considered. Since almost no attention has been given to numerical methods for nonholonomic DAE, other than that of Chapter 6, consideration here is limited to holonomic systems.

### 7.2.1 Index 1 DAE Formulation

The *Index 1 DAE formulation* of Eqs. (7.1.2) through (7.1.8) is to enforce the equations of motion, acceleration constraint equations, and initial conditions; i.e.,

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} &= -\left(\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - 2\Phi_{\dot{q}q}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) = -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (7.2.1) \\ \mathbf{q}(t^0) &= \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0 \end{aligned}$$

but to leave the configuration and velocity constraint equations

$$\begin{aligned}\Phi(\mathbf{q}, t) &= \mathbf{0} \\ \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t)\end{aligned}\quad (7.2.2)$$

unenforced, except to require that initial values  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  of Eq. (7.2.1) satisfy Eqs. (7.2.2).

### 7.2.2 Index 2 DAE Formulation

The *Index 2 DAE formulation* of Eqs. (7.1.2) through (7.1.8) is to enforce the equations of motion, velocity constraint equations, and initial conditions; i.e.,

$$\begin{aligned}\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t) \\ \mathbf{q}(t^0) &= \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0\end{aligned}\quad (7.2.3)$$

but to leave the configuration and acceleration constraint equations

$$\begin{aligned}\Phi(\mathbf{q}, t) &= \mathbf{0} \\ \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} &= -(\mathbf{q}, \dot{\mathbf{q}}, t)\end{aligned}\quad (7.2.4)$$

unenforced, except to require that the initial values  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  of Eq. (7.2.3) satisfy the second of Eqs. (7.2.3) and the first of Eqs. (7.2.4) and initial values  $\ddot{\mathbf{q}}^0$  and  $\lambda^0$  satisfy the first of Eqs. (7.2.3) and the second of Eqs. (7.2.4).

### 7.2.3 Index 3 DAE Formulation

The *Index 3 DAE formulation* of Eqs. (7.1.2) through (7.1.8) is to enforce the equations of motion, configuration constraint equations, and initial conditions; i.e.,

$$\begin{aligned}\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi(\mathbf{q}, t) &= \mathbf{0} \\ \mathbf{q}(t^0) &= \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0\end{aligned}\quad (7.2.5)$$

but to leave the velocity and acceleration constraint equations

$$\begin{aligned}\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t) \\ \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} &= -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t)\end{aligned}\quad (7.2.6)$$

unenforced, except to require that the initial values  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  of Eq. (7.2.5) satisfy the second of Eqs. (7.2.5) and the first of Eqs. (7.2.6) and  $\ddot{\mathbf{q}}^0$  and  $\lambda^0$  satisfy the first of Eqs. (7.2.5) and the second of Eqs. (7.2.6).

### 7.2.4 Correction of Constraint Errors

If errors in configuration, velocity, and/or acceleration constraints that are unenforced in one of the DAE formulations exceed prescribed tolerances, perturbations in the associated variables are often determined as corrections to predicted solutions at each time step that prevent the approximate solution from becoming physically meaningless.

If the configuration constraint is unenforced, it is first corrected by a perturbation  $\Delta\mathbf{q}$  of minimum norm is desired such that  $\Phi(\mathbf{q} + \Delta\mathbf{q}) = \mathbf{0}$ ; i.e.,

$$\text{minimize}(\Delta\mathbf{q}^T \Delta\mathbf{q} / 2), \text{ such that } \Phi(\mathbf{q} + \Delta\mathbf{q}) = \mathbf{0} \quad (7.2.7)$$

The necessary condition for a solution is existence of a multiplier vector  $\boldsymbol{\tau} \in \mathbb{R}^{n_{hc}}$  such that

$$\begin{aligned} (\Delta\mathbf{q}^T \Delta\mathbf{q} / 2 + \boldsymbol{\tau}^T \Phi(\mathbf{q} + \Delta\mathbf{q}))_{\Delta\mathbf{q}} &= \Delta\mathbf{q}^T + \boldsymbol{\tau}^T \Phi_q(\mathbf{q} + \Delta\mathbf{q}) = \mathbf{0} \\ \Phi(\mathbf{q} + \Delta\mathbf{q}) &= \mathbf{0} \end{aligned} \quad (7.2.8)$$

Since these are nonlinear equations in  $\boldsymbol{\tau}$  and  $\Delta\mathbf{q}$ , they must be solved using an iterative approach, such as the Newton-Raphson method. With an initial estimate  $\Delta\mathbf{q}^1 = \mathbf{0}$  and  $\boldsymbol{\tau}^1 = \mathbf{0}$ , the iterative solution equation is

$$\begin{aligned} \begin{bmatrix} \mathbf{I}_{n_{qc}} + \mathbf{P}4(\mathbf{q} + \Delta\mathbf{q}^i, \boldsymbol{\tau}^i) & \Phi_q(\mathbf{q} + \Delta\mathbf{q}^i)^T \\ \Phi_q(\mathbf{q} + \Delta\mathbf{q}^i) & \mathbf{0} \end{bmatrix} \delta \begin{bmatrix} \Delta\mathbf{q}^i \\ \boldsymbol{\tau}^i \end{bmatrix} &= - \begin{bmatrix} \Delta\mathbf{q}^i + \Phi_q^T(\mathbf{q} + \Delta\mathbf{q}^i)\boldsymbol{\tau}^i \\ \Phi(\mathbf{q} + \Delta\mathbf{q}^i) \end{bmatrix} \\ \begin{bmatrix} \Delta\mathbf{q}^{i+1} \\ \boldsymbol{\tau}^{i+1} \end{bmatrix} &= \begin{bmatrix} \Delta\mathbf{q}^i \\ \boldsymbol{\tau}^i \end{bmatrix} + \begin{bmatrix} \Delta\mathbf{q}^i \\ \boldsymbol{\tau}^i \end{bmatrix} \end{aligned} \quad (7.2.9)$$

$$i=0,1,\dots, \text{ until } \|\Phi(\mathbf{q} + \Delta\mathbf{q}^i)\| + \|\Delta\mathbf{q}^i + \Phi_q^T(\mathbf{q} + \Delta\mathbf{q}^i)\boldsymbol{\tau}^i\| \leq i \text{ ntol}$$

If the velocity constraint is unenforced, after a correction in  $\mathbf{q}$  if one has been made, a perturbation  $\Delta\dot{\mathbf{q}}$  of minimum norm is desired such that  $\Phi_q(\dot{\mathbf{q}} + \Delta\dot{\mathbf{q}}) + \Phi_t = \mathbf{0}$ ; i.e.,

$$\text{minimize}(\Delta\dot{\mathbf{q}}^T \Delta\dot{\mathbf{q}} / 2), \text{ such that } \Phi_q(\dot{\mathbf{q}} + \Delta\dot{\mathbf{q}}) + \Phi_t = \mathbf{0} \quad (7.2.10)$$

The necessary condition for a solution is existence of a multiplier vector  $\boldsymbol{\mu} \in \mathbb{R}^{n_{hc}}$  such that

$$\begin{aligned} (\Delta\dot{\mathbf{q}}^T \Delta\dot{\mathbf{q}} / 2 + \boldsymbol{\mu}^T (\Phi_q(\dot{\mathbf{q}} + \Delta\dot{\mathbf{q}}) + \Phi_t))_{\Delta\dot{\mathbf{q}}} &= \Delta\dot{\mathbf{q}}^T + \boldsymbol{\mu}^T \Phi_q = \mathbf{0} \\ \Phi_q(\dot{\mathbf{q}} + \Delta\dot{\mathbf{q}}) + \Phi_t &= \mathbf{0} \end{aligned} \quad (7.2.11)$$

with the solution

$$\Delta\dot{\mathbf{q}} = -\Phi_q^T \boldsymbol{\mu} \quad (7.2.12)$$

where  $\boldsymbol{\mu}$  is determined by the condition  $\Phi_q(\dot{\mathbf{q}} - \Phi_q^T \boldsymbol{\mu}) + \Phi_t = \mathbf{0}$ , or

$$\Phi_q \Phi_q^T \boldsymbol{\mu} = \Phi_q \dot{\mathbf{q}} + \Phi_t \quad (7.2.13)$$

If the acceleration constraint is unenforced, after corrections in  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  if one or more has been made, a perturbation  $\Delta\ddot{\mathbf{q}}$  of minimum norm is desired such that  $\Phi_q(\ddot{\mathbf{q}} + \Delta\ddot{\mathbf{q}}) + \gamma = \mathbf{0}$ ; i.e.,

$$\text{minimize}(\Delta\ddot{\mathbf{q}}^T \Delta\ddot{\mathbf{q}} / 2), \text{ such that } \Phi_q(\ddot{\mathbf{q}} + \Delta\ddot{\mathbf{q}}) + \gamma = \mathbf{0} \quad (7.2.14)$$

The necessary condition for a solution is existence of a multiplier vector  $\boldsymbol{v} \in \mathbb{R}^{n_{hc}}$  such that

$$\begin{aligned} \left( \Delta \ddot{\mathbf{q}}^T \Delta \ddot{\mathbf{q}} / 2 + v^T (\Phi_q(\ddot{\mathbf{q}} + \Delta \ddot{\mathbf{q}}) + \gamma) \right)_{\Delta \dot{\mathbf{q}}} &= \Delta \ddot{\mathbf{q}}^T + v^T \Phi_q = \mathbf{0} \\ \Phi_q(\ddot{\mathbf{q}} + \Delta \ddot{\mathbf{q}}) + \gamma &= \mathbf{0} \end{aligned} \quad (7.2.15)$$

with the solution

$$\Delta \ddot{\mathbf{q}} = -\Phi_q^T v \quad (7.2.16)$$

where  $v$  is determined by the condition  $\Phi_q(\ddot{\mathbf{q}} - \Phi_q^T v) + \gamma = \mathbf{0}$ , or

$$\Phi_q \Phi_q^T v = \Phi_q \ddot{\mathbf{q}} + \gamma \quad (7.2.17)$$

The results of these calculations yield perturbations to approximate solutions that satisfy any configuration, velocity and/or acceleration constraints that have been ignored, to within specified tolerances. The correction process, however, contributes errors to the approximate solution, beyond those already present in the numerical integration approximation process. One might be suspicious that the *Petzold Imperative* will dominate and poor approximations will result; i.e., the *Larry Bird Paradox*. While there is some theory in the literature to justify the correction approach in selected applications, numerical experiments are required to assess the validity of these approximations.

### 7.2.5 Numerical Approaches to Solution of DAE

In Section 7.2.1, the DAE of Eqs. (7.2.1) reduces to an ODE with one differentiation, hence it is of Index 1. Since one differentiation of the velocity constraint of the DAE of Eqs. (7.2.3) reduces to the DAE of Eqs. (7.2.1), the DAE of Eqs. (7.2.3) is of Index 2. Finally, the analysis of Section 7.1.2 has shown that the DAE of Eqs. (7.2.5) is of Index 3.

As shown in Section 7.1.2, the Index 1 form of the equations of motion of Eqs. (7.2.1) can be solved at time  $t_n$  for  $\ddot{\mathbf{q}}_n$ , which may be integrated using an explicit numerical integration formula, such as RKFN45 or Nystrom4 of Section 4.8. This constitutes an *explicit numerical integration approach for the Index 1 DAE formulation*. It says nothing, however, about the accuracy with which the unenforced position and velocity constraint equations of Eq. (7.2.2) are satisfied. The Index 2 and 3 formulations of Eqs. (7.2.3) and (7.2.5) do not yield to this explicit integration approach.

If an implicit ODE integration formula is applied to the acceleration variable  $\ddot{\mathbf{q}}$  and the result is substituted into Eqs. (7.2.1), (7.2.3), or (7.2.5), a system of  $ngc + nhc$  nonlinear equations in the  $ngc + nhc$  variables  $\ddot{\mathbf{q}}_n$  and  $\lambda_n$  is obtained at integration time step  $t_n$ . There is thus the prospect, but not assurance, that these equations can be solved for an approximate solution. There is no guarantee, however, that unenforced constraint equations of Eqs. (7.2.2), (7.2.4), or (7.2.6) are even approximately satisfied. This is one consequence of the Petzold Imperative that DAE are not ODE; i.e., that application of an ODE integrator to something that is not an ODE is not guaranteed to provide meaningful results.

Theoretical results have been presented showing that the direct approach to solution using the Index 1 formulation yields reasonable approximate solutions (Brenan, Campbell, and Petzold, 1989; Hairer and Wanner, 1996; Arnold, 2017) and that slightly less assurance is offered for solutions obtained using the Index 2 formulation. Severe warnings are provided that results obtained using the Index 3 formulation may be unreliable. Results obtained using the

Index 3 formulation and backward differentiation ODE integration formulas (Ascher and Petzold, 1998; Hairer and Wanner, 1996) in the landmark paper (Orlandea, Chace, and Calahan, 1977) has shown over time that, with great care, the direct approach using the Index 3 formulation can yield acceptable results. A more recent contribution (Negrut, Rampalli, Ottarsson, and Sajdak, 2007) presented a DAE integration formulation with the *HHT formulation* for structural dynamics (Hilber, Hughes, and Taylor, 1977) that has been shown to yield reasonable results for the Index 3 formulation of the equations of multibody dynamics (Arnold and Bruls, 2007).

In the HHT formulation (Negrut, Rampalli, Ottarsson, and Sajdak, 2007), the discretized equations of motion of Eq. (7.2.5) are modified by introducing a parameter  $\alpha$ ,  $-1/3 \leq \alpha \leq 0$ , at time  $t_n$  into the discrete form of the equations of motion,

$$(1/(1+\alpha))\mathbf{M}(\mathbf{q}_n)\ddot{\mathbf{q}}_n + \Phi_q^T(\mathbf{q}_n, t_n)\lambda_n - \mathbf{Q}^A(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) - \mathbf{S}(\mathbf{q}_n, \dot{\mathbf{q}}_n) \\ - ((1/(1+\alpha))(\Phi_q^T(\mathbf{q}_{n-1}, t_{n-1})\lambda_{n-1} - \mathbf{Q}^A(\mathbf{q}_{n-1}, \dot{\mathbf{q}}_{n-1}, t_{n-1}) - \mathbf{S}(\mathbf{q}_{n-1}, \dot{\mathbf{q}}_{n-1})) = \mathbf{0} \quad (7.2.18)$$

The trapezoidal numerical integration algorithm of Eq. (4.8.40) is similarly modified as

$$\begin{aligned} \mathbf{q}_n &= \mathbf{q}_{n-1} + h\dot{\mathbf{q}}_{n-1} + (h^2/2)((1-\alpha)\ddot{\mathbf{q}}_{n-1} + 2\alpha\ddot{\mathbf{q}}_n) \\ \dot{\mathbf{q}}_n &= \dot{\mathbf{q}}_{n-1} + h((1-\alpha)\ddot{\mathbf{q}}_{n-1} + \alpha\ddot{\mathbf{q}}_n) \end{aligned} \quad (7.2.19)$$

where

$$\begin{aligned} &= (1-\alpha)^2/4 \\ &= (1-2\alpha)/2 \end{aligned} \quad (7.2.20)$$

In the literature, the parameter  $\alpha$  used here is called  $\gamma$ . The modification in terminology used here is to avoid conflict with the function  $\gamma(\mathbf{q}, \dot{\mathbf{q}})$  that appears in the kinematic acceleration equations.

It is interesting that the HHT method uses an integration formula of Eq. (7.2.19), which is not technically valid as an ODE integrator, with Eq. (7.2.18) which is not a discretized form of the multibody system equation of motion. The result is a method that incorporates *numerical damping* to avoid divergence of the method when it is applied to the multibody system equations of motion (Negrut, Rampalli, Ottarsson, and Sajdak, 2007; Arnold and Bruls, 2007). It is also interesting that the HHT method reduces to the trapezoidal method when  $\alpha = 0$ . A recent paper (Schweizer and Pu, 2016) presents a similar approach that modifies Runge-Kutta ODE integration stage equations to enable satisfaction of all three forms of kinematic constraint, at the expense of failing to be a valid ODE integration formula.

The explicit Nystrom4 and RKFN45 integrators outlined above for the Index 1 formulation and the implicit trapezoidal and HHT integrators for Index 1, 2, and 3 formulations are implemented in the following sections, to enable systematic study of properties of each method.

### 7.2.6 Third Derivative for Error Control

For the case in which there is no explicit time dependence in the kinematic constraints and generalized forces, the Index 1 formulation of equations of motion of Eq. (7.2.1) may be differentiated with respect to time, to obtain

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\dot{\lambda} &= -\left(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - \left(\Phi_q(\mathbf{q}, t)^T \ddot{\lambda}\right)_q \dot{\mathbf{q}} + \mathbf{Q}_q^A(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{S}_q(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ &\quad + \mathbf{Q}_{\dot{\mathbf{q}}}^A(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + \mathbf{S}_{\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} \end{aligned} \quad (7.2.21)$$

$$\Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} = -\left(\Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - \gamma_q \dot{\mathbf{q}} - \gamma_{\dot{\mathbf{q}}} \ddot{\mathbf{q}}$$

Using the derivative operator notation of Section 5.3 and suppressing arguments of functions, in matrix form, this is

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} (-\mathbf{M}2(\mathbf{q}, \ddot{\mathbf{q}}) - \mathbf{P}4(\mathbf{q}, \lambda) + \mathbf{S}_q + \mathbf{Q}_q^A)\dot{\mathbf{q}} + (\mathbf{S}_{\dot{\mathbf{q}}} + \mathbf{Q}_{\dot{\mathbf{q}}}^A)\ddot{\mathbf{q}} \\ (-\mathbf{P}2(\mathbf{q}, \ddot{\mathbf{q}}) - \gamma_q)\dot{\mathbf{q}} - \gamma_{\dot{\mathbf{q}}}\ddot{\mathbf{q}} \end{bmatrix} \quad (7.2.22)$$

The third derivative of generalized coordinates computed using Eq. (7.2.22) can be used in error estimates, as in Section 4.8.4. They and the first derivatives of Lagrange multipliers can also be used in obtaining estimates to start iterative solution of discretized equations of motion.

### 7.2.7 Evaluation of Error in Approximate Solutions

Two forms of error in approximate solutions may be evaluated, without knowing an exact solution. First, if constraint errors are not corrected, using methods presented in Section 7.2.4, they may be easily evaluated and plotted. Second, and less definitive, error in satisfying the variational equations of motion may be assessed. Recall that

$$\delta\mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t)) = 0 \quad (7.2.23)$$

for all  $\delta\mathbf{q}$  such that

$$\Phi(\mathbf{q})\delta\mathbf{q} = \mathbf{0} \quad (7.2.24)$$

Geometrically, Eqs. (7.2.23) and (7.2.24) require the vector

$$\mathbf{R}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (7.2.25)$$

To be orthogonal to the null space of  $\Phi_q(\mathbf{q})$ . Defining a basis for the null space of  $\Phi_q(\mathbf{q})$  as the columns of a matrix  $\mathbf{V}(\mathbf{q})$  such that  $\Phi_q(\mathbf{q})\mathbf{V}(\mathbf{q}) = \mathbf{0}$  and  $\mathbf{V}^T(\mathbf{q})\mathbf{V}(\mathbf{q}) = \mathbf{I}$ , suppressing arguments, for the exact solution,  $\mathbf{R}$  must be orthogonal to the columns of  $\mathbf{V}$ ; i.e.,  $\mathbf{V}^T\mathbf{R} = \mathbf{0}$ . A vector measure of error for the approximate solution is thus

$$\mathbf{E}^T \mathbf{r} \mathbf{r} = \mathbf{V}^T \mathbf{R} \quad (7.2.26)$$

and a scalar measure of error is

$$E^2 \mathbf{r} \mathbf{r} = (\mathbf{V}^T \mathbf{R})^T (\mathbf{V}^T \mathbf{R}) = \mathbf{R}^T \mathbf{V} \mathbf{V}^T \mathbf{R} \quad (7.2.27)$$

The scalar measure can be misleading, since components of the vector  $\mathbf{V}^T \mathbf{R}$  may have different orders of magnitude, leading to domination by the largest. It is interesting that since  $\mathbf{V}\mathbf{V}^T \neq \mathbf{I}$ ,  $E^2 rr \neq \mathbf{R}^T \mathbf{R}$ . This makes sense, since  $\mathbf{R}$  does not include constraint reaction forces, so  $\mathbf{R} \equiv \mathbf{0}$  is to be expected.

In assessing error in approximate solution of holonomically constrained systems, it is perhaps most reasonable to first implement corrections in constraint errors in Section 7.2.4 and then to evaluate error in equations of motion, as in this section. A rationale for this approach is that violation of constraints means that motion of a fictitious system is being simulated.

Index 1, 2, and 3 formulations of holonomic DAE enforce only one of the three forms of kinematic constraint. Errors in unenforced constraints may be corrected, but at the cost of introducing further error in the approximate solution. Numerical approaches for approximate solutions presented in subsequent sections use error estimates based on third derivatives of solutions for variable step size error control.

## Key Formulas

### Index 1:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} &= -\left(\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - 2\Phi_{tq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \equiv -\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \mathbf{q}(t^0) = \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0 \end{aligned} \tag{7.2.1}$$

### Index 2:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t) \\ \mathbf{q}(t^0) = \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0 \end{aligned} \tag{7.2.3}$$

### Index 3:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi(\mathbf{q}, t) &= \mathbf{0} \\ \mathbf{q}(t^0) = \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0 \end{aligned} \tag{7.2.5}$$

## 7.3 Numerical Integration of Index 1, 2, and 3 DAE of Holonomic Systems

The Index 1 DAE is integrated using either explicit integration methods or the implicit HHT method, whereas the Index 2 and 3 DAE may only be treated using the implicit approach.

### 7.3.1 Integration of Index 1 DAE

*Numerical integration of the Index 1 DAE formulation* for holonomic systems of Section 7.2.1 can be implemented with explicit and implicit numerical integration algorithms. The integration algorithms presented are implemented for both planar and spatial systems, using adaptations of Codes 5.7 and 5.9 that are presented in Sections 5.7 and 5.9.

#### 7.3.1.1 Explicit Integration of Index 1 DAE

The first two equations of Eqs. 7.2.1, in matrix form, are

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \Phi_q^T(\mathbf{q}, t) \\ \Phi_q(\mathbf{q}, t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} \quad (7.3.1)$$

Beginning at  $t^0$  with initial values  $\mathbf{q}(t_0) = \mathbf{q}^0$  and  $\dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0$  of Eqs. (7.2.1), Eq. (7.3.1) is solved for  $\ddot{\mathbf{q}}^0$  and  $\boldsymbol{\lambda}^0$ , which are stored as solution variables. *For explicit numerical integration of the Index 1 DAE formulation*, an explicit numerical integrator, such as Nystrom 4 or RKFN45, is used to integrate for  $\mathbf{q}_1$  and  $\dot{\mathbf{q}}_1$ , the approximate solution at  $t_1 = t^0 + h$ . The process is repeated to solve Eq. (7.3.1), evaluated at  $\mathbf{q}_1$  and  $\dot{\mathbf{q}}_1$  to obtain  $\ddot{\mathbf{q}}_1$  and  $\boldsymbol{\lambda}_1$ , which are stored as solution variables. At time  $t_n$  with  $\mathbf{q}_n$  and  $\dot{\mathbf{q}}_n$  known, Eq. (7.3.1) is evaluated at  $\mathbf{q}_n$  and  $\dot{\mathbf{q}}_n$  and solved for  $\ddot{\mathbf{q}}_n$  and  $\boldsymbol{\lambda}_n$ , which are integrated for  $\mathbf{q}_{n+1}$  and  $\dot{\mathbf{q}}_{n+1}$  and are stored as solution variables. The process is repeated until the final time is reached or a singularity is encountered.

#### 7.3.1.2 Implicit Integration of Index 1 DAE

*Implicit numerical integration of the Index 1 DAE formulation* for holonomic systems of Section 7.2.1 may be implemented with implicit numerical integration formulas based on the HHT method of Section 7.2.5. The formulation is applicable for both planar and spatial systems, using the foundations of Codes 5.7 and 5.9.

The Index 1 equations of motion of Eq. (7.2.1) are written in *residual form* as

$$\mathbf{R} \equiv \begin{bmatrix} \mathbf{R1} \\ \mathbf{R2} \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\boldsymbol{\lambda} - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} + \gamma(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} = \mathbf{0} \quad (7.3.2)$$

For the HHT implicit numerical integration approach of Section 7.2.5, these equations are modified and written at time  $t_n$  in the form

$$\begin{aligned} \mathbf{R1} &= (1/(1+\alpha))\mathbf{M}(\mathbf{q}_n)\ddot{\mathbf{q}}_n + \Phi_q^T(\mathbf{q}_n, t_n)\boldsymbol{\lambda}_n - \mathbf{Q}^A(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) - \mathbf{S}(\mathbf{q}_n, \dot{\mathbf{q}}_n) \\ &\quad - ((1-\alpha)/(1+\alpha))(\Phi_q^T(\mathbf{q}_{n-1}, t_{n-1})\boldsymbol{\lambda}_{n-1} - \mathbf{Q}^A(\mathbf{q}_{n-1}, \dot{\mathbf{q}}_{n-1}, t_{n-1}) - \mathbf{S}(\mathbf{q}_{n-1}, \dot{\mathbf{q}}_{n-1})) \quad (7.3.3) \\ \mathbf{R2} &= \Phi_q(\mathbf{q}_n, t_n)\ddot{\mathbf{q}}_n + \gamma(\mathbf{q}_n, \dot{\mathbf{q}}_n) \end{aligned}$$

It should be noted that this reduces to direct application of the trapezoidal method applied to Eq. (7.3.2) when  $\alpha = 0$ .

To evaluate the Jacobian of these equations, using the derivative operators of Chapter 5,

$$\begin{aligned}
\mathbf{R1}_{\dot{\mathbf{q}}_n} &= (1/(1+\cdot))\mathbf{M}(\mathbf{q}_n) \\
\mathbf{R1}_{\ddot{\mathbf{q}}_n} &= -\mathbf{Q}_{\dot{\mathbf{q}}_n}^A(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) - \mathbf{S}_{\dot{\mathbf{q}}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n) \\
\mathbf{R1}_{\lambda_n} &= (1/(1+\cdot))(\mathbf{M}(\mathbf{q}_n)\ddot{\mathbf{q}}_n)_{\mathbf{q}_n} + (\Phi_q^T(\mathbf{q}_n, t_n)\lambda_n)_{\mathbf{q}_n} - \mathbf{Q}_{\dot{\mathbf{q}}_n}^A(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) - \mathbf{S}_{\dot{\mathbf{q}}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n) \\
&= (1/(1+\cdot))\mathbf{M}2(\mathbf{q}_n, \ddot{\mathbf{q}}_n) + \mathbf{P}4(\mathbf{q}_n, \lambda_n) - \mathbf{Q}_{\dot{\mathbf{q}}_n}^A(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) - \mathbf{S}_{\dot{\mathbf{q}}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n) \\
\mathbf{R1}_{\lambda_n} &= \Phi_q^T(\mathbf{q}_n, t_n) \\
\mathbf{R2}_{\dot{\mathbf{q}}_n} &= \Phi_q(\mathbf{q}_n, t_n) \\
\mathbf{R2}_{\dot{\lambda}_n} &= \gamma_{\dot{\mathbf{q}}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) \\
\mathbf{R2}_{\mathbf{q}_n} &= (\Phi_q(\mathbf{q}_n, t_n)\ddot{\mathbf{q}}_n)_{\mathbf{q}_n} + \gamma_{\mathbf{q}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) \\
&= \mathbf{P}2(\mathbf{q}_n, \ddot{\mathbf{q}}_n) + \gamma_{\mathbf{q}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) \\
\mathbf{R2}_{\lambda_n} &= \mathbf{0}
\end{aligned} \tag{7.3.4}$$

Using the HHT integration formulas of Eq. (7.2.15) and the chain rule of differentiation,

$$\begin{aligned}
\frac{d\mathbf{R1}}{d\ddot{\mathbf{q}}_n} &= (1/(1+\cdot))\mathbf{M}(\mathbf{q}_n) - h \left( \mathbf{Q}_{\dot{\mathbf{q}}_n}^A(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) + \mathbf{S}_{\dot{\mathbf{q}}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n) \right) \\
&\quad + h^2 \left( (1/(1+\cdot))\mathbf{M}2(\mathbf{q}_n, \ddot{\mathbf{q}}_n) + \mathbf{P}4(\mathbf{q}_n, \lambda_n) - \mathbf{Q}_{\dot{\mathbf{q}}_n}^A(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) - \mathbf{S}_{\dot{\mathbf{q}}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n) \right) \\
\frac{d\mathbf{R2}}{d\ddot{\mathbf{q}}_n} &= \Phi_q(\mathbf{q}_n, t_n) + h \gamma_{\dot{\mathbf{q}}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) + h^2 \left( \mathbf{P}2(\mathbf{q}_n, \ddot{\mathbf{q}}_n) + \gamma_{\mathbf{q}_n}(\mathbf{q}_n, \dot{\mathbf{q}}_n, t_n) \right)
\end{aligned} \tag{7.3.5}$$

The Jacobian of the residual is thus

$$\mathbf{J} = \begin{bmatrix} \frac{d\mathbf{R1}}{d\ddot{\mathbf{q}}_n} & \Phi_q^T(\mathbf{q}_n, t_n) \\ \frac{d\mathbf{R2}}{d\ddot{\mathbf{q}}_n} & \mathbf{0} \end{bmatrix} \tag{7.3.6}$$

Beginning with initial conditions of Eq. (7.2.1), Eq. (7.3.1) is solved for  $\ddot{\mathbf{q}}_0$  and  $\lambda_0$ . At time step  $t_n$ ,  $n=1, \dots, n$ , the solution variables  $\ddot{\mathbf{q}}_n$  and  $\lambda_n$  are estimated as  $\ddot{\mathbf{q}}_n^1 = \ddot{\mathbf{q}}_{n-1}$  and  $\lambda_n^1 = \lambda_{n-1}$ , or  $\ddot{\mathbf{q}}_n^1 = \ddot{\mathbf{q}}_{n-1}^1 + h\ddot{\mathbf{q}}_n^1$  and  $\lambda_n^1 = \lambda_{n-1}^1 + h\lambda_n^1$  if derivatives  $\ddot{\mathbf{q}}_n^1$  and  $\lambda_n^1$  are computed, as in Section 7.2.6. Generalized coordinates  $\mathbf{q}_n^1$  and  $\dot{\mathbf{q}}_n^1$  are evaluated using Eq. (7.2.15). The Jacobian of Eq. (7.3.6) and residual of Eq. (7.3.3) are evaluated as  $\mathbf{J}^1$  and  $\mathbf{R}^1$ . The first step in Newton - Raphson iteration is carried out by solving

$$\mathbf{J}^1 \mathbf{x} = -\mathbf{R}^1 \tag{7.3.7}$$

With  $\mathbf{P}_{\ddot{\mathbf{q}}} \equiv [\mathbf{I}_{ngc} \quad \mathbf{0}]$  and  $\mathbf{P}_{\lambda} \equiv [\mathbf{0} \quad \mathbf{I}_{nc}]$ ,  $\Delta\ddot{\mathbf{q}}_n^1 = \mathbf{P}_{\ddot{\mathbf{q}}} \mathbf{x}$  and  $\Delta\lambda_n^1 = \mathbf{P}_{\lambda} \mathbf{x}$ . The values of solution variables are thus updated as

$$\begin{aligned}\ddot{\mathbf{q}}_n^2 &= \ddot{\mathbf{q}}_n^1 + \Delta \ddot{\mathbf{q}}_n^1 \\ \lambda_n^2 &= \lambda_n^1 + \Delta \lambda_n^1\end{aligned}\quad (7.3.8)$$

With the Jacobian held fixed at  $\mathbf{J} = \mathbf{J}^1$ , the process is continued until  $\|\mathbf{R}^i\| \leq \text{intol}$ , the integration tolerance, to determine  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ ,  $\ddot{\mathbf{q}}_n$ , and  $\lambda_n$ . The process is repeated at  $t_{n+1}$  and continued until the final time is reached, or a singularity is encountered.

The trapezoidal integration algorithm is obtained by setting  $\alpha = 0$  in the preceding development.

### 7.3.2 Integration of Index 2 DAE

*Implicit numerical integration of the Index 2 DAE formulation* for holonomic systems of Section 7.2.2 is implemented with implicit numerical integration formulas based on the *HHT method*. The formulation is applicable for both planar and spatial systems, using the foundations of MATLAB Codes 5.7 and 5.9.

The Index 2 equations of motion of Eq. (7.2.3) are written in *residual form* as

$$\mathbf{R} \equiv \begin{bmatrix} \mathbf{R1} \\ \mathbf{R2} \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ (1/\delta h)(\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} + \Phi_t(\mathbf{q}, t)) \end{bmatrix} = \mathbf{0} \quad (7.3.9)$$

The factor  $1/(\delta h)$ , where  $\delta$  is given in Eq. (7.2.16), is introduced to enhance the condition of the Jacobian of the discretized equations. For HHT implicit integration, these equations are modified and written at time  $t_n$  in the form of Eqs. (7.3.3), where  $\mathbf{R1}$  is given in Eq. (7.3.3), and

$$\mathbf{R2} = (1/(\delta h))(\Phi_q(\mathbf{q}_n, t_n)\dot{\mathbf{q}}_n + \Phi_t(\mathbf{q}_n, t_n)) \quad (7.3.10)$$

To evaluate the Jacobian of Eq. (7.3.9), derivative operators of Section 5.3 are used for  $\mathbf{R1}$  and

$$\begin{aligned}\mathbf{R2}_{\dot{\mathbf{q}}_n} &= \mathbf{0} \\ \mathbf{R2}_{\ddot{\mathbf{q}}_n} &= (1/(\delta h))\Phi_q(\mathbf{q}_n, t_n) \\ \mathbf{R2}_{\mathbf{q}_n} &= (1/(\delta h))\left(\left(\Phi_q(\mathbf{q}_n, t_n)\dot{\mathbf{q}}_n\right)_{\mathbf{q}_n} + \Phi_{tq}(\mathbf{q}_n, t_n)\right) \\ &= (1/(\delta h))(\mathbf{P2}(\mathbf{q}_n, \dot{\mathbf{q}}_n) + \Phi_{tq}(\mathbf{q}_n, t_n)) \\ \mathbf{R2}_{\lambda_n} &= \mathbf{0}\end{aligned}\quad (7.3.11)$$

Using the HHT integration formulas of Eq. (7.2.15) and the chain rule of differentiation,  $\frac{d\mathbf{R1}}{d\dot{\mathbf{q}}_n}$  is as given in Eq. (7.3.5) and

$$\frac{d\mathbf{R2}}{d\dot{\mathbf{q}}_n} = \Phi_q(\mathbf{q}_n, t_n) + (h / \delta)(\mathbf{P2}(\mathbf{q}_n, \dot{\mathbf{q}}_n) + \Phi_{tq}(\mathbf{q}_n, t_n)) \quad (7.3.12)$$

The Jacobian of the residual of Eq. (7.3.9) is thus

$$\mathbf{J} = \begin{bmatrix} \frac{d\mathbf{R1}}{d\ddot{\mathbf{q}}_n} & \Phi_q^T(\mathbf{q}_n, t_n) \\ \frac{d\mathbf{R2}}{d\ddot{\mathbf{q}}_n} & \mathbf{0} \end{bmatrix} \quad (7.3.13)$$

Beginning with initial conditions of Eq. (7.2.1), Eq. (7.3.1) is solved for  $\ddot{\mathbf{q}}^0$  and  $\lambda^0$ . At time step  $t_n$ ,  $n=1, \dots, n$ , the solution variables  $\ddot{\mathbf{q}}_n$  and  $\lambda_n$  are estimated as  $\ddot{\mathbf{q}}_n^1 = \ddot{\mathbf{q}}_{n-1}$  and  $\lambda_n^1 = \lambda_{n-1}$ , or  $\ddot{\mathbf{q}}_n^1 = \ddot{\mathbf{q}}_{n-1} + h\ddot{\mathbf{q}}_n^1$  and  $\lambda_n^1 = \lambda_{n-1}^1 + h\dot{\lambda}_n^1$  if derivatives  $\ddot{\mathbf{q}}_n^1$  and  $\dot{\lambda}_n^1$  are computed, as in Section 7.2.6. Generalized coordinates  $\mathbf{q}_n^1$  and  $\dot{\mathbf{q}}_n^1$  are evaluated using Eq. (7.2.19). The Jacobian of Eq. (7.3.13) and residual of Eq. (7.3.10) are evaluated as  $\mathbf{J}^1$  and  $\mathbf{R}^1$ . The first step in Newton-Raphson iteration is carried out by solving

$$\mathbf{J}^1 \mathbf{x} = -\mathbf{R}^1 \quad (7.3.14)$$

With  $\mathbf{P}_{\dot{\mathbf{q}}} \equiv [\mathbf{I}_{ngc} \quad \mathbf{0}]$  and  $\mathbf{P}_{\lambda} \equiv [\mathbf{0} \quad \mathbf{I}_{nc}]$ ,  $\Delta\ddot{\mathbf{q}}_n^1 = \mathbf{P}_{\dot{\mathbf{q}}} \mathbf{x}$  and  $\Delta\lambda_n^1 = \mathbf{P}_{\lambda} \mathbf{x}$ . The values of solution variables are then updated as

$$\begin{aligned} \ddot{\mathbf{q}}_n^2 &= \ddot{\mathbf{q}}_n^1 + \Delta\ddot{\mathbf{q}}_n^1 \\ \lambda_n^2 &= \lambda_n^1 + \Delta\lambda_n^1 \end{aligned} \quad (7.3.15)$$

With the Jacobian held fixed at  $\mathbf{J} = \mathbf{J}^1$ , the process is continued until  $\|\mathbf{R}^i\| \leq \text{intol}$ , the integration tolerance, to determine  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ ,  $\ddot{\mathbf{q}}_n$ , and  $\lambda_n$ . The process is repeated at  $t_{n+1}$  and continued until the final time is reached, or a singularity is encountered.

### 7.3.3 Integration of Index 3 DAE

*Implicit numerical integration of the Index 3 DAE formulation* for holonomic systems of Section 7.2.3 is implemented with implicit numerical integration formulas based on the HHT method. The formulation is applicable for both planar and spatial systems, using the foundations of MATLAB Codes 5.7 and 5.9.

The Index 3 equations of motion of Eq. (7.2.5) are written in residual form as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R1} \\ \mathbf{R2} \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ (1/h^2)\Phi(\mathbf{q}, t) \end{bmatrix} = \mathbf{0} \quad (7.3.16)$$

The factor  $1/(h^2)$  is introduced to enhance the condition of the Jacobian of the discretized equations. For HHT implicit numerical integration, these equations are modified and written at time  $t_n$  in the form of Eq. (7.3.3), where  $\mathbf{R1}$  is given in Eq. (7.3.3), and

$$\mathbf{R2} = (1/h^2)\Phi(\mathbf{q}_n, t_n) \quad (7.3.17)$$

To evaluate the Jacobian, derivative operators of Section 5.3 for evaluation of derivatives of  $\mathbf{R1}$  and derivatives of  $\mathbf{R2}$  are

$$\begin{aligned}
\mathbf{R}2_{\ddot{\mathbf{q}}_n} &= \mathbf{0} \\
\mathbf{R}2_{\dot{\mathbf{q}}_n} &= \mathbf{0} \\
\mathbf{R}2_{\mathbf{q}_n} &= (1/h^2)\Phi_q(\mathbf{q}_n, t_n) \\
\mathbf{R}2_{\lambda_n} &= \mathbf{0}
\end{aligned} \tag{7.3.18}$$

Using the HHT integration formulas of Eq. (7.2.15) and the chain rule of differentiation,

$\frac{d\mathbf{R}1}{d\ddot{\mathbf{q}}_n}$  is as given in Eq. (7.3.5) and

$$\frac{d\mathbf{R}2}{d\ddot{\mathbf{q}}_n} = \Phi_q(\mathbf{q}_n, t_n) \tag{7.3.19}$$

The Jacobian of the residual is thus

$$\mathbf{J} = \begin{bmatrix} \frac{d\mathbf{R}1}{d\ddot{\mathbf{q}}_n} & \Phi_q^T(\mathbf{q}_n, t_n) \\ \frac{d\mathbf{R}2}{d\ddot{\mathbf{q}}_n} & \mathbf{0} \end{bmatrix} \tag{7.3.20}$$

Beginning with initial conditions of Eq. (7.2.1), Eq. (7.3.1) is solved for  $\ddot{\mathbf{q}}^0$  and  $\lambda^0$ . At time step  $t_n$ ,  $n = 1, \dots, n$ , the solution variables  $\ddot{\mathbf{q}}_n$  and  $\lambda_n$  are estimated as  $\ddot{\mathbf{q}}_n^1 = \ddot{\mathbf{q}}_{n-1}$  and  $\lambda_n^1 = \lambda_{n-1}$ , or  $\ddot{\mathbf{q}}_n^1 = \ddot{\mathbf{q}}_{n-1}^1 + h\ddot{\mathbf{q}}_n^1$  and  $\lambda_n^1 = \lambda_{n-1}^1 + h\lambda_n^1$  if derivatives  $\ddot{\mathbf{q}}_n^1$  and  $\lambda_n^1$  are computed, as in Section 7.2.6. Generalized coordinates  $\mathbf{q}_n^1$  and  $\dot{\mathbf{q}}_n^1$  are evaluated using Eq. (7.2.8). The Jacobian of Eq. (7.3.6) and residual of Eq. (7.3.17) are evaluated as  $\mathbf{J}^1$  and  $\mathbf{R}^1$ . The first step in Newton - Raphson iteration is carried out by solving

$$\mathbf{J}^1 \mathbf{x} = -\mathbf{R}^1 \tag{7.3.21}$$

With  $\mathbf{P}\ddot{\mathbf{q}} = [\mathbf{I}_{ngc} \quad \mathbf{0}]$  and  $\mathbf{P}\lambda = [\mathbf{0} \quad \mathbf{I}_{nc}]$ ,  $\Delta\ddot{\mathbf{q}}_n^1 = \mathbf{P}_{\ddot{\mathbf{q}}} \mathbf{x}$ ,  $\Delta\lambda_n^1 = \mathbf{P}_\lambda \mathbf{x}$ . Solution variables are updated as

$$\begin{aligned}
\ddot{\mathbf{q}}_n^2 &= \ddot{\mathbf{q}}_n^1 + \Delta\ddot{\mathbf{q}}_n^1 \\
\lambda_n^2 &= \lambda_n^1 + \Delta\lambda_n^1
\end{aligned} \tag{7.3.22}$$

With the Jacobian held fixed at  $\mathbf{J} = \mathbf{J}^1$ , the process is continued until  $\|\mathbf{R}^i\| \leq \text{intol}$ , the integration tolerance, to determine  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ ,  $\ddot{\mathbf{q}}_n$ , and  $\lambda_n$ . The process is repeated at  $t_{n+1}$  and continued until the final time is reached, or a singularity is encountered.

Explicit RKF45 and implicit HHT integration algorithms may be applied for the Index 1 formulation. For the Index 2 and 3 formulations, the HHT implicit algorithm is required. Setting  $\alpha = 0$  in the HHT algorithm yields the trapezoidal algorithm.

## 7.4 Code 7.4 for DAE Simulation of Planar Systems

A general-purpose MATLAB Code 7.4 has been developed that implements the DAE formulations of Sections 7.3.1, 7.3.2, and 7.3.3 for planar multibody system simulation. Revolute, translational, and distance kinematic constraints of Section 3.2 are included, with derivatives presented in Section 3.2 and Appendix 5.B that are required for implementation of explicit and implicit numerical integration methods of Section 4.8 for solution of Index 1, 2, and 3 DAE of this chapter. Applied, gravitational, and internal forces defined by translational- and rotational-*spring-damper-actuators* (*TSDA* and *RSDA*) presented in Section 4.5 are implemented, together with derivatives of Appendix 5.B. Centroidal body fixed reference frames for equations of motion derived in Section 4.2 are employed, with derivative expressions presented in Appendix 5.B. Fixed and variable time step explicit *RKFN45* and *Nystrom4* and implicit *HHT* and *trapezoidal* integration algorithms are implemented.

Integration, error control, and data input segments of the code are essentially identical to those of Code 5.7, presented in Section 5.7, so they are only briefly summarized here. Following an explanation of the computer formulation and MATLAB code in this section, numerical examples are presented in Section 7.5, including those treated with detailed derivations in Section 5.4 and with tangent space ODE Code 5.7 in Section 5.8. The focus in numerical examples is on comparison of results with ODE numerical solution methods.

Components of Code 7.4 of Appendix 7.A that interface with the user are presented in Section 7.4.1, followed by an outline of the body of the code, with which the user need not interact, in Section 7.4.2.

### 7.4.1 User Components of Code

The initial segment of code involves integration and error control parameters that underlie the DAE formulation and associated numerical integration methods. Lines 3 to 12 of Fig. 7.4.1 define parameters that are used to control error in the DAE formulations and numerical integration methods. Relatively tight error control parameters are presented as defaults, which may be modified to seek greater computer efficiency or minimize error. The initial time step is defined in line 14, followed by the maximum time step allowed in variable time step integrators in line 16. The final simulation time is defined in line 19. The integration option to be used is selected from among the four options shown in lines 21 to 24. If alpha is chosen as a nonzero value in the specified range, the implicit HHT algorithm is implemented. If alpha = 0 is selected, the implicit trapezoidal algorithm is implemented. For explicit algorithms, no value of alpha is used.

```
1 %AA_Planar_Multibody_Sim_DAE
2
3 %Integration and Error Control Parameters
4 intol=10^-6; %Tolerance in solving discretized equations of motion
5 Atol=10^-6; %Absolute error tolerance for variable step methods
6 MaxIntiter=8; %Limit on number of implicit integration iterations
7 MaxJcond=100; %Limit on magnitude of Jcond in Trap
8 R1nmax=3000; %Limit on residual in integration
9 MaxECond=20; %Limit on magnitude of ECond
10 PosConstrMax=10^-4; %Limit on position constraint error
11 VelConstrMax=10^-2; %Limit on velocity constraint error
```

```

12 AccConstrMax=10^20; %Limit on acceleration constraint error
13
14 h0=0.001; %Initial time step
15 h=h0;
16 hmax=0.01; %hvar=1, variable h; hvar=2, constant h
17 hvar=1; %hvar=1, variable h;hvar=2, constant h
18
19 tfinal=10; %Final simulation time
20
21 integ=3; %integ=1-ImplicitIndex1; alpha=0-Trapezoidal; alpha<0-HHT
22           %integ=2-ImplicitIndex2; alpha=0-Trapezoidal; alpha<0-HHT
23           %integ=3-ImplicitIndex3; alpha=0-Trapezoidal; alpha<0-HHT
24           %integ=4-ExplicitRKFN45
25
26 alpha=-1/3; %Enter -1/3<=alpha<0 for HHT
27 %alpha not defined for ExplicitNystrom4 or ExplicitRKFN45

```

Figure 7.4.1 Integration and Error Control Parameters

Application data are indexed in lines 31 to 41 of Fig. 7.4.2 to specific applications defined in the AppData Function that is defined in the following. The declaration in line 43 defines which application is implemented in the simulation. The function AppData and parameter definitions of lines 45 through 47 are used throughout the code to pass application data for each simulation. If required for definition of initial conditions that are consistent with constraint, the user may enter code following line 49, such as that for the quick return mechanism of app = 1. Initial accelerations and Lagrange multipliers needed as estimates in initiating numerical integration are computed in lines 71 to 86. Finally, output data desired are defined for each application, as in lines 241 to 248 for the quick return mechanism of app = 1.

```

31 %Application Data
32 %app=1, Quick Return
33 %app=2, Loader
34 %app=3, Slider-Crank
35 %app=4, Multiple Slider-Crank
36 %app=5, Pendulum-Distance constraint
37 %app=6, Pendulum-Revolute constraint
38 %app=7, Pendulum-TSDA
39 %app=8, Double Pendulum
40 %app=9, Lumped Mass Coil Spring-5 masses
41 %app=10, Lumped Mass Coil Spring-10 masses
42
43 app=1;
44
45[nb,ngc,nh,nc,NTSDA,NRSDA,PJDT,PMRD,PTSDAT,PRSDAT,q0,qd0]=AppData(a);
46
47 par=[nb;ngc;nh;nc;g:intol;Atol;h0;hvar;NTSDA;NRSDA];
48
49 %Initial condition calculation, if required
50 if app==1
51 phi2d0=6;
52 Phiq=PhiqEval(0,q0,PJDT,par);
53 EEE=[Phiq;zeros(1,5),1,zeros(1,6)];

```

```

54 RRHS=zeros(11,1);phi2d0];
55 qd0=EEE\RRHS;
56 end

71 %Calculate Initial Acceleration and Lagrange Multipliers
72 M=MEval(PMDT,par);
73 Phiq=PhiqEval(0,q0,PJDT,par);
74 QA=QAEval(0,q0,qd0,PMDT,PTSDAT,PRSDAT,par);
75 Gam=GamEval(0,q0,qd0,PJDT,par);
76 EE=[M,Phiq';Phiq,zeros(nc,nc)];
77 CondEE=cond(EE);
78 RHS=[QA;-Gam];
79 x=EE\RHS;
80 Pqdd=[eye/ngc),zeros/ngc,nc)];
81 PLam=[zeros(nc,ngc),eye(nc)];
82 qdd=Pqdd*x;
83 Lam=PLam*x;
84
85 Qdd(:,1)=qdd;
86 LLam(:,1)=Lam;

241 %Data of Interest (Enter for each application)
242
243 if app==1
244 phi1(n)=q(3);
245 phi2(n)=q(6);
246 x4(n)=q(10);
247 x4d(n)=qd(10);
248 end

```

Figure 7.4.2 Application Data in Main Code

The third component of user entered data is the *AppData Function*, which is identical to that employed in Codes 3.9 and 5.7 of Sections 3.9.1 and 5.7.1. The reader is referred to that section for details.

#### 7.4.2 Computational Components of Code

Computational flow in the main program, which requires no input from the user beyond line 88, is outlined in Fig. 7.4.3. Data that are required for integration methods of Section 7.4.1 are initialized following line 88. Code implementing the four numerical integration methods supported is outlined following line 108. Corrections to approximate solutions to compensate for unenforced velocity and position constraints are computed, using the results of Section 8.2.4, in lines 147 to 180. Total energy for systems simulated is computed following line 188. Output data of interest for each of the applications discussed in Section 7.4.1 are defined following line 241. Finally, position, velocity, and acceleration constraint errors are computed following line 350.

88 %Initialize Data For Integration

```

102 %Integration
103 while t(n)<tfinal
104 n=n+1;

```

```

105 t(n)=t(n-1)+h;
106 tn=t(n);
107 tnm=t(n-1);
108
110 if integ==1 %Implicit Index 1
111 [q,qd,qdd,Lam,R1n,Jiter,JCond]=...
112 ImplicitIndex1(n,tn,Q,Qd,Qdd,LLam,h,PMDT,PTSDAT,PRSDAT,PJDT,par,alpha);
116 end
117
118 if integ==2 %Implicit Index 2
119 [q,qd,qdd,Lam,R1n,Jiter,JCond]=...
120 ImplicitIndex2(n,tn,Q,Qd,Qdd,LLam,h,PMDT,PTSDAT,PRSDAT,PJDT,par,alpha);
124 end
125
126 if integ==3 %Implicit Index 3
127 [q,qd,qdd,Lam,R1n,Jiter,JCond]=...
128 ImplicitIndex3(n,tn,Q,Qd,Qdd,LLam,h,PMDT,PTSDAT,PRSDAT,PJDT,par,alpha);
132 end
133
134 if integ==4 %Explicit RKFN45
135 [q,qd,qdd,Lam,ECond]=ExplicitNystrom4(n,tn,Q,Qd,...
136 h,PMDT,PTSDAT,PRSDAT,PJDT,par);
138 end
146
147 %Corrections if velocity or position errors exceed tolerances
148 Phiq=PhiqEval(tn,q,PJDT,par);
149 if norm(Phiq*qd)>VelConstrMax
150 mu=(Phiq*Phiq')\Phiq*qd;
151 delqd=-Phiq'*mu;
152 qd=qd+delqd;
153 corvel=corvel+1;
154 corvelrpt(n)=corvel;
155 end
156
157 if norm(PhiEval(tn,q,PJDT,par))>PosConstrMax
158 z=zeros(nc,1);
159 nu=zeros(nc,1);
160 Pz=[eye(nc),zeros(nc,nc)];
161 Pnu=[zeros(nc,nc),eye(nc)];
162 err=intol+1;
163 i=0;
164 while err > intol
165 Phiq=PhiqEval(tn,q+z,PJDT,par);
166 Resid=[z+Phiq'*nu;PhiEval(tn,q+z,PJDT,par)];
167 I=eye(nc);
168 JJ=[I+P4Eval(tn,q+z,nu,PJDT,par),Phiq';Phiq,zeros(nc,nc)];
169 w=-JJ\Resid;
170 z=z+Pz*w;
171 nu=nu+Pnu*w;
172 err=norm(Resid);
173 i=i+1;
174 end
175 q=q+z;

```

```

176 corpos=corpos+1;
177 corposrp(n)=corpos;
178 corpositer(corpos)=i;
179 end
180 %End Corrections
181
182 %Record Solution
187
188 %Calculate Total Energy
241 %Data of Interest (Enter for each application)
350 %Calculate constraint error

```

Figure 7.4.3 Main Code Computational Flow

Computing functions that underlie the main code outlined in Fig. 7.4.3 are identified in Fig. 7.4.4. Computing subroutines include the *Add function* that enables adding nonzero submatrices to sparse matrices, below and to the right of the address of the (1,1) term in the submatrix added, in the underlying matrix that was initialized to zeroes. *ATran* evaluates the orientation transformation matrix. *GamEval* and *GamsqqdEval* evaluate terms in Eq. (5.3.18).

Vector partition functions *parPart*, *qPart*, and *xPart* support partitioning of vectors involved into components that are used in kinematic and kinetic computations. Similarly, the Constraint and Data Table Partition Functions listed provide access to elements of the data tables required to implement computation.

Constraint and derivative evaluation functions listed evaluate kinematic constraint expressions and the derivatives defined in Section 3.2 and Appendix 5.B that are used in implementing the formulation and numerical integration of the associated equations of motion. Care is taken to account for the fact that ground is designated by  $j = 0$  and its generalized coordinates are constant, yielding no derivative contribution, but to include geometric quantities that define the constraint of body  $i$  with ground. Similarly, the kinetic and derivative evaluation functions listed evaluate inertial and generalized force terms and derivatives that are defined in Sections 4.2 and 5.3 and Appendix 5.B and are required in formulation and solution of the equations of motion.

Finally, the numerical integration functions listed carry out the numerical integration process, where *ODEfunct* evaluates the Index 1 ODE of Section 7.3.1.1.

Internal details of the MATLAB Functions listed in Fig. 7.4.4 are not presented, since each of the Functions is documented internally and the user need not modify these functions.

#### Computing Functions

- Add
- ATran
- GamEval
- GamsqqdEval

#### User Input Function

- AppData

```

Vector Partition Functions
  parPart
  qPart
  xPart
Constraint and Data Table Partition Functions
  DistPart
  RevPart
  TranPart
  PTSDATPart
  PRSDATPart
Constraint and Derivative Evaluation Functions
  PhiEval
  PhiqEval
  P2Eval
  P3Eval
  P4Eval
  P5Eval
Kinetic and Derivative Evaluation Functions
  MEval
  M2Eval
  QAEval
  QAsqqd
Numerical Integration Functions
  ExplicitNystrom4
  ExplicitRKFN45
  ODEFunct
  ImplicitTanSpTrap
  ImplicitSDIRK54

```

Figure 7.4.4 Computing Functions

Code 7.4 of Appendix 7.A is outlined, to enable the reader to create planar models and carry out simulations using DAE formulations presented in Sections 7.2 and 7.3, without writing detailed computer code.

## 7.5 Planar System Simulation Using Code 7.4

Index 1, 2, and 3 simulations are carried out using the *HHT*, *trapezoidal*, and *RKFN45* algorithms for the double pendulum with unilateral spring presented in Section 5.8.1, to study the influence of stiffness in the model on DAE integration methods. The higher dimensional quick return mechanism of Section 5.8.2 provides for study of the effectiveness of DAE integration methods on a slightly more complex example that is not stiff.

### 7.5.1 Double Pendulum

The *planar double pendulum with unilateral spring* of Section 5.8.1 is used to evaluate effectiveness of the RKFN45, trapezoidal, and HHT integration algorithms, the latter with  $\gamma = -1/3$ , in Index 1, 2, and 3 formulations. The model with unilateral spring in Fig. 5.8.1 serves as a test case for *stiff* systems, as in Section 5.8.1. The *AppData* definition of the model in Code 7.4 of Appendix 7.A is given in Fig. 7.5.1.

```
305 if app==7 %Double Pendulum-Unilateral Spring
307 nb=2; %Number of bodies
308 ngc=3*nb; %number of generalized coordinates
309 nh=2; %Number of holonomic constraints
310 nc=4; %Number of constraint equations
311 nv=ngc-nc;
312 nu=nc;
313 NTSDA=1; %Number of TSDA force elements
314 NRSDA=2; %Number of RSDA force elements
316 ux=[1;0];
317 uy=[0;1];
318 zer=zeros(2,1);
320 %PJDT(12,nh) Joint Data Table
321 %PJDT(:,k)=[t;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
322 %t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
323 %si&sjpr=vectors to Pi&Pj, d=dist., vipr&vjpr=vectors along trans axis
324 PJDT(:,1)=[1;1;0;-ux;zer;0;zer;zer]; %Rev-Body 1 to Ground
325 PJDT(:,2)=[1;1;2;ux;-ux;0;zer;zer]; %Rev-Body 1 to Body 2
327 %PMDT(2,nb) Mass Data Table
328 %PMDT=[[m1;J1],[m2;J2],...,[mn;Jn]]
329 PMDT=[[1;0.3],[1;0.3]];
331 %PTSDAT(10,NTSDA) TSDA Data Table
332 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
333 %i&j=bodies conn., si&jpr=vectors to Pi&j, K=spring constant,
334 %C=damping coefficient, el0=spring free length, F=const. force
335 %Unilateral spring defined by setting spring constant in QA, QAsq,
336 %and QAsdd equal to  $K=K*(1-\text{sign}(q(1)))/2$ 
337 K3=10^5;
338 PTSDAT(:,1)=[1;0;zer;-ux-uy;K3;0;1;0];
340 %PRSDAT(6,NRSDA): RSDA Data Table
341 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
342 %i&j=bodies connected, K=spring constant,
343 %C=damping coefficient, phi0=spring free angle, T=constant torque
344 PRSDAT(:,1)=[1,0,0,0,pi/2,0];
345 PRSDAT(:,2)=[1,2,10^5,10^4,0,0];
```

```

347 %Initial generalized coordinates
349 q0=[1;0;0;3;0;0];
350 qd0=[0;0;0;0;0;0];
351
352 end

```

Figure 7.5.1 AppData, Double Pendulum with Unilateral Spring

In successful simulations with a variety of spring and damper values, results are consistent with those presented in Section 5.8.1. *Average step-size/equation error* or failure (denoted x) of 2 sec simulations carried out with Code 7.4 and data of Section 5.8.1, with intol = e-6, Atol = e-4, and hmax=e-3 are presented in Tables 7.5.1 through 7.5.3 for variable step size trapezoidal and HHT algorithms and Index 1, 2, and 3 formulations. In all cases, constraints that are not enforced in the DAE algorithm are corrected to levels of e-4, e-3, and e-2 for position, velocity, and acceleration constraints, respectively. Stiffness and damping parameters used in simulations are listed in the last three rows of Tables 7.5.1 through 7.5.3. Results are reported for the explicit RKFN45 algorithm in the Index 1 formulation in Table 7.5.1, where it succeeds only for moderately stiff models.

For the *Index 1 formulation*, both the trapezoidal and HHT algorithms provide approximate solutions for quite stiff systems, ultimately failing with the error control algorithm driving step size to zero. The trapezoidal and RKFN45 algorithms yield small equation error, whereas the HHT algorithm, with its algorithmically induced damping, yields larger error, especially of quite stiff systems.

Table 7.5.1 Index 1 Simulation Performance and Failure with Stiff Systems  
(*average step-size/equation error*)

RKFN45	8e-4/e-13	9e-5/e-14	x	x	x
Trapezoidal	e-3/e-11	9.8e-4/e-11	9e-4/e-10	5e-4/e-10	1.6e-4/e-9
HHT	e-3/6e-1	e-4/4	1e-5/5e1	x	x
$K_2$	e4	e5	e6	e7	e8
$C_2$	e3	e4	e5	e6	e7
$K_3$	e4	e5	e6	e7	e8

For the *Index 2 formulation*, the trapezoidal algorithm performs significantly better than the HHT algorithm, with trapezoidal able to treat substantially stiffer systems.

Table 7.5.2 Index 2 Simulation Performance and Failure with Stiff Systems

Trapezoidal	e-3/e-12	e-3/e-11	e-3/e-10	4e-4/e-10	e-4/e-8
HHT	8e-4/e-2	e-4/4	x	x	x
$K_2$	e4	e5	e6	e7	e8
$C_2$	e3	e4	e5	e6	e7
$K_3$	e4	e5	e6	e7	e8

For the *Index 3 formulation*, the *Petzold Imperative* appears to have taken control. As shown in Table 7.5.3, both algorithms fail for stiff cases.

Table 7.5.3 Index 3 Simulation Performance and Failure with Stiff Systems

Trapezoidal	e-3/e-11	e-3/e-12	4e-4/e-11	x
HHT	e-3/7e-4	e-3/2e-3	e-3/5e-3	x
$K_2$	e3	e4	e5	e6
$C_2$	e2	e3	e4	e5
$K_3$	e3	e4	e5	e6

The behavior portrayed in Table 7.5.3 is often used in the literature to suggest that the equations of motion of multibody dynamics are inherently stiff. This is only marginally true. The ODE formulation of Chapter 5, Table 5.8.1, and the Index 1 formulation herein, Table 7.5.1, perform well for far stiffer data sets than do the Index 3 formulations in Table 7.5.3. A better explanation of the Index 3 difficulties lies simply in the *Petzold Imperative*; one should think twice before applying ODE methods directly to Index 3 applications.

As a test of the ability of Index 1, 2, and 3 formulations to control constraint error, the medium stiffness case with  $K_2 = K_3 = e5$  and  $C_2 = e4$  is simulated with trapezoidal and HHT algorithms for 10 sec, with  $\text{intol} = e-6$ ,  $\text{Atol} = e-4$ , and  $\text{hmax}=e-3$ . Maximum errors in norms of position, velocity, and acceleration constraints are presented in Tables 7.5.4 and 7.5.5. For all unenforced constraints, maximum norms of constraint error are several orders of magnitude larger than errors reported for the same system using the tangent space ODE formulation. Whereas the trapezoidal algorithm with constraint error correction succeeded for this data set in the Index 3 formulation, without constraint correction, it failed. In light of theoretical and practical difficulties with the Index 3 formulation, these results suggest that the Index 2 formulation may be the preferable alternative.

Table 7.5.4 HHT Simulation Errors,  $K_2 = K_3 = e5$   $C_2 = e4$

	Position	Velocity	Acceleration
Index 1	4e-3	8e-4	6e-14
Index 2	3e-8	2.5e-15	1.2e-1
Index 3	7e-16	5.5e-6	1e-1

Table 7.5.5 Trapezoidal Simulation Errors,  $K_2 = K_3 = e5$   $C_2 = e4$

	Position	Velocity	Acceleration
Index 1	1.4e-3	3e-4	2e-13
Index 2	3.5e-6	2e-15	8e-2

### 7.5.2 Quick Return Mechanism

The *quick return mechanism* of Section 5.8.2 serves as a higher dimensional application to test numerical methods in Index 1, 2, and 3 DAE formulations without constraint correction. Model and initial condition data are as in Section 5.8.2, given in the *AppData function* of Code 7.4 in Appendix 7.A, shown in Fig. 7.5.2. Error control parameters are selected as  $\text{intol} = \text{e-8}$ ,  $\text{Atol} = \text{e-6}$ , and  $\text{hmax} = \text{e-3}$ . Simulation results with the Index 1 and 2 formulation and trapezoidal and HHT algorithms, for an initial angular velocity 6 rad/sec of the crank, are as in Fig. 5.8.8 of Section 5.8.2. Identical results are obtained with the explicit *RKFN45* algorithm applied to the Index 1 formulation, confirming the conclusion of Section 5.8.2 that this system is *not stiff*. Simulations with the Index 3 formulation and the *trapezoidal* algorithm with constant step size  $h = 0.001$  and variable step size failed. The *HHT* algorithm with constant step size  $h = 0.002$  succeeded, but with warnings that integration error may be large. The fact that the system is not stiff suggests that the *Petzold Imperative* has again intervened, emphasizing that the Index 3 formulation is indeed problematic.

```

4 if app==1 %Quick Return
6 nb=4; %Number of bodies
7 ngc=3*nb; %number of generalized coordinates
8 nh=6; %Number of holonomic constraints
9 nc=11; %Number of constraint equations
10 nv=ngc-nc; %Number of independent coordinates
11 nu=nc; %Number of dependent coordinates
12 NTSDA=0; %Number of TSDA force elements
13 NRSDA=0; %Number of RSDA force elements
17 zer=zeros(2,1);
19 %PJDT(12,nh): Joint Data Table
20 %PJTD(:,k)=[t;i;j;sipr;sjpr;d;vipr;vjpr]; k=joint No.,
21 %t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
22 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis
23 PJDT(:,1)=[1;1;0;-2*ux;zer;0;zer;zer]; %Revolute-bar to ground
24 PJDT(:,2)=[1;2;0;zer;2*uy;0;zer;zer]; %Revolute-crank to ground
25 PJDT(:,3)=[1;2;3;1.5*ux;zer;0;zer;zer]; %Revolute-crank to key
26 PJDT(:,4)=[2;1;3;zer;zer;0;ux;ux]; %Trans.-bar to key
27 PJDT(:,5)=[2;4;0;zer;4*uy;0;ux;ux]; %Trans.-cutter to ground
28 PJDT(:,6)=[3;1;4;2*ux;zer;2.5298;zer;zer]; %Dist.-bar to cutter
30 %PMDT(2,nb): Mass Data Table
31 %PMDT=[[m1;J1],[m2;J2],...,[mn;Jn]];
32 PMDT=[[100;100],[1000;1000],[1;1],[50;50]];
34 %PTSDAT(10,NTSDA): TSDA Data Table
35 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
36 %i&j=bodies connected, si&jpr=vectors to Pi&j, K=spring constant,
37 %C=damping coefficient,el0=spring free length,F=constant force
38 PTSDAT=zeros(10,1);
40 %PRSDAT(6,NRSDA): RSDA Data Table
41 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
42 %i&j=bodies connected, K=spring constant,
43 %C=damping coefficient,phi0=spring free angle,T=constant torque
44 PRSDAT=zeros(6,1);
46 %Initial generalized coordinates
47 q10=[1.2;1.6;0.9273];

```

```

48 q20=[0;2;0];
49 q30=[1.5;2;0.9273];
50 q40=[0;4;0];
52 q0=[q10;q20;q30;q40];
53 qd0=zeros(12,1); %Placeholder, qd0 calculated in main program
55 end

```

Figure 7.5.2 AppData, Quick Return Mechanism

To evaluate the influence of varying initial crank angular velocity on integration algorithm performance, simulations were run with the foregoing data and initial crank angular velocities 6, 12, and 18 rad/sec. Results for initial angular velocity 6 rad/sec were identical to those presented in Fig. 5.8.7. Results for initial angular velocity 12 and 18 rad/sec are with twice and three times the frequency, the same amplitude, twice and three times the peak velocities, and four and nine times the peak accelerations. Data provided in Tables 7.5.6 and 7.5.7 indicate that both algorithms perform well, even for the more extreme dynamic response associated with the higher initial angular velocities. In the Index 1 formulation, the trapezoidal algorithm satisfies unenforced constraints two orders of magnitude more precisely than the HHT formulation.

Table 7.5.6 Constraint Error and Average Step Size, Index 1 Formulation

$\dot{\phi}_2(0)$	Pos err	Vel err	Acc err	h-ave
Trapezoidal				
6	4.5 e-4	5 e-4	5 e-13	5 e-4
12	3.3 e-3	2.5 e-3	1.1 e-12	2.6 e-4
18	9 e-3	7 e-3	2.5e-12	1.7 e-4
HHT				
6	9 e-2	6 e-2	2 e-13	2.1 e-4
12	3 e-1	2.4 e-1	9 e-12	9.3 e-5
18	6.5 e-1	5 e-1	3.5e-12	5.7 e-5

Table 7.5.7 Constraint Error and Average Step Size, Index 2 Formulation

$\dot{\phi}_2(0)$	Pos err	Vel err	Acc err	h-ave
Trapezoidal				
6	4.5 e-5	2 e-14	2 e-2	5.1 e-4
12	1.2 e-4	4 e-14	6 e-2	2.4 e-4
18	1.8 e-4	9e-14	1.5e-1	1.5 e-4
HHT				
6	3.5 e-5	2 e-14	3.5 e-1	2.1 e-4
12	4 e-5	4 e-14	1.5	9.6 e-5
18	4.5e-5	8 e-14	3	6.1 e-5

To gain insight into difficulties with the Index 3 formulation, the HHT algorithm is applied with initial angular velocity of the crank equal to 6 rad/sec, error control parameters defined above,  $h = 0.001$ , and  $t_{final} = 0.5$  sec. The plot of  $\ddot{x}_4 = x_{4dd}$  for the slider in Fig. 7.5.3 shows that it is initially highly oscillatory, but damps to a response similar to that predicted in the Index 1 and 2 formulations in Fig. 7.5.4, obtained with HHT, trapezoidal, and RKF45 algorithms. With a step size  $h = 0.002$ , the initial transient is much less severe, suggesting that the Index 3 formulation is sensitive to small step size.

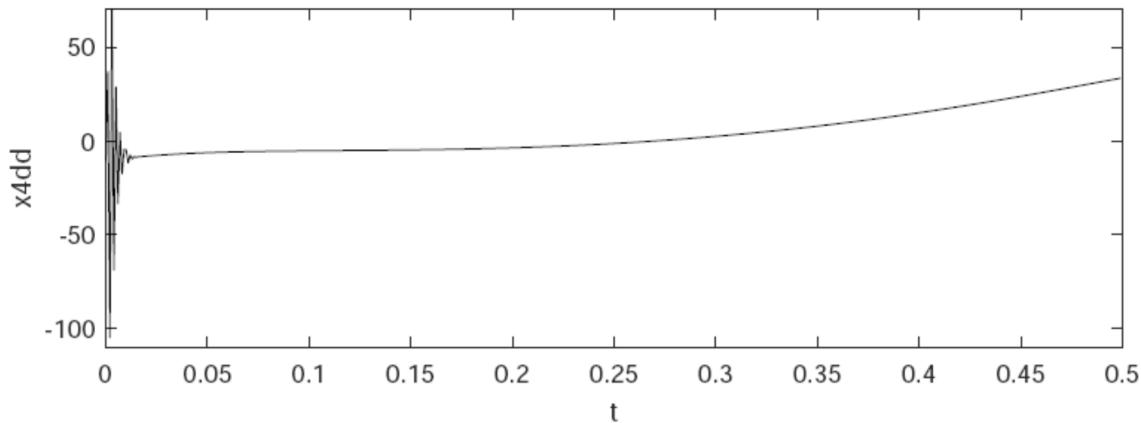


Figure 7.5.3  $x_{4dd}$  vs Time, HHT Index 3,  $\alpha = -1/3$

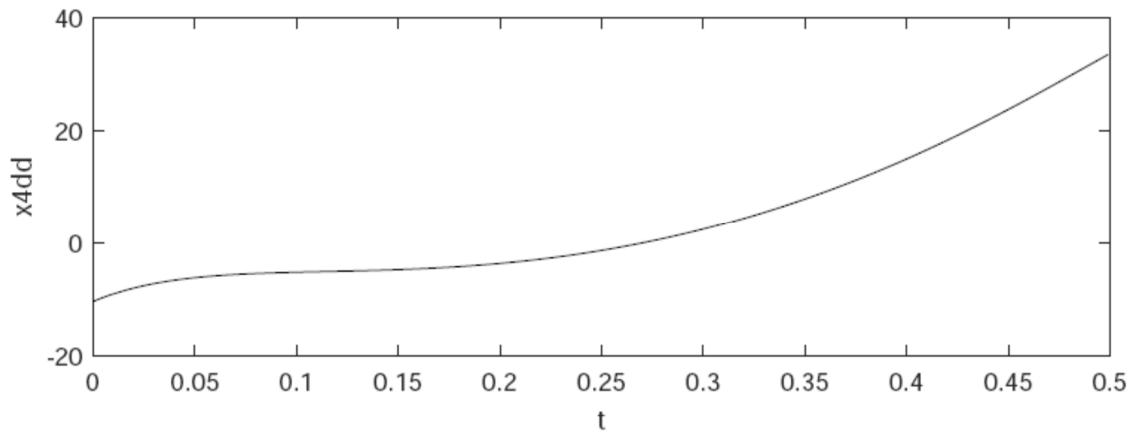


Figure 7.5.4  $x_{4dd}$  vs Time, HHT Index 2,  $\alpha = -1/3$

Simulation results for increasingly stiff models of the double pendulum with unilateral spring show that both the HHT and trapezoidal algorithms yield reliable results with Index 1 and 2 formulations. With the Index 3 formulation, the trapezoidal algorithm with variable step size fails, whereas the HHT algorithm marginally succeeds. For the higher dimensional quick return mechanism, all is well in the Index 1 and 2 formulations, but all simulations fail or display large error in the Index 3 formulation.

## 7.6 Code 7.6 for DAE Simulation of Spatial Systems

A general-purpose Code 7.6 has been created that implements the DAE formulation of Sections 7.3.1, 7.3.2, and 7.3.3 for spatial multibody system simulation. Kinematic constraints of Section 3.3 are included, with derivatives presented in Section 3.3 and Appendix 5.B that are required to implement explicit and implicit numerical integration methods of Section 4.8 for solution of Index 1, 2, and 3 DAE of this chapter. Applied, gravitational, and internal forces defined by *translational-spring-damper-actuators* (*TSDA*) presented in Section 4.5 are implemented. Centroidal body fixed reference frames for equations of motion derived in Section 4.3 are employed. Fixed time step explicit *RKFN45* and implicit *HHT* and *trapezoidal* methods are implemented.

Integration and error control and data input segments of the code are essentially identical to those of Code 7.4, so they are only briefly summarized here. Following an explanation of the computer formulation and MATLAB code in this section, numerical examples are presented in Section 7.7, including those treated with detailed derivations in Section 5.4 and with tangent space ODE Code 5.9 in Section 5.10. The focus in numerical examples is to compare results with ODE numerical solution methods.

Components of Code 7.6 of Appendix 7.A that interface with the user are presented in Section 7.6.1, followed by an outline of the body of the code, with which the user need not interact, in Section 7.6.2.

### 7.6.1 User Components of Code

The initial segment of code involves integration and error control parameters that underlie the DAE formulation and associated numerical integration methods in lines 1 to 26. Since this segment of code is essentially identical to that in Section 7.4.1, the reader is referred to that section.

Application data are indexed in lines 30 to 36 of Fig. 7.6.1 to specific applications defined in the App Data function defined in the following. The declaration in line 38 defines which application is implemented in the simulation. The AppData and parameter definitions of lines 31 through 36 are used throughout the code to pass application data for each simulation. If required for definition of initial conditions that are consistent with constraint, the user may enter code following line 44. Initial accelerations and Lagrange multipliers that are needed as estimates in initiating numerical integration are computed in lines 60 to 76. Finally, output data desired are defined for each application, as in lines 214 to 220 for the pendulum of app = 1.

```
30 %Application Data
31 %app=1, Pendulum, Spherical to Ground
32 %app=2, Top, Spherical to Ground
33 %app=3, One Body Pendulum, Dist. to Ground
34 %app=4 Two Body Pendulum
35 %app=5, One Body Cylindrical with Spring
36 %app=6, Spatial Slider-Crank
37
38 app=2;
39
40 [nb,ngc,nh,nc,NTSDA,SJDT,SMDT,STSDAT,q0,qd0]=AppData(app);
```

```

41
42 par=[nb;ngc;nh;nc;g;intol;Atol;h0;hvar;NTSDA];
43
44 %Initial condition calculation, if required

60 %Calculate Initial Acceleration and Lagrange Multipliers
61 M=MEval(q0,SMDT,par);
62 Phiq0=PhiqEval(0,q0,SJDT,par);
63 QA=QAEval(0,q0,qd0,SMDT,STSDAT,par);
64 S=SEval(q0,qd0,SMDT,par);
65 Gam=GamEval(0,q0,qd0,SJDT,par);
66 EE=[M,Phiq0';Phiq0,zeros(nc,nc)];
67 EEcond=cond(EE);
68 RHS=[QA+S;-Gam];
69 x=EE\RHS;
70 Pqdd=[eye(ngc),zeros(ngc,nc)];
71 PLam=[zeros(nc,ngc),eye(nc)];
72 qdd=Pqdd*x;
73 Lam=PLam*x;
74
75 Qdd(:,1)=qdd;
76 LLam(:,1)=Lam;

214 %Data of Interest (Enter for each application)
215
216 if app==1
217 x1(n)=q(1);
218 y1(n)=q(2);
219 z1(n)=q(3);
220 end

```

Figure 7.6.1 Application Data in Main Code

The third component of user entered code is the *AppData Function*, which is identical to that employed in Code 5.7 and defined in Fig. 5.7.2 of Section 5.7.1. The reader is referred to that section for details.

### 7.6.2 Computational Components of Code

Computational flow in the main program, which requires no input from the user, is identical to that of Code 7.1 and is outlined in Fig. 7.4.2. The reader is referred to Section 7.4.2 for details.

Computing functions that underlie the main code outlined in Fig. 7.4.2 are identified in Fig. 7.6.2. Computing subroutines include the *Add function* that enables adding nonzero submatrices to sparse matrices, below and to the right of the address of the (1,1) term in the submatrix added, in the underlying matrix that was initialized to zeroes. *ATran* evaluates the orientation transformation matrix. *GamEval* and *GamsqqdEval* evaluate terms in Eq. (5.3.18).

Vector partition functions *parPart*, *qPart*, and *xPart* support partitioning of vectors involved into components that are used in kinematic and kinetic computations. Similarly, the Constraint and Data Table Partition Functions listed provide access to elements of the data tables required to implement computation.

Constraint and derivative evaluation functions listed evaluate kinematic constraint expressions and derivatives defined in Section 3.3 and Appendix 5.B that are needed in implementing the formulation and numerical integration of the associated equations of motion. A substantial number of functions are included for building block constraints under the heading bbxxx. Care is taken to account for the fact that ground is designated by  $j=0$  and its generalized coordinates are constant, yielding no derivative contribution, but including geometric quantities that define a constraint of body  $i$  with ground. Similarly, the kinetic and derivative evaluation functions listed evaluate inertial and generalized force terms and derivatives defined in Sections 4.3 and 5.3 and Appendix 5.B that are required in the DAE formulation and solution of the equations of motion.

Finally, numerical integration functions listed carry out the numerical integration process, where *ODEfunct* evaluates the Index 1 ODE of Section 7.3.1.1.

Internal details of the MATLAB Functions listed in Fig. 7.6.2 are not presented, since each of the Functions is documented internally and the user need not modify these functions in applications.

#### Computing Functions

- Add
- atil
- EEval
- GEval
- KEval
- BTran
- ATran
- GamEval
- GamsqqdEval

#### User Input Function

- AppData

#### Vector Partition Functions

- parPart
- qPart
- xPart
- pNormPart

#### Constraint and Data Table Partition Functions

- CylPart
- DistPart
- RevPart
- TranPart
- StrutPart
- RevSphPart
- UnivPart
- STSDATPart

Constraint and Derivative Evaluation Functions

bbxxx  
PhiEval  
PhiqEval  
P2Eval  
P3Eval  
P4Eval  
P5Eval

Kinetic and Derivative Evaluation Functions

MEval  
M2Eval  
QAEval  
QAsqqd

Numerical Integration Functions

ExplicitNystrom4  
ExplicitRKFN45  
ODEfunct  
ImplicitTanSpTrap  
ImplicitSDIRK54  
Jacob  
Resid

Figure 7.6.2 Computing Functions

Code 7.6 of Appendix 7.A is outlined, to enable the reader to create spatial models and carry out simulations, without writing detailed computer code.

## 7.7 Spatial System Simulation Using DAE Code 7.6

Three spatial examples that are simulated using tangent space ODE Code 5.9 of Appendix 5.A are treated here using DAE Code 7.6. Kinematic and kinetic characteristics of the models are defined in *AppData Functions*, just as in Section 5.9. Code 7.6 is then run and results compared with those of the ODE formulation.

### 7.7.1 Spin Stabilized Top

The *spin stabilized top* studied in Sections 5.1, 5.4.2.1, 5.6.1.1, and 5.10.1 is simulated using the DAE approach in Code 7.6. Plots of the x-y trajectory and of *total energy* vs time for initial angular velocity of 13.5 rad/sec are presented in Fig. 7.7.1. The plot of x-y trajectory at the bottom left that is obtained using the Index 1 formulation with the trapezoidal integrator is virtually identical to results presented in Figs. 5.1.3 and 5.10.2. Consistent with the conservative character of this system, the total energy plot at the bottom right of Fig. 7.7.1 shows that conservation of energy is accurately accounted for by the trapezoidal integrator. Identical results are obtained using the *trapezoidal* integrator with the Index 2 formulation and the *RKFN45* integrator with the Index 1 formulation. The system is therefore clearly *not stiff*. Nevertheless, the trapezoidal integrator fails in the Index 3 formulation. This is a clear indicator that instability of the trapezoidal approach in the Index 3 DAE formulation is a result of the *Petzold Imperative*; i.e., DAE are not ODE.

In contrast to the foregoing, results presented in the first three rows of Fig. 7.7.1 using the *HHT* integrator in all three Index formulations are completely inaccurate. This is due to the dominant effect of *numerical damping* introduced in the HHT formulation to enhance numerical stability, resulting in the loss in total energy shown in the plots on the right as time proceeds. Constraint error data shown in the text box of the right plots also explain the poor performance of the HHT integrator over long time simulations. These difficulties arise in this application, due in part to the *long simulation time* that is required to establish convergence to the spin stabilized vertical trajectory. In an attempt to compensate for these difficulties, the velocity correction scheme of Section 7.2.4 is incorporated in the Index 1 HHT formulation. While this leads to a reduction in loss of total energy, the approximate solution is totally corrupted. This is likely a result of the *Larry Bird Paradox* of Section 7.2.

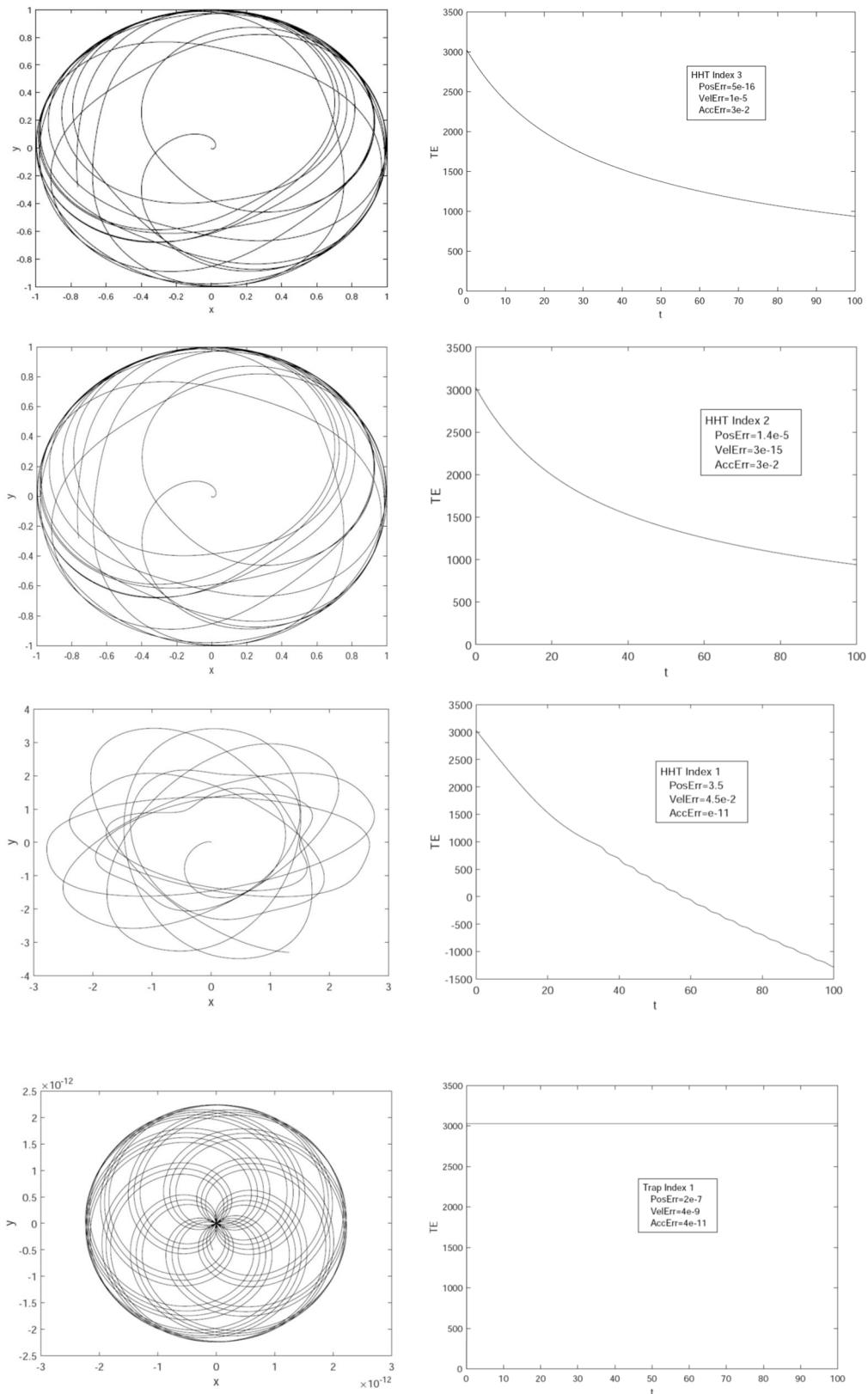


Figure 7.7.1 Spin Stabilized Top with  $\omega_0 = 13.5 \text{ rad/sec}$

### 7.7.2 Double Pendulum

The *spatial double pendulum* studied in Sections 5.4.4, 5.6.2, and 5.10.2 is simulated using the DAE approach in Code 7.6. Results using  $\text{intol} = \text{e-6}$ ,  $\text{Atol} = \text{e-5}$ , and  $\text{hmax} = \text{e-3}$  over a 20 sec simulation with *RKFN45*, *trapezoidal*, and *HHT* integrators in the Index 1 formulation, the trapezoidal and HHT integrators in the Index 2 formulation, and the HHT integrator in the Index 3 formulation are essentially identical to those reported using tangent space ODE and Index 0 DAE formulations in Sections 5.4.4, 5.6.2, and 5.10.2. The fact that the explicit RKFN45 integrator succeeds indicates that the system is not stiff. Nevertheless, the trapezoidal integrator fails in the Index 3 formulation. This is another invocation of the *Petzold Imperative*.

To assess the ability of integrators in the various Index formulations to successfully enforce constraints, maximum norms of position (configuration), velocity, and acceleration constraint errors are presented in Table 7.7.1. Interestingly, the high order explicit RKFN45 integrator led to several orders of magnitude less error in unenforced constraints in the Index 1 formulation, contrasted with the poor performance of the HHT integrator. In the Index 2 formulation, the trapezoidal and HHT formulations exhibited similar performance. In the Index 3 formulation, the HHT integrator exhibited good results and the trapezoidal integrator failed.

Table 7.7.1 Spatial Double Pendulum Simulation Constraint Error

	RKFN Ind 1	HHT Ind 1	Trap Ind 1	HHT Ind 2	Trap Ind 2	HHT Ind 3
Pos Err	3e-8	1.2e-1	9e-6	4e-6	3.8e-6	8e-16
Vel Err	2.7e-9	1.3e-2	6e-5	8e-16	1e-15	4e-5
Acc Err	4e-13	2.5e-14	2.5e-14	1.3e-1	9e-3	1.4e-1

### 7.7.3 Slider-Crank

The *spatial slider-crank* studied in Section 5.10.4 is simulated using the DAE approach in Code 7.6. Results using  $\text{intol} = \text{e-6}$ ,  $\text{Atol} = \text{e-5}$ , and  $\text{hmax} = \text{e-3}$  over a 0.25 sec simulation with RKFN45, trapezoidal, and HHT integrators in the Index 1 formulation, the trapezoidal and HHT integrators in the Index 2 formulation, and the HHT integrator in the Index 3 formulation are essentially identical to those reported in Section 5.10.4, using tangent space ODE and Index 0 DAE formulations. The fact that the explicit RKFN45 integrator succeeded indicates that the system is not stiff. Nevertheless, the trapezoidal integrator failed in the Index 3 formulation; another invocation of the *Petzold Imperative*.

Maximum norms of position, velocity, and acceleration constraint errors presented in Tables 7.7.2 and 3 show the same trends as in Table 7.7.1. Perhaps a bit surprising, the integrators performed well even for the near singular system with  $d_3 = 0.23$  m, according to data in Table 7.7.4.

Table 7.7.2 Spatial Slider-Crank Simulation Constraint Error, d3=0.4 m

	RKFN Ind 1	HHT Ind 1	Trap Ind 1	HHT Ind 2	Trap Ind 2	HHT Ind 3
Pos Err	6e-8	1e-3	8e-4	4.6e-7	7.5e-6	1.1e-16
Vel Err	3e-7	9e-3	6e-3	1e-14	1.2e-14	2e-5
Acc Err	1e-12	9e-13	1.4e-12	6e-1	3e-1	7e-1

Table 7.7.3 Spatial Slider-Crank Simulation Constraint Error, d3=0.3 m

	RKFN Ind 1	HHT Ind 1	Trap Ind 1	HHT Ind 2	Trap Ind 2	HHT Ind 3
Pos Err	1.3e-7	6.5e-4	1.5e-3	4.3e-7	1.7e-5	1.1e-16
Vel Err	9e-7	9e-3	1.2e-2	1e-14	1.2e-14	4e-5
Acc Err	9e-13	1.4e-12	1.4e-12	1	3e-1	1

Table 7.7.4 Spatial Slider-Crank Simulation Constraint Error, d3=0.23 m

	RKFN Ind 1	HHT Ind 1	Trap Ind 1	HHT Ind 2	Trap Ind 2	HHT Ind 3
Pos Err	2e-6	7.5e-3	1.6e-3	2.5e-6	1.3e-5	2e-16
Vel Err	1.8e-5	6e-2	1.4e-2	1.3e-14	1.6e-14	1.2e-4
Acc Err	1.6e-12	1.8e-12	1.6e-12	3	3.7	3.5

Long time simulation results for the spinning top show that numerical damping embedded in the HHT integrator to enhance stability leads to erroneous results, in contrast to good results obtained with the explicit RKFN45 and implicit trapezoidal integrators.

Simulation results for the spatial double pendulum and slider-crank models show that RKFN45, HHT, and trapezoidal integrators yield reliable results with the Index 1 formulation, as do HHT and trapezoidal integrators in the Index 2 formulation. With the Index 3 formulation, the trapezoidal and RKFN45 algorithms fail, whereas the HHT integrator succeeds, as suggested by the Petzold Imperative.

## **Appendix 7.A DAE Solution Code**

*Code 7.4 Planar Multibody Simulation-DAE  
Code 7.6 Spatial Multibody Simulation-DAE*

## Appendix 7.B Key Formulas, Chapter 7

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q(\mathbf{q}, t)^T \lambda = \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \quad \begin{aligned} \Phi(\mathbf{q}^0, t^0) &= \mathbf{0} \\ \Phi_q(\mathbf{q}^0, t^0) \dot{\mathbf{q}}^0 &= -\Phi_t(\mathbf{q}^0, t^0) \end{aligned} \quad (7.1.4) \quad (7.1.5)$$

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \Phi_q(\mathbf{q}, t)^T \\ \Phi_q(\mathbf{q}, t) & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ -\gamma \end{bmatrix} \quad (7.1.6)$$

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi_q(\mathbf{q}, t)\ddot{\mathbf{q}} &= -\left(\Phi_q(\mathbf{q}, t)\dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - 2\Phi_{qq}(\mathbf{q}, t)\dot{\mathbf{q}} - \Phi_{tt}(\mathbf{q}, t) \equiv -\gamma(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \mathbf{q}(t^0) &= \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0 \end{aligned} \quad (7.2.1)$$

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi_q(\mathbf{q}, t)\dot{\mathbf{q}} &= -\Phi_t(\mathbf{q}, t) \\ \mathbf{q}(t^0) &= \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0 \end{aligned} \quad (7.2.3)$$

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\lambda &= \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \\ \Phi(\mathbf{q}, t) &= \mathbf{0} \\ \mathbf{q}(t^0) &= \mathbf{q}^0; \quad \dot{\mathbf{q}}(t^0) = \dot{\mathbf{q}}^0 \end{aligned} \quad (7.2.5)$$

## CHAPTER 8

### Simulation of Friction in Multibody Dynamics

#### 8.0 Introduction

Modeling friction in multibody system simulation is a daunting task that has been pursued for decades, with no comprehensive solution to date. Progress in this regard is impeded by the complexity of quantifying constraint reaction forces and representing friction forces as functions of constraint forces, relative velocities between contact points and surfaces in joints, and constitutive relations involving material interaction due to contact and viscous effects of lubricants. A second factor impeding progress in simulation of multibody systems with friction, as reported in the literature, is apparent stiffness of DAE of motion, due to friction effects.

Approaches used in this chapter to address some of the challenges noted include (1) implementing a continuous approximation of friction effects, including stiction; (2) evaluating continuous friction approximations relative to models of discontinuous Coulomb friction; and (3) utilizing the index 0 DAE formulation to avoid apparent stiffness.

A *continuous model of friction* is used in Section 8.1 with the tangent space Index 0 DAE formulation of Section 5.5 for simulation of a three-mass model problem. Analytical *criteria for stiction* are derived for the model problem in Section 8.2, using the *discontinuous Coulomb friction* to define the onset of and departure from stiction events with redundant equations of constraint. The tangent space formulation with implicit trapezoidal integration is applied to this analytical model to compute dynamic response, determine ranges of constraint forces that may occur during periods of stiction, and demonstrate that dynamic response is a discontinuous function of model parameters when stiction occurs. Cartesian coordinate models of higher dimension are presented in Section 8.3, for three and four mass model problems with *continuous reparameterization of friction* that encounter redundancy in constraints during periods of stiction.

Formulations for modeling and simulation of friction effects in planar and spatial multibody systems are presented in Sections 8.4 and 8.5, using kinematic constraints presented in Sections 3.2 and 3.3. Constraint reaction forces on bodies that are connected by joints that support friction are derived as functions of Lagrange multipliers. Friction forces that act on bodies are calculated as functions of joint geometry, constraint reaction forces that are functions of Lagrange multipliers and relative velocities at constraint contact points that are determined by system kinematics. Friction forces are introduced into the Index 0 DAE formulation of system equations of motion in Section 8.6.

General-purpose Codes 8.7. and 8.9 for simulation of planar and spatial systems with friction are presented in Sections 8.7 and 8.9, respectively. Numerical examples of planar and spatial system simulation with friction are presented in Sections 8.8 and 8.10, reinforcing conclusions drawn in simulation of more modest model problems in prior sections. MATLAB computer codes for all applications treated are presented in Appendix 8.A and key formulas for the chapter are tabulated in Appendix 8.B.

## 8.1 Modeling Friction in Multibody Dynamics

Modeling friction in multibody system simulation is a daunting task. For deformable bodies, the challenge is of monumental proportion, as outlined in a literature review by Berger (2002). Even for rigid bodies, many models have been proposed (Marques, Flores, Pimenta Claro, and Lankarani, 2016; Pennestri, Rossi, Salvini, and Velentini, 2016; Brown and McPhee, 2016), but their effective implementation in multibody dynamics remains an open challenge. Progress in this regard is impeded by the complexity of quantifying *constraint reaction force* and representing *friction force* as a function of *constraint geometry* and reaction forces, relative velocities between contact points and surfaces in joints, and *constitutive relations* involving material interaction due to contact and viscous effects of lubricants.

A second factor impeding progress in simulation of multibody systems with friction, as reported in the literature cited above, is apparent stiffness of differential-algebraic equations (DAE) of motion due to friction effects. Resolving these issues, in the context of spatial multibody dynamics, is a near-impossible task. The objective of this Section is to formulate and analyze multibody static and dynamic model problems that embody challenges in modeling and simulation of friction effects, without the extreme geometric and analytical complexity of spatial multibody systems. Specific challenges addressed include (1) implementing continuous approximations of friction effects, including stiction; (2) evaluating continuous approximations relative to models that represent discontinuous effects; and (3) utilizing the tangent space Index 0 DAE formulation of Chapter 5 to account for nonlinearities due to friction.

### 8.1.1 Static Coulomb Friction in a Constrained System

As an elementary precursor to analysis of friction in a dynamic setting, consider two blocks on a horizontal plane that are connected by a massless bar and acted on by an applied force  $F \geq 0$ , as shown in Fig. 8.1.1. With *Coulomb friction* acting between the blocks and plane and with the blocks at rest, the objective is to determine the smallest value of  $F$  that leads to breaking the *static friction*, or *stiction*, bonds between the blocks and plane and leads to sliding motion.

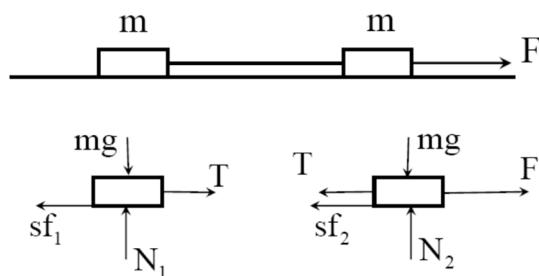


Figure 8.1.1 Constrained Blocks on Horizontal Plane

For blocks at rest, with free body diagrams shown in Fig. 8.1.1,  $T$  is tension in the bar that connects the blocks, and normal forces between the blocks and plane are  $N_1 = N_2 = mg$ . Necessary and sufficient conditions for the blocks to remain at rest are equilibrium conditions  $T - sf_1 = 0$  and  $F - T - sf_2 = 0$  and *Coulomb friction inequalities*  $|sf_i| / |N_i| \leq \mu_s$ ,  $i = 1, 2$ , where  $\mu_s$  is the *coefficient of static friction*. *Stiction forces* are thus  $sf_1 = T$  and  $sf_2 = F - T$  and *Coulomb friction limits on stiction forces* are

$$\begin{aligned}|T| &= |sf_1| \leq \mu_s |N_1| = \mu_s mg \\ |F - T| &= |sf_2| \leq \mu_s |N_2| = \mu_s mg\end{aligned}\quad (8.1.1)$$

For a given applied force  $F > 0$ , stiction occurs if and only if the set of  $T$  that satisfy the inequalities of Eq. (8.1.1) is nonempty. Otherwise the set of feasible values of  $T$  is empty and stiction cannot be supported. This may be viewed as an implication of *Newton's first law*; i.e., bodies at rest tend to stay at rest, only making the transition to nonzero motion if applied forces are sufficiently large.

Inequalities equivalent to those of Eq. (8.1.1) are

$$\begin{aligned}-\mu_s mg &\leq T \leq \mu_s mg \\ F - \mu_s mg &\leq T \leq F + \mu_s mg\end{aligned}\quad (8.1.2)$$

The set of all  $T$  that satisfy the inequalities of Eq. (8.1.2), equivalently Eq. (8.1.1), is defined by the inequalities  $\max(-\mu_s mg, F - \mu_s mg) \leq T \leq \min(\mu_s mg, F + \mu_s mg)$ . Since  $F \geq 0$ , a necessary and sufficient condition for equilibrium is

$$F - \mu_s mg \leq T \leq \mu_s mg \quad (8.1.3)$$

If  $F > 2\mu_s mg$ , the set of  $T$  that satisfy the Coulomb friction inequalities of Eqs. (8.1.3) is empty, stiction is not possible, and the blocks will slide. If  $F \leq 2\mu_s mg$ , the set of  $T$  that satisfy the Coulomb friction inequalities of Eqs. (8.1.3) is not empty, the blocks are in equilibrium, acceleration is zero, and the blocks remain at rest; i.e., stiction occurs.

A point of interest with this example is that the largest value of  $F$  such that the set defined by Eqs. (8.1.3) is not empty causes both inequalities of Eqs. (8.1.1) to be equalities; i.e., with  $F = 2\mu_s mg$  and  $T = \mu_s mg$  from Eq. (8.1.3), the inequalities of Eqs. (8.1.1) reduce to

$$\begin{aligned}|sf_1| &= |T| = \mu_s mg = \mu_s |N_1| \\ |sf_2| &= |F - T| = |2\mu_s mg - \mu_s mg| = \mu_s mg = \mu_s |N_2|\end{aligned}\quad (8.1.4)$$

### 8.1.2 Three Mass Model with Friction and Stiction

#### 8.1.2.1 Mechanism Model

The mechanism shown in Fig. 8.1.2 is comprised of three lumped masses that slide along orthogonal  $x$  and  $y$  axes in a vertical plane, acted on by a distance constraint between masses  $m_1$  and  $m_2$ , linear springs, gravitational force in the negative  $y$  direction, and Coulomb friction.

Generalized coordinates are  $\mathbf{q} = [q_1 \ q_2 \ q_3]^T$  shown in Fig 8.1.2, constrained by a massless rod of length  $\ell$  between masses one and two; i.e., the holonomic constraint

$$(q) = (q_1^2 + q_2^2 - \ell^2) / 2 = 0 \quad (8.1.5)$$

The Jacobian of this constraint is  $\mathbf{q}' = [q_1 \ q_2 \ 0]$ .

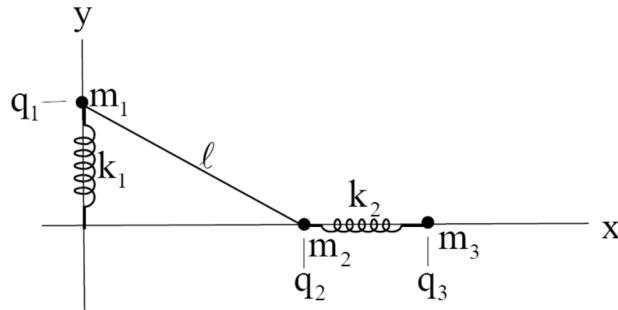


Figure 8.1.2 Three Mass Planar Mechanism

From Eq. (4.10.8), without Euler parameters, the generalized force that acts on the system due to the constraint between masses one and two is

$$\mathbf{Q}^C = -\lambda \Phi_q^T = - [q_1 \quad q_2 \quad 0]^T \quad (8.1.6)$$

Since  $\mathbf{W} = {}^T \mathbf{q} \mathbf{Q}^C = -\lambda q_1 \dot{q}_1 - \lambda q_2 \dot{q}_2$ , force  $-\lambda q_1$  acts on mass  $m_1$  along the  $y$  axis and force  $-\lambda q_2$  acts on mass  $m_2$  along the  $x$  axis. These constraint forces, gravitational forces, spring forces acting on the masses, and reaction forces that act on the massless bar are shown in Fig. 8.1.3. Since the bar is massless, it is required that reaction forces on the bar due to contact with masses one and two are  $F_1 = -q_2$  and  $F_2 = -q_1$ . When the masses are in motion, with a dynamic coefficient of friction  $\mu_d$  acting between the axes and masses, Coulomb friction forces on masses one, two, and three are  $-N_1 \text{sign}(\dot{q}_1)$ ,  $-N_2 \text{sign}(\dot{q}_2)$ , and  $-m_3 g \text{sign}(\dot{q}_3)$ , respectively. Free body diagrams of the masses and the massless bar in Fig 8.1.3 yield

$$\begin{aligned} N_1 &= F_1 = -q_2 \\ N_2 &= F_2 + m_2 g = m_2 g - q_1 \end{aligned} \quad (8.1.7)$$

and the equations of motion are

$$\mathbf{M}\ddot{\mathbf{q}} + \begin{bmatrix} q_1 \\ q_2 \\ 0 \end{bmatrix} = \begin{bmatrix} -m_1 g - k_1 q_1 \\ k_2(q_3 - q_2 - 1) \\ -k_2(q_3 - q_2 - 1) \end{bmatrix} + \begin{bmatrix} -\mu_d |q_2| \text{sign}(\dot{q}_1) \\ -\mu_d |q_1 - m_2 g| \text{sign}(\dot{q}_2) \\ -\mu_d m_3 g \text{sign}(\dot{q}_3) \end{bmatrix} \equiv \mathbf{Q}^A \quad (8.1.8)$$

where  $\mathbf{M} = \text{diag}(m_1, m_2, m_3)$ .

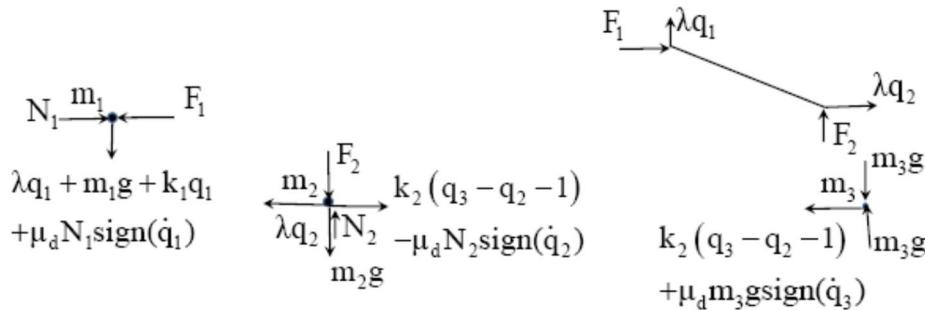


Figure 8.1.3 Free Body Diagrams

### 8.1.2.2 Continuous Friction-Slip Velocity Function

In order to avoid difficulties with the physically unrealistic discontinuous friction effects when sliding velocity is near zero (Brown and McPhee, 2016; Marques, Flores, Claro, and Lankarani, 2016; Pennestri, Rossi, Salvini, and Velentini, 2016), a *continuous friction approximation* with local variation in magnitude near  $\dot{q} = 0$  is used. The model adopted was developed by Brown and McPhee (2016) to account for phenomena that lead to static and dynamic friction. As a function of model and system parameters, the *continuous friction force* in this model is

$$F_f = -F_n(\mathbf{q}, \lambda)S(v, \boldsymbol{\mu}) \quad (8.1.9)$$

where  $F_n(\mathbf{q}, \lambda)$  is the absolute value of normal force of contact, as a function of generalized coordinates and Lagrange multipliers that depend on system kinematics and the DAE of motion, and

$$S(v, \boldsymbol{\mu}) = \mu_d \tanh(4v/v_t) + (\mu_s - \mu_d)(v/v_t)/((v/2v_t)^2 + 3/4)^2 \quad (8.1.10)$$

where  $v$  is slip velocity,  $v_t$  is a transition velocity ( $2v_t$  is the difference between velocities at maximum and minimum transition friction force), and  $\boldsymbol{\mu} = [\mu_s \quad \mu_d]^T$ , where  $\mu_d$  is the *coefficient of dynamic friction* and  $\mu_s$  is the *coefficient of static friction*. The graph of this function, for  $F_n(\mathbf{q}, \lambda) = 1$ ,  $v_t = 0.01$ ,  $\mu_d = 0.3$ , and  $\mu_s = 0.5$  is presented in Fig. 8.1.4, showing that friction force is negative for positive slip velocity and positive for negative slip velocity.

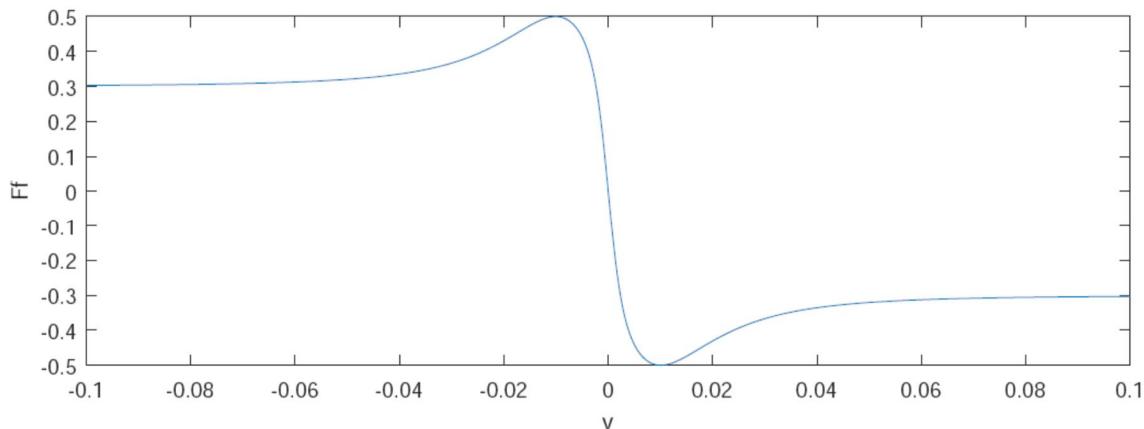


Figure 8.1.4 Continuous Friction Force Function

This model is more realistic for mechanical systems with some lubrication than the classical discontinuous Coulomb friction model. With some form of lubricant in zones of contact for realistic mechanical systems, the Coulomb discontinuous model is a fantasy. While the continuous model suggested is an idealization of physical reality, it is superior to *unrealistic discontinuous Coulomb friction*.

The derivative of the *friction force function* of Eq. (8.1.9) with respect to  $v$ , using subscript notation for derivative, is

$$F_f v = -F_n(\mathbf{q}, \lambda) S'(v, \mu) \quad (8.1.11)$$

where

$$\begin{aligned} S'(v, \mu) \equiv S(v, \mu)_v &= (4\mu_d / v_t) \left( 1 - \tanh^2(4v/v_t) \right) + (\mu_s - \mu_d)(1/v_t) / \left( v/(2v_t)^2 + 3/4 \right)^2 \\ &\quad - (\mu_s - \mu_d)(v^2/v_t^3) / \left( v/(2v_t)^2 + 3/4 \right)^3 \end{aligned} \quad (8.1.12)$$

As in Eq. (8.1.8), the normal force  $F_n(\mathbf{q}, \lambda)$  is often the absolute value of an expression in  $\mathbf{q}$  and  $\lambda$ ; e.g., in Eq. (8.1.7),  $F_n(\mathbf{q}, \lambda) = |N_1(\mathbf{q}, \lambda)| = |\lambda q_2|$ . Since  $|x| = x \text{sign}(x)$ , the derivative  $d|x|/dx$  is discontinuous. A continuously differentiable approximation of the function  $\text{sign}(x)$ , consistent with the friction force approximation of Eq. (8.1.10), is

$$|x| \approx c \text{sign}(x) \equiv \begin{cases} \text{sign}(x), & \text{if } |x| > v_t \\ \sin\left(\frac{\pi x}{2v_t}\right), & \text{if } |x| \leq v_t \end{cases} \quad (8.1.13)$$

with derivative

$$d|x|/dx \approx d \text{csign}(x) \equiv \begin{cases} 0, & \text{if } |x| > v_t \\ \frac{\pi}{2v_t} \cos\left(\frac{\pi x}{2v_t}\right), & \text{if } |x| \leq v_t \end{cases} \quad (8.1.14)$$

### 8.1.2.3 Tangent Space Index 0 DAE Model with Friction

With the *friction approximation* of Section 8.1.2.2, equations of motion with friction that generalize those of Section 8.1.2.1 are

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \Phi_q(\mathbf{q})^T \lambda - \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \lambda) = \mathbf{0} \quad (8.1.15)$$

with kinematic configuration, velocity, and acceleration constraints

$$\begin{aligned} \Phi(\mathbf{q}) &= \mathbf{0} \\ \Phi_q \dot{\mathbf{q}} &= \mathbf{0} \\ \Phi_q \ddot{\mathbf{q}} + (\Phi_q \ddot{\mathbf{q}})_q \dot{\mathbf{q}} &\equiv \Phi_q \ddot{\mathbf{q}} + \gamma = \mathbf{0} \end{aligned} \quad (8.1.16)$$

The tangent space parameterization of Section 5.2 remains valid and is not influenced by the dependence of generalized force on Lagrange multipliers. Basic relations of Section 5.2 are summarized here, for use in treating systems with friction. Since the constraint Jacobian  $\Phi_q$  must have full rank, *singular value decomposition* may be used at the initial configuration  $(t^0, \mathbf{q}^0)$  to define an orthogonal basis  $\mathbf{V}$  for its null space,  $\Phi_q(\mathbf{q}^0)\mathbf{V} = \mathbf{0}$  and  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$ . The columns of  $\mathbf{V}$  and those of  $\mathbf{U} \equiv \Phi_q(\mathbf{q}^0)^T$  form a basis for configuration space, so any vector of generalized coordinates can be uniquely written in the tangent space form of Eq. (5.2.9),

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V} \mathbf{v} - \mathbf{U} \mathbf{u} \quad (8.1.17)$$

In order that  $\mathbf{q}$  satisfies the first of Eqs. (8.1.16), and noting that the matrix  $\Phi_q(\mathbf{q}^0)\Phi_q^T(\mathbf{q}^0)$  is nonsingular,  $\mathbf{B}(\mathbf{q}) = (\Phi_q(\mathbf{q}, t)\mathbf{U})^{-1}$  is defined in a neighborhood of  $\mathbf{q}^0$  and  $\mathbf{u}$  is uniquely determined as the solution of

$$\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}) = \mathbf{0} \quad (8.1.18)$$

which exists and is unique. Since  $\Phi_u(\mathbf{q}) = -\Phi_q(\mathbf{q})\mathbf{U} \equiv -\mathbf{B}(\mathbf{q})^{-1}$  is nonsingular, a unique solution exists, as a function of  $\mathbf{v}$ ,

$$\mathbf{u} = \mathbf{h}(\mathbf{v}) \quad (8.1.19)$$

Using this result in Eq. (8.1.17),

$$\mathbf{q} = \mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}) \quad (8.1.20)$$

The matrix  $\mathbf{B}$  and solution  $\mathbf{u} = \mathbf{h}(\mathbf{v})$  are evaluated using Newton-Raphson iteration in Eqs. (5.2.20) and (5.2.17), as follows:

$$\mathbf{B}^{i+1} = 2\mathbf{B}^i - \mathbf{B}^i\Phi_q\mathbf{U}\mathbf{B}, \quad i=1,2,\dots, \text{ until } \|\Phi_q\mathbf{U}\mathbf{B}^{i+1} - \mathbf{I}\| \leq \text{Btol} \quad (8.1.21)$$

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \mathbf{B}(\mathbf{q}, t)\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^i, t), \quad i = 1,2,\dots, \text{ until } \|\Phi(\mathbf{q}^0 + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^{i+1}, t)\| \leq \text{utol} \quad (8.1.22)$$

Linear algebra relations derived in Section 5.2 yield velocities and accelerations of Eqs. (5.2.34) and (5.2.35) that satisfy the second and third of Eqs. (8.1.16),

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\Phi_t \\ \ddot{\mathbf{q}} &= \mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\gamma \end{aligned} \quad (8.1.23)$$

where

$$\mathbf{D} \equiv (\mathbf{I} - \mathbf{U}\mathbf{B}\Phi_q)\mathbf{V} \quad (8.1.24)$$

The results of Eqs. (8.1.20) and (8.1.23) may be substituted into the equations of motion of Eq. (8.1.15), written in residual form as

$$\mathbf{R}(\ddot{\mathbf{v}}, \dot{\mathbf{v}}, \mathbf{v}, \lambda) \equiv \mathbf{M}\mathbf{D}\ddot{\mathbf{v}} + \Phi_q^T\lambda - \mathbf{M}\mathbf{U}\mathbf{B}\gamma - \mathbf{Q}^A = \mathbf{0} \quad (8.1.25)$$

Equation (8.1.25) appears to be an Index 0 DAE, as was Eq. (5.5.11) of Section 5.5. Unfortunately, it is not equivalent to an ODE, because  $\mathbf{Q}^A$  is a function of the Lagrange multiplier, so it is technically not an Index 0 DAE. Due to the similarity of formulation with Section 5.5 and the fact that all three forms of constraint are satisfied, the term Index 0 DAE continues to be used. The reality that the equations of multibody dynamics with friction are not ODE and are not equivalent to an ODE will be reflected in numerical behavior of solution methods.

#### 8.1.2.4 Explicit Integration of the Index 0 DAE

For explicit integration of Eq. (8.1.25), which is linear in  $\dot{\mathbf{v}}$  and nonlinear in  $\lambda$  due the dependence  $\mathbf{Q}^A = \mathbf{Q}^A(\mathbf{v}, \dot{\mathbf{v}}, \lambda, t)$ , the derivative  $\mathbf{Q}_\lambda^A$  is required. The Jacobian of the residual of Eq. (8.1.25) is

$$\mathbf{J}_{\ddot{\mathbf{v}}, \lambda} = \begin{bmatrix} \mathbf{MD} & \Phi_q^T - \mathbf{Q}_\lambda^A \end{bmatrix} \quad (8.1.26)$$

With known values of  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  at a given time step and estimates  $\ddot{\mathbf{v}}^j$  and  $\lambda^j$ , Newton-Raphson iteration is

$$\begin{aligned} \mathbf{J}_{\ddot{\mathbf{v}}, \lambda} \begin{bmatrix} \Delta \ddot{\mathbf{v}}^j \\ \Delta \lambda^j \end{bmatrix} &= -\mathbf{R}(\ddot{\mathbf{v}}^j, \dot{\mathbf{v}}, \mathbf{v}, \lambda^j) \\ \begin{bmatrix} \ddot{\mathbf{v}}^{j+1} \\ \lambda^{j+1} \end{bmatrix} &= \begin{bmatrix} \ddot{\mathbf{v}}^j \\ \lambda^j \end{bmatrix} + \begin{bmatrix} \Delta \ddot{\mathbf{v}}^j \\ \Delta \lambda^j \end{bmatrix} \end{aligned} \quad (8.1.27)$$

Until convergence to  $\ddot{\mathbf{v}}$  and  $\lambda$  is achieved.

An explicit integrator such as Nystrom4 is then used to integrate for  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  at the next time step. The process is continued until the desired final time is reached. This explicit method is implemented in Code 8.1 of Appendix 8.A.

### 8.1.2.5 Implicit integration of the Index 0 DAE

In order to solve Eq. (8.1.25) using an implicit numerical integration formula, the following quantities, defined in Sections 5.2 and 5.3, are required:

$$\begin{aligned} \mathbf{P}2(\mathbf{q}, \chi) &\equiv (\Phi_q(\mathbf{q}, t)\ddot{\chi})_q \\ \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) &\equiv (\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}})_q \\ \mathbf{P}4(\mathbf{q}, \lambda) &\equiv (\Phi_q(\mathbf{q}, t)^T \ddot{\lambda})_q \\ \mathbf{M}2(\mathbf{q}, \beta) &\equiv (\mathbf{M}(\mathbf{q})\ddot{\beta})_q \\ \gamma &= \mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ \gamma_q &= \mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) \\ \gamma_{\dot{\mathbf{q}}} &= 2\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}}) \end{aligned} \quad (8.1.28)$$

where  $\chi$  has the same dimension as  $\mathbf{q}$ . Using *trapezoidal integration formulas* of Eq. (4.8.40),

$$\begin{aligned} \mathbf{v}_n &= \mathbf{v}_{n-1} + h\dot{\mathbf{v}}_{n-1} + (h^2/4)(\dot{\mathbf{v}}_{n-1} + \mathbf{V}^T \dot{\mathbf{q}}_n) \\ \dot{\mathbf{v}}_n &= \dot{\mathbf{v}}_{n-1} + (h/2)(\dot{\mathbf{v}}_{n-1} + \mathbf{V}^T \dot{\mathbf{q}}_n) \end{aligned} \quad (8.1.29)$$

Derivatives of Eq. (8.1.25) with respect to solution variables  $\ddot{\mathbf{v}}$  and  $\lambda$  are

$$\begin{aligned} d\mathbf{R} / d\ddot{\mathbf{v}} &= \mathbf{MD} + (h/2)\mathbf{R}_{\dot{\mathbf{v}}} + (h^2/2)\mathbf{R}_{\mathbf{v}} \\ \mathbf{R}_\lambda &= \Phi_q^T + \mathbf{Q}_\lambda^A \end{aligned} \quad (8.1.30)$$

where partial derivatives of the residual, in terms of the relations of Eq. (8.1.28), are given in Eq. (5.5.18) as

$$\begin{aligned}\mathbf{R}_v &= \left[ \begin{array}{c} \mathbf{M2}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUBP2}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{MUB}\gamma_q + \mathbf{P4}(\mathbf{q}, \lambda) \\ -\mathbf{S}_q - \mathbf{Q}_q^A + (\mathbf{MUB}\gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \mathbf{UB}(\mathbf{P2}(\mathbf{q}, \dot{\mathbf{q}}) + \Phi_{tq}) \end{array} \right] \mathbf{D} \quad (8.1.31) \\ \mathbf{R}_{\dot{v}} &= -(\mathbf{MUB}\gamma_{\dot{q}} + \mathbf{S}_{\dot{q}} + \mathbf{Q}_{\dot{q}}^A) \mathbf{D}\end{aligned}$$

Newton-Raphson iteration to solve Eq. (8.1.25) is carried out at each time step, using

$$\begin{aligned} \begin{bmatrix} d\mathbf{R} / d\ddot{\mathbf{v}} & \mathbf{R}_\lambda \end{bmatrix} \begin{bmatrix} \Delta \dot{\mathbf{v}}_n^i \\ \Delta \lambda_n^i \end{bmatrix} &= -\mathbf{R}(\ddot{\mathbf{v}}_n^i, \lambda_n^i) \\ \begin{bmatrix} \ddot{\mathbf{v}}_n^{i+1} \\ \lambda_n^{i+1} \end{bmatrix} &= \begin{bmatrix} \ddot{\mathbf{v}}_n^i \\ \lambda_n^i \end{bmatrix} + \begin{bmatrix} \Delta \dot{\mathbf{v}}_n^i \\ \Delta \lambda_n^i \end{bmatrix}, \quad i=1,2,\dots \end{aligned} \quad (8.1.32)$$

Since the tangent space parameterization of Eq. (8.1.20) is valid only locally, criteria such as the *condition number* of the coefficient matrix on the left of the first of Eqs. (8.1.32) is monitored to determine when a new parameterization is required. For more detail on the Index 0 DAE algorithm and its implementation, the reader is referred to Section 5.5.

### 8.1.3. Simulation of Three Mass planar Mechanism

To implement the tangent space algorithm of Section 5.5 for the mechanism of Section 8.1.2.1, using the continuously differentiable friction model of Section 8.1.2.2, the applied generalized force of Eq. (8.1.8) and its derivatives are

$$\begin{aligned} \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \lambda) &= \begin{bmatrix} -m_1g - k_1q_1 - q_2\text{csign}(q_2)S(\dot{q}_1) \\ k_2(q_3 - q_2 - 1) - (q_1 - m_2g)\text{csign}(q_1 - m_2g)S(\dot{q}_2) \\ -k_2(q_3 - q_2 - 1) - m_3gS(\dot{q}_3) \end{bmatrix} \\ \mathbf{Q}_q^A &= \begin{bmatrix} -k_1 & -(\text{csign}(q_2) + \lambda^2 q_2 \text{dcsign}(q_2))S(\dot{q}_1) & 0 \\ -aS(\dot{q}_2) & -k_2 & k_2 \\ 0 & k_2 & -k_2 \end{bmatrix} \\ a &= \text{csign}(q_1 - m_2g) + \lambda(\text{q}_1 - m_2g)\text{dcsign}(q_1 - m_2g) \\ \mathbf{Q}_{\dot{q}}^A &= \begin{bmatrix} -q_2\text{csign}(q_2)S'(\dot{q}_1) & 0 & 0 \\ 0 & -(q_1 - m_2g)\text{csign}(q_1 - m_2g)S'(\dot{q}_2) & 0 \\ 0 & 0 & -m_3gS'(\dot{q}_3) \end{bmatrix} \\ \mathbf{Q}^A &= \begin{bmatrix} -(q_2\text{csign}(q_2) + \lambda q_2^2 \text{dcsign}(q_2))S(\dot{q}_1) \\ -(q_1\text{csign}(q_1 - m_2g) + q_1(q_1 - m_2g)\text{dcsign}(q_1 - m_2g))S(\dot{q}_2) \\ 0 \end{bmatrix}\end{aligned}$$

Since  $\mathbf{M}$  of Eq. (8.1.8) does not depend on  $\mathbf{q}$ ,  $\mathbf{M}2(\mathbf{q}, \hat{\beta}) \equiv (\mathbf{M}(\mathbf{q})\hat{\beta})_{\mathbf{q}} = \mathbf{0}$ . The remaining kinematic derivatives of Eqs. (8.1.28) are  $\mathbf{P}2(\mathbf{q}, \chi) = \left( \begin{smallmatrix} & \\ q \ddot{\chi} & \end{smallmatrix} \right)_{\mathbf{q}} = [\chi_1 \quad \chi_2 \quad 0]$ ,  $\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) = \left( \mathbf{P}2(\mathbf{q}, \ddot{\mathbf{q}}) \ddot{\mathbf{q}} \right)_{\mathbf{q}} = \mathbf{0}$ , and  $\mathbf{P}4(\mathbf{q}, \lambda) = \left( \begin{smallmatrix} & \\ q^T & \end{smallmatrix} \right)_{\mathbf{q}} = \text{diag}(\lambda, \lambda, 0)$ .

For numerical simulation, model parameters are selected as  $m_1 = 6.3 \text{ kg}$ ,  $m_2 = m_3 = 2 \text{ kg}$ ,  $g = 9.8 \text{ m/sec}^2$ , step size is  $h = e-4 \text{ sec}$ ,  $v_t = 20h \text{ sec}$ ,  $d = 0.3$ ,  $s = 0.5$ ,  $k_1 = k_2 = 10 \text{ N/m}$ , and  $\ell = 5 \text{ m}$ . Initial conditions are  $\mathbf{q}^0 = [0 \quad 5 \quad 6]^T \text{ m}$  and  $\dot{\mathbf{q}} = [0 \quad 0 \quad -1]^T \text{ m/sec}$ . Numerical solutions are obtained using MATLAB Code 8.1 of Appendix 8.A, with the implicit trapezoidal integration algorithm of Section 5.5.4 and the explicit Nystrom4 integration algorithm of Section 4.8. Plots of the solution in Fig. 8.1.5 show positions  $q_i$  and velocities  $\dot{q}_i \equiv q_i d$  of the three masses vs time. Essentially identical solutions are obtained with both implicit and explicit integrators. This shows that this dynamic simulation model with friction is *not stiff*.

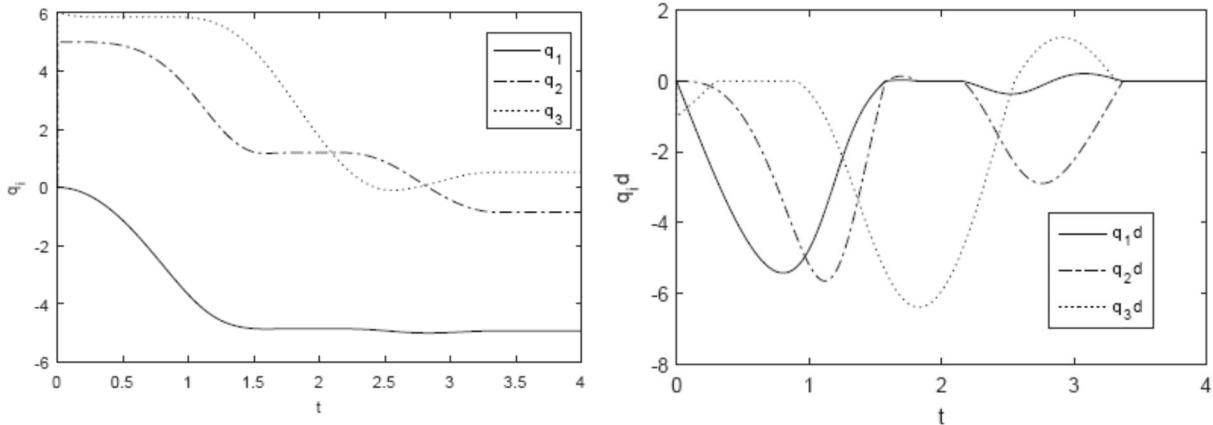


Figure 8.1.5  $q_i$  and  $\dot{q}_i \equiv q_i d$  vs Time,  $m_1 = 6.3 \text{ kg}$

As shown on the right of Fig. 8.1.5, mass three is nearly stationary (*stiction*) from approximately 0.25 to 0.85 sec and after 3.3 sec. At approximately 1.5 sec, the velocities of masses one and two are simultaneously zero, but stiction does not occur. As shown more clearly in the inset of Fig. 8.1.6,  $\dot{q}_2$  is zero for a short period and then varies until approximately 1.8 sec. From 1.8 to 2.2 sec, masses one and two are simultaneously stationary; i.e., stiction. This behavior is consistent with the kinematic velocity constraint  ${}_q \dot{\mathbf{q}} = q_1 \dot{q}_1 + q_2 \dot{q}_2 = 0$ . Provided  $q_1 \neq 0 \neq q_2$ ,  $\dot{q}_1 = 0$  implies  $\dot{q}_2 = 0$  and vice versa. This is the source of the redundant stiction constraint identified by Wojtyra (2016). Throughout periods of predicted stiction, the variation in slip velocity is less than  $e-3 \text{ m/sec}$ . At approximately 3.3 sec, the velocities of all three masses are zero, after which no motion occurs.

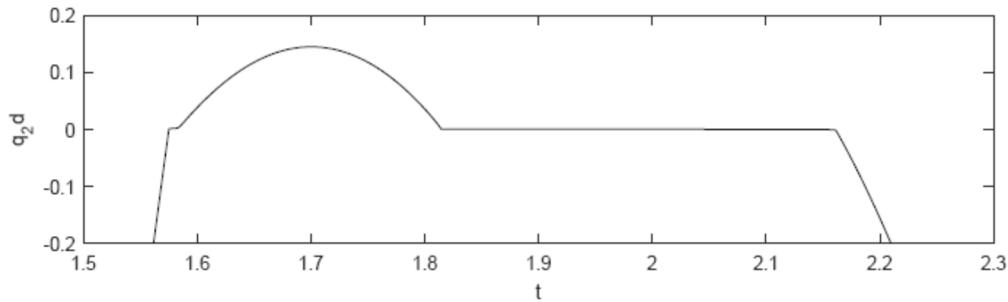


Figure 8.1.6 Stiction Interval for  $m_i = 6.3 \text{ kg}$

To confirm the validity of simulation results, integration step size  $h$  and parameter  $v_t$  are reduced to  $e-6$  sec and  $2e-5$  m/sec, respectively. The dissipation of energy due to slip velocity during the period of approximate stiction of masses one and two is  $1.2e-4$  N.m, a small fraction of the total frictional loss of  $187$  N.m over the period of the simulation. This reflects the limit of  $v_t = 2e-5$  m/sec on slip velocity during periods of approximate stiction. Only 20 *reparameterizations* in  $4e6$  time steps are required ( $2e5$  time steps per reparameterization). Further, the equations of motion are satisfied to within an *integration tolerance* of  $e-6$  and the maximum norms of error in position, velocity, and acceleration constraints over the simulation period are  $2e-13$ ,  $5e-13$ , and  $2e-12$ , respectively. It has been reported by Marques, Flores, Claro, and Lankarani (2016) that the *Index 3 equations of motion* they employ to evaluate a similar model of friction appear to be *stiff*. In contrast, the tangent space Index 0 DAE formulation with explicit Nystrom4 and implicit trapezoidal integration methods succeeded herein, show that the model is *not stiff*.

To investigate the sensitivity of stiction behavior to variations in model parameters, the mass of body 1 is increased to  $m_i = 6.4 \text{ kg}$  and yields the response shown in Fig. 8.1.7. In this case, velocities of masses one and two go simultaneously to zero at approximately 1.5 sec and, in contrast to the behavior shown in Figs. 8.1.5 and 8.1.6, stiction of masses one and two occurs and persists until approximately 2.2 sec.

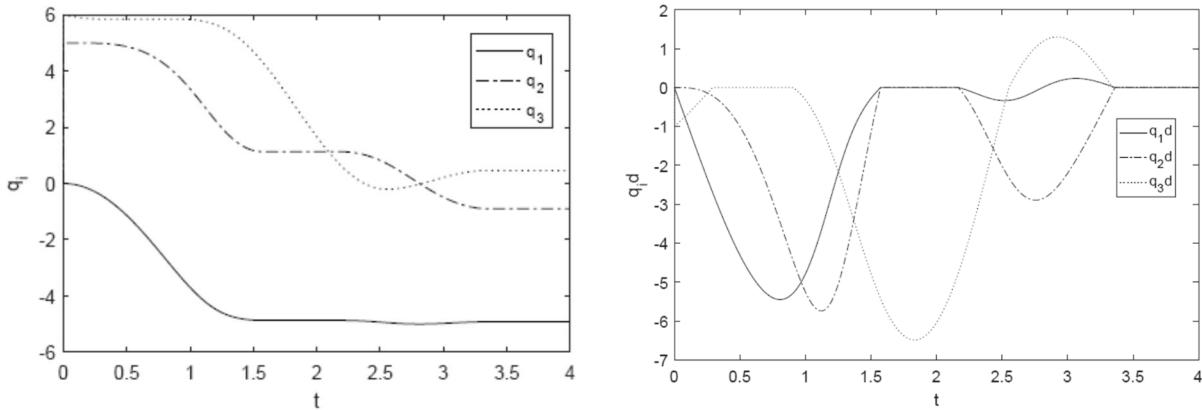


Figure 8.1.7  $q_i$  and  $\dot{q}_i \equiv q_i^d$  vs Time,  $m_i = 6.4 \text{ kg}$

A parametric study confirmed the characteristic delay in onset of stiction shown in Figs. 8.1.5 and 8.1.6. With mass one increased from 6.3 kg to 6.3466602 kg, the magnitude of delay in onset of stiction shown in Figs 8.1.5 and 8.1.6 and intervening slip velocities are essentially invariant. With mass one equal to 6.3466603 kg, a 1.5e-6 % increase, the instant initiation of stiction shown in Fig. 8.1.7 occurs. A similar change in qualitative behavior occurred due to variations in other model parameters.

The Index 0 DAE for dynamic systems with no friction are shown in Chapter 5 to be equivalent to ODE, hence *well posed*; i.e., they have unique solutions that depend continuously on problem data. It has been shown, under more restrictive assumptions (Blumentals, Brogliato, and Bertails-Descoubes, 2016), that systems under the influence of friction forces with nonzero slip velocities; i.e., without stiction, are well behaved. The foregoing example, however, has solutions that depend discontinuously on data. The result is a clear indicator that multibody dynamic response in the presence of stiction may be a discontinuous function of model parameters. Thus, *problems of multibody dynamics with friction are not well posed*. It is interesting to note that this discontinuous behavior was reported (Haug, 2017c) with the transition value of  $m_1$  that was 1.2% higher than reported here. The difference is due to use of the continuously differentiable absolute value function here, whereas the basic absolute value function was used in the reference. Using the explicit *Nystrom4* integrator with the algorithm of Section 5.5.2, which is implemented in Code 8.1 of Appendix 8.1, the transition occurs with a value of  $m_1$  that is within 0.007% of that reported above. This is significant for two reasons. First, the explicit integrator is more sensitive to discontinuous derivatives than the trapezoidal integrator. Second, and more important, if there was stiffness in the model, the explicit integrator would have failed, which it did not. This supports the contention that *models with friction are not necessarily stiff*.

A partial explanation of the discontinuous dependence of dynamic response on model parameters is had by studying the effective coefficients of friction between sliding surfaces and masses one and two in the vicinity of a stiction event. Define the *effective coefficient of friction* as the friction force that is predicted by the *continuous friction model* divided by the associated normal force. Effective coefficients of friction obtained from simulation of the planar three mass model problem are plotted in Fig. 8.1.8. As shown in the upper plot with  $m_1 = 6.4$  kg, when sliding velocities of the bodies are simultaneously zero at approximately 1.58 sec, both coefficients of friction rise from  $\mu_d$  to  $\mu_s$ , but stiction is not achieved. Both masses continue to slide until just after 1.8 sec, when both effective coefficients of friction simultaneously reach the value  $\mu_s$ . At this point, stiction occurs and continues until approximately 2.17 sec. At this time, both effective coefficients of friction again rise to  $\mu_s$  and the stiction bond is broken, leading to sliding with an effective coefficient of friction  $\mu_d$ . In contrast, as shown in the lower plot of Fig. 8.1.8 with  $m_1 = 6.5$  kg, the effective coefficients of friction for both bodies simultaneously reach the value  $\mu_s$  at approximately 1.58 sec, after which stiction is initiated and continues until approximately 2.17 sec. As with the prior case, both effective coefficients of friction simultaneously achieve the value  $\mu_s$  at 2.17 sec, the stiction bond is broken, and the masses slide with nonzero slip velocity. This is consistent with the slip velocity behavior shown in Figs. 8.1.5, 8.1.6, and 8.1.7. This behavior appears to be too systematic to be an accident. It is reminiscent of

Newton's First Law, as noted in Section 4.1.1; i.e., bodies at rest tend to stay at rest (stiction), until external effects lead to motion.

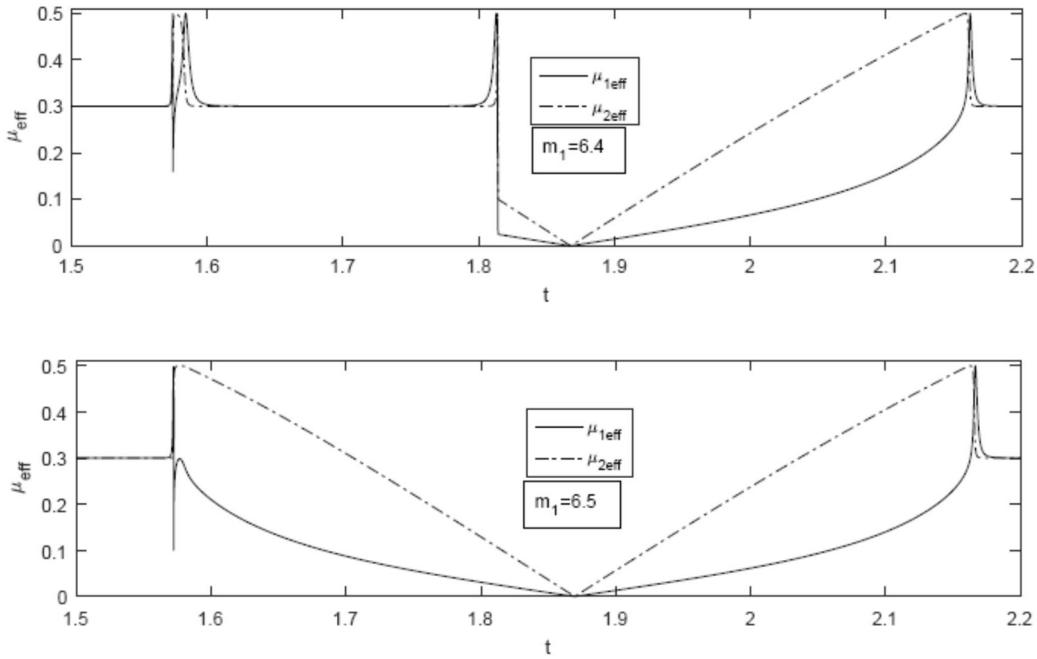


Figure 8.1.8 Effective Coefficients of Friction for Masses One and Two

The tangent space Index 0 DAE formulation, with both implicit trapezoidal and explicit Nystrom4 numerical integration algorithms, yields reliable results in the three mass model problem and shows no sign of stiffness, in contrast to stiffness reported in the literature with Index 3 DAE formulations. These results demonstrate the potential for use of the continuously differentiable friction approximation approach with the tangent space Index 0 DAE formulation for modeling and simulation of friction and stiction in multibody dynamics.

A theoretically interesting point is that the equations of motion with friction, even with the continuous model used herein, are not well posed.

### Key Formulas

$$F_f = -F_n(q, \lambda)S(v, \mu) \quad (8.1.9)$$

$$S(v, \mu) = \mu_d \tanh(4v/v_t) + (\mu_s - \mu_d)(v/v_t)/\left((v/2v_t)^2 + 3/4\right)^2 \quad (8.1.10)$$

$$M(q)\ddot{q} + \Phi_q^T(q)\lambda - Q^A(q, \dot{q}, \lambda) = 0 \quad (8.1.15)$$

$$\Phi(q) = 0$$

$$\Phi_q \dot{q} + \left(\Phi_q \ddot{q}\right)_q \dot{q} \equiv \Phi_q \ddot{q} + \gamma = 0 \quad (8.1.16)$$

$$\Phi_q \ddot{q} + \left(\Phi_q \ddot{q}\right)_q \dot{q} \equiv \Phi_q \ddot{q} + \gamma = 0$$

## 8.2 Analytical Stiction Criteria and Ranges of Stiction Forces

Inequalities that characterize discontinuous Coulomb friction lead to complex logic that defines criteria for the onset of *static friction*, or *stiction*. These inequalities are studied in this section for the three-mass model problem, leading to criteria that, with considerable effort, yield a general solution that defines *admissible ranges of stiction forces*. A numerical solution is presented for the model problem and is shown to agree with the continuous model introduced in Section 8.1. The complexity of the equations encountered for even this simple model problem with discontinuous Coulomb friction suggests that analytical solutions of more realistic applications are unlikely. The reader who is primarily interested in numerical methods for simulation of systems with friction may wish to bypass this section, or to review the results without suffering all the details.

### 8.2.1 Analytical Stiction Criteria for the Three Mass Model Problem

In order to determine whether stiction will occur during a finite time interval, beyond a time  $\bar{t}$  at which  $\dot{q}_1(\bar{t}) = \dot{q}_2(\bar{t}) = 0$ , the conditions  $\ddot{q}_1(t) = \ddot{q}_2(t) = 0$ ,  $q_1(t) = \bar{q}_1$ , and  $q_2(t) = \bar{q}_2$  are imposed during the interval. *Stiction forces*  $sf_1$  and  $sf_2$  are presumed to act in the negative coordinate directions on masses one and two in Fig. 8.1.2, replacing the sliding friction forces on masses one and two in Eq. (8.1.8). The first two equations of motion of Eq. (8.1.8) at  $t = \bar{t}$  thus reduce to

$$\begin{aligned} sf_1 &= -q_1 - m_1 g - k_1 q_1 \\ sf_2 &= -q_2 + k_2 (q_3 - q_2 - 1) \end{aligned} \quad (8.2.1)$$

where the over bar is suppressed for notational convenience. Necessary and sufficient conditions for *static equilibrium*; i.e., stiction, to occur during the subsequent time interval with a static coefficient of friction  $\mu_s$  are that the *Coulomb friction inequalities*  $|sf_i| \leq \mu_s |N_i|$ ,  $i=1,2$ , are satisfied. Using Eqs. (8.1.7) and (8.2.1), this is

$$\begin{aligned} |-q_1 - m_1 g - k_1 q_1| &\leq |\mu_s q_2| \\ |-q_2 + k_2 (q_3 - q_2 - 1)| &\leq |\mu_s (m_2 g - q_1)| \end{aligned} \quad (8.2.2)$$

It is interesting to note that the same criteria may be obtained by imposing nonholonomic constraints that require  $\dot{q}_1(t) = \dot{q}_2(t) = 0$ , with associated Lagrange multipliers playing the role of  $sf_1$  and  $sf_2$  in Eq. (8.2.1) (Wojtyra, 2016).

With  $q_1$ ,  $q_2$ , and  $q_3$  evaluated at  $\bar{t}$ , the inequalities of Eqs. (8.2.2) are functions of  $\lambda$ . If the set of  $\lambda$  that satisfy both inequalities of Eq. (8.2.2) is nonempty, then stiction can occur during a subsequent interval of time. If no value of  $\lambda$  satisfies Eqs. (8.2.2), stiction cannot occur and the equations of motion can be integrated beyond  $\bar{t}$ . If stiction does occur, there are generally many values of the *Lagrange multiplier* that yield the same solution of the constrained equations of motion. This is consistent with the *Coulomb friction law* that is *set valued* during stiction.

The following absolute value inequalities for scalar functions  $a(\lambda)$  and  $b(\lambda)$  are key to defining analytical criteria for stiction:

$$\begin{aligned}
|a(\lambda)| \leq |b(\lambda)| &\Leftrightarrow a(\lambda) \leq |b(\lambda)| \text{ and } -a(\lambda) \leq |b(\lambda)| \\
a(\lambda) \leq |b(\lambda)| &\Leftrightarrow a(\lambda) \leq b(\lambda) \text{ or } b(\lambda) \leq -a(\lambda) \\
-a(\lambda) \leq |b(\lambda)| &\Leftrightarrow -a(\lambda) \leq b(\lambda) \text{ or } b(\lambda) \leq a(\lambda)
\end{aligned} \tag{8.2.3}$$

In terms of sets

$$\begin{aligned}
A(\lambda) &\equiv \{\lambda : |a(\lambda)| \leq |b(\lambda)|\} \\
D(\lambda) &\equiv \{\lambda : a(\lambda) \leq b(\lambda)\} \\
E(\lambda) &\equiv \{\lambda : b(\lambda) \leq -a(\lambda)\} \\
F(\lambda) &\equiv \{\lambda : -a(\lambda) \leq b(\lambda)\} \\
G(\lambda) &\equiv \{\lambda : b(\lambda) \leq a(\lambda)\}
\end{aligned} \tag{8.2.4}$$

the set of solutions of the inequalities of Eq. (8.2.3) is

$$A(\lambda) = (D(\lambda) \cup E(\lambda)) \cap (F(\lambda) \cup G(\lambda)) \tag{8.2.5}$$

where  $\cup$  and  $\cap$  denote *union* and *intersection* of sets, respectively. Let  $A_1(\cdot)$  and  $A_2(\cdot)$  be sets of values of  $\lambda$  that satisfy the first and second inequalities of Eqs. (8.2.2), respectively, at a time  $\bar{t}$  for which  $\dot{q}_1(\bar{t}) = \dot{q}_2(\bar{t}) = 0$ . If  $A_1(\cdot) \cap A_2(\cdot)$  is nonempty, both inequalities of Eqs. (8.2.2) are satisfied for each  $\lambda$  in  $A_1(\cdot) \cap A_2(\cdot)$  and stiction can occur in a subsequent interval of time. Conversely, if  $A_1(\cdot) \cap A_2(\cdot)$  is empty, one or both of the inequalities in Eqs. (8.2.2) is not satisfied for any  $\lambda$ , and stiction cannot occur in a subsequent interval of time.

In the present example, for the first inequality of Eqs. (8.2.2),

$$\begin{aligned}
D_1(\cdot) &= \{\lambda : (\mu_s q_2 - q_1) \leq m_1 g + k_1 q_1\} = \{\lambda : (\alpha) \leq \gamma\} \\
E_1(\cdot) &= \{\lambda : (-\mu_s q_2 - q_1) \leq m_1 g + k_1 q_1\} = \{\lambda : (-\beta) \leq \gamma\} \\
F_1(\cdot) &= \{\lambda : (q_1 + \mu_s q_2) \leq -m_1 g - k_1 q_1\} = \{\lambda : (\beta) \leq -\gamma\} \\
G_1(\cdot) &= \{\lambda : (q_1 - \mu_s q_2) \leq -m_1 g - k_1 q_1\} = \{\lambda : (-\alpha) \leq -\gamma\}
\end{aligned} \tag{8.2.6}$$

where  $\alpha = \mu_s q_2 - q_1$ ,  $\beta = -\mu_s q_2 - q_1$ , and  $\gamma = m_1 g + k_1 q_1$ . For the second inequality,

$$\begin{aligned}
D_2(\cdot) &= \{\lambda : -(q_2 + \mu_s q_1) \leq -k_2(q_3 - q_2 - 1) - \mu_s m_2 g\} = \{\lambda : (\varepsilon) \leq -\tau\} \\
E_2(\cdot) &= \{\lambda : (\mu_s q_1 - q_2) \leq -k_2(q_3 - q_2 - 1) + \mu_s m_2 g\} = \{\lambda : (-\delta) \leq -\eta\} \\
F_2(\cdot) &= \{\lambda : (q_2 - \mu_s q_1) \leq k_2(q_3 - q_2 - 1) - \mu_s m_2 g\} = \{\lambda : (\delta) \leq \eta\} \\
G_2(\cdot) &= \{\lambda : (\mu_s q_1 + q_2) \leq k_2(q_3 - q_2 - 1) + \mu_s m_2 g\} = \{\lambda : (-\varepsilon) \leq \tau\}
\end{aligned} \tag{8.2.7}$$

where  $\delta = \mu_s q_1 + q_2$ ,  $\varepsilon = \mu_s q_1 - q_2$ ,  $\eta = k_2(q_3 - q_2 - 1) + \mu_s m_2 g$ , and  $\tau = k_2(q_3 - q_2 - 1) - \mu_s m_2 g$ .

### 8.2.2 Analytical Solution of Stiction Inequalities

Solutions of the linear inequalities on the right of Eqs. (8.2.6) and (8.2.7) may be tabulated and the sets of solutions  $A_1(\lambda)$  and  $A_2(\lambda)$  of Eq. (8.2.5) and their intersection may be evaluated, to establish criteria for the occurrence of stiction. Solutions based on the algebraic signs of parameters  $\alpha$  and  $\beta$  in Eq. (8.2.6) and  $\delta$  and  $\varepsilon$  in Eqs. (8.2.7) are evaluated in the following.

For Eqs. (8.2.5) and (8.2.6);

$$\text{If } \alpha > 0 \text{ and } \beta < 0, D_1(\lambda) = \{ \lambda : \leq \gamma/\alpha \}, E_1(\lambda) = \{ \lambda : \leq -\gamma/\beta \}, F_1(\lambda) = \{ \lambda : \geq -\gamma/\beta \}, \\ G_1(\lambda) = \{ \lambda : \geq \gamma/\alpha \}, D_1(\lambda) \cup E_1(\lambda) = \{ \lambda : \leq \max(\gamma/\alpha, -\gamma/\beta) \},$$

$$F_1(\lambda) \cup G_1(\lambda) = \{ \lambda : \geq \min(\gamma/\alpha, -\gamma/\beta) \}, \text{ and}$$

$$A_1(\lambda) = \{ \lambda : \min(\gamma/\alpha, -\gamma/\beta) \leq \lambda \leq \max(\gamma/\alpha, -\gamma/\beta) \} \quad (8.2.8)$$

$$\text{If } \alpha > 0 \text{ and } \beta > 0, D_1(\lambda) = \{ \lambda : \leq \gamma/\alpha \}, E_1(\lambda) = \{ \lambda : \geq -\gamma/\beta \}, F_1(\lambda) = \{ \lambda : \leq -\gamma/\beta \},$$

$$G_1(\lambda) = \{ \lambda : \geq \gamma/\alpha \}, D_1(\lambda) \cup E_1(\lambda) = \{ \lambda : \leq \gamma/\alpha \text{ or } \geq -\gamma/\beta \},$$

$$F_1(\lambda) \cup G_1(\lambda) = \{ \lambda : \leq -\gamma/\beta \text{ or } \geq -\gamma/\alpha \}, \text{ and}$$

$$A_1(\lambda) = \{ \lambda : \leq \min(\gamma/\alpha, -\gamma/\beta) \text{ or } \geq \max(\gamma/\alpha, -\gamma/\beta) \} \quad (8.2.9)$$

$$\text{If } \alpha < 0 \text{ and } \beta > 0, D_1(\lambda) = \{ \lambda : \geq \gamma/\alpha \}, E_1(\lambda) = \{ \lambda : \geq -\gamma/\beta \}, F_1(\lambda) = \{ \lambda : \leq -\gamma/\beta \},$$

$$G_1(\lambda) = \{ \lambda : \leq \gamma/\alpha \}, D_1(\lambda) \cup E_1(\lambda) = \{ \lambda : \geq \min(\gamma/\alpha, -\gamma/\beta) \},$$

$$F_1(\lambda) \cup G_1(\lambda) = \{ \lambda : \leq \max(\gamma/\alpha, -\gamma/\beta) \}, \text{ and}$$

$$A_1(\lambda) = \{ \lambda : \min(\gamma/\alpha, -\gamma/\beta) \leq \lambda \leq \max(\gamma/\alpha, -\gamma/\beta) \} \quad (8.2.10)$$

$$\text{If } \alpha < 0 \text{ and } \beta < 0, D_1(\lambda) = \{ \lambda : \geq \gamma/\alpha \}, E_1(\lambda) = \{ \lambda : \leq -\gamma/\beta \}, F_1(\lambda) = \{ \lambda : \geq -\gamma/\beta \},$$

$$G_1(\lambda) = \{ \lambda : \leq \gamma/\alpha \}, D_1(\lambda) \cup E_1(\lambda) = \{ \lambda : \leq -\gamma/\beta \text{ or } \geq \gamma/\alpha \},$$

$$F_1(\lambda) \cup G_1(\lambda) = \{ \lambda : \leq \gamma/\alpha \text{ or } \geq -\gamma/\beta \}, \text{ and}$$

$$A_1(\lambda) = \{ \lambda : \leq \min(\gamma/\alpha, -\gamma/\beta) \text{ or } \geq \max(\gamma/\alpha, -\gamma/\beta) \} \quad (8.2.11)$$

In summary, apart from cases with  $\alpha = 0$  or  $\beta = 0$ ,

$$A_1(\lambda) = \{ \lambda : \min(\gamma/\alpha, -\gamma/\beta) \leq \lambda \leq \max(\gamma/\alpha, -\gamma/\beta) \}, \quad \text{if } \alpha\beta < 0 \quad (8.2.12) \\ A_1(\lambda) = \{ \lambda : \leq \min(\gamma/\alpha, -\gamma/\beta) \text{ or } \geq \max(\gamma/\alpha, -\gamma/\beta) \}, \quad \text{if } \alpha\beta > 0$$

For Eqs. (8.2.5) and (8.2.7);

$$\text{If } \delta > 0 \text{ and } \varepsilon < 0, D_2(\lambda) = \{ \lambda : \geq -\tau/\varepsilon \}, E_2(\lambda) = \{ \lambda : \geq \eta/\delta \}, F_2(\lambda) = \{ \lambda : \leq \eta/\delta \},$$

$$G_2(\lambda) = \{ \lambda : \leq -\tau/\varepsilon \}, D_2(\lambda) \cup E_2(\lambda) = \{ \lambda : \geq \min(\eta/\delta, -\tau/\varepsilon) \},$$

$$F_2(\lambda) \cup G_2(\lambda) = \{ \lambda : \leq \max(\eta/\delta, -\tau/\varepsilon) \}, \text{ and}$$

$$A_2(\cdot) = \left\{ \cdot : \min(\eta/\delta, -\tau/\varepsilon) \leq \cdot \leq \max(\eta/\delta, -\tau/\varepsilon) \right\} \quad (8.2.13)$$

If  $\delta > 0$  and  $\varepsilon > 0$ ,  $D_2(\cdot) = \left\{ \cdot : \cdot \leq -\tau/\varepsilon \right\}$ ,  $E_2(\cdot) = \left\{ \cdot : \cdot \geq \eta/\delta \right\}$ ,  $F_2(\cdot) = \left\{ \cdot : \cdot \leq \eta/\delta \right\}$ ,  $G_2(\cdot) = \left\{ \cdot : \cdot \geq -\tau/\varepsilon \right\}$ ,  $D_2(\cdot) \cup E_2(\cdot) = \left\{ \cdot : \cdot \leq -\tau/\varepsilon \text{ or } \cdot \geq \eta/\delta \right\}$ ,  $F_2(\cdot) \cup G_2(\cdot) = \left\{ \cdot : \cdot \leq \eta/\delta \text{ or } \cdot \geq -\tau/\varepsilon \right\}$ , and

$$A_2(\cdot) = \left\{ \cdot : \cdot \leq \min(\eta/\delta, -\tau/\varepsilon) \text{ or } \cdot \geq \max(\eta/\delta, -\tau/\varepsilon) \right\} \quad (8.2.14)$$

If  $\delta < 0$  and  $\varepsilon > 0$ ,  $D_2(\cdot) = \left\{ \cdot : \cdot \leq -\tau/\varepsilon \right\}$ ,  $E_2(\cdot) = \left\{ \cdot : \cdot \leq \eta/\delta \right\}$ ,  $F_2(\cdot) = \left\{ \cdot : \cdot \geq \eta/\delta \right\}$ ,  $G_2(\cdot) = \left\{ \cdot : \cdot \geq -\tau/\varepsilon \right\}$ ,  $D_2(\cdot) \cup E_2(\cdot) = \left\{ \cdot : \cdot \leq \max(\eta/\delta, -\tau/\varepsilon) \right\}$ ,  $F_2(\cdot) \cup G_2(\cdot) = \left\{ \cdot : \cdot \geq \min(\eta/\delta, -\tau/\varepsilon) \right\}$ , and

$$A_2(\cdot) = \left\{ \cdot : \min(\eta/\delta, -\tau/\varepsilon) \leq \cdot \leq \max(\eta/\delta, -\tau/\varepsilon) \right\} \quad (8.2.15)$$

If  $\delta < 0$  and  $\varepsilon < 0$ ,  $D_2(\cdot) = \left\{ \cdot : \cdot \geq -\tau/\varepsilon \right\}$ ,  $E_2(\cdot) = \left\{ \cdot : \cdot \leq \eta/\delta \right\}$ ,  $F_2(\cdot) = \left\{ \cdot : \cdot \geq \eta/\delta \right\}$ ,  $G_2(\cdot) = \left\{ \cdot : \cdot \leq -\tau/\varepsilon \right\}$ ,  $D_2(\cdot) \cup E_2(\cdot) = \left\{ \cdot : \cdot \leq \eta/\delta \text{ or } \cdot \geq -\tau/\varepsilon \right\}$ ,  $F_2(\cdot) \cup G_2(\cdot) = \left\{ \cdot : \cdot \leq -\tau/\varepsilon \text{ or } \cdot \geq \eta/\delta \right\}$ , and

$$A_2(\cdot) = \left\{ \cdot : \cdot \leq \min(\eta/\delta, -\tau/\varepsilon) \text{ or } \cdot \geq \max(\eta/\delta, -\tau/\varepsilon) \right\} \quad (8.2.16)$$

In summary, apart from cases with  $\delta = 0$  or  $\varepsilon = 0$ ,

$$\begin{aligned} A_2(\cdot) &= \left\{ \cdot : \min(\eta/\delta, -\tau/\varepsilon) \leq \cdot \leq \max(\eta/\delta, -\tau/\varepsilon) \right\}, & \text{if } \delta\varepsilon < 0 \\ A_2(\cdot) &= \left\{ \cdot : \cdot \leq \min(\eta/\delta, -\tau/\varepsilon) \text{ or } \cdot \geq \max(\eta/\delta, -\tau/\varepsilon) \right\}, & \text{if } \delta\varepsilon > 0 \end{aligned} \quad (8.2.17)$$

Apart from zero values of  $\alpha$ ,  $\beta$ ,  $\delta$ , or  $\varepsilon$  that influence only potentially infinite ranges of admissible  $\cdot$ , Eqs. (8.2.12) and (8.2.17) yield four possible cases. Defining  $\min 1 = \min(\gamma/\alpha, -\gamma/\beta)$ ,  $\max 1 = \max(\gamma/\alpha, -\gamma/\beta)$ ,  $\min 2 = \min(\eta/\delta, -\tau/\varepsilon)$ , and  $\max 2 = \max(\eta/\delta, -\tau/\varepsilon)$ , solutions for  $\cdot$  in the four cases shown in Fig. 8.2.1 are as follows:

(1) If  $\alpha\beta < 0$  and  $\delta\varepsilon < 0$  (Fig. 8.2.1(1)),

$$A_1(\cdot) \cap A_2(\cdot) = \begin{cases} \emptyset, & \text{iff } \max(\min 1, \min 2) > \min(\max 1, \max 2) \\ \left\{ \cdot : \max(\min 1, \min 2) \leq \cdot \leq \min(\max 1, \max 2) \right\}, & \text{otherwise} \end{cases} \quad (8.2.18)$$

where  $\emptyset$  denotes the null, or empty, set and iff reads *if and only if*.

(2) If  $\alpha\beta > 0$  and  $\delta\varepsilon < 0$  (Fig. 8.2.1(2)),

$$A_1(\cdot) \cap A_2(\cdot) = \begin{cases} \emptyset, & \text{iff } \min 2 > \min 1 \text{ and } \max 2 < \max 1 \\ \left\{ \cdot : \min 2 \leq \cdot \leq \min 1 \text{ or } \max 1 \leq \cdot \leq \max 2 \right\}, & \text{otherwise} \end{cases} \quad (8.2.19)$$

(3) If  $\alpha\beta < 0$  and  $\delta\varepsilon > 0$  (Fig. 8.2.1(3)),

$$A_1(\ ) \cap A_2(\ ) = \begin{cases} \emptyset, & \text{iff } \min 2 < \min 1 \text{ and } \max 2 > \max 1 \\ \{ : \min 1 \leq \cdot \leq \min 2 \text{ or } \max 2 \leq \cdot \leq \max 1 \}, & \text{otherwise} \end{cases} \quad (8.2.20)$$

(4) If  $\alpha\beta > 0$  and  $\delta\varepsilon > 0$  (Fig. 8.2.1(4)),

$$A_1(\ ) \cap A_2(\ ) = \{ : \cdot \leq \min(\min 1, \min 2) \text{ or } \cdot \geq \max(\max 1, \max 2)\} \neq \emptyset \quad (8.2.21)$$

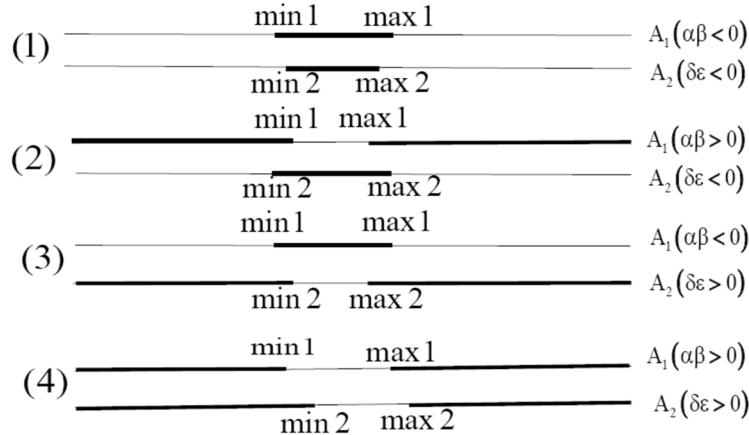


Figure 8.2.1 Admissible Subsets of  $A_1(\ )$  and  $A_2(\ )$  as Dark Lines

### 8.2.3 Admissible Ranges of Stiction Forces

Admissible ranges of *stiction forces* for the three mass example are determined by Eq. (8.2.1) and the ranges of values of Lagrange multipliers defined in Eqs. (8.2.18) to (8.2.21), as follows:

(1) If  $\max(\min 1, \min 2) \leq \min(\max 1, \max 2)$  in Eq. (8.2.18), bounds on stiction forces that are specified in Eqs. (8.2.1) are

$$\begin{cases} -\min(\max 1, \max 2)q_1 - m_1g - k_1q_1 \leq sf_1 \leq -\max(\min 1, \min 2)q_1 - m_1g - k_1q_1, & \text{if } q_1 \geq 0 \\ -\max(\min 1, \min 2)q_1 - m_1g - k_1q_1 \leq sf_1 \leq -\min(\max 1, \max 2)q_1 - m_1g - k_1q_1, & \text{if } q_1 < 0 \end{cases}$$

$$\begin{cases} -\min(\max 1, \max 2)q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\max(\min 1, \min 2)q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 \geq 0 \\ -\max(\min 1, \min 2)q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\min(\max 1, \max 2)q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 < 0 \end{cases} \quad (8.2.22)$$

(2) If  $\min 2 \leq \min 1$  in Eq. (8.2.19), bounds on stiction forces that are specified in Eqs. (8.2.1) are

$$\begin{cases} -\min 1 q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\min 2 q_1 - m_1 g - k_1 q_1, & \text{if } q_1 \geq 0 \\ -\min 2 q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\min 1 q_1 - m_1 g - k_1 q_1, & \text{if } q_1 < 0 \end{cases}$$

$$\begin{cases} -\min 1 q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\min 2 q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 \geq 0 \\ -\min 2 q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\min 1 q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 < 0 \end{cases} \quad (8.2.23)$$

or, if  $\max 2 \geq \max 1$  in Eq. (8.2.19), bounds on stiction forces that are specified in Eqs. (8.2.1) are

$$\begin{cases} -\max 2q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\max 1q_1 - m_1 g - k_1 q_1, & \text{if } q_1 \geq 0 \\ -\max 1q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\max 2q_1 - m_1 g - k_1 q_1, & \text{if } q_1 < 0 \end{cases} \quad (8.2.24)$$

$$\begin{cases} -\max 2q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\max 1q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 \geq 0 \\ -\max 1q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\max 2q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 < 0 \end{cases}$$

(3) If  $\min 2 \geq \min 1$  in Eq. (8.2.20), bounds on stiction forces that are specified in Eqs. (8.2.1) are

$$\begin{cases} -\min 2q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\min 1q_1 - m_1 g - k_1 q_1, & \text{if } q_1 \geq 0 \\ -\min 1q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\min 2q_1 - m_1 g - k_1 q_1, & \text{if } q_1 < 0 \end{cases} \quad (8.2.25)$$

$$\begin{cases} -\min 2q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\min 1q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 \geq 0 \\ -\min 1q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\min 2q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 < 0 \end{cases}$$

or, if  $\max 2 \leq \max 1$  in Eq. (8.2.20), bounds on stiction forces that are specified in Eqs. (8.2.1) are

$$\begin{cases} -\max 1q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\max 2q_1 - m_1 g - k_1 q_1, & \text{if } q_1 \geq 0 \\ -\max 2q_1 - m_1 g - k_1 q_1 \leq sf_1 \leq -\max 1q_1 - m_1 g - k_1 q_1, & \text{if } q_1 < 0 \end{cases} \quad (8.2.26)$$

$$\begin{cases} -\max 1q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\max 2q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 \geq 0 \\ -\max 2q_2 + k_2(q_3 - q_2 - 1) \leq sf_2 \leq -\max 1q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 < 0 \end{cases}$$

(4) Ranges of stiction forces, as a result of Eq. (8.2.21) and Eqs. (8.2.1), are

$$\begin{cases} -\min(\min 1, \min 2)q_1 - m_1 g - k_1 q_1 \leq sf_1, & \text{if } q_1 \geq 0 \\ sf_1 \leq -\min(\min 1, \min 2)q_1 - m_1 g - k_1 q_1, & \text{if } q_1 < 0 \end{cases} \quad (8.2.27)$$

$$\begin{cases} -\min(\min 1, \min 2)q_2 + k_2(q_3 - q_2 - 1) \leq sf_2, & \text{if } q_2 \geq 0 \\ sf_2 \leq -\min(\min 1, \min 2)q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 < 0 \end{cases}$$

or,

$$\begin{cases} sf_1 \leq -\max(\max 1, \max 2)q_1 - m_1 g - k_1 q_1, & \text{if } q_1 \geq 0 \\ -\max(\max 1, \max 2)q_1 - m_1 g - k_1 q_1 \leq sf_1, & \text{if } q_1 < 0 \end{cases} \quad (8.2.28)$$

$$\begin{cases} sf_2 \leq -\max(\max 1, \max 2)q_2 + k_2(q_3 - q_2 - 1), & \text{if } q_2 \geq 0 \\ -\max(\max 1, \max 2)q_2 + k_2(q_3 - q_2 - 1) \leq sf_2, & \text{if } q_2 < 0 \end{cases}$$

Since conditions for occurrence of stiction in Eqs. (8.2.18) through (8.2.21) depend only on problem data and generalized coordinates in solutions of the equations of motion, they are only necessary conditions for stiction of masses one and two to occur. They are not sufficient conditions for onset of stiction, since slip velocities must be zero in order for stiction to follow. Once stiction occurs, any stiction force in the ranges defined in Eqs. (8.2.22) through (8.2.28) is equivalent for stiction of both masses.

To determine the solution for generalized coordinates in the three mass model problem of Section 8.1.2, a constraint may be imposed on the position of either mass one or mass two during a stiction period that involves these two masses, for solution of the equations of motion. The resulting generalized coordinates must be identical for either choice of a single stiction constraint, since enforcing a position constraint on either mass imposes a constraint on the position of the other mass, due to the holonomic constraint of Eq. (8.2.5). Lagrange multipliers determined in the alternative solutions will be different and will generally not yield stiction forces on the individual masses that satisfy Eqs. (8.2.2). However, stiction forces determined by Eqs. (8.2.22) through (8.2.28), using results from either solution, will satisfy Eqs. (8.2.2).

Treatment of a stiction event that involves mass three is much simpler than for masses one and two, since it is not subject to a kinematic constraint. A *stiction constraint*  $q_3(t) - \bar{q}_3(\bar{t}) = 0$  is imposed for  $t > \bar{t}$  if  $\dot{q}_3(\bar{t}) = 0$  and the associated Lagrange multiplier  $\lambda_3(t)$  is the stiction force that must satisfy the condition  $|\lambda_3(t)| \leq s m_3 g$  for continuation of stiction, after which the stiction constraint is deleted.

#### 8.2.4 Numerical Solution of Coulomb Stiction Criteria

Equations of motion for the example of Section 8.1.2, with the same data, are implemented with stiction constraints on first mass one and then mass two. Simulations are carried out using MATLAB Code 8.2 in Appendix 8.A, with *implicit trapezoidal integration* and the tangent space Index 0 DAE formulation with  $m_1 = 5$  and  $7$  kg,  $h = e-4$ , other data as in Section 8.1.2, and a tolerance  $tol = e-3$  on the conditions  $\max(|\dot{q}_1|, |\dot{q}_2|) \leq tol$  and  $|\dot{q}_3| \leq tol$  for onset of stiction for masses one and two and for mass three, respectively. Results with a *holonomic stiction constraint* on mass one or two yield identical solutions for generalized coordinate velocities shown in Fig. 8.2.2.

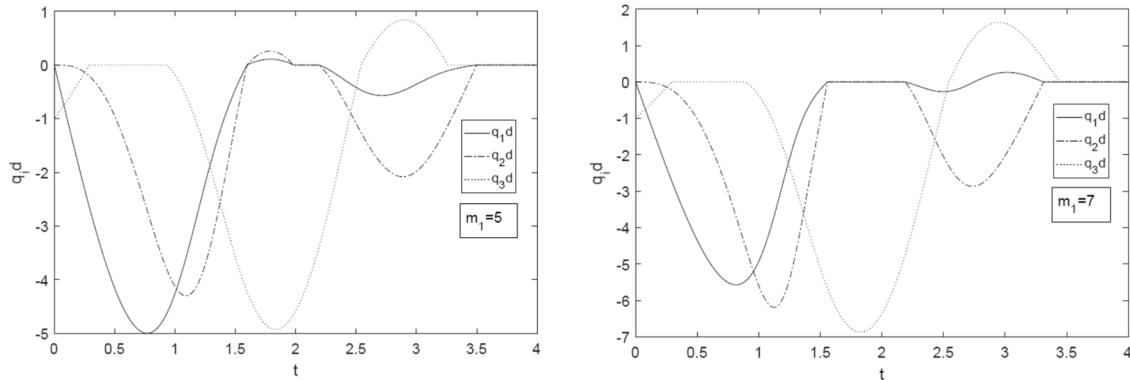


Figure 8.2.2  $\dot{q}_i$  ( $q_i d$ ) vs. Time for  $m_1 = 5$  and  $7$

Generalized coordinate histories are similar to those presented in Figs. 8.1.5 and 8.1.7. Only 20 *reparameterizations* were required in 40,000 time steps (2000 time steps per reparameterization), and there was *no sign of stiffness* in the tangent space equations of motion. To compare solutions based on the *continuous friction model* presented in Section 8.1.2.2 with the *constraint addition-deletion* analysis of this section, norms of differences in solutions for  $m_1 = 8.5$  are evaluated. The mean norms of differences in positions, velocities, and accelerations over 40,000 time steps are 0.0042, 0.0054, and 0.159, respectively.

The large range of values of  $m_1$  for the results of Fig 8.2.2 was selected to enable clear graphical explanation of the theoretical basis for the difference in stiction response modes. Based on the criteria of Eqs. (8.2.18) through (8.2.21),  $\text{stict} = 0$  if stiction cannot occur in the present state of the mechanism and  $\text{stict} = 1$  if stiction can occur; i.e., the necessary conditions are satisfied. If the necessary conditions for stiction are satisfied and the slip velocities are zero at a given time, stiction occurs and mode = 2 holds until the necessary conditions fail to be satisfied, at which time the value of stict is set to 0 and mode is set to 1. Values of these functions are shown in Fig. 8.2.3.

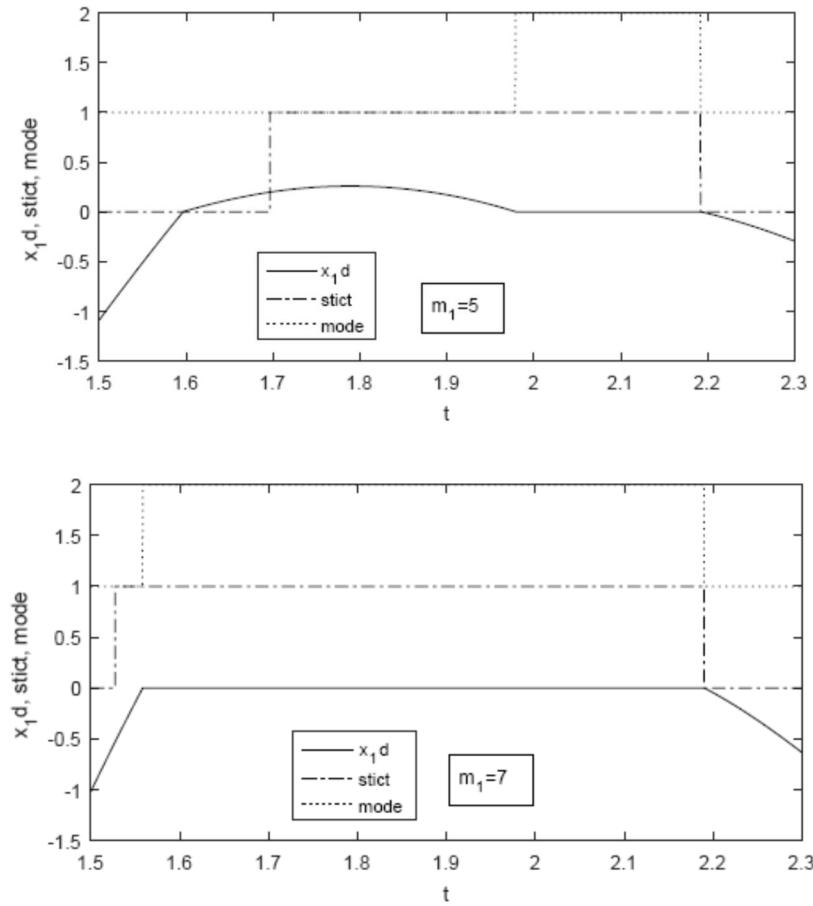


Figure 8.2.3 Triggers for Onset and Termination of Stiction Event

As shown in the top plot of Fig. 8.2.3, for  $m_1 = 5$ , the necessary conditions for stiction are not satisfied and  $\text{stict} = 0$  at 1.58 sec when the slip velocities go to zero. Thus, the state of the system precludes the onset of stiction, and nonzero *slip velocities* persist until just before 2 sec, when they go to zero. At this time, the necessary conditions are satisfied and  $\text{stict} = 1$ . Therefore, stiction is initiated and continues until approximately 2.2 sec, at which time the necessary conditions for stiction fail to be satisfied and sliding resumes. In contrast to stiction behavior for  $m_1 = 5$ , with  $m_1 = 7$ , the lower plot of Fig. 8.2.3 shows that the necessary conditions for stiction to occur are satisfied and  $\text{stict} = 1$  when the slip velocities first go to zero. Stiction is thus initiated and continues until 2.2 sec, when it is terminated, as in the preceding case.

Maximum and minimum limits on stiction forces that act on masses one and two, from Eqs. (8.2.22) through (8.2.28) that satisfy Eq. (8.2.2), are obtained using generalized coordinates from individual mass stiction simulations and are shown in Fig. 8.2.4. These plots show the set-valued nature of *Coulomb friction force*, with any stiction force values between the maximum and minimum bounds satisfying Eq. (8.2.2) and yielding the same solution of the equations of motion for generalized coordinates. Note that, at the beginning and end of stiction periods, both inequalities of Eqs. (8.2.2) are equalities, as shown in simulations in Section 8.1.2.3.

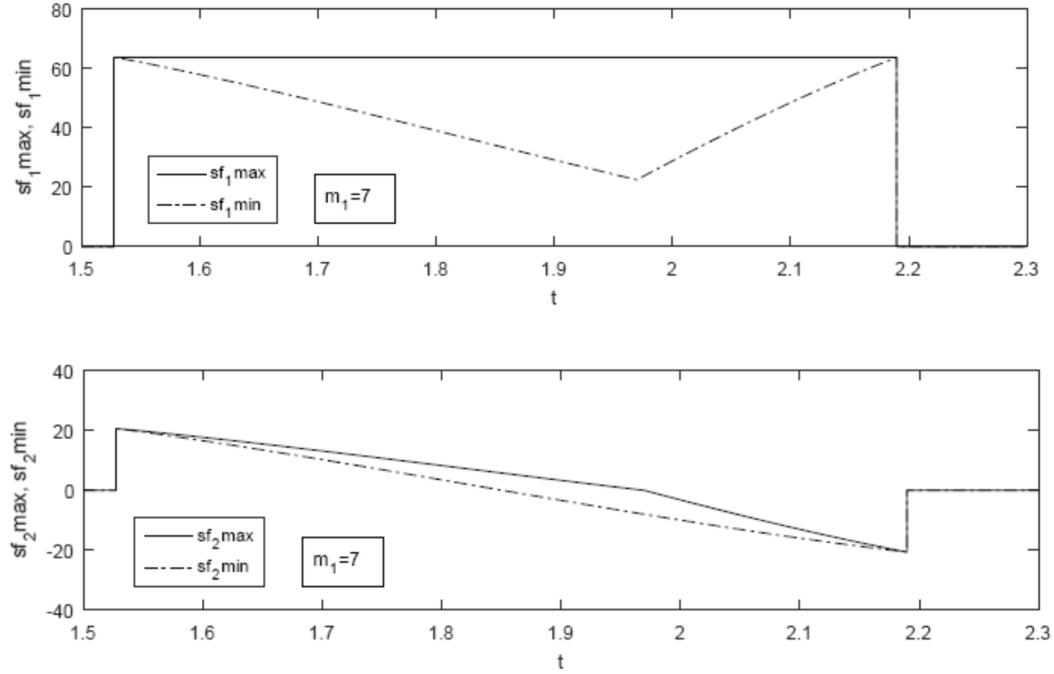


Figure 8.2.4 Coulomb Stiction Force Range on Masses 1 and 2

It is not clear how to extend the foregoing analytical criteria to general multibody systems. However, analysis presented for the three mass planar model problem suggests that simulation using the tangent space Index 0 DAE formulation with the continuous friction model and the trapezoidal integration method presented in Section 8.1 provide meaningful results for analysis of more realistic multibody systems with friction, including stiction events.

Analysis of the three-mass model problem of Sections 8.1 and 8.2 shows that friction and stiction models using a continuous friction model and an analytical Coulomb friction model that accounts for stiction effects yield comparable results.

### 8.3 Cartesian Coordinate Simulation of Model Problems

As a first step in extension of the analysis presented in Section 8.1 to general multibody systems, which involves detailed models of constraint geometry, constraint reaction forces, and friction forces, Cartesian coordinate models of the three-mass system of Sections 8.1 and 8.2 and a four mass spatial system with a higher degree of stiction redundancy are studied.

#### 8.3.1 Cartesian Coordinate Simulation of a Three Mass System

For the planar three mass model problem of Fig. 8.1.2, vectors  $\mathbf{r}_1 = [x_1 \ y_1]^T$  and  $\mathbf{r}_2 = [x_2 \ y_2]^T$  that locate masses one and two and the  $x_3$  coordinate of mass three are used as Cartesian coordinates; i.e.,  $\mathbf{q} = [x_1 \ y_1 \ x_2 \ y_2 \ x_3]^T \in \mathbb{R}^5$ . Constraints that act on masses one and two and the associated Jacobian are expressed in terms of these coordinates as

$$\Phi(\mathbf{q}) = \begin{bmatrix} q_1 \\ q_4 \\ ((q_3 - q_1)^2 + (q_4 - q_2)^2 - \ell^2) / 2 \end{bmatrix} = \mathbf{0} \quad (8.3.1)$$

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ q_1 - q_3 & q_2 - q_4 & q_3 - q_1 & q_4 - q_2 & 0 \end{bmatrix}$$

Constraint forces between masses one and two and the axes along which they move are Lagrange multipliers  $\lambda_1$  and  $\lambda_2$  associated with the first two constraints of Eq. (8.3.1), so system generalized forces are

$$\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) = \begin{bmatrix} 0 \\ -m_1 g - k_1 q_2 - \lambda_1 \text{csign}(\lambda_1) S(\dot{q}_2) \\ k_2(q_5 - q_3 - 1) - \lambda_2 \text{csign}(\lambda_2) S(\dot{q}_3) \\ -m_2 g \\ -k_2(q_5 - q_3 - 1) - m_3 g S(\dot{q}_5) \end{bmatrix} \quad (8.3.2)$$

With  $\mathbf{M} = \text{diag}(m_1 \ m_1 \ m_2 \ m_2 \ m_3)$ , the equations of motion are

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q})\boldsymbol{\lambda} = \mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) \quad (8.3.3)$$

While the same stiction behavior observed with the reduced dimension model of Sections 8.1 and 8.2 should occur with this model, the analytical approach used in Section 8.2 is not applicable. The Cartesian coordinate model has three Lagrange multipliers associated with holonomic constraints, whereas the reduced dimension model used in Section 8.2 has only one. This special case allowed stiction condition inequalities to be solved as functions of a single variable. The goal here is to apply the *continuous friction model* to the higher dimensional Cartesian coordinate formulation for numerical simulation and compare results with previous simulations.

Quantities that are required to implement the tangent space *Index 0 DAE* algorithm are  $\mathbf{M}2(\mathbf{q}, \mathbf{x}) = \mathbf{0}$ ,  $\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ , and

$$\begin{aligned}\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) &= (\Phi_{\mathbf{q}}(\mathbf{q})\ddot{\boldsymbol{\chi}})_{\mathbf{q}} = \begin{bmatrix} & & \mathbf{0}_{2 \times 5} \\ \chi_1 - \chi_3 & \chi_2 - \chi_4 & \chi_3 - \chi_1 & \chi_4 - \chi_2 & 0 \end{bmatrix} \\ \mathbf{P}4(\mathbf{q}, \boldsymbol{\lambda}) &= (\Phi_{\mathbf{q}}(\mathbf{q})^T \ddot{\boldsymbol{\lambda}})_{\mathbf{q}} = \begin{bmatrix} \lambda_3 & 0 & -\lambda_3 & 0 & 0 \\ 0 & \lambda_3 & 0 & -\lambda_3 & 0 \\ -\lambda_3 & 0 & \lambda_3 & 0 & 0 \\ 0 & -\lambda_3 & 0 & \lambda_3 & 0 \\ & & & \mathbf{0}_{1 \times 5} & \end{bmatrix} \\ \mathbf{Q}_{\mathbf{q}}^A &= \begin{bmatrix} & \mathbf{0}_{1 \times 5} \\ 0 & -k_1 & 0 & 0 & 0 \\ 0 & 0 & -k_2 & 0 & k_2 \\ & \mathbf{0}_{1 \times 5} \\ 0 & 0 & k_2 & 0 & -k_2 \end{bmatrix} \\ \mathbf{Q}_{\dot{\mathbf{q}}}^A &= - \begin{bmatrix} & \mathbf{0}_{1 \times 5} \\ 0 & \lambda_1 \text{csign}(\lambda_1) S'(\dot{q}_2) & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 \text{csign}(\lambda_2) S'(\dot{q}_3) & 0 & 0 \\ & \mathbf{0}_{1 \times 5} \\ 0 & 0 & 0 & 0 & m_3 g S'(\dot{q}_5) \end{bmatrix} \\ \mathbf{Q}_{\boldsymbol{\lambda}}^A &= - \begin{bmatrix} & \mathbf{0}_{1 \times 3} & & \\ (\text{csign}(\lambda_1) + \lambda_1 \text{dcsign}(\lambda_1)) S(\dot{q}_2) & 0 & & 0 \\ 0 & (\text{csign}(\lambda_2) + \lambda_2 \text{dcsign}(\lambda_2)) S(\dot{q}_3) & 0 & \\ & \mathbf{0}_{2 \times 3} & & \end{bmatrix}\end{aligned}$$

Simulations carried out with this model in MATLAB Code 8.3.1 of Appendix 8.A, with  $m_2 = m_3 = 2 \text{ kg}$ ,  $\ell = 5 \text{ m}$ ,  $\mu_d = 0.3$ ,  $\mu_s = 0.5$ ,  $g = 9.8 \text{ m/s}$ ,  $\text{Maxvnorm} = 0.7$ ,  $\text{utol} = \text{Btol} = \text{e-8}$ ,  $\text{intol} = \text{e-6}$ , and varying values of  $m_1$  give results that are qualitatively like those of Section 8.1. Numerical results using *implicit trapezoidal* and *explicit Nystrom4* integrators with this formulation and the formulation presented in Section 8.1 are essentially identical. Even the sensitivity of the discontinuous response event to variations in  $m_1$  differs by only 0.01% with the two integrators. Since the most practical indicator of *stiffness* in a model is failure of an explicit integrator to carry out simulations, this system is definitely *not stiff*. Even with the increased dimension of the *Index 0 DAE* of motion, only twenty *reparameterizations* were required in 4e4 time steps (2,000 time steps per reparameterization) for both implicit and explicit integrators. This reinforces the effectiveness of the continuous friction model and tangent space integrators in approximating the solution.

### 8.3.2. Cartesian Coordinate Simulation of a Four Mass System

As a second example of friction and stiction simulation using a Cartesian coordinate model, consider the *four-mass spatial system* shown in Fig. 8.3.1. As in the three mass model problem of Section 8.3.1, *stiction yields redundancy* in this model. To see this, note that if the velocity of mass one is zero, so will be the velocity of mass two, due to the distance constraint between them. In turn, the velocity of mass 3 will be zero, due to the distance constraint between it and mass two. This yields a *third order stiction redundancy*. The analytical approach used in Section 8.2 would be defeated for two reasons; (1) there are eight Lagrange multipliers, rather than one, and (2) three absolute value inequalities must be enforced.

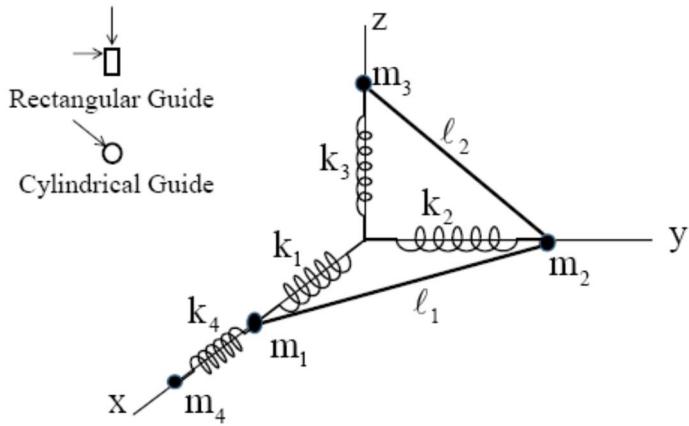


Figure 8.3.1 Four Mass Spatial Model Problem

To illustrate the effect of *constraint geometry* on system dynamics, two *contact geometries* between masses and coordinate axis guides are modeled. The rectangular guide along each axis involves two orthogonal reaction forces, as shown at the upper left in Fig. 8.3.1, defined by *Lagrange multipliers*  $\lambda_1^k$  and  $\lambda_2^k$  for axis  $k$ . The cylindrical guide shown at the lower left of Fig 8.3.1, in contrast, involves line contact between the masses and cylinders along which they move. The resulting contact force, shown at the lower left in Fig. 8.3.1, is the vector sum of the components acting on cylindrical guide  $k$ , so its magnitude is  $\sqrt{\lambda_i^{k^2} + \lambda_j^{k^2}}$ .

Generalized coordinates for this model are 3-vectors  $\mathbf{r}_i$ ,  $i = 1, 2, 3$ , that locate masses one through three and the  $x$  coordinate of mass 4,  $X_4$ . This is a set of 10 generalized coordinates, with  $\mathbf{r}_1 = [q_1 \quad q_2 \quad q_3]^T$ ,  $\mathbf{r}_2 = [q_4 \quad q_5 \quad q_6]^T$ ,  $\mathbf{r}_3 = [q_7 \quad q_8 \quad q_9]^T$ , and  $X_4 = q_{10}$ . With the three-dimensional identity matrix  $\mathbf{I}_3$ , the system mass matrix is  $\mathbf{M} = \text{diag}(\mathbf{m}_1\mathbf{I}_3 \quad \mathbf{m}_2\mathbf{I}_3 \quad \mathbf{m}_3\mathbf{I}_3 \quad \mathbf{m}_4)$ . Holonomic constraints are

$$\Phi(\mathbf{q}) = \begin{bmatrix} q_2 \\ q_3 \\ q_4 \\ q_6 \\ q_7 \\ q_8 \\ ((q_6 - q_3)^2 + (q_5 - q_2)^2 + (q_4 - q_1)^2 - \ell_1^2)/2 \\ ((q_9 - q_6)^2 + (q_8 - q_5)^2 + (q_7 - q_4)^2 - \ell_2^2)/2 \end{bmatrix} = \mathbf{0} \quad (8.3.4)$$

Using the rectangular guide model, the Lagrange multipliers corresponding to the first six constraints of Eq. (8.3.4) are *constraint reaction forces* acting on faces of the rectangular guides shown in Fig. 8.3.1 and generalized forces acting on the system are

$$\mathbf{Q}^A(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) = \begin{bmatrix} k_4(q_{10} - q_1 - 1) - k_1 q_1 - (\lambda_1 \text{csign}(\lambda_1) + \lambda_2 \text{csign}(\lambda_2)) S(\dot{q}_1) \\ 0 \\ -m_1 g \\ 0 \\ -k_2 q_5 - (\lambda_3 \text{csign}(\lambda_3) + \lambda_4 \text{csign}(\lambda_4)) S(\dot{q}_5) \\ -m_2 g \\ 0 \\ 0 \\ -m_3 g - k_3 q_9 - (\lambda_5 \text{csign}(\lambda_5) + \lambda_6 \text{csign}(\lambda_6)) S(\dot{q}_9) \\ -k_4(q_{10} - q_1 - 1) - m_4 g S(\dot{q}_{10}) \end{bmatrix} \quad (8.3.5)$$

The remaining task in implementing the tangent space *Index 0 DAE friction formulation* with the trapezoidal integrator is to calculate derivatives of constraint and generalized force related terms. Since the resulting matrices are even more sparse than their counterparts in Section 8.3.1, only nonzero terms are presented. Nonzero entries  $a_{ij}$  in the ij positions in the constraint Jacobian  $\Phi_q(\mathbf{q})$  are  $a_{12}=1, a_{23}=1, a_{34}=1, a_{46}=1, a_{57}=1, a_{68}=1, a_{71}=q_1-q_4, a_{72}=q_2-q_5,$

$a_{73}=q_3-q_6, a_{74}=q_4-q_1, a_{75}=q_5-q_2, a_{76}=q_6-q_3, a_{84}=q_4-q_7, a_{85}=q_5-q_8,$   
 $a_{86}=q_6-q_9, a_{87}=q_7-q_4, a_{88}=q_8-q_5$ , and  $a_{89}=q_9-q_6$ . Nonzero entries  $b_{ij}$  in the ij positions of  $\mathbf{P}2(\mathbf{q}, \boldsymbol{\chi}) \equiv (\Phi_q(\mathbf{q})\boldsymbol{\chi})_q$  are  $b_{71}=\chi_1-\chi_4, b_{72}=\chi_2-\chi_5, b_{73}=\chi_3-\chi_6,$   
 $b_{74}=\chi_4-\chi_1, b_{75}=\chi_5-\chi_2, b_{76}=\chi_6-\chi_3, b_{84}=\chi_4-\chi_7, b_{85}=\chi_5-\chi_8,$   
 $b_{86}=\chi_6-\chi_9, b_{87}=\chi_7-\chi_4, b_{88}=\chi_8-\chi_5$ , and  $b_{89}=\chi_9-\chi_6$ . Nonzero entries  $c_{ij}$  in the ij positions of  $(\Phi_q^T(\mathbf{q})\boldsymbol{\lambda})_q$  are  $c_{11}=\lambda_7, c_{14}=-\lambda_7, c_{22}=\lambda_7, c_{25}=-\lambda_7, c_{33}=\lambda_7,$

$c_{36} = -\lambda_7$ ,  $c_{41} = -\lambda_7$ ,  $c_{44} = \lambda_7 + \lambda_8$ ,  $c_{47} = -\lambda_8$ ,  $c_{52} = -\lambda_7$ ,  $c_{55} = \lambda_7 + \lambda_8$ ,  $c_{58} = -\lambda_8$ ,  
 $c_{63} = -\lambda_7$ ,  $c_{66} = \lambda_7 + \lambda_8$ ,  $c_{69} = -\lambda_8$ ,  $c_{74} = -\lambda_8$ ,  $c_{77} = \lambda_8$ ,  $c_{85} = -\lambda_8$ ,  $c_{88} = \lambda_8$ ,  $c_{96} = -\lambda_8$ , and  
 $c_{99} = \lambda_8$ . Nonzero entries  $d_{ij}$  in the  $ij$  positions of  $\mathbf{Q}_q^A(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda})$  are  
 $d_{11} = -k_4 - k_1$ ,  $d_{1,10} = k_4$ ,  $d_{55} = -k_2$ ,  $d_{99} = -k_3$ ,  $d_{10,1} = k_4$ , and  $d_{10,10} = -k_4$ . Nonzero entries  $e_{ij}$   
in the  $ij$  positions of  $\mathbf{Q}_q^A(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda})$  are  $e_{11} = -(\lambda_1 \text{csign}(\lambda_1) + \lambda_2 \text{csign}(\lambda_2)) S'(\dot{q}_1)$ ,  
 $e_{55} = -(\lambda_3 \text{csign}(\lambda_3) + \lambda_4 \text{csign}(\lambda_4)) S'(\dot{q}_5)$ ,  $e_{99} = -(\lambda_5 \text{csign}(\lambda_5) + \lambda_6 \text{csign}(\lambda_6)) S'(\dot{q}_9)$ , and  
 $e_{10,10} = -m_4 g S'(\dot{q}_{10})$ . Finally, nonzero entries  $f_{ij}$  in the  $ij$  positions of  $\mathbf{Q}_\lambda^A(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda})$  are  
 $f_{11} = -(\text{csign}(\lambda_1) + \lambda_1 \text{dcsign}(\lambda_1)) S(\dot{q}_1)$ ,  $f_{12} = -(\text{csign}(\lambda_2) + \lambda_2 \text{dcsign}(\lambda_2)) S(\dot{q}_1)$ ,  
 $f_{53} = -(\text{csign}(\lambda_3) + \lambda_3 \text{dcsign}(\lambda_3)) S(\dot{q}_5)$ ,  $f_{54} = -(\text{csign}(\lambda_4) + \lambda_4 \text{dcsign}(\lambda_4)) S(\dot{q}_5)$ ,  
 $f_{95} = -(\text{csign}(\lambda_5) + \lambda_5 \text{dcsign}(\lambda_5)) S(\dot{q}_9)$ , and  $f_{96} = -(\text{csign}(\lambda_6) + \lambda_6 \text{dcsign}(\lambda_6)) S(\dot{q}_9)$ .

Problem data are  $\ell_1 = 5 \text{ m}$ ,  $\ell_2 = 7 \text{ m}$ ,  $m_1 = m_2 = 2 \text{ kg}$ ,  $m_3 = m_4 = 6 \text{ kg}$ ,  $g = 9.8 \text{ m/sec}^2$ ,  
 $k_1 = k_2 = k_3 = 2 \text{ N/m}$ ,  $k_4 = 10 \text{ N/m}$ ,  $\mu_d = 0.13$ ,  $\mu_s = 0.2$ , and  $v_t = 20 \text{ h}$ . Initial conditions are  
 $\mathbf{q}^0 = [4 \ 0 \ 0 \ 0 \ 3 \ 0 \ 0 \ 0 \ 6.32 \ 5]^T$  and  $\dot{\mathbf{q}}^0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1]^T$ . Integration tolerances in MATLAB Code 8.3.2 of Appendix 8.A that is used for simulation are  $\text{utol} = \text{Btol} = \text{e-8}$ ,  $\text{intol} = \text{e-6}$ , and  $\text{Maxv} = 0.7$ . With a step size  $h = \text{e-4}$ , simulation is carried out over a 4 sec period. Results are plotted in Fig. 8.3.2. Stiction is initiated simultaneously upon onset of zero values of slip velocities of bodies one, two, and three at approximately 1.3 sec and continues over a finite period of time. Fifteen *reparameterizations* were required in 4e4 time steps (2,666 time steps per reparameterization). To assess accuracy of simulation results, the maximum position, velocity, and acceleration constraint error norms over the simulated time interval are 7e-9, 2e-9, and e-8, respectively. Essentially identical results are obtained with explicit Nystrom4 and implicit trapezoidal integration algorithms of Sections 8.1.2.4 and 8.1.2.5. This confirms that the system is *not stiff*.

With a slight reduction of the dynamic coefficient of friction to  $\mu_d = 0.1$ , position plots very similar to the top of Fig. 8.3.2 are obtained, but the characteristic delay in *onset of the stiction event* shown in Fig. 8.3.3 is obtained.

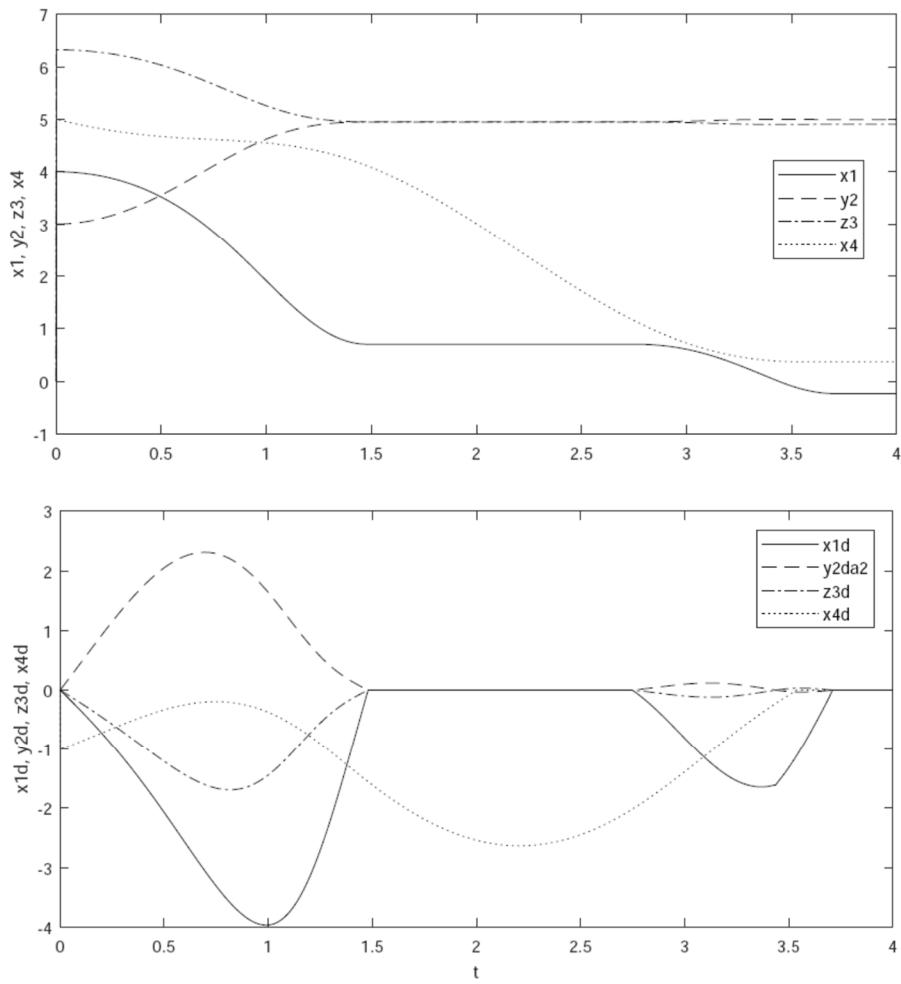


Figure 8.3.2 Positions and Velocities vs. Time, Translational Guide,  $\mu_d = 0.13$ ,  $\mu_s = 0.18$

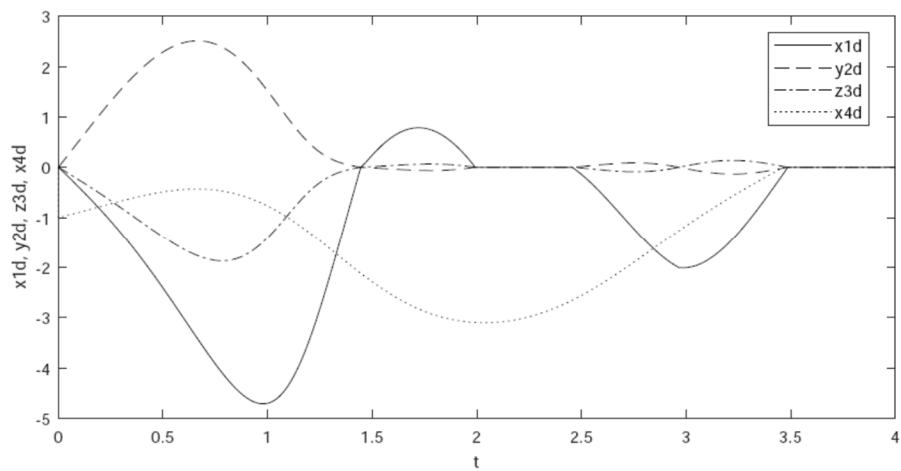


Figure 8.3.3 Velocities vs. Time, Translational Guide,  $\mu_d = 0.10$ ,  $\mu_s = 0.18$

To simulate friction with the cylindrical guide shown at the lower left of Fig. 8.3.1, the terms  $(\lambda_i \text{csign}(\lambda_i) + \lambda_j \text{csign}(\lambda_j))$  in generalized applied force of Eq. (8.3.5) are replaced by  $\sqrt{\lambda_i^2 + \lambda_j^2}$ . Further, the derivative with respect to  $\lambda_i$  in the rectangular form,  $(\text{csign}(\lambda_i) + \lambda_i d\text{csign}(\lambda_i))$ , is replaced by the corresponding derivative in the cylindrical form,  $\lambda_i / \sqrt{\lambda_i^2 + \lambda_j^2}$ , in foregoing expressions above. Since, in general,  $(|\lambda_i^k| + |\lambda_j^k|) \geq \sqrt{\lambda_i^{k2} + \lambda_j^{k2}}$ , friction force with the rectangular guide is greater than that with the cylindrical guide. Results for velocities from simulations with the cylindrical guide and the same data as above for the rectangular guide are shown on the left of Fig. 8.3.4. Note the *delay in onset of stiction*, similar to that occurring in the planar three mass model and the above results with the rectangular guide. Increasing friction coefficients to  $\mu_d = 0.16$  and  $\mu_s = 0.2$ , the response shown on the right of Fig. 8.3.4 is realized.

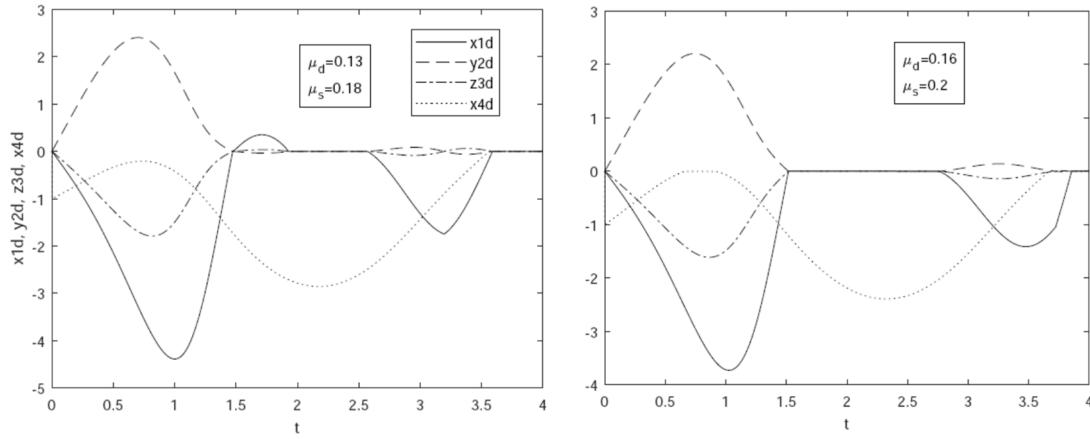


Figure 8.3.4 Velocities vs. Time, Cylindrical Guide

Cartesian coordinate formulations of the three mass planar model problem and a four mass spatial model problem demonstrate the ability of the continuous friction model to account for friction and stiction effects, even for stiction constraint redundancies of degree higher than two. Success in simulations with the explicit Nystrom4 integrator confirms that the systems are not stiff.

## 8.4 Modeling Planar Multibody Systems with Friction

Constraint reaction forces and associated friction forces in planar revolute and translational joints are derived and incorporated in generalized forces for use in the Index 0 DAE form of planar system equations of motion. Derivatives of generalized forces that are required for explicit and implicit numerical integration are derived. The approach is an extension of that for particles in Section 8.3, with moderate complexity in the planar multibody dynamics formulation.

### 8.4.1 Constraint Reaction Forces and Friction Generalized Forces

*Constraint reaction force* and *constraint reaction torque* on body  $i$  due to joint  $k$  shown in Fig. 8.4.1, represented in the  $x'_i$ - $y'_i$  reference frame, are given by Eq. (4.10.25) as

$$\begin{bmatrix} \mathbf{F}'^k_i \\ T_i^k \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_i^T \Phi_{r_i}^{kT} \lambda^k \\ -(\mathbf{s}'^{P_kT} \mathbf{A}_i^T \mathbf{P} \Phi_{r_i}^{kT} + \Phi_{\phi_i}^{kT}) \lambda^k \end{bmatrix} \quad (8.4.1)$$

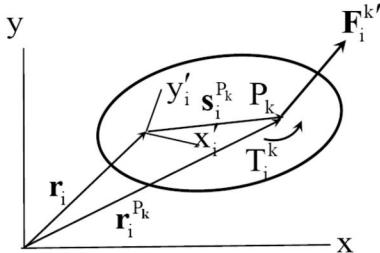


Figure 8.4.1 Constraint Reaction Force and Torque on Planar Body  $i$

#### 8.4.1.1 Revolute Joint Reaction and Friction Forces

For the *revolute joint* of Fig. 3.2.2, constraint Jacobians of Eq. (3.2.9) are

$$\begin{aligned} \Phi_{r_i}^{\text{revk}} &= -\mathbf{I} \\ \Phi_{\phi_i}^{\text{revk}} &= -\mathbf{P} \mathbf{A}_i \mathbf{s}'^{P_k} \end{aligned} \quad (8.4.2)$$

So, Eq. (8.4.1) yields

$$\begin{aligned} \mathbf{F}'^k_i &= \mathbf{A}_i^T \lambda^k \\ T_i^k &= \mathbf{s}'^{P_kT} \mathbf{A}_i^T \mathbf{P} \lambda^k - \mathbf{s}'^{P_kT} \mathbf{A}_i^T \mathbf{P} \lambda^k = 0 \end{aligned} \quad (8.4.3)$$

In the global frame,

$$\mathbf{F}_i^k = \lambda^k \quad (8.4.4)$$

which acts at revolute joint definition point  $P_i^k$  on body  $i$ , as shown in Fig. 8.4.1.

The angular velocity of body  $j$  relative to body  $i$  about revolute joint definition point  $P_i^k$ , shown in Fig. 8.4.2, is  $\omega_{ij} = \dot{\phi}_j - \dot{\phi}_i$ . With joint radius  $R_k$ , the velocity of the contact point on

body j relative to body i, shown in Fig. 8.4.2, is  $v_{ij} = R_k(\dot{\phi}_j - \dot{\phi}_i)$ . The *friction force* on body j at the contact point is given by Eq. (8.1.9) as

$$F_{fj}^k = \|F_i^k\| S(v_{ij}, \mu) \quad (8.4.5)$$

and the friction force on body i is  $F_{fi}^k = -F_{fj}^k$ .

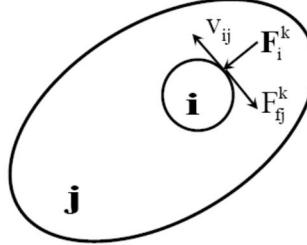


Figure 8.4.2 Constraint and Friction Forces and Relative Velocity in Revolute Joint

From Eq. (8.4.4),  $\|F_i^k\| = \|\lambda^k\|$  and Eq. (8.4.5) is

$$F_{fj}^k = -\|\lambda^k\| S(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu) \quad (8.4.6)$$

The force due to friction on body i is  $F_{fi}^k = -F_{fj}^k = \|\lambda^k\| S(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu)$ , so the counterclockwise torque acting on body i is

$$T_{fi}^k = R_k \|\lambda^k\| S(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu) \quad (8.4.7)$$

and the reaction torque on body j is  $T_{fj}^k = -R_k \|\lambda^k\| S(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu)$ . *Generalized revolute joint friction forces* are thus

$$\begin{aligned} \mathbf{Q}_i^{Arevf_k} &= \begin{bmatrix} \mathbf{0} \\ R_k \|\lambda^k\| S(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu) \end{bmatrix} \\ \mathbf{Q}_j^{Arevf_k} &= \begin{bmatrix} \mathbf{0} \\ -R_k \|\lambda^k\| S(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu) \end{bmatrix} \end{aligned} \quad (8.4.8)$$

### 8.4.1.2 Translational Joint Reaction and Friction Forces

For the *translational joint* of Fig. 3.2.3, constraint Jacobians of Eq. (3.2.15) are

$$\begin{aligned} \Phi_{r_i}^{trans} &= \begin{bmatrix} \mathbf{v}'^T \mathbf{P} \mathbf{A}_i^T \\ \mathbf{0} \end{bmatrix} \\ \Phi_{\phi_i}^{trans} &= \begin{bmatrix} -\mathbf{v}'^T \mathbf{s}'^{P_k} - \mathbf{v}'^T \mathbf{A}_i^T \mathbf{d}_{ij} \\ -\mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} = \begin{bmatrix} -\mathbf{v}'^T \mathbf{A}_i^T (\mathbf{r}_j + \mathbf{A}_j \mathbf{s}'^{P_k} - \mathbf{r}_i) \\ -\mathbf{v}'^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j \end{bmatrix} \end{aligned} \quad (8.4.9)$$

The *constraint force* and *constraint torque* of Eq. (8.4.1) are thus

$$\begin{aligned}
\mathbf{F}'^k_i &= -\mathbf{A}_i^T [-\mathbf{A}_i \mathbf{P} \mathbf{v}'_i \quad \mathbf{0}] \lambda^k = \lambda^k \mathbf{P} \mathbf{v}'_i \\
\mathbf{T}_i^k &= -\mathbf{s}'^{P_k T} \mathbf{A}_i^T \mathbf{P} [-\mathbf{A}_i \mathbf{P} \mathbf{v}'_i \quad \mathbf{0}] \lambda^k + \left[ (\mathbf{r}_j^T + \mathbf{s}'^{P_k T} \mathbf{A}_j^T - \mathbf{r}_i^T) \mathbf{A}_i \mathbf{v}'_i - \mathbf{v}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j \right] \lambda^k \quad (8.4.10) \\
&= -\mathbf{s}'^{P_k T} \mathbf{v}'_i \lambda^k + \left[ (\mathbf{r}_j^T + \mathbf{s}'^{P_k T} \mathbf{A}_j^T - \mathbf{r}_i^T) \mathbf{A}_i \mathbf{v}'_i - \mathbf{v}'_i^T \mathbf{A}_i^T \mathbf{A}_j \mathbf{v}'_j \right] \lambda^k
\end{aligned}$$

where Eqs. (2.3.32) and (2.3.33) have been used. Clearly,  $\mathbf{F}'^k_i$  is perpendicular to the axis of relative translation, acting at translational joint definition point  $P_i^k$ .

For the geometry of the translational joint, with a key in body i shown in Fig. 8.4.3 that slides in a keyway in body j, where  $\|\mathbf{v}'_i\| = d_i$  and using Eq. (8.4.10), static equivalence of force and torque systems requires that

$$\begin{aligned}
f_1^k + f_2^k + \mathbf{F}'^{k T} ((1/d_i) \mathbf{P} \mathbf{v}'_i) &= f_1^k + f_2^k + (\lambda^k / d_i) \mathbf{v}'_i^T \mathbf{P}^T \mathbf{P} \mathbf{v}'_i \\
&= f_1^k + f_2^k + d_i \lambda^k = 0 \quad (8.4.11) \\
d_i f_2^k + T_i^k &= 0
\end{aligned}$$

Thus,

$$\begin{aligned}
f_1^k &= (1/d_i) T_i^k - d_i \lambda^k \\
f_2^k &= -(1/d_i) T_i^k \quad (8.4.12)
\end{aligned}$$

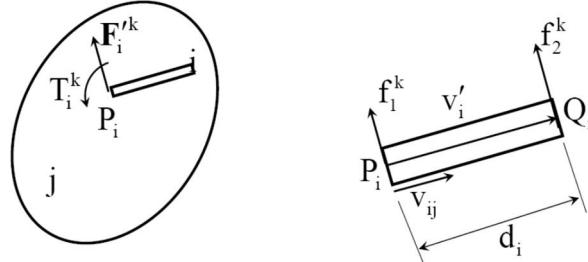


Figure 8.4.3 Constraint Forces and Relative Velocity in Translational Joint

The velocity of points on body j in the translational joint axis, relative to body i, is

$$\begin{aligned}
v_{ij} &= (1/d_i) \mathbf{v}'_i^T \mathbf{d}_{ij} = (1/d_i) \mathbf{v}'_i^T \mathbf{A}_i^T (\dot{\mathbf{r}}_j + \dot{\phi}_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j^{P_k} - \dot{\mathbf{r}}_i - \dot{\phi}_i \mathbf{P} \mathbf{A}_i \mathbf{s}'_i^{P_k}) \\
&= (1/d_i) \mathbf{v}'_i^T \mathbf{A}_i^T (\dot{\mathbf{r}}_j + \dot{\phi}_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j^{P_k} - \dot{\mathbf{r}}_i) - \dot{\phi}_i (1/d_i) \mathbf{v}'_i^T \mathbf{P} \mathbf{s}'_i^{P_k} \quad (8.4.13)
\end{aligned}$$

and the velocity of points on body i relative to body j is  $-v_{ij}$ . The *friction force* that acts on body i, along the axis of relative translation, is thus

$$F_{f_i}^k = -(|f_1^k| + |f_2^k|) S(v_{ij}, \mu) \approx -\left(f_1^k \text{csign}(f_1^k) + f_2^k \text{csign}(f_2^k)\right) S(v_{ij}, \mu) \quad (8.4.14)$$

The virtual displacement of points on body j relative to body i on the axis of translation is

$$\begin{aligned}
\delta s_{ij} &= (1/d_i) \mathbf{v}'_i^T \delta \mathbf{d}_{ij} = (1/d_i) \mathbf{v}'_i^T \mathbf{A}_i^T (\delta \mathbf{r}_j + \delta \phi_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j^{P_k} - \delta \mathbf{r}_i - \delta \phi_i \mathbf{P} \mathbf{A}_i \mathbf{s}'_i^{P_k}) \\
&= (1/d_i) \mathbf{v}'_i^T \mathbf{A}_i^T (\delta \mathbf{r}_j + \delta \phi_j \mathbf{P} \mathbf{A}_j \mathbf{s}'_j^{P_k} - \delta \mathbf{r}_i) - (1/d_i) \mathbf{v}'_i^T \mathbf{P} \mathbf{s}'_i^{P_k} \delta \phi_i \quad (8.4.15)
\end{aligned}$$

and the virtual displacement of points on body i relative to body j is  $-\delta s_{ij}$ . The virtual work done by the friction force on body i is thus

$$\begin{aligned}\delta W^k &= -F_{f_i}^k \delta s_{ij} \\ &= -(1/d_i) F_{f_i}^k \left\{ (\delta r_j^T - \delta r_i^T) A_i v'_i + (\delta \phi_j v_i'^T A_i^T P A_j s_j'^{P_k} - \delta \phi_i v_i'^T P s_i'^{P_k}) \right\}\end{aligned}\quad (8.4.16)$$

where Eq. (2.3.33) has been used. *Generalized translational joint friction forces* are thus

$$\begin{aligned}Q_i^{Atranf_k} &= (1/d_i) F_{f_i}^k \begin{bmatrix} A_i v'_i \\ v_i'^T P s_i'^{P_k} \end{bmatrix} \\ Q_j^{Atranf_k} &= (1/d_i) F_{f_i}^k \begin{bmatrix} -A_i v'_i \\ -v_i'^T A_i^T P A_j s_j'^{P_k} \end{bmatrix}\end{aligned}\quad (8.4.17)$$

#### 8.4.2 Derivatives of Friction Generalized Forces

For the revolute joint, differentiating Eq. (8.4.8) with respect to  $\dot{q}_{ij}$  and  $\lambda^k$ ,

$$Q_{i\dot{q}_{ij}}^{Arevf_k} = Q_{j\dot{q}_{ij}}^{Arevf_k} = \mathbf{0} \text{ and}$$

$$Q_{i\lambda^k}^{Arevf_k} = -Q_{j\lambda^k}^{Arevf_k} = \begin{bmatrix} \mathbf{0} \\ (R_k / \|\lambda^k\|) S(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu) \lambda^{kT} \end{bmatrix}\quad (8.4.18)$$

where  $\|\lambda^k\| = (\lambda^{kT} \lambda^k)^{1/2}$  and  $\|\lambda^k\|_{\lambda^k} = (1/\|\lambda^k\|) \lambda^{kT}$  have been used. Derivatives with respect to  $\dot{q}_{ij}$  are

$$Q_{i\dot{q}_{ij}}^{Arevf_k} = -Q_{j\dot{q}_{ij}}^{Arevf_k} = R_k^2 \|\lambda^k\| \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -S'(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu) & \mathbf{0} & S'(R_k(\dot{\phi}_j - \dot{\phi}_i), \mu) \end{bmatrix}\quad (8.4.19)$$

For the translational joint, the absolute values  $|f_\ell^k|$  and  $|f_1^k|$  of  $f_\ell^k$  and  $f_1^k$  defined in Eq. (8.4.12) are approximated using Eq. (8.1.13) as

$$|f_\ell^k| = f_\ell^k \operatorname{sign}(f_\ell^k) \approx f_\ell^k c \operatorname{sign}(f_\ell^k)\quad (8.4.20)$$

$\ell = 1, 2$ . Derivatives of these expressions, using Eqs. (8.1.14) and (8.4.12), are

$$\begin{aligned}|f_\ell^k|_{\dot{q}_{ij}} &= (\operatorname{csign}(f_\ell^k) + f_\ell^k d \operatorname{csign}(f_\ell^k)) f_{\ell\dot{q}_{ij}}^k \\ |f_\ell^k|_{\lambda^k} &= (\operatorname{csign}(f_\ell^k) + f_\ell^k d \operatorname{csign}(f_\ell^k)) f_{\ell\lambda^k}^k\end{aligned}\quad (8.4.21)$$

and  $|f_\ell|_{\dot{q}_{ij}} = \mathbf{0}$ ,  $\ell = 1, 2$ . From Eq. (8.4.12),

$$\begin{aligned}f_{1\dot{q}_{ij}}^k &= -f_{2\dot{q}_{ij}}^k = (1/d_i) T_{i\dot{q}_{ij}}^k \\ f_{1\lambda^k}^k &= (1/d_i) T_{i\lambda^k}^k - d_i [1 \quad 0] \\ f_{2\lambda^k}^k &= -(1/d_i) T_{i\lambda^k}^k\end{aligned}\quad (8.4.22)$$

Finally, from the second of Eqs. (8.4.10),

$$\begin{aligned} T_{iq_{ij}}^k &= \begin{bmatrix} -v_i'^T A_i^T & a_1 & v_i'^T A_i^T & a_2 \end{bmatrix} \\ T_{ik^k}^k &= -s_i'^{P_k T} v_i' [1 \ 0] + \left[ (r_j^T + s_j'^{P_k T} A_j^T - r_i^T) A_i v_i' \quad v_i'^T A_i^T A_j v_j' \right] \end{aligned} \quad (8.4.23)$$

where  $a_1 = \left[ (r_j^T + s_j'^{P_k T} A_j^T - r_i^T) P A_i v_i' \quad -v_i'^T A_i^T P A_j v_j' \right] \lambda^k$ ,  $a_2 = \left[ -s_j'^{P_k T} A_j^T P A_i v_i' \quad v_i'^T A_i^T P A_j v_j' \right] \lambda^k$ , Eqs. (2.3.31) have been used, and from Eq. (8.4.13),

$$\begin{aligned} v_{ijq_{ij}} &= \begin{bmatrix} \mathbf{0} & a_3 & \mathbf{0} & a_4 \end{bmatrix} \\ v_{ij\dot{q}_{ij}} &= (1/d_i) \begin{bmatrix} -v_i'^T A_i^T & -v_i'^T P s_i'^{P_k} & v_i'^T A_i^T & v_i'^T A_i^T P A_j s_j'^{P_k} \end{bmatrix} \\ v_{ij\lambda^k} &= \mathbf{0} \end{aligned} \quad (8.4.24)$$

where  $a_3 = -(1/d_i) v_i'^T A_i^T P (r_j + \dot{\phi}_j P A_j s_j'^{P_k} - r_i)$  and  $a_4 = -\dot{\phi}_j (1/d_i) v_i'^T A_i^T A_j s_j'^{P_k}$ .

Derivatives of translational joint generalized friction forces of Eq. (8.4.17) are

$$\begin{aligned} Q_{iq_{ij}}^{Atranf_k} &= (1/d_i) S(v_{ij}, \mu) \begin{bmatrix} A_i v_i' \\ v_i'^T P s_i'^{P_k} \end{bmatrix} \left( |f_1^k|_{q_{ij}} + |f_2^k|_{q_{ij}} \right) \\ &\quad + (1/d_i) \left( |f_1^k| + |f_2^k| \right) \begin{bmatrix} A_i v_i' \\ v_i'^T P s_i'^{P_k} \end{bmatrix} S'(v_{ij}, \mu) v_{ijq_{ij}} \\ &\quad + (1/d_i) \left( |f_1^k| + |f_2^k| \right) S(v_{ij}, \mu) \begin{bmatrix} \mathbf{0} & P A_i v_i' & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 0 \end{bmatrix} \\ Q_{jq_{ij}}^{Atranf_k} &= (1/d_i) S(v_{ij}, \mu) \begin{bmatrix} -A_i v_i' \\ -v_i'^T A_i^T P A_j s_j'^{P_k} \end{bmatrix} \left( |f_1^k|_{q_{ij}} + |f_2^k|_{q_{ij}} \right) \\ &\quad + (1/d_i) \left( |f_1^k| + |f_2^k| \right) \begin{bmatrix} -A_i v_i' \\ -v_i'^T A_i^T P A_j s_j'^{P_k} \end{bmatrix} S'(v_{ij}, \mu) v_{ijq_{ij}} \\ &\quad + (1/d_i) \left( |f_1^k| + |f_2^k| \right) S(v_{ij}, \mu) \begin{bmatrix} \mathbf{0} & -P A_i v_i' & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -v_i'^T A_i^T A_j s_j'^{P_k} & \mathbf{0} & v_i'^T A_i^T A_j s_j'^{P_k} \end{bmatrix} \\ Q_{iq_{ij}}^{Atranf_k} &= (1/d_i) \left( |f_1^k| + |f_2^k| \right) \begin{bmatrix} A_i v_i' \\ v_i'^T P s_i'^{P_k} \end{bmatrix} S'(v_{ij}, \mu) v_{ij\dot{q}_{ij}} \\ Q_{jq_{ij}}^{Atranf_k} &= (1/d_i) \left( |f_1^k| + |f_2^k| \right) \begin{bmatrix} -A_i v_i' \\ -v_i'^T A_i^T P A_j s_j'^{P_k} \end{bmatrix} S'(v_{ij}, \mu) v_{ij\dot{q}_{ij}} \\ Q_{ik^k}^{Atranf_k} &= (1/d_i) S(v_{ij}, \mu) \begin{bmatrix} A_i v_i' \\ v_i'^T P s_i'^{P_k} \end{bmatrix} \left( |f_1^k|_{\lambda^k} + |f_2^k|_{\lambda^k} \right) \\ Q_{jk^k}^{Atranf_k} &= (1/d_i) S(v_{ij}, \mu) \begin{bmatrix} -A_i v_i' \\ -v_i'^T A_i^T P A_j s_j'^{P_k} \end{bmatrix} \left( |f_1^k|_{\lambda^k} + |f_2^k|_{\lambda^k} \right) \end{aligned} \quad (8.4.25)$$

which may be evaluated using Eqs. (8.4.21), (8.4.22), (8.4.23), and (8.4.24).

Constraint reaction forces that act on planar bodies connected by kinematic constraints depend on the geometry of the joints and Lagrange multipliers that appear in the Index 0 equations of motion. Expressions for constraint contact forces are used with continuous friction force models presented in Section 8.1 to obtain expressions for generalized friction forces that are incorporated in Index 0 DAE of motion. Finally, derivative expressions that are required to implement explicit and implicit numerical integration methods for multibody simulation of planar systems with friction are presented.

## 8.5 Modeling Spatial Multibody Systems with Friction

To extend results presented for model problems with particles and planar systems, the library of spatial joints presented in Section 3.3 is used to determine *constraint reaction forces* and associated *friction forces* for spatial multibody dynamic simulation. Derivatives that are required in the Index 0 DAE formulation and for numerical integration of the equations of motion are presented. The similar geometry of cylindrical, revolute, and translational joints defined in Section 3.3 is exploited for formulation and coding efficiencies, but requires care in evaluating expressions that arise in each of the three constraints.

### 8.5.1 Constraint Contact Reaction Forces

*Constraint reaction force* and *constraint reaction torque* on body i due to joint k shown in Fig. 8.5.1, represented in the  $x'_i - y'_i - z'_i$  frame, are given by Eq. (4.10.17) as

$$\begin{bmatrix} \mathbf{F}'^k_i \\ \mathbf{T}'^k_i \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_{r_i}^T \Phi_{r_i}^{k T} \lambda^k \\ -(1/2)\mathbf{G}(\mathbf{p}_i) \Phi_{p_i}^{k T} - \tilde{\mathbf{s}}_i^{k T} \mathbf{A}_{r_i}^T \Phi_{r_i}^{k T} \end{bmatrix} \quad (8.5.1)$$

It is important to note that, whereas *Lagrange multipliers* associated with *Euler parameter normalization conditions* appear in the Index 0 DAE of Eq. (5.5.12), they do not appear in the constraint reaction force and torque of Eq. (8.5.1). While Eq. (8.5.1) holds for all joints, it is valid only if constraint Jacobian terms  $\Phi_{r_i}^k$  and  $\Phi_{p_i}^k$  and associated Lagrange multipliers  $\lambda^k$  are properly defined for each joint.

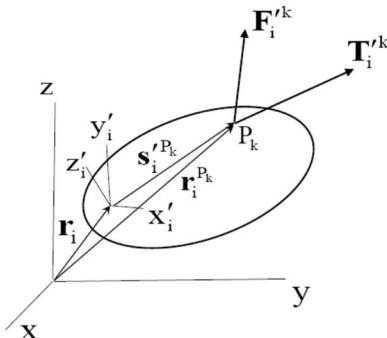


Figure 8.5.1 Constraint Reaction Force and Torque on Spatial Body i

*Constraint surface contacts* and associated physical *forces of constraint* depend on the *geometry of joints*. For the *cylindrical joint* of Section 3.3.3, shown in Fig. 8.5.2, rotational and translational bearings are located in body i at endpoints  $P_i^k$  and  $Q_i^k$  that are in contact with the cylindrical bar in body j. Constraint contact force components in cylindrical joint k that act at ends  $P_i^k$  and  $Q_i^k$  in body i are  $f_x'^{lk}$ ,  $f_y'^{lk}$ ,  $f_x'^{2k}$ , and  $f_y'^{2k}$ , as shown in Fig. 8.5.2. Contact force  $f_z'^{lk}$  is the contribution from a revolute joint thrust bearing and does not act in the cylindrical joint.

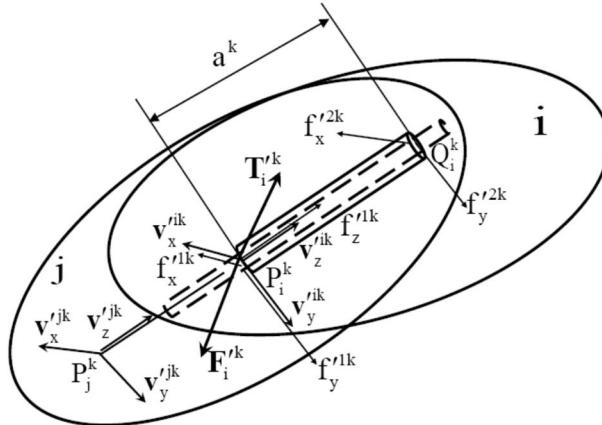


Figure 8.5.2 Constraint Forces for Cylindrical and Revolute Joints

Cylindrical joint constraint reaction force  $\mathbf{F}'^k$  and torque  $\mathbf{T}'^k$  shown in Fig. 8.5.2, that act on body  $i$  at point  $P_i^k$  and are represented in the body  $x'_i - y'_i - z'_i$  reference frame, are determined by Eq. (8.5.1), using submatrices of the Jacobian of the cylindrical constraint of Eq. (3.3.24). These subJacobians are obtained for the cylindrical joint, using Jacobians of *building block constraints* in Section 3.3.2. Software implementation is included in Code 5.8 of Appendix 5.A.

Force  $\mathbf{F}'^k$  and torque  $\mathbf{T}'^k$  must be converted to statically equivalent physical contact force components  $f'_x^{\text{lk}}$ ,  $f'_y^{\text{lk}}$ ,  $f'_x^{2k}$ , and  $f'_y^{2k}$  shown in Fig. 8.5.2. Summing forces on body  $i$  in the  $\mathbf{v}_x^{\text{rik}}$  and  $\mathbf{v}_y^{\text{rik}}$  directions,

$$\begin{aligned}\mathbf{v}_x^{\text{rik}^T} \mathbf{F}'^k + f'_x^{\text{lk}} + f'_x^{2k} &= 0 \\ \mathbf{v}_y^{\text{rik}^T} \mathbf{F}'^k + f'_y^{\text{lk}} + f'_y^{2k} &= 0\end{aligned}\quad (8.5.2)$$

Summing torques about  $\mathbf{v}_x^{\text{rik}}$  and  $\mathbf{v}_y^{\text{rik}}$ ,

$$\begin{aligned}\mathbf{v}_x^{\text{rik}^T} \mathbf{T}'^k - a^k f'_y^{2k} &= 0 \\ \mathbf{v}_y^{\text{rik}^T} \mathbf{T}'^k + a^k f'_x^{2k} &= 0\end{aligned}\quad (8.5.3)$$

The solution of Eqs. (8.5.2) and (8.5.3) is contact forces that are statically equivalent to  $\mathbf{F}'^k$  and  $\mathbf{T}'^k$ ,

$$\begin{aligned}f'_x^{\text{lk}} &= -\mathbf{v}_x^{\text{rik}^T} \mathbf{F}'^k + (1/a^k) \mathbf{v}_y^{\text{rik}^T} \mathbf{T}'^k \\ f'_y^{\text{lk}} &= -\mathbf{v}_y^{\text{rik}^T} \mathbf{F}'^k - (1/a^k) \mathbf{v}_x^{\text{rik}^T} \mathbf{T}'^k \\ f'_x^{2k} &= -(1/a^k) \mathbf{v}_y^{\text{rik}^T} \mathbf{T}'^k \\ f'_y^{2k} &= (1/a^k) \mathbf{v}_x^{\text{rik}^T} \mathbf{T}'^k\end{aligned}\quad (8.5.4)$$

The magnitudes of resultant *contact forces* on body  $i$  at points  $P_i^k$  and  $Q_i^k$  are

$$f'^{\ell k} = \sqrt{(f'_x^{\ell k})^2 + (f'_y^{\ell k})^2}, \ell = 1, 2 \quad (8.5.5)$$

For the *revolute joint*,  $\mathbf{F}'^k_i$  and  $\mathbf{T}'^k_i$  are calculated using Eq. (8.5.1), with submatrices of the Jacobian of the revolute constraint of Eq. (3.3.26) defined in Section 3.3 and evaluated using software in Code 5.8. In addition to the lateral forces of Eqs. (8.5.4) and (8.5.5) that are defined using the revolute constraint Jacobian, the contact force  $\mathbf{f}'^{1k}$  shown in Fig. 8.5.2 is the thrust due to the constraint of Eq. (3.3.25),

$$\mathbf{f}'^{1k} = \mathbf{V}_z'^{ikT} \mathbf{F}'^k_i \quad (8.5.6)$$

For the *translational joint*,  $\mathbf{F}'^k_i$  and  $\mathbf{T}'^k_i$  are calculated using Eq. (8.5.1), with submatrices of the Jacobian of the translational constraint of Eq. (3.3.28) defined in Section 3.3 and evaluated using software in Code 5.8. A key of length  $b^k$  along the negative  $\mathbf{v}_y'^{ik}$  axis attached at point  $P^k$  rides in a slot in body j that is parallel to  $\mathbf{v}_z'^{ik}$ , to enforce the constraint that precludes relative rotation of bodies i and j about the axis defined by  $\mathbf{v}_z'^{ik}$ , as shown in Fig. 8.5.3.

The effect of the rotational constraint of Eq. (3.3.27) and associated constraint torque about vector  $\mathbf{v}_z'^{ik}$  determines the reaction force  $\mathbf{f}_x'^{3k}$  shown in Fig. 8.5.3. Assuming that a bearing surface on the  $\mathbf{v}_y'^{ik}$  axis that precludes relative displacement in the  $\mathbf{v}_x'^{ik}$  direction, hence precluding relative rotation about  $\mathbf{v}_z'^{ik}$ , is at a distance  $b^k$  from point  $P_i^k$ , the moment condition about  $\mathbf{v}_z'^{ik}$  is  $\mathbf{V}_z'^{ikT} \mathbf{T}_i'^k + b^k \mathbf{f}_x'^{3k} = 0$ . Thus,

$$\mathbf{f}_x'^{3k} = -(1/b^k) \mathbf{V}_z'^{ikT} \mathbf{T}_i'^k \quad (8.5.7)$$

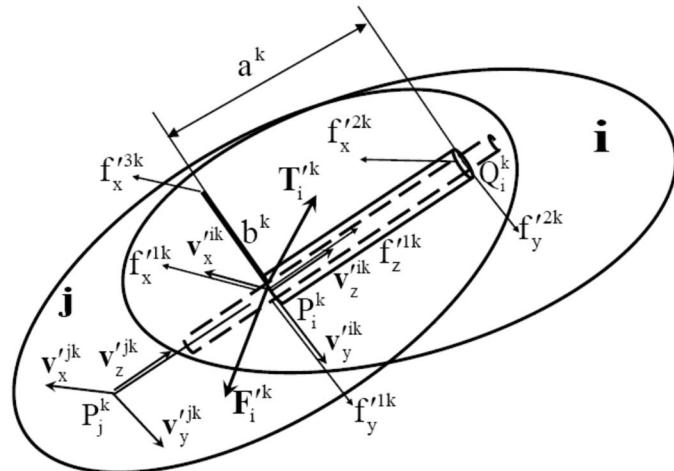


Figure 8.5.3 Constraint Forces for Translational Joint

Summing forces on body i in the  $\mathbf{v}_x'^{ik}$  and  $\mathbf{v}_y'^{ik}$  directions,

$$\begin{aligned}\mathbf{v}_x'^{ikT} \mathbf{F}_i'^k + f_x'^{1k} + f_x'^{3k} + f_x'^{2k} &= 0 \\ \mathbf{v}_y'^{ikT} \mathbf{F}_i'^k + f_y'^{1k} + f_y'^{2k} &= 0\end{aligned}\quad (8.5.8)$$

and summing torques about  $\mathbf{v}_x'^{ik}$  and  $\mathbf{v}_y'^{ik}$ ,

$$\begin{aligned}\mathbf{v}_x'^{ikT} \mathbf{T}_i'^k - a^k f_y'^{2k} &= 0 \\ \mathbf{v}_y'^{ikT} \mathbf{T}_i'^k + a^k f_x'^{2k} &= 0\end{aligned}\quad (8.5.9)$$

The solution of Eqs. (8.5.8) and (8.5.9) is contact forces that are statically equivalent to  $\mathbf{F}_i'^k$  and  $\mathbf{T}_i'^k$ ,

$$\begin{aligned}f_x'^{1k} &= -\mathbf{v}_x'^{ikT} \mathbf{F}_i'^k + ((1/a^k) \mathbf{v}_y'^{ikT} + (1/b^k) \mathbf{v}_z'^{ikT}) \mathbf{T}_i'^k \\ f_y'^{1k} &= -\mathbf{v}_y'^{ikT} \mathbf{F}_i'^k - (1/a^k) \mathbf{v}_x'^{ikT} \mathbf{T}_i'^k \\ f_x'^{2k} &= -(1/a^k) \mathbf{v}_y'^{ikT} \mathbf{T}_i'^k \\ f_y'^{2k} &= (1/a^k) \mathbf{v}_x'^{ikT} \mathbf{T}_i'^k\end{aligned}\quad (8.5.10)$$

Clearly, *joint contact forces* depend on the geometry of the translational joint. A two contact point geometry that is added to the cylindrical constraint is presented in (Haug, 2018c), yielding different results, as would more intricate models of contacts that occur in alternative translational *joint designs*.

### 8.5.2 Friction Forces in Joints

In order to avoid difficulties with the discontinuous sign function of slip velocity in the Coulomb friction model and to account for friction effects when sliding velocity is near zero, the *continuous friction* model of Section 8.1.2.2 is used. Since spherical joints are generally lubricated and subject to low friction effects, friction is considered here for only cylindrical, revolute, and translational joints.

For the *cylindrical joint*, constraint reaction forces defined in Section 8.5.1.1 are first computed, using the cylindrical constraint Jacobian. Since the vector  $\mathbf{d}_{ij}^k$  in Fig. 3.3.4 is in the opposite direction of  $\mathbf{v}_z^{jk}$ , the velocity of contact points in body i relative to body j in the direction  $\mathbf{v}_z^{jk}$  is

$$\dot{s}_{ij}^k = -\mathbf{v}_z^{ikT} \dot{\mathbf{d}}_{ij}^k = -\mathbf{v}_z^{ikT} \mathbf{A}_i^T (\dot{\mathbf{r}}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k) \dot{\mathbf{p}}_j - \dot{\mathbf{r}}_i - \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k) \dot{\mathbf{p}}_i)\quad (8.5.11)$$

The friction force that acts on body i in the direction  $\mathbf{v}_z^{jk}$  of translation is

$$f_{ij}^{kcyfr} = -F_{ij}^{kcy} S(\dot{s}_{ij}^k, \mu_t) \quad (8.5.12)$$

where  $\mu_t$  contains *translational coefficients of friction* and the sum of normal contact forces at contact points  $P_i^k$  and  $Q_i^k$ , from Eq. (8.5.5), is

$$F_{ij}^{k cyl} \equiv f'^{ik} + f'^{2k} \quad (8.5.13)$$

Using the identity  $\omega = 2E(\mathbf{p})\dot{\mathbf{p}}$  of Eq. (2.8.49), the angular velocity of body i relative to body j, about the axis defined by  $\mathbf{v}_z^{ik}$ , is

$$\omega_{ij}^k = \mathbf{v}_z^{ikT} (\boldsymbol{\omega}_i - \boldsymbol{\omega}_j) = 2\mathbf{v}_z'^{ikT} \mathbf{A}_i^T (E(\mathbf{p}_i)\dot{\mathbf{p}}_i - E(\mathbf{p}_j)\dot{\mathbf{p}}_j) \quad (8.5.14)$$

The *torque due to friction* on body i relative to body j about  $\mathbf{v}_z^{ik}$ , where  $R^k$  is the radius of the rotational bearing in the joint, and rotation induced slip velocity is  $R^k \omega_{ij}^k$ , is

$$\tau_{ij}^{k cylfr} = -R^k F_{ij}^{k cyl} S(R^k \omega_{ij}^k, \mu_r) \quad (8.5.15)$$

where  $\mu_r$  contains rotational coefficients of friction.

Replacing  $\omega$  with  $\delta\pi$  in Eq. (2.8.68) and derivative with differential in Eqs. (8.5.11) and (8.5.14), virtual displacement and rotation of body i relative to body j, along and about  $\mathbf{v}_z^{ik}$ , are

$$\begin{aligned} \delta s_{ij}^k &= -\mathbf{v}_z^{ikT} \delta \mathbf{d}_{ij}^k = -\mathbf{v}_z'^{ikT} \mathbf{A}_i^T (\delta \mathbf{r}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k) \delta \mathbf{p}_j - \delta \mathbf{r}_i - \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k) \delta \mathbf{p}_i) \\ \delta \pi_{ij}^k &= \mathbf{v}_z^{ikT} (\delta \pi_i - \delta \pi_j) = 2\mathbf{v}_z'^{ikT} \mathbf{A}_i^T (E(\mathbf{p}_i) \delta \mathbf{p}_i - E(\mathbf{p}_j) \delta \mathbf{p}_j) \end{aligned} \quad (8.5.16)$$

The virtual work of friction forces that act in cylindrical joint k is thus

$$\begin{aligned} \delta W^k &= \mathbf{Q}_i^{k cylfrT} \begin{bmatrix} \delta \mathbf{r}_i^T & \delta \mathbf{p}_i^T \end{bmatrix}^T + \mathbf{Q}_j^{k cylfrT} \begin{bmatrix} \delta \mathbf{r}_j^T & \delta \mathbf{p}_j^T \end{bmatrix}^T = \delta s_{ij}^k f_{ij}^{k cylfr} + \delta \pi_{ij}^k \tau_{ij}^{k cylfr} \\ &= -\mathbf{v}_z'^{ikT} \mathbf{A}_i^T (\delta \mathbf{r}_j + \mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k) \delta \mathbf{p}_j - \delta \mathbf{r}_i - \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k) \delta \mathbf{p}_i) f_{ij}^{k cylfr} \\ &\quad + 2\mathbf{v}_z'^{ikT} \mathbf{A}_i^T (E(\mathbf{p}_i) \delta \mathbf{p}_i - E(\mathbf{p}_j) \delta \mathbf{p}_j) \tau_{ij}^{k cylfr} \end{aligned} \quad (8.5.17)$$

so generalized forces acting on bodies i and j due to friction in the cylindrical joint are

$$\begin{aligned} \mathbf{Q}_i^{k cylfr} &= \begin{bmatrix} \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij}^{k cylfr} \\ \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij}^{k cylfr} + 2E(\mathbf{p}_i)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ij}^{k cylfr} \end{bmatrix} \\ \mathbf{Q}_j^{k cylfr} &= \begin{bmatrix} -\mathbf{A}_j \mathbf{v}_z'^{ik} f_{ij}^{k cylfr} \\ -\mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k)^T \mathbf{A}_j \mathbf{v}_z'^{ik} f_{ij}^{k cylfr} - 2E(\mathbf{p}_j)^T \mathbf{A}_j \mathbf{v}_z'^{ik} \tau_{ij}^{k cylfr} \end{bmatrix} \end{aligned} \quad (8.5.18)$$

For the *revolute joint*, constraint reaction forces defined in Section 8.5.1.2 are first computed, using the revolute constraint Jacobian. The friction torque for the revolute joint is the torque of Eq. (8.5.15) for the cylindrical joint, since  $\dot{s}_{ij}^k = \dot{\delta s}_{ij}^k = 0$  as a result of the thrust constraint, plus the effect of thrust reaction force  $f_z'^{ik} = \mathbf{v}_z'^{ikT} \mathbf{F}_i^k$  of Eq. (8.5.6). Assuming that the *thrust bearing* has radius  $R^k$ , the slip velocity is  $R^k \omega_{ij}^k$  and the friction torque acting on body i about the  $\mathbf{v}_z^{ik}$  axis due to the thrust bearing is

$$\begin{aligned}\tau_{ij}^{krev5fr} &= -R^k |f_z'^{ik}| S(R^k \omega_{ij}^k, \mu_r) \\ &\approx -R^k f_z'^{ik} \text{csign}(f_z'^{ik}) S(R^k \omega_{ij}^k, \mu_r)\end{aligned}\quad (8.5.19)$$

The *virtual work* due to this torque and that of Eq. (8.5.15), computed using revolute constraint subJacobians, is

$$\delta W^k = \delta \pi_{ij}^k (\tau_{ij}^{krev5fr} + \tau_{ij}^{kcylfr}) = 2v_z'^{ikT} A_i^T (E(p_i) \delta p_i - E(p_j) \delta p_j) (\tau_{ij}^{krev5fr} + \tau_{ij}^{kcylfr})$$

so generalized forces that act on bodies i and j due to friction in the revolute joint are

$$\begin{aligned}Q_i^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ 2E^T(p_i) A_i v_z'^{ik} (\tau_{ij}^{krev5fr} + \tau_{ij}^{kcylfr}) \end{bmatrix} \\ Q_j^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ -2E^T(p_j) A_i v_z'^{ik} (\tau_{ij}^{krev5fr} + \tau_{ij}^{kcylfr}) \end{bmatrix}\end{aligned}\quad (8.5.20)$$

For the *translational joint*, constraint reaction forces defined in Section 8.5.1.3 are first computed, using the translational constraint Jacobian. The friction force for the translational joint has only a translational component, since  $\omega_{ij}^k = \delta \pi_{ij}^k = 0$ , due to the rotational constraint. Since the translational joint contact geometry tends to be rectangular, in contrast to the rotational bearings in cylindrical and revolute joints, each of the contact force components of Eqs. (8.5.10) and (8.5.7) acts independently. This yields the friction translational force in the  $v_z^{ik}$  direction,

$$f_{ij}^{ktranfr} = -F_{ij}^{ktran} S(\dot{s}_{ij}^k, \mu_t) \quad (8.5.21)$$

where

$$\begin{aligned}F_{ij}^{ktran} &= |f_x'^{ik}| + |f_y'^{ik}| + |f_x'^{2k}| + |f_y'^{2k}| + |f_x'^{3k}| \\ &\approx f_x'^{ik} \text{csign}(f_x'^{ik}) + f_y'^{ik} \text{csign}(f_y'^{ik}) + f_x'^{2k} \text{csign}(f_x'^{2k}) \\ &\quad + f_y'^{2k} \text{csign}(f_y'^{2k}) + f_x'^{3k} \text{csign}(f_x'^{3k})\end{aligned}\quad (8.5.22)$$

The virtual work of this force is

$$\delta W^{ktranfr} = \delta s_{ij}^k f_{ij}^{ktranfr} = -v_z'^{ikT} A_i^T (\delta r_j + B(p_j, s_j'^k) \delta p_j - \delta r_i - B(p_i, s_i'^k) \delta p_i) f_{ij}^{ktranfr}$$

so generalized forces acting on bodies i and j due to friction in the translational joint are

$$\begin{aligned}Q_i^{ktranfr} &= \begin{bmatrix} A_i v_z'^{ik} \\ B^T(p_i, s_i'^k) A_i v_z'^{ik} \end{bmatrix} f_{ij}^{ktranfr} \\ Q_j^{ktranfr} &= \begin{bmatrix} -A_i v_z'^{ik} \\ -B^T(p_j, s_j'^k) A_i v_z'^{ik} \end{bmatrix} f_{ij}^{ktranfr}\end{aligned}\quad (8.5.23)$$

### 8.5.3 Derivatives of Friction Forces

For each of the three joints analyzed, using the associated constraint Jacobian, Eq. (8.5.1) is differentiated with respect to  $\mathbf{q}_{ij} = [\mathbf{q}_i^T \quad \mathbf{q}_j^T]^T$  and  $\lambda^k$  to obtain

$$\begin{aligned}\mathbf{F}'^k_{iq_{ij}} &= -\mathbf{A}_i^T \left( \Phi_{r_i}^{kT} \dot{\lambda}^k \right)_{q_{ij}} - \left[ \mathbf{0} \quad \mathbf{C}(\mathbf{p}_i, \Phi_{r_i}^{kT} \lambda^k) \quad \mathbf{0} \quad \mathbf{0} \right] \\ \mathbf{T}'^k_{iq_{ij}} &= - \left( (1/2)\mathbf{G}(\mathbf{p}_i) \left( \Phi_{p_i}^{kT} \dot{\lambda}^k \right)_{q_{ij}} - \tilde{s}'^k_i \mathbf{A}_i^T \left( \Phi_{r_i}^{kT} \dot{\lambda}^k \right)_{q_{ij}} \right) \\ &\quad - \left[ \mathbf{0} \quad -(1/2)\mathbf{G}(\Phi_{p_i}^{kT} \lambda^k) - \tilde{s}'^k_i \mathbf{C}(\mathbf{p}_i, \Phi_{r_i}^{kT} \lambda^k) \quad \mathbf{0} \quad \mathbf{0} \right] \\ \mathbf{F}'^k_{i\lambda^k} &= -\mathbf{A}_i^T \Phi_{r_i}^{kT} \\ \mathbf{T}'^k_{i\lambda^k} &= - \left( (1/2)\mathbf{G}(\mathbf{p}_i) \Phi_{p_i}^{kT} - \tilde{s}'^k_i \mathbf{A}_i^T \Phi_{r_i}^{kT} \right)\end{aligned}\quad (8.5.24)$$

where Eqs. (2.8.30) and (2.8.44) and identities of Section 2.6 have been used. The terms  $\left( \Phi_{r_i}^{kT} \dot{\lambda}^k \right)_{q_{ij}}$  and  $\left( \Phi_{p_i}^{kT} \dot{\lambda}^k \right)_{q_{ij}}$  in Eqs. (8.5.24) are rows of

$$\begin{aligned}\left( \Phi_{q_i}^{kT} \dot{\lambda}^k \right)_{q_{ij}} &= \left[ \left( \Phi_{q_i}^{kT} \dot{\lambda}^k \right)_{q_i} \quad \left( \Phi_{q_i}^{kT} \dot{\lambda}^k \right)_{q_j} \right] \\ &= \left[ \left( \Phi_{r_i}^{kT} \dot{\lambda}^k \right)_{r_i} \quad \left( \Phi_{r_i}^{kT} \dot{\lambda}^k \right)_{p_i} \quad \left( \Phi_{r_i}^{kT} \dot{\lambda}^k \right)_{r_j} \quad \left( \Phi_{r_i}^{kT} \dot{\lambda}^k \right)_{p_j} \right] \\ &= \left[ \left( \Phi_{p_i}^{kT} \dot{\lambda}^k \right)_{r_i} \quad \left( \Phi_{p_i}^{kT} \dot{\lambda}^k \right)_{p_i} \quad \left( \Phi_{p_i}^{kT} \dot{\lambda}^k \right)_{r_j} \quad \left( \Phi_{p_i}^{kT} \dot{\lambda}^k \right)_{p_j} \right]\end{aligned}\quad (8.5.25)$$

which are defined in Eq. (5.3.34) as the matrix  $\mathbf{P4}(q_{ij}, \lambda^k)$ , for each of the spatial joints in Appendix 5.B, and evaluated using software in Code 5.8.

For the cylindrical joint, using the associated constraint Jacobian and matrix  $\mathbf{P4}(q_{ij}, \lambda^k)$ , derivatives of terms in Eqs. (8.5.4) with respect to  $\mathbf{q}_{ij}$  are obtained using the first two of Eqs. (8.5.24),

$$\begin{aligned}f'^{1k}_{xq_{ij}} &= -\mathbf{v}_x'^{ikT} \mathbf{F}'^k_{iq_{ij}} + (1/a^k) \mathbf{v}_y'^{ikT} \mathbf{T}'^k_{iq_{ij}} \\ f'^{1k}_{yq_{ij}} &= -\mathbf{v}_y'^{ikT} \mathbf{F}'^k_{iq_{ij}} - (1/a^k) \mathbf{v}_x'^{ikT} \mathbf{T}'^k_{iq_{ij}} \\ f'^{2k}_{xq_{ij}} &= -(1/a^k) \mathbf{v}_y'^{ikT} \mathbf{T}'^k_{iq_{ij}} \\ f'^{2k}_{yq_{ij}} &= (1/a^k) \mathbf{v}_x'^{ikT} \mathbf{T}'^k_{iq_{ij}}\end{aligned}\quad (8.5.26)$$

Expressions for derivatives with respect to  $\lambda^k$  are obtained by replacing subscript  $q_{ij}$  with  $\lambda^k$  in Eqs. (8.5.26), using the third and fourth of Eqs. (8.5.24).

Derivatives of Eqs. (8.5.13) with respect to  $\mathbf{q}_{ij}$  and  $\lambda^k$  are

$$\begin{aligned}
F_{ijq_{ij}}^{k cyl} &= f_{q_{ij}}'^{1k} + f_{q_{ij}}'^{2k} \\
&= (1/f'^{1k}) \left( f_x'^{1k} f_{xq_{ij}}'^{1k} + f_y'^{1k} f_{yq_{ij}}'^{1k} \right) + (1/f'^{2k}) \left( f_x'^{2k} f_{xq_{ij}}'^{2k} + f_y'^{2k} f_{yq_{ij}}'^{2k} \right) \\
F_{ij\lambda^k}^{k cyl} &= (1/f'^{1k}) \left( f_x'^{1k} f_{x\lambda^k}'^{1k} + f_y'^{1k} f_{y\lambda^k}'^{1k} \right) + (1/f'^{2k}) \left( f_x'^{2k} f_{x\lambda^k}'^{2k} + f_y'^{2k} f_{y\lambda^k}'^{2k} \right)
\end{aligned} \quad (8.5.27)$$

Derivatives of Eqs. (8.5.11), and (8.5.14) with respect to  $q_{ij}$  are

$$\begin{aligned}
\dot{s}_{ijq_{ij}}^k &= -\mathbf{v}_z'^{ikT} \left\{ \begin{bmatrix} \mathbf{0} & C(\mathbf{p}_i, (\dot{\mathbf{r}}_j + \mathbf{B}(\mathbf{p}_j, s_j'^k) \dot{\mathbf{p}}_j - \dot{\mathbf{r}}_i - \mathbf{B}(\mathbf{p}_i, s_i'^k) \dot{\mathbf{p}}_i)) & \mathbf{0} & \mathbf{0} \\ + A_i^T [\mathbf{0} & -\mathbf{B}(\dot{\mathbf{p}}_i, s_i'^k) & \mathbf{0} & \mathbf{B}(\dot{\mathbf{p}}_j, s_j'^k)] \end{bmatrix} \right\} \\
\omega_{ijq_{ij}}^k &= 2\mathbf{v}_z'^{ikT} [\mathbf{0} & C(\mathbf{p}_i, (E(\mathbf{p}_i) \dot{\mathbf{p}}_i - E(\mathbf{p}_j) \dot{\mathbf{p}}_j)) - A_i^T E(\dot{\mathbf{p}}_i) & \mathbf{0} & A_i^T E(\dot{\mathbf{p}}_j)]
\end{aligned} \quad (8.5.28)$$

where Eqs (2.8.14) and (2.8.28) have been used. Since  $\dot{s}_{ij}^k$  and  $\omega_{ij}^k$  do not depend on  $\lambda^k$ ,

$\dot{s}_{ij\lambda^k}^k = \omega_{ij\lambda^k}^k = \mathbf{0}$ . Using these results in Eqs. (8.5.12) and (8.5.15), derivatives with respect to  $q_{ij}$  are

$$\begin{aligned}
f_{ijq_{ij}}^{k cylfr} &= -F_{ij}^{k cyl} S'(\dot{s}_{ij}^k, \mu_t) \dot{s}_{ijq_{ij}}^k - S(\dot{s}_{ij}^k, \mu_t) F_{ijq_{ij}}^{k cyl} \\
ijq_{ij}^{k cylfr} &= -\left(R^k\right)^2 F_{ij}^{k cyl} S'(R^k \omega_{ij}^k, \mu_r) \omega_{ijq_{ij}}^k - R^k S(R^k \omega_{ij}^k, \mu_r) F_{ijq_{ij}}^{k cyl}
\end{aligned} \quad (8.5.29)$$

Since  $\dot{s}_{ij}^k$  and  $\omega_{ij}^k$  do not depend on  $\lambda^k$ ,

$$\begin{aligned}
f_{ij\lambda^k}^{k cylfr} &= -S(\dot{s}_{ij}^k, \mu_t) F_{ij\lambda^k}^{k cyl} \\
ij\lambda^k^{k cylfr} &= -R^k S(R^k \omega_{ij}^k, \mu_r) F_{ij\lambda^k}^{k cyl}
\end{aligned} \quad (8.5.30)$$

To account for derivatives with respect to  $\dot{q}_{ij}$ , from Eqs. (8.5.11) and (8.5.14),

$$\begin{aligned}
\dot{s}_{ij\dot{q}_{ij}}^k &= -\mathbf{v}_z'^{ikT} A_i^T [-\mathbf{I} & -\mathbf{B}(\mathbf{p}_i, s_i'^k) & \mathbf{I} & \mathbf{B}(\mathbf{p}_j, s_j'^k)] \\
\omega_{ij\dot{q}_{ij}}^k &= 2\mathbf{v}_z'^{ikT} A_i^T [\mathbf{0} & E(\mathbf{p}_i) & \mathbf{0} & -E(\mathbf{p}_j)]
\end{aligned} \quad (8.5.31)$$

Since  $F_{ij}^{k cyl}$  does not depend on  $q_{ij}$ , from Eqs. (8.5.12) and (8.5.15),

$$\begin{aligned}
f_{ij\dot{q}_{ij}}^{k cylfr} &= -F_{ij}^{k cyl} S'(\dot{s}_{ij}^k, \mu_t) \dot{s}_{ij\dot{q}_{ij}}^k \\
ij\dot{q}_{ij}^{k cylfr} &= -\left(R^k\right)^2 F_{ij}^{k cyl} S'(R^k \omega_{ij}^k, \mu_r) \omega_{ij\dot{q}_{ij}}^k
\end{aligned} \quad (8.5.32)$$

The foregoing enable calculation of derivatives of generalized friction forces of Eq. (8.5.18) with respect to  $q_{ij}$ ,  $\lambda^k$ , and  $\dot{q}_{ij}$ , using derivative identities of Section 2.6, as

$$\begin{aligned}
\mathbf{Q}_{iq_{ij}}^{k cylfr} &= \begin{bmatrix} \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ijq_{ij}}^{k cylfr} + \mathbf{a}_1 \\ \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ijq_{ij}}^{k cylfr} + 2\mathbf{E}(\mathbf{p}_i)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ijq_{ij}}^{k cylfr} + \mathbf{a}_2 \end{bmatrix} \\
\mathbf{Q}_{jq_{ij}}^{k cylfr} &= \begin{bmatrix} -\mathbf{A}_i \mathbf{v}_z'^{ik} f_{ijq_{ij}}^{k cylfr} - \mathbf{a}_1 \\ -\mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ijq_{ij}}^{k cylfr} - 2\mathbf{E}(\mathbf{p}_j)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ijq_{ij}}^{k cylfr} - \mathbf{a}_3 \end{bmatrix} \\
\mathbf{a}_1 &= [\mathbf{0} \quad \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik} f_{ij}^{k cylfr}) \quad \mathbf{0} \quad \mathbf{0}] \\
\mathbf{a}_2 &= [\mathbf{0} \quad \mathbf{K}(\mathbf{s}_i'^k, \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij}^{k cylfr}) + \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik} f_{ij}^{k cylfr}) \quad \mathbf{0} \quad \mathbf{0}] \\
&\quad + 2[\mathbf{0} \quad \mathbf{R}(\mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ij}^{k cylfr}) + \mathbf{E}(\mathbf{p}_i)^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik} \tau_{ij}^{k cylfr}) \quad \mathbf{0} \quad \mathbf{0}]
\end{aligned} \tag{8.5.33}$$

$$\begin{aligned}
\mathbf{Q}_{i\lambda^k}^{k cylfr} &= \begin{bmatrix} \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\lambda^k}^{k cylfr} \\ \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\lambda^k}^{k cylfr} + 2\mathbf{E}^T(\mathbf{p}_i) \mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ij\lambda^k}^{k cylfr} \end{bmatrix} \\
\mathbf{Q}_{j\lambda^k}^{k cylfr} &= \begin{bmatrix} -\mathbf{A}_i \mathbf{u}_z'^{ik} f_{ij\lambda^k}^{k cylfr} \\ -\mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\lambda^k}^{k cylfr} - 2\mathbf{E}(\mathbf{p}_j)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ij\lambda^k}^{k cylfr} \end{bmatrix}
\end{aligned} \tag{8.5.34}$$

$$\begin{aligned}
\mathbf{Q}_{i\dot{q}_{ij}}^{k cylfr} &= \begin{bmatrix} \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\dot{q}_{ij}}^{k cylfr} \\ \mathbf{B}^T(\mathbf{p}_i, \mathbf{s}_i'^k) \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\dot{q}_{ij}}^{k cylfr} + 2\mathbf{E}^T(\mathbf{p}_i) \mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ij\dot{q}_{ij}}^{k cylfr} \end{bmatrix} \\
\mathbf{Q}_{j\dot{q}_{ij}}^{k cylfr} &= \begin{bmatrix} -\mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\dot{q}_{ij}}^{k cylfr} \\ -\mathbf{B}^T(\mathbf{p}_j, \mathbf{s}_j'^k) \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\dot{q}_{ij}}^{k cylfr} - 2\mathbf{E}^T(\mathbf{p}_j) \mathbf{A}_i \mathbf{v}_z'^{ik} \tau_{ij\dot{q}_{ij}}^{k cylfr} \end{bmatrix}
\end{aligned} \tag{8.5.35}$$

For the revolute joint, using the associated constraint Jacobian and matrix  $\mathbf{P}4(\mathbf{q}_{ij}, \lambda^k)$ , derivatives of Eq. (8.5.26) are evaluated. Expressions for derivatives with respect to  $\lambda^k$  are likewise obtained by replacing subscript  $\mathbf{q}_{ij}$  with  $\lambda^k$  in Eqs. (8.5.26). The derivative of Eq. (8.5.6) is

$$f_{zq_{ij}}'^{lk} = \mathbf{v}_z'^{ikT} \mathbf{F}_{iq_{ij}}'^k \tag{8.5.36}$$

which is evaluated using the first of Eqs. (8.5.24). The associated derivative with respect to  $\lambda^k$  is evaluated using the third of Eqs. (8.5.24). Similarly, the derivatives of Eqs. (8.5.27) are evaluated. Since relative translation is zero, only the second of Eqs. (8.5.30) and (8.5.31) are germane. Finally, the derivative of the torque of Eq. (8.5.19) with respect to  $\mathbf{q}_{ij}$  is

$$\begin{aligned}
\tau_{ijq_{ij}}^{k rev5fr} &= -\left(R^k\right)^2 f_z'^{lk} \text{csign}(f_z'^{lk}) S'(R^k \omega_{ij}^k, \mu_r) \omega_{ijq_{ij}}^k \\
&\quad - R^k S(R^k \omega_{ij}^k, \mu_r) (\text{csign}(f_z'^{lk}) + f_z'^{lk} \text{dcsign}(f_z'^{lk})) f_{zq_{ij}}'^{lk}
\end{aligned} \tag{8.5.37}$$

Since  $\omega_{ij}^k$  does not depend on  $\lambda^k$ , the derivative of this quantity with respect to  $\lambda^k$  is

$$\tau_{ij\lambda^k}^{krevfr} = -R^k S(R^k \omega_{ij}^k, \mu_r) (csign(f_z'^{lk}) + f_z'^{lk} dcsign(f_z'^{lk})) f_z'^{lk} \quad (8.5.38)$$

To account for derivatives of Eq. (8.5.19) with respect to  $\dot{q}_{ij}$ , from Eq. (8.5.31),

$$\tau_{ij\dot{q}_{ij}}^{krevfr} = -\left(R^k\right)^2 f_z'^{lk} csign(f_z'^{lk}) S'(R^k \omega_{ij}^k, \mu_r) \omega_{ij\dot{q}_{ij}}^k \quad (8.5.39)$$

The foregoing enables calculation of derivatives of revolute generalized friction forces of Eqs. (8.5.20) with respect to  $q_{ij}$ ,  $\lambda^k$ , and  $\dot{q}_{ij}$  as

$$\begin{aligned} \mathbf{Q}_{iq_{ij}}^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ 2E(\mathbf{p}_i)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \left( \tau_{ijq_{ij}}^{krevfr} + \frac{k cylfr}{ijq_{ij}} \right) + \mathbf{b}_1 \end{bmatrix} \\ \mathbf{Q}_{jq_{ij}}^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ -2E(\mathbf{p}_j)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \left( \tau_{ijq_{ij}}^{krevfr} + \frac{k cylfr}{ijq_{ij}} \right) - \mathbf{b}_2 \end{bmatrix} \\ \mathbf{b}_1 &= 2 \begin{bmatrix} \mathbf{0} & R(\mathbf{A}_i \mathbf{v}_z'^{ik}) + E(\mathbf{p}_i)^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik}) & \mathbf{0} & \mathbf{0} \end{bmatrix} \left( \tau_{ij}^{krevfr} + \frac{k cylfr}{ij} \right) \\ \mathbf{b}_2 &= 2 \begin{bmatrix} \mathbf{0} & E(\mathbf{p}_j)^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik}) & \mathbf{0} & R(\mathbf{A}_i \mathbf{v}_z'^{ik}) \end{bmatrix} \left( \tau_{ij}^{krevfr} + \frac{k cylfr}{ij} \right) \end{aligned} \quad (8.5.40)$$

$$\begin{aligned} \mathbf{Q}_{i\lambda^k}^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ 2E(\mathbf{p}_i)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \end{bmatrix} \left( \tau_{ij\lambda^k}^{krevfr} + \frac{k cylfr}{ij\lambda^k} \right) \\ \mathbf{Q}_{j\lambda^k}^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ -2E(\mathbf{p}_j)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \end{bmatrix} \left( \tau_{ij\lambda^k}^{krevfr} + \frac{k cylfr}{ij\lambda^k} \right) \end{aligned} \quad (8.5.41)$$

$$\begin{aligned} \mathbf{Q}_{i\dot{q}_{ij}}^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ 2E(\mathbf{p}_i)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \left( \tau_{ij\dot{q}_{ij}}^{krevfr} + \frac{k cylfr}{ij\dot{q}_{ij}} \right) \end{bmatrix} \\ \mathbf{Q}_{j\dot{q}_{ij}}^{krevfr} &= \begin{bmatrix} \mathbf{0} \\ -2E(\mathbf{p}_j)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \left( \tau_{ij\dot{q}_{ij}}^{krevfr} + \frac{k cylfr}{ij\dot{q}_{ij}} \right) \end{bmatrix} \end{aligned} \quad (8.5.42)$$

For the translational joint, using the associated constraint Jacobian and matrix  $\mathbf{P}4(q_{ij}, \lambda^k)$ , derivatives of Eq. (8.5.10) with respect to  $q_{ij}$  are

$$\begin{aligned} f_{xq_{ij}}'^{lk} &= -\mathbf{v}_x'^{ikT} \mathbf{F}_{iq_{ij}}'^k + ((1/a^k) \mathbf{v}_y'^{ikT} + (1/b^k) \mathbf{v}_z'^{ikT}) \mathbf{T}_{iq_{ij}}'^k \\ f_{yq_{ij}}'^{lk} &= -\mathbf{v}_y'^{ikT} \mathbf{F}_{iq_{ij}}'^k - (1/a^k) \mathbf{v}_x'^{ikT} \mathbf{T}_{iq_{ij}}'^k \\ f_{xq_{ij}}'^{2k} &= -(1/a^k) \mathbf{v}_y'^{ikT} \mathbf{T}_{iq_{ij}}'^k \\ f_{yq_{ij}}'^{2k} &= (1/a^k) \mathbf{v}_x'^{ikT} \mathbf{T}_{iq_{ij}}'^k \end{aligned} \quad (8.5.43)$$

Expressions for derivatives with respect to  $\lambda^k$  are likewise obtained by replacing subscript  $\mathbf{q}_{ij}$  with  $\lambda^k$  in Eqs. (8.5.43). The derivative of Eq. (8.5.7) with respect to  $\mathbf{q}_{ij}$  is

$$f'_{x\mathbf{q}_{ij}} = (1/b^k) \mathbf{v}_z'^{ikT} \mathbf{T}_{i\mathbf{q}_{ij}}^{ik} \quad (8.5.44)$$

which is evaluated using the second of Eqs. (8.5.24). The associated derivative with respect to  $\lambda^k$  is evaluated using the fourth of Eqs. (8.5.24).

The derivative of Eq. (8.5.22) with respect to  $\mathbf{q}_{ij}$  is

$$\begin{aligned} F_{ij\mathbf{q}_{ij}}^{ktran} \equiv & \left( \text{csign}(f_x'^{ik}) + f_x'^{ik} \text{dcsign}(f_x'^{ik}) \right) f_{x\mathbf{q}_{ij}}^{ik} + \left( \text{csign}(f_y'^{ik}) + f_y'^{ik} \text{dcsign}(f_y'^{ik}) \right) f_{y\mathbf{q}_{ij}}^{ik} \\ & + \left( \text{csign}(f_x'^{2k}) + f_x'^{2k} \text{dcsign}(f_x'^{2k}) \right) f_{x\mathbf{q}_{ij}}^{2k} + \left( \text{csign}(f_y'^{2k}) + f_y'^{2k} \text{dcsign}(f_y'^{2k}) \right) f_{y\mathbf{q}_{ij}}^{2k} \\ & + \left( \text{csign}(f_x'^{3k}) + f_x'^{3k} \text{design}(f_x'^{3k}) \right) f_{x\mathbf{q}_{ij}}^{3k} \end{aligned} \quad (8.5.45)$$

and similarly with respect to  $\lambda^k$ . Derivatives of Eq. (8.5.21) are thus

$$\begin{aligned} f_{ij\dot{\mathbf{q}}_{ij}}^{ktranfr} &= -F_{ij}^{ktran} S'(\dot{s}_{ij}^k, \mu_t) \dot{s}_{ij\mathbf{q}_{ij}}^k - S(\dot{s}_{ij}^k, \mu_t) F_{ij\mathbf{q}_{ij}}^{ktran} \\ f_{ij\lambda^k}^{ktranfr} &= -S(\dot{s}_{ij}^k, \mu_t) F_{ij\lambda^k}^{ktran} \\ f_{ij\dot{\lambda}_{ij}}^{ktranfr} &= -F_{ij}^{ktran} S'(\dot{s}_{ij}^k, \mu_t) \dot{s}_{ij\dot{\lambda}_{ij}}^k \end{aligned} \quad (8.5.46)$$

The foregoing enables calculation of derivatives of translational generalized friction forces of Eqs. (8.5.23) with respect to  $\mathbf{q}_{ij}$ ,  $\lambda^k$ , and  $\dot{\mathbf{q}}_{ij}$  as

$$\begin{aligned} \mathbf{Q}_{i\mathbf{q}_{ij}}^{ktranfr} &= \begin{bmatrix} \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\mathbf{q}_{ij}}^{ktranfr} + \mathbf{c}_1 \\ \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\mathbf{q}_{ij}}^{ktranfr} + \mathbf{c}_2 \end{bmatrix} \\ \mathbf{Q}_{j\mathbf{q}_{ij}}^{ktranfr} &= - \begin{bmatrix} -\mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\mathbf{q}_{ij}}^{ktranfr} - \mathbf{c}_1 \\ -\mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} f_{ij\mathbf{q}_{ij}}^{ktranfr} - \mathbf{c}_3 \end{bmatrix} \\ \mathbf{c}_1 &= [\mathbf{0} \quad \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik}) \quad \mathbf{0} \quad \mathbf{0}] f_{ij}^{ktranfr} \\ \mathbf{c}_2 &= [\mathbf{0} \quad \mathbf{K}(\mathbf{s}_i'^k, \mathbf{A}_i \mathbf{v}_z'^{ik}) + \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik}) \quad \mathbf{0} \quad \mathbf{0}] f_{ij}^{ktranfr} \end{aligned} \quad (8.5.47)$$

$$\mathbf{c}_3 = [\mathbf{0} \quad \mathbf{B}(\mathbf{p}_j, \mathbf{s}_i'^k)^T \mathbf{B}(\mathbf{p}_i, \mathbf{v}_z'^{ik}) \quad \mathbf{0} \quad \mathbf{K}(\mathbf{s}_j'^k, \mathbf{A}_i \mathbf{v}_z'^{ik})] f_{ij}^{ktranfr} \quad (8.5.48)$$

$$\begin{aligned} \mathbf{Q}_{i\lambda^k}^{ktranfr} &= \begin{bmatrix} \mathbf{A}_i \mathbf{v}_z'^{ik} \\ \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \end{bmatrix} f_{ij\lambda^k}^{ktranfr} \\ \mathbf{Q}_{j\lambda^k}^{ktranfr} &= \begin{bmatrix} -\mathbf{A}_i \mathbf{v}_z'^{ik} \\ -\mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \end{bmatrix} f_{ij\lambda^k}^{ktranfr} \end{aligned} \quad (8.5.48)$$

$$\begin{aligned}\mathbf{Q}_{i\dot{\mathbf{q}}_{ij}}^{ktranfr} &= \begin{bmatrix} \mathbf{A}(\mathbf{p}_i)\mathbf{v}_z'^{ik} \\ \mathbf{B}(\mathbf{p}_i, \mathbf{s}_i'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \end{bmatrix} \mathbf{f}_{i\dot{\mathbf{q}}_{ij}}^{ktranfr} \\ \mathbf{Q}_{j\dot{\mathbf{q}}_{ij}}^{ktranfr} &= \begin{bmatrix} -\mathbf{A}(\mathbf{p}_i)\mathbf{v}_z'^{ik} \\ -\mathbf{B}(\mathbf{p}_j, \mathbf{s}_j'^k)^T \mathbf{A}_i \mathbf{v}_z'^{ik} \end{bmatrix} \mathbf{f}_{j\dot{\mathbf{q}}_{ij}}^{ktranfr}\end{aligned}\quad (8.5.49)$$

Constraint reaction forces at joint definition points derived using results of Section 4.10 are converted to equivalent contact forces in spatial joints that are defined in Section 3.3. Relative velocities at contact points are used with contact forces and the continuous friction model of Section 8.1.2.2 to obtain generalized friction forces that act on the multibody system. Expressions for derivatives of friction forces that are required for implementation of Index 0 DAE integration algorithms are derived. Thankfully, these results hold in perpetuity.

## 8.6 Index 0 DAE Formulation for Systems with Friction

As shown in Sections 8.1 and 8.3, the *Index 0 DAE formulation* of Section 5.5 is well suited for simulation of multibody systems with friction. Its formulation includes *Lagrange multipliers* that determine *constraint contact forces* that are used in Sections 8.4 and 8.5 to derive expressions for *generalized friction forces* for planar and spatial systems, respectively. Once generalized friction forces are available, implementation of the Index 0 formulation for solution of the equations of motion of planar and spatial systems is identical, as presented in this section.

### 8.6.1 Index 0 Equations of Motion

Including the effects of friction forces derived in Sections 8.4 and 8.5, system *equations of motion with friction* are of the form

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_q^T(\mathbf{q}, t)\boldsymbol{\lambda} = \mathbf{Q}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \quad (8.6.1)$$

where the term  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$  is zero for planar systems that are modeled with centroidal generalized coordinates. Substituting the relation  $\ddot{\mathbf{q}} = \mathbf{D}(\mathbf{q}, t)\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{q})\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t)$  of Eq. (5.2.31) into Eq. (8.6.1) yields the *Index 0 equations of motion*

$$\mathbf{M}(\mathbf{q}, t)\mathbf{D}(\mathbf{q}, t)\ddot{\mathbf{v}} + \Phi_q^T(\mathbf{q}, t)\boldsymbol{\lambda} = \mathbf{M}(\mathbf{q}, t)\mathbf{U}\mathbf{B}(\mathbf{q}, t)\boldsymbol{\gamma}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{Q}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \quad (8.6.2)$$

with constraint equations in Cartesian generalized coordinates that are satisfied by the tangent space parameterization of Section 5.2,

$$\begin{aligned} \Phi(\mathbf{q}, t) &= \mathbf{0} \\ \Phi_q \dot{\mathbf{q}} &= -\Phi_t = \mathbf{v} \\ \Phi_q \ddot{\mathbf{q}} &= -\left( \left( \Phi_q \ddot{\mathbf{q}} \right)_q \dot{\mathbf{q}} + 2\Phi_{qt} \dot{\mathbf{q}} + \Phi_{tt} \right) \equiv -\boldsymbol{\gamma} \end{aligned} \quad (8.6.3)$$

where arguments of functions that appear in the last two equations have been suppressed. With the exception of dependence of  $\mathbf{Q}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$  on  $\boldsymbol{\lambda}$  and  $\boldsymbol{\mu}$ , the Index 0 DAE of Eqs. (8.6.2) and (8.6.3) are as presented in Section 5.5. This exception, however, leads to significant challenges in numerical solution of the *equations of motion with friction*. First, equivalence of the DAE of Eqs. (8.6.2) and (8.6.3) with an ODE does not follow from the analysis of Section 5.3, so justification for direct application of an ODE integrator presented in Sections 5.5.4 and 5.5.5 does not hold.

Second, the derivative of  $\mathbf{Q}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$  with respect to  $\boldsymbol{\lambda}$  is required for use of even *explicit numerical integration* methods, and certainly for use of *implicit numerical integration* methods. To resolve these issues, explicit and implicit numerical integration algorithms are presented in Sections 8.6.2 and 8.6.3, respectively. Extensions of MATLAB Codes 5.7 and 5.9 to account for friction in planar and spatial systems are presented in Sections 8.7 and 8.9, respectively.

### 8.6.2 Explicit Numerical Integration Algorithm

The *tangent space parameterization* of holonomic constraints of Eqs (8.6.3) is not influenced by dependence of  $\mathbf{Q}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$  on  $\boldsymbol{\lambda}$ . Thus, all kinematically related terms are available for use in numerical computation of  $\ddot{\mathbf{v}}$  and  $\boldsymbol{\lambda}$ , once  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  are determined by initial

conditions or prior integration results. Equation (8.6.2), however, is nonlinear in  $\lambda$  and may be written in residual form as

$$\mathbf{R}(\ddot{\mathbf{v}}, \lambda) \equiv \mathbf{MD}\ddot{\mathbf{v}} + \Phi_q^T \lambda - \mathbf{Q}^A - \mathbf{MUB}\gamma - \mathbf{S} = \mathbf{0} \quad (8.6.4)$$

with  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ , hence  $\mathbf{q}$  and  $\dot{\mathbf{q}}$ , known. The Jacobian of  $\mathbf{R}$  with respect to  $\ddot{\mathbf{v}}$  and  $\lambda$  is

$$\mathbf{R}_{\ddot{\mathbf{v}}, \lambda} = \begin{bmatrix} \mathbf{MD} & \Phi_q^T - \mathbf{Q}^A \end{bmatrix} \quad (8.6.5)$$

While the matrix  $\begin{bmatrix} \mathbf{MD} & \Phi_q^T \end{bmatrix}$  is shown in Section 5.5.1 to be nonsingular, inclusion of the term  $-\mathbf{Q}^A$  in Eq. (8.6.5) raises the possibility that  $\mathbf{R}_{\ddot{\mathbf{v}}, \lambda}$  is singular. If so, friction effects may induce *singular behavior*. This possibility must be monitored during simulation.

To determine  $\ddot{\mathbf{v}}_n$  and  $\lambda_n$  for a time  $t_n$  at which  $\mathbf{v}_n$  and  $\dot{\mathbf{v}}_n$  are known, Newton-Raphson iteration for  $\ddot{\mathbf{v}}_n$  and  $\lambda_n$  is

$$\begin{aligned} \mathbf{R}_{\ddot{\mathbf{v}}_i} \begin{bmatrix} \Delta \ddot{\mathbf{v}}^i \\ \Delta \lambda^i \end{bmatrix} &= -\mathbf{R}(\ddot{\mathbf{v}}^i, \lambda^i) \\ \begin{bmatrix} \ddot{\mathbf{v}}^{i+1} \\ \lambda^{i+1} \end{bmatrix} &= \begin{bmatrix} \ddot{\mathbf{v}}^i \\ \lambda^i \end{bmatrix} + \begin{bmatrix} \Delta \ddot{\mathbf{v}}^i \\ \Delta \lambda^i \end{bmatrix}, \quad i = 1 \dots \text{until } \|\mathbf{R}\| \leq \text{intol} \end{aligned} \quad (8.6.6)$$

where  $\ddot{\mathbf{v}}^1$  and  $\lambda^1$  are solution estimates at  $t_n$ , perhaps  $\ddot{\mathbf{v}}^1 = \ddot{\mathbf{v}}_{n-1}$  and  $\lambda^1 = \lambda_{n-1}$ , where  $\ddot{\mathbf{v}}_{n-1}$  and  $\lambda_{n-1}$  are the solution at time step  $t_{n-1}$ .

Values of  $\ddot{\mathbf{v}}^1$  and  $\lambda^1$  at  $t_1 = t^0$  that satisfy Eq. (8.6.2) are difficult to obtain, since an estimate of their values is not readily available. Equation (8.6.2) may be solved for  $\ddot{\mathbf{v}}^0$  and  $\lambda^0$  using  $\mathbf{Q}^A(t^0, \mathbf{q}^0, \dot{\mathbf{q}}^0, \mathbf{0}, \mathbf{0})$ , where  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  are initial values that satisfy the first two of Eqs. (8.6.3). With this value of generalized force, Eq. (8.6.2) is linear in  $\ddot{\mathbf{v}}^0$  and  $\lambda^0$ , with the nonsingular coefficient matrix  $\begin{bmatrix} \mathbf{MD} & \Phi_q^T \end{bmatrix}$ . Since these values of  $\ddot{\mathbf{v}}^0$  and  $\lambda^0$  may not be accurate enough to assure convergence in Eq. (8.6.6), a *parameterized system of equations* may be used. Setting  $\boldsymbol{\mu}(w) = w\boldsymbol{\mu}$ ,  $0 \leq w \leq 1$ ,  $\ddot{\mathbf{v}}^0$  and  $\lambda^0$  are the solution of Eq. (8.6.2) with  $w = 0$ . Indexing  $w^k = k/N$ ,  $k = 1, \dots, N$ , Eq. (8.6.6) may be used with  $w = 1/N$  and  $\ddot{\mathbf{v}}^0$  and  $\lambda^0$ . With this estimate, convergence of Eq. (8.6.6) yields  $\ddot{\mathbf{v}}^1$  and  $\lambda^1$ . The process may be continued with  $w^k = k/N$ ,  $k = 2, \dots, N$ , to obtain initial values  $\ddot{\mathbf{v}}^N = \ddot{\mathbf{v}}_1$  and  $\lambda^N = \lambda_1$  that satisfy Eq. (8.6.2) at  $t^0$ .

It is interesting that the foregoing sequence of computations to obtain initial values of  $\ddot{\mathbf{v}}$  and  $\lambda$  is required in the friction problem, due to the fact that Eq. (8.6.2) cannot be reduced to an ODE. This is in contrast to the situation in Section 5.5 without friction, in which only kinematically admissible initial conditions on position and velocity are required to initiate the numerical integration process. The distinction is that Index 0 DAE without friction are equivalent to ODE, but those with friction are not.

With convergence of Eq. (8.6.6),  $\ddot{\mathbf{v}}_n$  is defined as  $\ddot{\mathbf{v}}_n = \text{ODEfunct}(\mathbf{v}_n, \dot{\mathbf{v}}_n)$ , and an explicit numerical integration algorithm may be applied to advance to time step  $t_{n+1}$  with  $\mathbf{v}_{n+1}$  and  $\dot{\mathbf{v}}_{n+1}$ . Bounds on the number of iterations required in Eq. (8.6.6) and the *condition number* of  $\mathbf{R}_{\dot{\mathbf{v}}_n}$  are monitored as criteria for *reparameterization*. If unbounded growth in the condition number of  $\mathbf{R}_{\dot{\mathbf{v}}_n}$  occurs, the possibility of singular behavior of the system must be considered.

The *explicit Index 0 DAE numerical integration algorithm with friction* is as follows:

- (1) Define initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t^0$  that satisfy kinematic configuration and velocity constraints of Eq. (8.6.3). Evaluate the constraint Jacobian  $\Phi_q(\mathbf{q}^0, t^0)$  and matrices  $\mathbf{U}$  and  $\mathbf{V}$  in Eqs. (5.2.5) and (5.2.6). Obtain initial conditions  $\mathbf{v}^0 = \mathbf{0}$  and  $\dot{\mathbf{v}}^0 = \mathbf{V}^T \dot{\mathbf{q}}^0$  from Eqs. (5.2.51). Evaluate  $\mathbf{B}^0$  of Eq. (5.2.12) and update  $\mathbf{B}$  in Eq. (5.2.20). Evaluate  $\mathbf{u} = \mathbf{h}(\mathbf{v}, t)$  of Eq. (5.2.14) and  $\mathbf{q}$  of Eq. (5.2.33). Evaluate  $\mathbf{D}$  of Eq. (5.2.24) and  $\dot{\mathbf{q}}$  of Eq. (5.2.34). Evaluate  $\gamma$  of Eq. (5.2.35).
- (2) Evaluate  $\mathbf{R}_{\dot{\mathbf{v}}, \lambda}$  of Eq. (8.6.5) and iteratively determine  $\ddot{\mathbf{v}}_n$  and  $\lambda_n$  in Eq. (8.6.6). Apply an explicit numerical integrator to determine  $\mathbf{v}_{n+1}$  and  $\dot{\mathbf{v}}_{n+1}$ . Use Eqs. (5.2.33) and (5.2.34) to determine  $\mathbf{q}_{n+1}$  and  $\dot{\mathbf{q}}_{n+1}$ .
- (3) Monitor the condition number of  $\mathbf{R}_{\dot{\mathbf{v}}, \lambda}$ , the norm of  $\mathbf{v}$ , and the number of iterations required to evaluate  $\ddot{\mathbf{v}}_i$ ,  $\mathbf{u}$ , and  $\mathbf{B}$ . If tolerances are exceeded, define a new time  $\bar{t}^0$  and associated  $\bar{\mathbf{q}}^0$  and  $\bar{\dot{\mathbf{q}}}^0$ . Repeat calculations in Step (1) to define a new parameterization and initial conditions  $\bar{\mathbf{v}}^0$  and  $\bar{\dot{\mathbf{v}}}^0$ . This process follows the trajectory shown in Fig. 5.2.2, moving smoothly across *charts* on the *constraint manifold*.
- (4) Continue the process until the final time  $tf$  is reached, or a singularity occurs.

While Lagrange multipliers are not used in application of the explicit integrator, they are computed in Eq. (8.6.6) and may be used to determine constraint reaction forces in the system.

### 8.6.3 Implicit Numerical Integration Algorithm

An approach for applying implicit Runge-Kutta methods, including the trapezoidal method, for solution of Eq. (8.6.2) has been suggested by Gear and Petzold (1984;1986). They show that direct application of Runge-Kutta methods for solution of *Index 1 DAE*, which includes Eq. (8.6.2), yields acceptable results. The approach is a formal; i.e., without rigorous justification, application of Runge-Kutta formulas to Eq. (8.6.2), similar to the direct introduction of trapezoidal formulas in Section 5.5.4. This is direct application of an ODE integrator to equations that are not ODE and, in the present case, not even equivalent to an ODE.

For the second order ODE of Eq. (4.8.13),

$$\ddot{\mathbf{v}} = \mathbf{g}(t, \mathbf{v}, \dot{\mathbf{v}}) \quad (8.6.7)$$

and the Runge-Kutta formula of Eq. (4.8.27) is

$$\mathbf{k}_i = \mathbf{g} \left( t_i + c_i h, \mathbf{v}_n + h c_i \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^i A_{ij} \mathbf{k}_j, \dot{\mathbf{v}}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}_j \right) \quad (8.6.8)$$

Defining *intermediate variables*  $\mathbf{v}_i$  and  $\dot{\mathbf{v}}_i$  (Petzold, 1986)

$$\begin{aligned} \mathbf{v}_i &\equiv \mathbf{v}_n + h c_i \dot{\mathbf{v}}_n + \sum_{j=1}^i A_{ij} \mathbf{k}_j \\ \dot{\mathbf{v}}_i &\equiv \dot{\mathbf{v}}_n + h \sum_{j=1}^i a_{ij} \mathbf{k}_j \end{aligned} \quad (8.6.9)$$

Eq. (8.6.8) is written as

$$\mathbf{k}_i = \mathbf{g}(t_n + c_i h, \mathbf{v}_i, \dot{\mathbf{v}}_i) \quad (8.6.10)$$

Using this formal manipulation, Eq. (8.6.2) is discretized as

$$\begin{aligned} \mathbf{R}_n &\equiv \mathbf{M}(\mathbf{q}_i) \mathbf{D}(\mathbf{q}_i, t_n + c_i h) \mathbf{k}_i + \Phi_q^T(\mathbf{q}_i, t_n + c_i h) \boldsymbol{\lambda}_i \\ &- \mathbf{M}(\mathbf{q}_i) \mathbf{U} \mathbf{B}(\mathbf{q}_i, t_n + c_i h) \gamma(\mathbf{q}_i, \dot{\mathbf{q}}_i) - \mathbf{S}(\mathbf{q}_i, \dot{\mathbf{q}}_i) - \mathbf{Q}^A(t_n + c_i h, \mathbf{q}_i, \dot{\mathbf{q}}_i, \boldsymbol{\lambda}_i, \mu) = \mathbf{0} \end{aligned} \quad (8.6.11)$$

where Eqs. (5.2.17) and (5.2.25) define  $\mathbf{q}_i$  and  $\dot{\mathbf{q}}_i$  as functions of  $\mathbf{v}_i$  and  $\dot{\mathbf{v}}_i$ ,

$$\begin{aligned} \mathbf{q}_i &= \mathbf{q}^0 + \mathbf{V} \mathbf{v}_i - \mathbf{U} \mathbf{h}(\mathbf{v}_i, t_n + c_i h) \\ \dot{\mathbf{q}}_i &= \mathbf{D}(\mathbf{q}_i, t_n + c_i h) \dot{\mathbf{v}}_i - \mathbf{U} \mathbf{B}(\mathbf{q}_i, t_n + c_i h) \Phi_t(\mathbf{q}_i, t_n + c_i h) \end{aligned} \quad (8.6.12)$$

With Eqs. (8.6.9) and (8.6.12), Eq.(8.6.11) is a system of equations in the unknown  $\mathbf{k}_i$  that may be solved numerically,  $i = 1, \dots, s$ . The approximate solution of Eq. (8.6.2) is, from Eq. (4.8.28),

$$\begin{aligned} \mathbf{v}_{n+1} &= \mathbf{v}_n + h \dot{\mathbf{v}}_n + h^2 \sum_{j=1}^s B_j \mathbf{k}_j \\ \dot{\mathbf{v}}_{n+1} &= \dot{\mathbf{v}}_n + h \sum_{j=1}^s b_j \mathbf{k}_j \end{aligned} \quad (8.6.13)$$

As in the Index 0 DAE formulation of Section 5.5, the Jacobian of the residual of Eq. (8.6.11) is  $\mathbf{J}_n^{RK} = [\partial \mathbf{R}_n / \partial \dot{\mathbf{v}} \quad \Phi_q^T]$ , where  $\partial \mathbf{R}_n / \partial \dot{\mathbf{v}}$  is given by Eqs. (5.5.34) and (5.5.17) and the iterative solution algorithm is

$$\begin{aligned} \mathbf{J}_n^{RK} \begin{bmatrix} \Delta \mathbf{k}_i^j \\ \Delta \boldsymbol{\lambda}_i^j \end{bmatrix} &= -\mathbf{R}(\mathbf{k}_i^j, \boldsymbol{\lambda}_i^j) \\ \begin{bmatrix} \mathbf{k}_i^{j+1} \\ \boldsymbol{\lambda}_i^{j+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{k}_i^j \\ \boldsymbol{\lambda}_i^j \end{bmatrix} + \begin{bmatrix} \Delta \mathbf{k}_i^j \\ \Delta \boldsymbol{\lambda}_i^j \end{bmatrix} \quad j=1, 2, \dots \text{ until } \|\mathbf{R}\| \leq \text{intol} \end{aligned} \quad (8.6.14)$$

for stages  $i = 1, \dots, s$ . The solution for  $\mathbf{v}_n$  and  $\dot{\mathbf{v}}_n$  is given by Eqs. (8.6.13) and for  $\mathbf{q}_n$  and its derivatives by Eqs. (5.2.33) through (5.2.35).

Finally, it is noted that with Eqs. (8.6.9) and (8.6.13) replaced by Eqs. (4.8.40) and with only a single stage equation of Eq. (8.6.11) at  $t_{n+1}$ , the trapezoidal method is obtained.

For a given application, a multibody model is defined by generalized coordinates  $\mathbf{q} = [\mathbf{q}_1^T \quad \dots \quad \mathbf{q}_{nb}^T]^T$ ; kinematic constraints of Eqs. (8.6.3); constraint Jacobian  $\Phi_q(\mathbf{q}, t)$  and derivative terms  $\mathbf{P}2(\mathbf{q}, \chi)$ ,  $\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}})$ , and  $\mathbf{P}4(\mathbf{q}, \lambda)$ ; kinetic terms  $\mathbf{M}(\mathbf{q})$ ,  $\mathbf{Q}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \lambda, \mu)$ , and  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$  and their derivatives  $\mathbf{M}2(\mathbf{q}, \mu)$ ,  $\mathbf{Q}_q^A(t, \mathbf{q}, \dot{\mathbf{q}}, \lambda, \mu)$ ,  $\mathbf{Q}_{\dot{\mathbf{q}}}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \lambda, \mu)$ ,  $\mathbf{Q}_{\lambda}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \lambda, \mu)$ ,  $\mathbf{S}_q(\mathbf{q}, \dot{\mathbf{q}})$ , and  $\mathbf{S}_{\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}})$ ; and coefficients of friction in the friction model of Section 8.1.2.2.

The numerical solution process, based on the Index 0 DAE formulation of Section 5.5, is the *implicit Index 0 DAE numerical integration algorithm with friction*, as follows

- (1) Define initial conditions  $\mathbf{q}^0$  and  $\dot{\mathbf{q}}^0$  at  $t_0$  that satisfy the first two of Eqs. (8.6.3). Evaluate matrices  $\mathbf{U}$  and  $\mathbf{V}$  that define the tangent space in the Index 0 DAE formulation of Section 5.5 and initial conditions on independent tangent space generalized coordinates  $\mathbf{v}^0$  and  $\dot{\mathbf{v}}^0$ .
- (2) At time step  $n$ , with known values of  $\mathbf{q}_{n-1}$  and  $\dot{\mathbf{q}}_{n-1}$  and estimates  $\mathbf{q} = \mathbf{q}_n$  and  $\dot{\mathbf{q}} = \dot{\mathbf{q}}_n$ , evaluate kinetic terms  $\mathbf{M}(\mathbf{q})$ ,  $\mathbf{Q}^A(t, \mathbf{q}, \dot{\mathbf{q}}, \lambda, \mu)$ , and  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$  and their derivatives with respect to  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\lambda$ . Evaluate derivatives  $(\Phi_q^T(\mathbf{q})\ddot{\lambda})_q$ ,  $\mathbf{P}2(\mathbf{q}, \chi) = (\Phi_q(\mathbf{q})\ddot{\chi})_q$ , and  $\mathbf{P}3(\mathbf{q}, \dot{\mathbf{q}}) = (\mathbf{P}2(\mathbf{q}, \dot{\mathbf{q}})\ddot{\dot{\mathbf{q}}})_q$  of constraints in Appendix 5.B. Apply an *implicit Runge-Kutta* or *trapezoidal* numerical integrator to the Index 0 DAE to determine  $\mathbf{k}_n$ ,  $\mathbf{v}_n$  and  $\dot{\mathbf{v}}_n$ , and evaluate solutions  $\mathbf{q}_n$ ,  $\dot{\mathbf{q}}_n$ , and  $\ddot{\mathbf{q}}_n$  using Eqs. (5.2.33) through (5.2.35). In the first time step, the parameterization of coefficients of friction with variable  $w$  presented in Section 8.8.1 is used to obtain initial conditions  $\dot{\mathbf{v}}_0$  and  $\lambda_0$ .
- (3) Monitor the condition number of the Jacobian  $\mathbf{J}_n$ , the number of Newton-Raphson iterations required in Step 2, the magnitude of  $\mathbf{v}$ , and the number of iterations required to evaluate tangent space integration variables. If tolerances are exceeded, define a new time  $\bar{t}_0$  and associated  $\bar{\mathbf{q}}^0$  and repeat calculations in Step 1 to define a new tangent space parameterization and initial conditions  $\bar{\mathbf{v}}^0$  and  $\dot{\bar{\mathbf{v}}}^0$ . Otherwise, continue with the same parameterization.
- (4) Continue the process until the final time  $tf$  is reached or a singularity is encountered.

Numerical integration of the Index 0 DAE of motion for systems with friction introduces challenges that are not encountered in systems without friction. First, the equations cannot be reduced to an ODE. Second, special care must be taken to determine initial values of accelerations and Lagrange multipliers that satisfy the equations of motion at the initial time, for use in iterative solution of the discretized equations of motion.

## 8.7 Code 8.7 for Simulation of Planar Systems with Friction

The *Index 0 DAE* components of planar tangent space multibody simulation Code 5.7 of Appendix 5.A are extended to account for friction in revolute and translational joints, using the formulation and derivatives presented in Section 8.4 and the solution algorithms of Section 8.6. The general-purpose MATLAB computer Code 8.7 of Appendix 8.A implements the tangent space Index 0 DAE formulations of Sections 5.2 and 5.5 for planar multibody systems, including the effects of friction in revolute and translational joints. All aspects of planar systems incorporated in Code 5.7, except noncentroidal coordinates, are included, with *friction generalized forces* and their derivatives of Section 8.4 and explicit and implicit numerical integration algorithms of Section 8.6 added. Explicit fixed time step Nystrom4 and variable time step RKFN45 algorithms and implicit variable time step trapezoidal and SDIRK54 algorithms are implemented for numerical integration of tangent space Index 0 DAE of *planar system dynamics with friction*.

Following an explanation of Code 8.7 in this section, numerical examples are presented in Section 8.8, including those treated in Section 5.8 without friction. To the author's knowledge, the friction representation presented in this chapter and included in Code 8.5 is not yet available in commercial dynamic simulation software.

### 8.7.1 User Components of Code

Components of Code 8.7 that interface with the user, with the exception of the *AppData Function* that includes definition of friction parameters, are identical to Code 5.7 and are presented in Section 5.7.1. Following a brief explanation of extensions of the AppData function, an outline of the body of the code, with which the user need not interact, is presented in Section 8.7.2.

Lines 14 through 23 that define the *Joint Data Table* in the AppData function of Fig. 5.7.3 for the three-body translating model are modified to include friction data in Fig. 8.7.1. As defined in line 15 of Fig. 8.7.1, three elements are added to define *friction data*, the radius  $R$  of revolute joints and static and dynamic coefficients of friction  $\mu_s = \text{mus}$  and  $\mu_d = \text{mud}$ , respectively. For the translational joint, the length of the key in body  $i$  that is  $d_i$  in Eq. (8.4.17) of the friction model is the length of vector  $\mathbf{v}'_i = \text{vpr}$  that defines orientation of the joint in body  $i$ . For the three-body translating model the AppData data set of Fig 8.7.1 includes the lengths of keys in bodies 1, 2, and 3 in translational joints 1, 2, and 3 that are defined in lines 20 through 22. Data in any field that are not required may be entered as zeros; e.g., the radius  $R$  in a translational joint in line 27. Note that Code 8.7, in its current form, is limited to models defined with centroidal coordinates.

```
4 if app==1 %Three Body Translating Model
5 nb=3; %Number of bodies
6 ngc=3*nb; %number of generalized coordinates
7 NTSDA=2; %Number of TSDA force elements
8 NRSDA=0; %Number of RSDA force elements
9
10 ux=[1;0];
11 uy=[0;1];
```

```

12 z2=zeros(2,1);
13
14 %PJDT(17,nh): Joint Data Table
15 %PJDT(:,k)=[t;i;j;sipr;sjpr;d;vipr;vjpr;R;mus;mud;ms;nm];
16 %k=joint No., t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
17 %sipr&sjpr=vectors to Pi&Pj, d=dist., vipr&vjpr=vectors along trans axis,
18 %length of vi is di in Eq. (8.4.17), R=rad Rev, mus&mud=FrCoefs,
19 %ms=Lagrange muplt.start address, nm=no. of mujlt.
20 PJDT(:,1)=[2;1;0;z2;z2;0;0.1*ux;uy;0.1;0.5;0.3;1;2];%Tran-Bod1toGrd
21 PJDT(:,2)=[2;2;0;z2;z2;0;0.1*ux;ux;0.1;0.5;0.3;3;2];%Tran-Bod2toGrd
22 PJDT(:,3)=[2;3;0;z2;z2;0;0.1*ux;ux;0.1;0.5;0.3;5;2];%Tran-Bod3toGrd
23 PJDT(:,4)=[3;1;2;z2;z2;5;z2;z2;zeros(3,1);7;1]; %Dist.-Bod1 to Bod2
24 nh=4; %Number of holonomic constraints
25 nc=7; %Number of constraint equations
26 nv=ngc-nc;
27 nu=nc;
28
29 %PMDT(2,nb) Mass Data Table(Centroidal coordinates)
30 %PMDT=[[m1;J1],[m2;J2],...,[mnb;Jnb]]
31 PMDT=[[6;1],[2;1],[2;1]];
32
33 %PTSDAT(10,NTSDA) TSDA Data Table
34 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
35 %i&j=bodies conn.,si&jpr=vectors to Pi&j, K=spring constant,
36 %C=damping coefficient,el0=spring free length,F=const. force
37 PTSDAT(:,1)=[1;0;z2;-10*uy;10;0;10;0]; %Bod1 to Grnd
38 PTSDAT(:,2)=[2;3;z2;10*ux;10;0;11;0]; %Bod2 to bod3
39
40 %PRSDAT(6,NRSDA): RSDA Data Table
41 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
42 %i&j=bodies connected, K=spring constant,
43 %C=damping coefficient,phi0=spring free angle,T=constant torque
44 PRSDAT=zeros(6,1);
45
46 %Initial generalized coordinates
47
48 q0=[0;0;pi/2;5*ux;0;6*ux;0];
49 qd0=[0;0;0;0;0;-1;0;0];
50
51 end

```

Figure 8.7.1 AppData Function, Three Body Translating Model

Lines 14 through 27 in the AppData function of Fig. 8.7.1 are code that define addresses of Lagrange multipliers in the array LLam in Code 8.7. The user need not modify these lines of code.

Figure 8.7.2 presents code in the main program that defines initial values of  $\dot{v} = vdd$  and  $\lambda = Lam$  that are required as initial estimates for iterative computation defined in Section 8.5.2, for use in both explicit and implicit integration algorithms with friction. The user may modify the value N in line 117 that defines the number of increments in Lagrange multipliers in the transition from zero coefficients of friction to the specified values in the AppData function of

Fig. 8.7.1. The default value 10 has been found adequate for applications treated to date, but may need to be increased if very large coefficients of friction are considered.

```

98 %Initial vdd and Lam Estimate with no friction
99 M=MEval(PMDT,par);
100 Gam=GamEval(0,q0,qd0,PJDT,par);
101 D=(eye(ncg)-U*B*Phiq)*V;
102 QA0=QAEval(0,q0,qd0,zeros(nc),PMDT,PJDT,LamInd,PTSDAT,PRSDAT,par);
103
104 EE=[M*D,Phiq'];
105 CondEE1=cond(EE);
106 RHS=M*U*B*Gam+QA0;
107 x=EE\RHS;
108 Pvdd=[eye(nv),zeros(nv,nc)];
109 PLam=[zeros(nc,nv),eye(nc)];
110 vdd=Pvdd*x;
111 Lam=PLam*x;
112 vdd0=vdd;
113 Lam0=Lam;
114
115 %Increment friction coefficients to obtain initial conditions on Lam and
116 %vdd
117 N=10;
118 w=1;
119 %QAwsLam=QAwsLamEval(q0,qd0,Lam,PJDT,LamInd,par,w,N);
120 while w<=N
121 [vdd,Lam,jodeiter,ECond]=FrODEfunct0w(q0,qd0,vdd,Lam, ...
122 V,U,B,PJDT,LamInd,PMDT,PTSDAT,PRSDAT,par,w,N);
123 Econd0(w)=ECond;
124 jodeiter0(w)=jodeiter;
125 Vvdd0(:,w)=vdd;
126 LLam0(:,w)=Lam;
127
128 w=w+1;
129 end
130
131 qdd=D*vdd-U*B*Gam;
132 Qdd(:,1)=qdd;
C:\Users\echau\Documents\CMMD\Chapter 8. Simulation of F...\AA_Planar_Ind0_Multibody_Sim_Friction.m Page
3
133 Vvdd(:,1)=vdd;
134 LLam(:,1)=Lam;

```

Figure 8.7.2 Calculation of Initial Values of  $\ddot{v} = vdd$  and  $\lambda = Lam$

### 8.7.2 Computational Components of Code

Computational components of the main code are essentially identical to those presented in Section 5.7.2, with two distinctions. First, only four integration options are supported for the Index 0 DAE formulation, two explicit and two implicit. Second, functions  $\mathbf{Q}^A = QAEval$  and  $\mathbf{Q}_q^A$  and  $\mathbf{Q}_{q\dot{q}}^A$  in  $QAsqqdEval$  define generalized force and its derivatives with respect to generalized coordinates that include the effects of friction that are defined in Section 8.4. In

addition, derivatives of generalized force with respect to Lagrange multipliers are defined in function  $\mathbf{Q}_\lambda^A = QAsLamEval$ .

Code 8.7 is an extension of Code 5.7 that includes effects of friction in revolute and translational joints. User supplied data are as in Code 5.7, with the exception of coefficients of static and dynamic friction, the radius of the revolute joint, and the length of the key in the translational joint. These data are provided in the AppData function.

Code that implements friction generalized forces defined in Section 8.4 is included in functions QAEval, QAsqqdEval, and QAsLamEval that need not be modified by the user.

## 8.8 Planar System Simulation with Friction Using Code 8.7

Five planar systems that have previously been modeled without friction are simulated with friction effects modeled. A key issue addressed is the presence of *stiffness* in the resulting equations of motion, as has been suggested in the literature, based on use of *DAE* in models with friction (Marques, Flores, Pimenta Claro, and Lankarani, 2016; Pennestri, Rossi, Salvini, and Valentini, 2016; Brown and McPhee, 2016). To resolve this issue, performance of explicit numerical integration algorithms in simulating systems with friction is considered. If explicit algorithms fail, the systems may indeed be stiff. If they succeed, the systems are clearly not stiff.

### 8.8.1 Three-Slider Mechanism

The *three-slider mechanism* shown in Fig. 8.8.1 is comprised of translational joints between each of the bodies and one of the axes in ground, a distance constraint between bodies one and two, and springs between body one and ground and between bodies two and three. It is identical to the model of Section 5.8.4 and similar to the *three-particle model with friction* of Section 8.1. The body fixed  $x'$  axis in each body is along the long dimension of the body. Vectors that define the joint between body one and ground are  $\mathbf{v}'_1 = \mathbf{u}_x$  and  $\mathbf{v}'_0 = \mathbf{u}_y$  and vectors that define the joints between bodies two and three and ground are  $\mathbf{v}'_2 = \mathbf{v}'_3 = \mathbf{u}_x$  and  $\mathbf{v}'_0 = \mathbf{u}_x$ . The length of the distance constraint is  $\ell = 5$  m, the lengths of keys in translational joints of bodies 1, 2, and 3 are 0.1 m, spring constants are  $k_1 = k_2 = 10$  N/m, masses and moments of inertia of the bodies are  $m_1 = 5$  kg,  $m_2 = m_3 = 2$  kg, and  $J_1 = J_2 = J_3 = 1$  kgm<sup>2</sup>, coefficients of friction are  $\mu_s = 0.5$  and  $\mu_d = 0.3$ , and initial conditions are  $\mathbf{q}_1 = [0 \ 0 \ \pi/2]^T$ ,  $\mathbf{q}_2 = [5 \ 0 \ 0]^T$ ,  $\mathbf{q}_3 = [6 \ 0 \ 0]^T$ ,  $\dot{\mathbf{q}}_1 = \dot{\mathbf{q}}_2 = \mathbf{0}$ , and  $\dot{\mathbf{q}}_3 = [-1 \ 0 \ 0]^T$ . The data set for simulation of this mechanism using Code 8.7 is presented in Fig. 8.8.2, which differs from that used in Code 5.7, as presented in Fig. 5.8.15, only by addition of friction data.

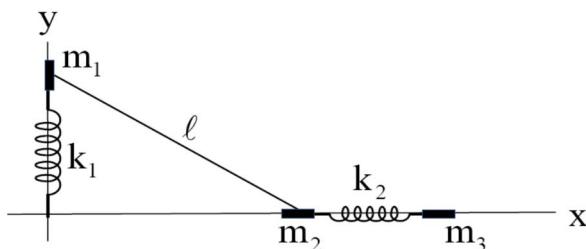


Figure 8.8.1 Three-Slider Mechanism

```

4 if app==1 %Three Body Translating Model
5 nb=3; %Number of bodies
6 ngc=3*nb; %number of generalized coordinates
7 NTSDA=2; %Number of TSDA force elements
8 NRSDA=0; %Number of RSDA force elements
9
10 ux=[1;0];
11 uy=[0;1];
12 z2=zeros(2,1);
13

```

```

14 %PJDT(17,nh): Joint Data Table
15 %PJTd(:,k)=[t;ij;sipr;sjpr;d;vipr;vjpr;R;mus;mud;ms;nm];
16 %k=joint No., t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
17 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis,
18 %length of vi is di in Eq. (8.4.17), R=rad Rev, mus&mud=FrCoefs,
19 %ms=Lagrange muplt.start address, nm=no. of mujlt.
20 PJDT(:,1)=[2;1;0;z2;z2;0;0.1*ux;uy;0.1;0.5;0.3;1;2]; %Tran-Bod1 to ground
21 PJDT(:,2)=[2;2;0;z2;z2;0;0.1*ux;ux;0.1;0.5;0.3;3;2]; %Tran-Bod2 to ground
22 PJDT(:,3)=[2;3;0;z2;z2;0;0.1*ux;ux;0.1;0.5;0.3;5;2]; %Tran-Bod3 to ground
23 PJDT(:,4)=[3;1;2;z2;z2;5;z2;z2;zeros(3,1);7;1]; %Dist.-Bod1 to Bod2
24 nh=4; %Number of holonomic constraints
25 nc=7; %Number of constraint equations
26 nv=ngc-nc;
27 nu=nc;
28
29 %PMDT(2,nb) Mass Data Table(Centroidal coordinates)
30 %PMDT=[[m1;J1],[m2;J2],...,[mnb;Jnb]]
31 PMDT=[[6;1],[2;1],[2;1]];
32
33 %PTSDAT(10,NTSDA) TSDA Data Table
34 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
35 %i&j=bodies conn., si&jpr=vectors to Pi&j, K=spring constant,
36 %C=damping coefficient, el0=spring free length, F=const. force
37 PTSDAT(:,1)=[1;0;z2;-10*uy;10;0;10;0]; %Bod1 to Grnd
38 PTSDAT(:,2)=[2;3;z2;10*ux;10;0;11;0]; %Bod2 to bod3
39
40 %PRSDAT(6,NRSDA): RSDA Data Table
41 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
42 %i&j=bodies connected, K=spring constant,
43 %C=damping coefficient, phi0=spring free angle, T=constant torque
44 PRSDAT=zeros(6,1);
45
46 %Initial generalized coordinates
47
48 q0=[0;0;pi/2;5*ux;0;6*ux;0];
49 qd0=[0;0;0;0;0;-1;0;0];
50
51 end

```

Figure 8.8.2 AppData Function, Three Slider Mechanism

Plots of position and velocity of each of the bodies predicted by Code 8.7 with the foregoing data are presented in Fig. 8.8.3. Note the *delay in onset of stiction* in this model, as occurred in the model problem of Section 8.1, Fig. 8.1.5. With an increased mass of body one to  $m_1 = 6\text{kg}$ , Fig. 8.8.4 shows that stiction occurs at the first possible time, as in Fig. 8.1.7 of Section 8.1. The magnitudes of mass that trigger the distinct modes of stiction response are different than in the particle model, as expected due to the fundamental difference in the particle and rigid body system models. Nevertheless, the qualitative behavior is identical.

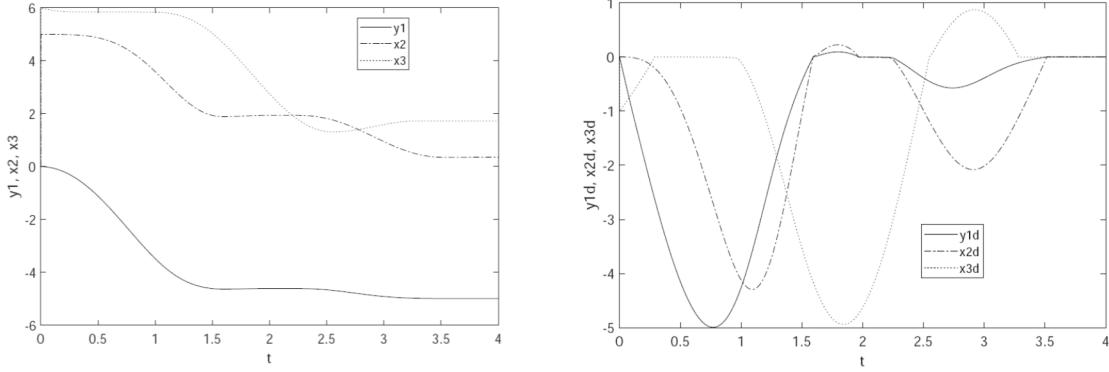


Figure 8.8.3 Position and Velocity,  $m_1 = 5 \text{ kg}$

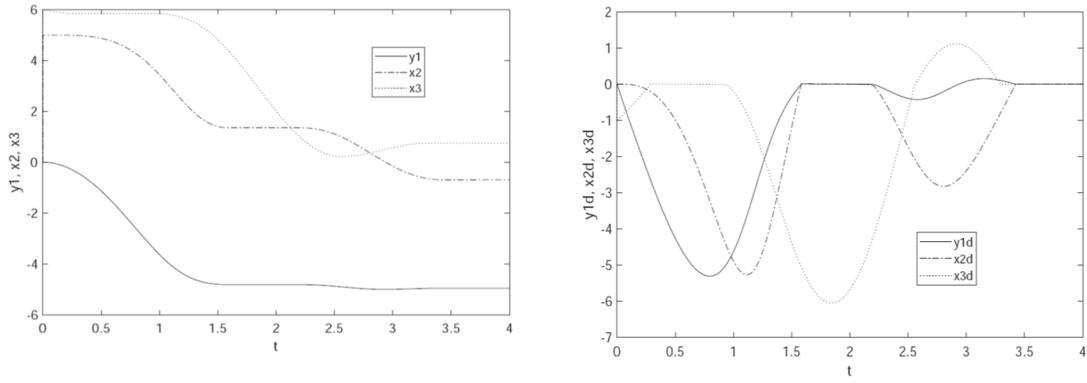


Figure 8.8.4 Position and Velocity,  $m_1 = 6 \text{ kg}$

As a check on validity of solutions obtained with the *trapezoidal* integration algorithm, maximum norms of position, velocity, and acceleration constraint errors were  $5\text{e-}15$ ,  $1.2\text{e-}14$ , and  $7\text{e-}14$ , respectively. Comparable results were obtained using the *SDIRK54* implicit integration algorithm and the *Nystrom4* and *RKFN45* explicit integration algorithms. Success in simulation with the explicit integration algorithms shows that the system is *not stiff*.

## 8.8.2 Slider-Crank

Dynamics of the two body model of the *slider-crank* mechanism of Fig. 8.8.5 that was analyzed kinematically in Section 3.2.3.3 and dynamically in Section 5.8.5 is analyzed here with the addition of friction. The radius of the crank is 1 m, masses and moments of inertia of bodies one and two are  $m_1 = 5 \text{ kg}$ ,  $J_1 = 5 \text{ kgm}^2$ ,  $m_2 = 1 \text{ kg}$ , and  $J_2 = 1 \text{ kgm}^2$ , coefficients of friction are  $\mu_s = 0.25$  and  $\mu_d = 0.2$ , and the length of the key in the translational joint and the radius of the revolute joint are 0.1 m. Initial conditions with  $\ell = 1.5 \text{ m}$  are  $\mathbf{q}_1 = [0 \ 0 \ 0]^T$ ,  $\mathbf{q}_2 = [2.5 \ 0 \ 0]^T$ ,  $\dot{\mathbf{q}}_1 = [0 \ 0 \ 100]^T$ , and  $\dot{\mathbf{q}}_2 = [0 \ 0 \ 0]^T$ . The data set for simulation of this mechanism with Code 8.7 is presented in Fig. 8.8.6.

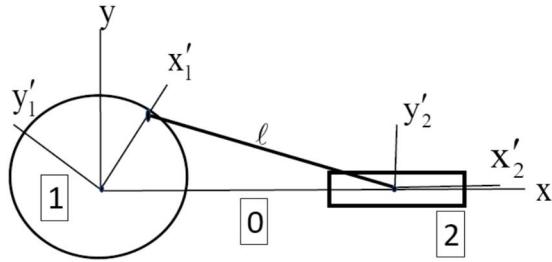


Figure 8.8.5 Slider-Crank

```

54 if app==2 %Slider-Crank
55
56 nb=2; %Number of bodies
57 ngc=3*nb; %number of generalized coordinates
58 NTSDA=0; %Number of TSDA force elements
59 NRSDA=0; %Number of RSDA force elements
60
61 ux=[1;0];
62 uy=[0;1];
63 z2=zeros(2,1);
64
65 %PJDT(17,nh): Joint Data Table
66 %PJTD(:,k)=[t;i;j;sipr;sjpr;d;vipr;vjpr;R;mus;mud;ms;nm];
67 %k=joint No., t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
68 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis,
69 %length of vi is di in Eq. (8.4.17), R=rad Rev, mus&mud=FrCoefs,
70 %ms=Lagrange muplt.start address, nm=no. of mujlt.
71 PJDT(:,1)=[1;1;0;z2;z2;0;z2;z2;0.1;0.25;0.2;1;2]; %Revolute-crank to ground
72 PJDT(:,2)=[2;2;0;z2;z2;0;0.1*ux;ux;0.1;0.25;0.2;3;2]; %Trans.-slider2 to ground
73 PJDT(:,3)=[3;1;2;ux;z2;1.25;z2;z2;zeros(3,1);5;1]; %Dist.-crank to slider2
74 nh=3; %Number of holonomic constraints
75 nc=5; %Number of holonomic constraint equations
76 nv=ngc-nc; %Number of independent coordinates
77 nu=nc; %Number of dependent coordinates
78
79 %PMDT(2,nb) Mass Data Table(Centroidal coordinates)
80 %PMDT=[[m1;J1],[m2;J2],...,[mnb;Jnb]]
81 PMDT=[[5;5],[1;1]];
82
83 %PTSDAT(10,NTSDA) TSDA Data Table
84 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
85 %i&j=bodies conn., si&jpr=vectors to Pi&j, K=spring constant,
86 %C=damping coefficient, el0=spring free length, F=const. force
87 PTSDAT=zeros(10,1);
88
89 %PRSDAT(6,NRSDA): RSDA Data Table
90 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=RSDA No.,
91 %i&j=bodies connected, K=spring constant,
92 %C=damping coefficient, phi0=spring free angle, T=constant torque
93 PRSDAT=zeros(6,1);
94

```

```

95 %Initial generalized coordinates
96
97 q0=[0;0;0;2.25*ux;0];
98 qd0=[0;0;100;0;0;0];
99
100 end

```

Figure 8.8.6 AppData Function, Two Body Slider-Crank

Plots of angular acceleration of the crank ( $\omega_{\text{crank}}$ ) and total energy divided by 20 ( $\text{TE}/20$ ) vs time for the basic connecting rod length ( $el$ ) 1.5 m and for 1.25 m and 1.15 m, obtained using Code 8.7 are presented in Fig. 8.8.7. Angular acceleration, hence constraint force, increases as connecting rod length is decreased toward the singular configuration  $l = 1$ . Accordingly, kinetic and total energy decrease 56%, 71%, and 79%, respectively, over the 0.5 sec simulation.

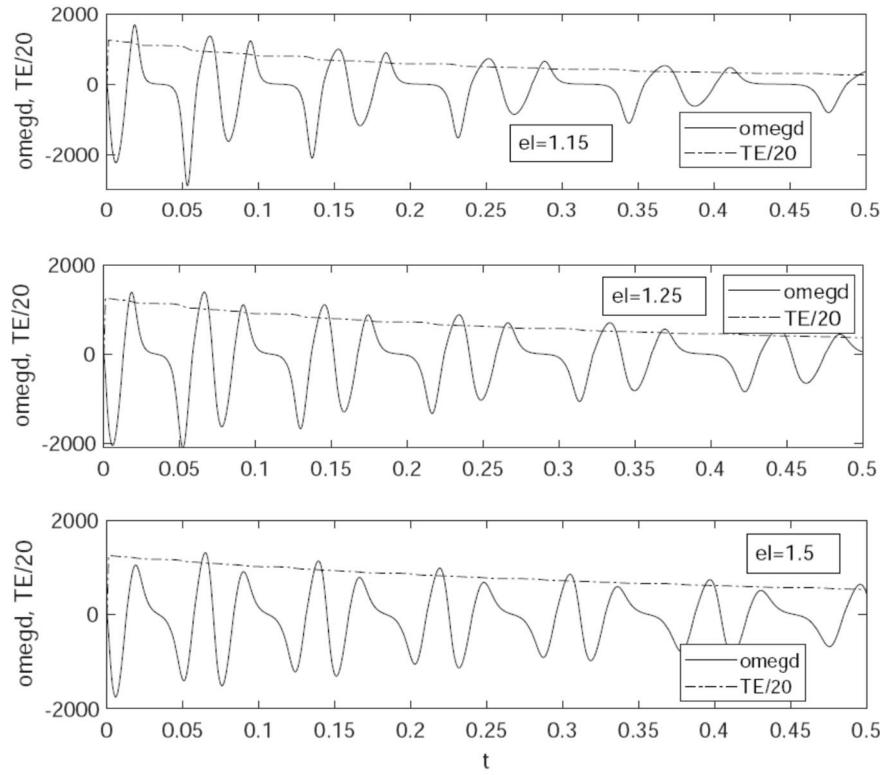


Figure 8.8.7 Crank Angular Acceleration and Total Energy/20 vs Time

As a check on validity of solutions obtained with the *trapezoidal* integration algorithm, maximum norms of position, velocity, and acceleration constraint errors are  $4e-15$ ,  $1e-12$ , and  $7e-10$ , respectively. Comparable results are obtained using the *SDIRK54* implicit integration algorithm and the *RKFN45* explicit integration algorithms. The constant step size Nystrom4 integration algorithm failed in all three simulations, likely due to the extreme accelerations involved. Success in simulation with the explicit RKFN45 integration algorithm shows that the system is not stiff.

### 8.8.3 Quick Return Mechanism

The *quick return mechanism* shown in Fig. 5.8.6 of Section 5.8.2 is a somewhat more complicated mechanism that is based on revolute, translational, and distance constraints. It undergoes large acceleration, hence large constraint forces that lead to significant *friction effects*. Data for the mechanism are as presented in Section 5.8.2. Rather than simulating the system with no external input and observing decay of motion, an *RSDA* is imposed on the crank with an applied 10,000 Nm torque that was arrived at to maintain approximately periodic motion, in the presence of *friction coefficients*  $\mu_s = 0.3$  and  $\mu_d = 0.25$  in all joints. The AppData function for simulation of the system, using Code 8.7 is provided in Fig. 8.8.8.

```

103 if app==3 %Quick Return
104
105 nb=4; %Number of bodies
106 nge=3*nb; %number of generalized coordinates
107 NTSDA=0; %Number of TSDA force elements
108 NRSDA=1; %Number of RSDA torque elements
109
110 ux=[1;0];
111 uy=[0;1];
112 z2=zeros(2,1);
113
114 %PJDT(17,nh): Joint Data Table
115 %PJTd(:,k)=[t;i;j;sipr;sjpr;d;vipr;vjpr;R;mus;mud;ms;nm];
116 %k=joint No., t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
117 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis,
118 %length of vi is di in Eq. (8.4.17), R=rad Rev, mus&mud=FrCoefs,
119 %ms=Lagrange muplt.start address, nm=no. of mujlt.
120 PJDT(:,1)=[1;1;0;-2*ux;z2;0;z2;z2;0.1;0.3;0.25;1;2]; %Revolute-bar to ground
121 PJDT(:,2)=[1;2;0;z2;2*uy;0;z2;z2;0.1;0.3;0.25;3;2]; %Revolute-crank to ground
122 PJDT(:,3)=[1;2;3;1.5*ux;z2;0;z2;z2;0.1;0.3;0.25;5;2]; %Revolute-crank to key
123 PJDT(:,4)=[2;3;1;z2;z2;0;0.2*ux;ux;0;0.3;0.25;7;2]; %Trans.-bar to key
124 PJDT(:,5)=[2;4;0;z2;4*uy;0;0.1*ux;ux;0;0.3;0.25;9;2]; %Trans.-cutter to ground
125 PJDT(:,6)=[3;1;4;2*ux;z2;2.5298;z2;z2=zeros(3,1);11;1]; %Dist.-bar to cutter
126
127 nh=6; %Number of holonomic constraints
128 nc=11; %Number of holonomic constraint equations
129 nv=nge-nc; %Number of independent coordinates
130 nu=nc; %Number of dependent coordinates
131
132 %PMDT(2,nb): Mass Data Table(Centroidal coordinates)
133 %PMDT=[[m1;J1],[m2;J2],...,[mn;Jn]]
134 PMDT=[[100;100],[1000;1000],[1;1],[50;50]];
135
136 %PTSDAT(10,NTSDA): TSDA Data Table
137 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
138 %i&j=bodies connected, si&jpr=vectors to Pi&j, K=spring constant,
139 %C=damping coefficient,el0=spring free length,F=constant force
140 PTSDAT=zeros(10,1);
141
142 %PRSDAT(6,NRSDA): RSDA Data Table

```

```

143 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=RSDA No.,
144 %i&j=bodies connected, K=spring constant,
145 %C=damping coefficient,phi0=spring free angle,T=constant torque
146 PRSDAT(:,1)=[2;0;0;0;0;5000];
147
148 %Initial generalized coordinates
149 q10=[1.2;1.6;0.9273];
150 q20=[0;2;0];
151 q30=[1.5;2;0.9273];
152 q40=[0;4;0];
153
154 q0=[q10;q20;q30;q40];
155 qd0=zeros(12,1); %Placeholder, qd0 calculated in main program
156
157 end

```

Figure 8.8.8 AppData Function, Quick Return Mechanism

Plots of the horizontal coordinate ( $x_4$ ) of the tool and angular velocity ( $\omega_2$ ) of the crank in the lower graph of Fig. 8.8.9 show that near steady state motion is achieved. The upper graph of the figure represents response to a torque of 5,000 Nm, which leads to a loss of 31% in kinetic energy over the 5 second simulation, 26% of the loss occurring in the first 2 seconds. Maximum norms of position, velocity, and acceleration constraint errors in these simulations, using the trapezoidal integration algorithm, were  $5.5e-15$ ,  $7e-14$ , and  $1.5e-12$ , respectively. Comparable results were obtained using the SDIRK54 implicit integration algorithm and the Nystrom4 and RKFN45 explicit integration algorithms, confirming that the system is not *stiff*.

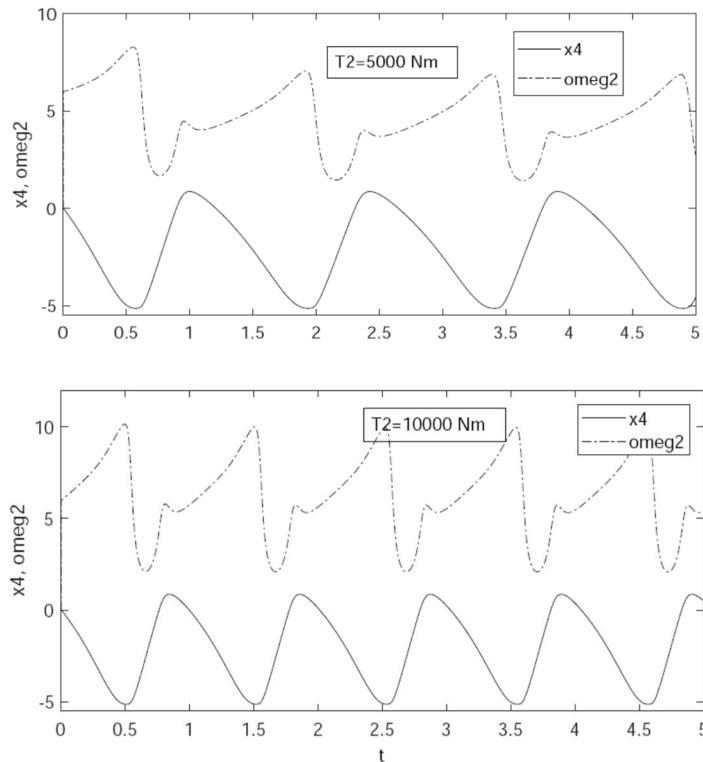


Figure 8.8.9 Motion of Crank and Body 4 with Varying Applied Torque

#### 8.8.4 Surge Waves in a Coil Spring

The *lumped mass spring model* of Section 5.8.3, dividing the spring into 5 equal lengths 0.2 m, each with stiffness  $k = 5K$ , and approximating the mass as 5 lumped masses,  $m = M/5$  each, is modified to include friction as the spring slides in a horizontal guide. For computation,  $M = 1 \text{ kg}$ ,  $K = 1000 \text{ N/m}$ , the spring has length  $L = 1 \text{ m}$ , and coefficients of friction  $\mu_s = 0.2$  and  $\mu_d = 0.19$  act between the spring masses and the horizontal guide. The initial position of mass  $i$  is  $x_i = 0.2i \text{ m}$ ,  $i = 1, \dots, 5$ . To study the effect of friction on surge wave behavior, motion of the system is initiated by giving mass 5 an initial velocity 1 m/sec in the negative x direction, which is imparted by impact with another body. The initial velocity of the first four masses is zero.

Kinematics of the system is defined by prescribing a translational joint between each of the masses and ground. Each spring is attached at the centers of mass of the bodies it connects. The mass of each body is  $m = 0.2 \text{ kg}$ , and each spring has spring rate  $k = 5000 \text{ N/m}$  and free length  $\ell_0 = 0.2 \text{ m}$ . Data for simulation of the system with Code 8.7 are given in the *AppData Function* in Fig. 8.8.10.

```

159 if app==4 %Lumped Mass Coil Spring-5 masses
160
161 nb=5; %Number of bodies
162 ngc=3*nb; %number of generalized coordinates
163 NTSDA=5; %Number of TSDA force elements
164 NRSDA=0; %Number of RSDA force elements
169
170 %PJDT(17,nh): Joint Data Table
171 %PJTd(:,k)=[t;i;j;sipr;sjpr;d;vipr;vjpr;R;mus;mud;ms;nm];
172 %k=joint No., t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
173 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis,
174 %length of vi is di in Eq. (8.4.17), R=rad Rev, mus&mud=FrCoefs,
175 %ms=Lagrange mult.start address, nm=no. of mult.
176 PJDT(:,1)=[2;1;0;z2;z2;0;0.1*ux;ux;0;0.2;0.19;1;2]; %Tran.-Body 1 to Ground
177 PJDT(:,2)=[2;2;0;z2;z2;0;0.1*ux;ux;0;0.2;0.19;3;2]; %Tran.-Body 2 to Ground
178 PJDT(:,3)=[2;3;0;z2;z2;0;0.1*ux;ux;0;0.2;0.19;5;2]; %Tran.-Body 3 to Ground
179 PJDT(:,4)=[2;4;0;z2;z2;0;0.1*ux;ux;0;0.2;0.19;7;2]; %Tran.-Body 4 to Ground
180 PJDT(:,5)=[2;5;0;z2;z2;0;0.1*ux;ux;0;0.2;0.19;9;2]; %Tran.-Body 5 to Ground
181
182 nh=5; %Number of holonomic constraints
183 nc=10; %Number of constraint equations
184 nv=ngc-nc;
185 nu=nc;
186
187 %PMDT=[[m1;J1],[m2;J2],...,[mn;Jn]] Mass Data Table(Centroidal coordinates)
188 PMDT(:,1)=[0.2;0.1];
189 PMDT(:,2)=[0.2;0.1];
190 PMDT(:,3)=[0.2;0.1];
191 PMDT(:,4)=[0.2;0.1];
192 PMDT(:,5)=[0.2;0.1];
193
194 %PTSDAT(10,NTSDA) Planar TSDA Data Table

```

```

195 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
196 %i&j=bodies conn.,si&jpr=vectors to Pi&j, K=spring constant,
197 %C=damping coefficient,el0=spring free length,F=const. force
198 PTSDAT(:,1)=[1;0;z2;z2;5000;0;0.2;0]; %Body 1 to Ground
199 PTSDAT(:,2)=[1;2;z2;z2;5000;0;0.2;0]; %Body 1 to 2
200 PTSDAT(:,3)=[2;3;z2;z2;5000;0;0.2;0]; %Body 2 to 3
201 PTSDAT(:,4)=[3;4;z2;z2;5000;0;0.2;0]; %Body 3 to 4
202 PTSDAT(:,5)=[4;5;z2;z2;5000;0;0.2;0]; %Body 4 to 5
203
204 %PRSDAT(6,NRSDA): RSDA Data Table
205 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
206 %i&j=bodies connected, K=spring constant,
207 %C=damping coefficient,phi0=spring free angle,T=constant torque
208 PRSDAT=zeros(6,1);
209
210 %Initial generalized coordinates
211
212 q0=[0.2;0;0;0.4;0;0;0.6;0;0;0.8;0;0;1;0;0];
213 qd0=[zeros(12,1);-1;0;0];

```

Figure 8.8.10 AppData Function, Five Lumped Mass Coil Spring Model

Numerical results of dynamic simulation carried out with Code 8.7 include plots of displacements  $\text{delxi} = x_i - x_i^0$ ,  $i = 1, \dots, 5$ , versus time for each lumped mass in Fig.

8.8.11. In comparison with results of Fig. 5.8.12 that were obtained for the same system without friction, there is a loss of 40% in total energy in a single pass of the wave, and the sharpness of wave behavior is diminished.

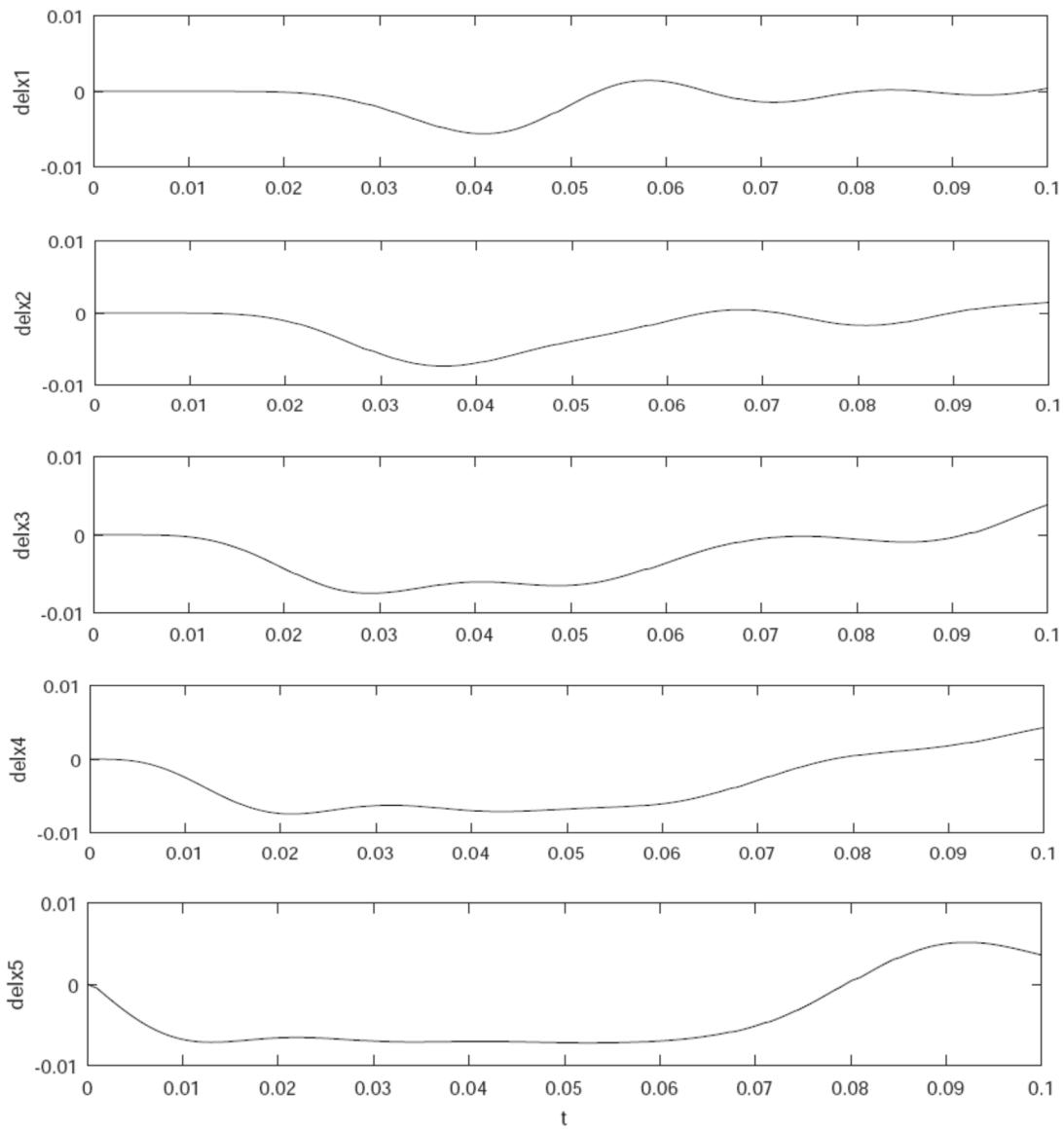


Figure 8.8.11 Lumped Mass Displacements Versus Time -5 Mass Model

### 8.8.5 Rotating Disk with Translating Body

The disk (body 1) in Fig. 8.8.12 rotates in ground (body 0) about the origin of the x-y global frame. Body 2 translates in body 1 along a guide that is parallel to the body fixed  $y'_1$  axis, one unit to the right of the  $y'_1$  axis. A TSDA with spring constant  $K = 10 \text{ N/m}$ , damping constant  $C = 0$ , and free length of  $10.1 \text{ m}$  acts between the origin of the  $x'_2$ - $y'_2$  reference frame and a point  $\mathbf{u}_x^{1'} + 10\mathbf{u}_y^{1'}$  in body 1. Static and dynamic coefficients are  $\mu_s = 0.06$  and  $\mu_s = 0.05$ , respectively, and masses and moments of inertia of the bodies are  $m_1 = 10 \text{ kg}$ ,  $J_1 = 10 \text{ kg} \cdot \text{m}^2$ ,  $m_2 = 5 \text{ kg}$ , and  $J_2 = 5 \text{ kg} \cdot \text{m}^2$ . Initial conditions for the system are  $\mathbf{q}_1^0 = [0 \ 0 \ 0]^T$ ,  $\dot{\mathbf{q}}_1^0 = [0 \ 0 \ 0]^T$ ,  $\mathbf{q}_2^0 = [1 \ 0 \ 0]^T$ , and

$\dot{\mathbf{q}}_2^0 = [0 \ 0 \ 0]^T$ . To simulate motion in a horizontal plane, the acceleration due to gravity is set to zero; i.e.,  $\mathbf{g} = 0$ .

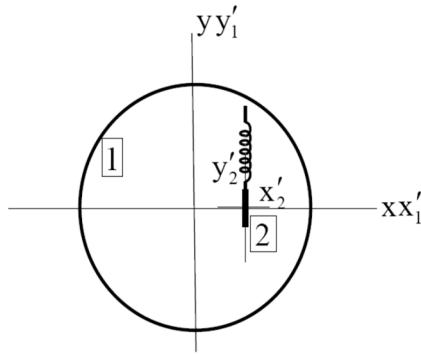


Figure 8.8.12 Rotating Disk with Translating Body

The *AppData Function* for simulation of this system using Code 8.7 is shown in Fig. 8.8.13. Simulations are carried out over a 10 sec interval, with plots of rotation of body 1,  $\phi_1 = \text{phi1}$ , and translation of body 2 relative to body 1,  $y'_2 = \text{dely2pr}$ , shown in Figs. 8.8.14 and 8.8.15. Identical results were obtained with all four integration formulations. For the smaller damping coefficients in Fig. 8.8.14, total energy decreased by 96% over the simulation and kinetic energy at the end of the simulation was 7.3e-4 Nm. For the larger damping coefficients in Fig. 8.8.15, total energy decreased by 99.7% and kinetic energy at the end was 9.8e-5. For larger coefficients of friction, kinetic energy reached near zero before the end of the simulation and integrators were unable to continue the simulation.

```

218 if app==5 %Rotating Disk with Translating Body
219
220 nb=2; %Number of bodies
221 ngc=3*nb; %number of generalized coordinates
222 NTSDA=1; %Number of TSDA force elements
223 NRSDA=0; %Number of RSDA force elements
224
225 ux=[1;0];
226 uy=[0;1];
227 z2=zeros(2,1);
228
229 %PJDT(17,nh): Joint Data Table
230 %PJTd(:,k)=[t;i;j;sipr;sjpr;d;vipr;vjpr;R;mus;mud;ms;nm];
231 %k=joint No., t=joint type(1=Rev,2=Tran,3=Dist), i&j=bodies conn.,
232 %si&jpr=vectors to Pi&j, d=dist., vi&jpr=vectors along trans axis,
233 %length of vi is di in Eq. (8.4.17), R=rad Rev, mus&mud=FrCoefs,
234 %ms=Lagrange mult.start address, nm=no. of mult.
235 PJDT(:,1)=[1;1;0;z2;z2;0;z2;z2;0.1;0.06;0.05;1;2]; %Revolute-1 to ground
236 PJDT(:,2)=[2;1;2;ux;z2;0;0.1*uy;uy;0.1;0.06;0.05;3;2]; %Trans.-1 to 2
237
238 nh=2; %Number of holonomic constraints
239 nc=4; %Number of holonomic constraint equations

```

```

240 nv=ngc-nc; %Number of independent coordinates
241 nu=nc; %Number of dependent coordinates
242
243 %PMDT(2,nb) Mass Data Table(Centroidal coordinates)
244 %PMDT=[[m1;J1],[m2;J2],...,[mnb;Jnb]]
245 PMDT=[[10;10],[5;5]];
246
247 %PTSDAT(10,NTSDA) TSDA Data Table
248 %PTSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F]; T=TSDA No.,
249 %i&j=bodies conn.,si&jpr=vectors to Pi&j, K=spring constant,
250 %C=damping coefficient,el0=spring free length,F=const. force
251 PTSDAT(:,1)=[1;2;ux+10*uy;z2;10;0;10.1;0];
252
253 %PRSDAT(6,NRSDA): RSDA Data Table
254 %PRSDAT(:,R)=[i;j;K;C;phi0;T]; R=TSDA No.,
255 %i&j=bodies connected, K=spring constant,
256 %C=damping coefficient,phi0=spring free angle,T=constant torque
257 PRSDAT=zeros(6,1);
258
259 %Initial generalized coordinates
260
261 q0=[0;0;0;ux;0];
262 qd0=[0;0;0;0;0];
263
264 end

```

Figure 8.8.13 AppData Function, Rotating Disk with Translating Body

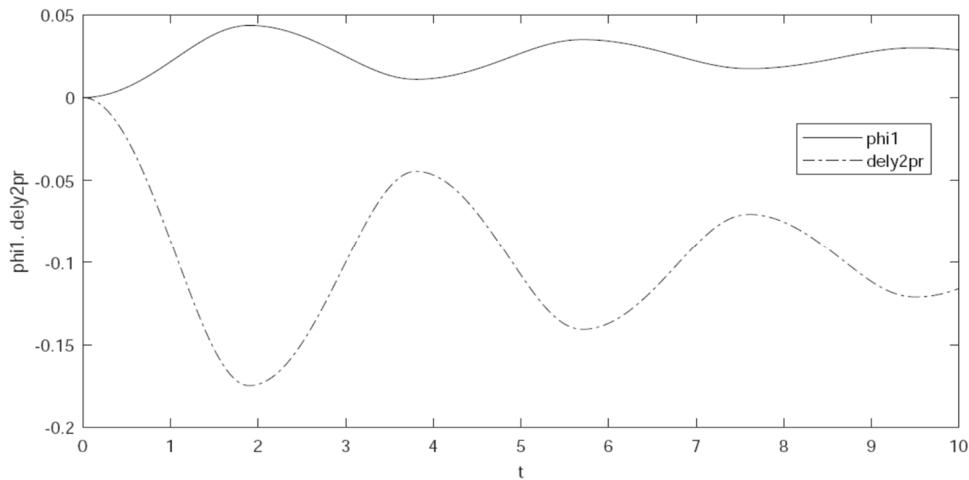


Figure 8.8.14  $\phi_1 = \text{phi1}$  and  $\delta y'_2 = \text{dely2pr}$ ,  $\mu_s = 0.06$  and  $\mu_d = 0.05$

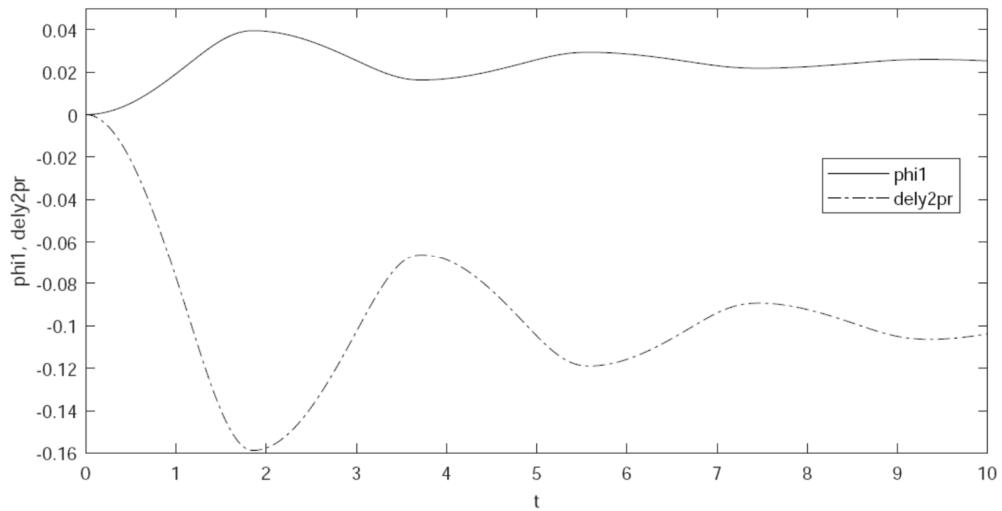


Figure 8.8.15  $\phi_1 = \text{phi1}$  and  $\delta y'_2 = \text{dely2pr}$ ,  $\mu_s = 0.08$  and  $\mu_d = 0.07$

The Index 0 DAE formulation and numerical integration algorithms presented for planar systems in Sections 8.4 and 8.6, implemented in Code 8.7, are shown to be effective in simulation of planar systems with friction. Of particular importance is the demonstrated ability of explicit integrators to solve the equations of motion. This demonstrates that friction effects do not necessarily imply that the resulting equations of multibody dynamics are stiff.

## 8.9 Code 8.9 for Simulation of Spatial Systems with Friction

The Index 0 DAE components of spatial tangent space multibody simulation Code 5.9 of Appendix 5.A are extended to account for friction in cylindrical, revolute, and translational joints, using the formulation and derivatives presented in Section 8.5 and the solution algorithms of Section 8.6. The general-purpose MATLAB computer Code 8.9 of Appendix 8.A implements the tangent space Index 0 DAE formulations of Section 5.5 for spatial multibody systems that are modeled with centroidal coordinates, including the effects of friction in cylindrical, revolute, and translational joints. All aspects of spatial systems incorporated in Code 5.9 are included, except noncentroidal coordinates, with *friction generalized forces* and their derivatives of Section 8.5 and explicit and implicit numerical integration algorithms of Section 8.6 added. Explicit fixed time step *Nystrom4* and variable time step *RKFN45* algorithms and implicit variable time step *trapezoidal* and *SDIRK54* algorithms are implemented for numerical integration of tangent space *Index 0 DAE* of spatial system dynamics with friction.

Following an explanation of Code 8.9 in this section, numerical examples are presented in Section 8.10, including those treated in Section 5.10, with friction effects added. To the author's knowledge, the friction representation presented in this chapter and included in Code 8.9 is not yet available in commercial dynamic simulation software.

### 8.9.1 User Components of Code

Components of Code 8.9 that interface with the user, with the exception of the *AppData Function* that includes definition of friction related parameters, are identical to Code 5.9 and are presented in Section 5.9.1. Following a brief explanation of extensions of the AppData function, an outline of the body of the code, with which the user need not interact, is presented in Section 8.9.2.

Lines 17 through 24 that define the *Joint Data Table* in the AppData function of Fig. 8.9.1 for the One Body Cylindrical Joint with Ground of Fig. 5.10.5 are modified from Fig. 5.10.6 to include friction related data. As defined in lines 17 through 23 of Fig. 8.9.1, six elements of data are added to define friction; radius R of revolute joint, dimension b of the translational joint, static and dynamic coefficients of friction  $\mu_s = \text{mus}$  and  $\mu_d = \text{mud}$ , and parameters ms and nm that define the address of the first *Lagrange multiplier* associated with the joint and the number of constraint equations for the joint, respectively. The last two parameters are used in friction force calculations to select the Lagrange multipliers that determine *friction forces* for the joint. Data in any field that are not required may be entered as zeros. Note that, in its present form, Code 8.9 is limited to use of centroidal coordinates and a diagonal inertia matrix.

```
10 if app==1 %Body in Cylindrical Joint with Ground
11
12 nb=1; %Number of bodies
13 ngc=7*nb; %number of generalized coordinates
14
15 NTSDA=1; %Number of TSDA force elements
16
17 %SJDT(28,nh): Joint Data Table
18 %SJTd(:,k)=[t;i;j;sipr;sjpr;d;vxipr;vzipr;vxjpr;vzjpr;a;b or R;mus;mud;mc;nm];
```

```

19 %k=joint No.; t=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
20 %6=Univ, 7=Strut, 8=Rev-Sph); i&j=bodies conn.,i>0;
21 %si&jpr=vectors to Pi&j; d=dist.; vxipr, vzipr, vxjpr, vzjpr;
22 %a,b or R=joint dimensions; mus,mud=coefficients of stat. &dyn. frict.
23 %ms=address of first Lag. Mult.,nm= No. of Constr. Eq.
24 SJDT(:,1)=[3;1;0;z3;z3;0;ux;uz;ux;uz;0.1;0.05;0.24;0.2;1;4];%Cyl - Body1 to Ground
25
26 nh=1; %Number of holonomic constraints
27 nhc=4; %Number of holonomic constraint equations
28 nc=nhc+nb; %Number of constraint equations
29 nv=ngc-nc;
30 nu=nc;
31
32 %SMDT(4,nb): Mass Data Table (Centroidal with diagonal inertia matrix)
33 %SMDT=[[m1;J11;J12;J13],...,[mnb;Jnb1;Jnb2;Jnb3]]
34 SMDT=[[1;0.1;0.1;0.1]];
35
36 %STSDAT(12,1): TSDA Data Table
37 if NTSDA==0
38 STSDAT=zeros(12,NTSDA);
39 end
40 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
41 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
42 %C=damping coefficient; el0=spring free length; F=const. force
43 STSDAT(:,1)=[1;0;uy+uz;-ux+uy+uz;10;0;1;0];
44
45 %Initial generalized coordinates
46 r10=[0;0;0];
47 p10=[1;0;0;0];
48 q0=[r10;p10];
49 r1d0=[0;0;0];
50 p1d0=0.5*EEval(p10)'*10*uz;
51 qd0=[r1d0;p1d0];
52
53 end

```

Figure 8.9.1 Friction in AppData Function for Body in Cylindrical Joint with Ground

Figure 8.9.2 presents code in the main program and function Ind0EC that defines initial values of  $\ddot{\mathbf{v}} = \mathbf{v}_{dd}$  and  $\lambda = \text{Lam}$  that are required as initial estimates for iterative computation defined in Section 8.5.2, for use in both explicit and implicit integration algorithms with friction. The user may modify the value N in line 117 that defines the number of increments in Lagrange multipliers in the transition from zero coefficients of friction to the specified values in the AppData function of Fig. 8.9.1. The default value 10 has been found adequate for applications treated to date, but may need to be increased if very large coefficients of friction are considered.

```

83 %Calculation of initial values of qdd and Lam to start integration
84 %Coefficients of friction are Indexed by w/N, w=0,1,...N in Function
85 %Ind0IC to calculate a sequence of qdd and Lam that converge to the
86 %desired coefficients of friction. The user may select N.
88 N=10;
89 [qdd,Lam,Qdd0,LLam0,w]=Ind0IC(q0,qd0,SMDT,SJDT,STSDAT,par,N);
91 Qdd(:,1)=qdd;

```

```

92 LLam(:,1)=Lam;

1 function [qdd,Lam,Qdd0,LLam0,w]=Ind0IC(q0,qd0,SMDT,SJDT,STSDAT,par,N)

3 [nb,ngc,nh,nc,nv,nu,g,utol,Btol,intol,Atol,Vtol,hvar,NTSDA,vt]=...
4 parPart(par);
5
6 %Evaluate Initial Acceleration and Lagrange Multipliers;
8 %Initial Parameterization
9 Phiq=PhiqEval(0,q0,SJDT,par);
10 U=Phiq';
11 B=inv(U'*U);
12 V=null(U');
13 v=zeros(nv,1);
14 vd=V'*qd0;
15 u=zeros(nu,1);
16 Vv(:,1)=v;
17 Vvd(:,1)=vd;
18 Uu(:,1)=u;
19
20 %Increment friction coefficients to obtain initial conditions on Lam and
21 %vdd
22 w=0;
23 Lam=zeros(nc,1);
24 vdd=zeros(nv,1);
25 while w<=N
26 [vdd,Lam,jodeiter,ECond]=FrODEfunct0w(q0,qd0,vdd,Lam,....
27 V,U,B,SJDT,SMDT,STSDAT,par,w,N);
28 w=w+1;
29 Gam=GamEval(0,q0,qd0,SJDT,par);
30 qdd=V*vdd-U*B*Gam;
31 Econd0(w)=ECond;
32 jodeiter0(w)=jodeiter;
33 Qdd0(:,w)=qdd;
34 LLam0(:,w)=Lam;
35 end

```

Figure 8.9.2 Calculation of Initial Values of  $\ddot{v} = vdd$  and  $\lambda = Lam$

### 8.9.2 Computational Components of Code

Computational components of the main code are essentially identical to those presented in Section 5.9.2, with two distinctions. First, only four integration options are supported for the Index 0 formulation. Second, functions  $\mathbf{Q}^A = QAEval$  and  $\mathbf{Q}_q^A$  and  $\mathbf{Q}_{\dot{q}}^A$  in  $QAsqqdEval$  define generalized force and its derivatives with respect to generalized coordinates that include the effects of friction defined in Section 8.5. In addition, derivatives of generalized force with respect to Lagrange multipliers are defined in function  $\mathbf{Q}_\lambda^A = QAsLamEva$ .

Code 8.9 is an extension of Code 5.9 that includes the effects of friction in cylindrical, revolute, and translational joints. User supplied data are as in Code 5.9, with the exception of coefficients of static and dynamic friction, the radius of the revolute joint, and the distance b in the translational joint. These data are provided in the AppData function.

Code that implements friction generalized forces that are defined in Section 8.5 is included in functions QAEval, QAsqqdEval, and QAsLamEval that need not be modified by the user.

## 8.10 Spatial System Simulation with Friction Using Code 8.9

Four spatial constrained systems are modeled using Code 8.9 of Appendix 8.A. Stiction is shown to occur in some examples.

### 8.10.1 Body in Cylindrical Joint with Ground

A single body  $i = 1$  shown in Fig. 8.10.1 is constrained to rotate and translate in a cylindrical joint, with axis along the global z axis in ground body  $j = 0$ . A spring connects point A (0,1,1) on the body with point B (-1,1,1) in ground. With the joint reference frame, initially at the origin of and aligned with the global x-y-z frame,  $\mathbf{s}_0^1 = \mathbf{s}_1^1 = \mathbf{0}$  and  $\mathbf{C}_0^1 = \mathbf{C}_1^1 = \mathbf{I}$ . With ground fixed in space,  $\mathbf{r}_0 = \mathbf{0}$  and  $\mathbf{p}_0 = [1 \ 0 \ 0 \ 0]^T$ . The initial position and orientation of body 1 are  $\mathbf{r}_1(0) = \mathbf{0}$  and  $\mathbf{p}_1(0) = [1 \ 0 \ 0 \ 0]^T$ . The dimensions of the cylindrical joint are

$a^1 = 0.1 \text{ m}$  and  $R^1 = 0.05 \text{ m}$ , the mass and moments of inertia of body 1 are 1 kg and  $0.1 \text{ kg}\cdot\text{m}^2$ , gravitational acceleration is  $9.8 \text{ m/sec}^2$  in the negative z direction, the free length of the spring is  $\ell_0 = 1$ , and the spring constant is  $k = 10 \text{ N/m}$ . Coefficients of translational and rotational friction are  $\mu_{dt} = \mu_{dr} = 0.2$  and  $\mu_{st} = \mu_{sr} = 0.24$ . The *AppData Function* in Code 8.9 for this system is presented in Fig. 8.9.1.

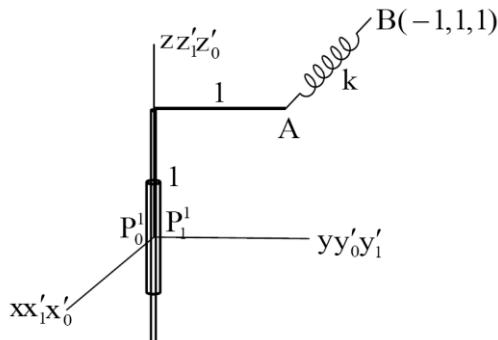


Figure 8.10.1 Body in Cylindrical Joint with Ground

Simulations are carried out with all four integration methods, using the initial configuration shown in Fig. 8.10.1,  $\dot{\mathbf{r}}_1(0) = \mathbf{0}$ , angular velocity  $\omega(0) = 10 \text{ rad/sec}$ , an integration step size of  $h = 0.0001 \text{ sec}$ , and  $v_t = 20h$  in Eq. (8.2.10). All four integration methods yielded essentially identical results. Using the trapezoidal implicit integrator, constraint reaction force components of Eq. (8.5.4), with  $f_p \equiv f'$ , are shown in Fig. 8.10.2. Vertical *friction force* and angular *friction torque* of Eqs. (8.5.12) and (8.5.15) are shown in Fig. 8.10.3. Vertical and angular velocity of the body are shown in Fig. 8.10.4. It is interesting to note that, even though the friction forces shown in Fig. 8.10.3 are approximately discontinuous, due to changes in sign of slip velocities, the constraint contact forces of Fig. 8.10.2 that lead to the friction forces are continuous.

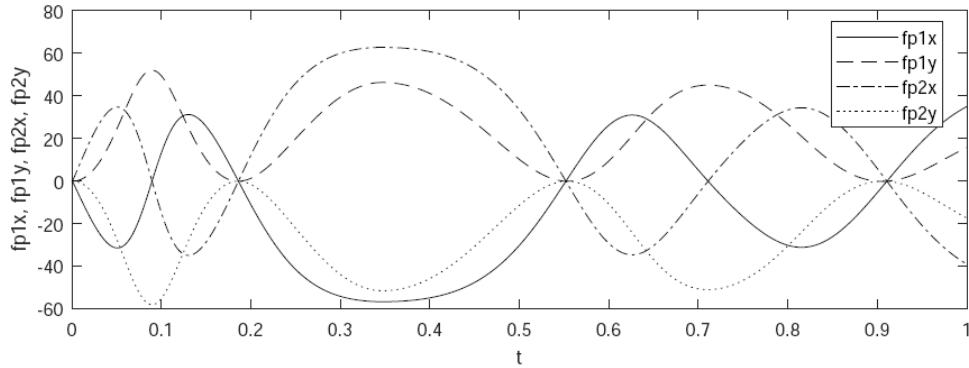


Figure 8.10.2 Components of Constraint Contact Force

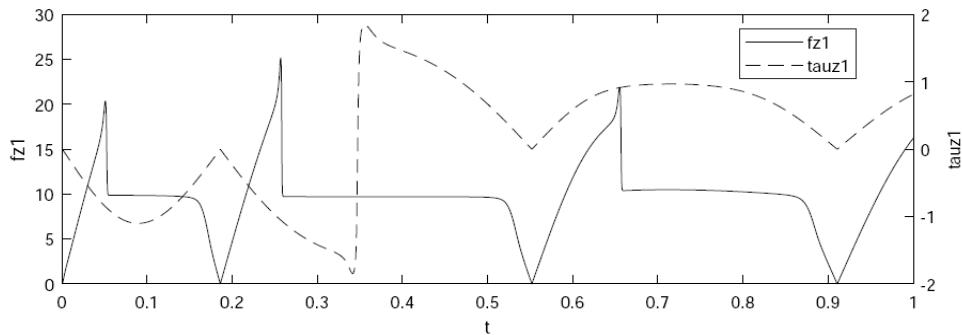


Figure 8.10.3 Vertical Friction Force and Friction Torque

What appear to be severe discontinuities in friction force and torque in Fig. 8.10.3 are in fact continuous when viewed on an expanded time scale. These forces are represented by the *continuous friction model*. As in model problems studied in Sections 8.1 through 8.3, the continuous formulation provides an effective method for modeling and simulation of dynamics with friction and stiction.

Another interesting behavior is the three *periods of stiction* in the vertical direction, shown by the plot of vertical velocity  $\dot{z}_1 = z_{1d}$  in Fig. 8.10.4, corresponding to periods of stiction force equal to the weight (9.8 N) of the body in Fig. 8.10.3. No stiction occurs in angular motion, which is possible with the cylindrical bearing model presented in Section 3.3.3 that decouples translational and rotational motion. Kinetic coupling of rotational and angular motion occurs as a result of interaction induced by the spring. If a design based on point contact at each end of the cylindrical sleeve in body i were adopted, friction forces would be coupled and this form of stiction could not occur.

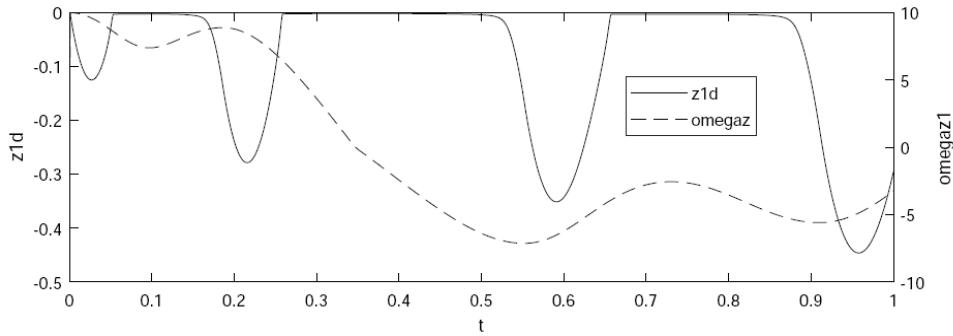


Figure 8.10.4 Vertical and Angular Velocity of Body 1

As a check on accuracy of simulation results, the equations of motion were satisfied to within an error tolerance of  $e\text{-}6$  and maximum position, velocity, and acceleration constraint error norms over the simulation period were  $1.5e\text{-}11$ ,  $2.5e\text{-}10$ , and  $e\text{-}7$ , respectively.

To see if the equations of motion with friction are stiff, the simulation is run with the *explicit Nystrom4 integrator* and a step size  $h = 0.001$ . The *condition number* of the ODE solver in explicit integration and the Jacobian in implicit integration varied between 6 and 15, and only one *reparameterization* was required in 10,000 time steps. Successful simulation with the explicit Nystrom4 integrator shows conclusively that there is *no stiffness* in the *Index 0 DAE* of motion, unlike indications of stiffness reported in the literature for the *Index 3 DAE* formulation with friction (Marques, Flores, Pimenta Claro, and Lankarani, 2016; Pennestri, Rossi, Salvini, and Velentini, 2016; Brown and McPhee, 2016). This is likely due to the fact that integrators in the Index 0 DAE formulation are applied to just two independent generalized coordinates, whereas Index 3 DAE integrators apply ODE integration formulas to all seven generalized coordinates.

### 8.10.2 Slider-Crank

The *spatial slider-crank* mechanism shown in Fig. 8.10.5 has been used as a computational example in kinematics and dynamics without friction in Sections 3.3.8.2 and 3.7.8.3 and with friction (Marques, Flores, Pimenta Claro, and Lankarani, 2016). It involves potentially large constraint contact forces and near *singular configurations* that may be dominated by *friction effects*. Body 1 rotates about an axis parallel to and offset from the global  $z$  axis in ground (body 0), with a crank in the  $x$ - $y$  plane of radius 0.08 m. The slider (body 2) translates in a guide along the global  $z$  axis in ground. It has an offset of length  $c_2$  in the positive  $y_2$  direction for distance constraint attachment, which increases constraint reaction forces in the translational joint. A distance constraint of length  $d^3$  connects the top of the offset in body 2 with the end of the crank in body 1. Data are given in the mks system of units.

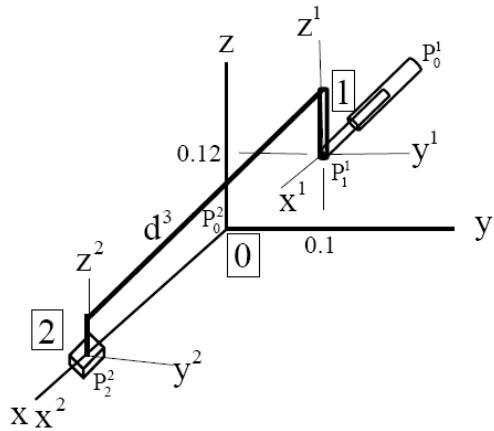


Figure 8.10.5 Spatial Slider-Crank

As in Section 5.10.4, since the  $x'$ - $y'$ - $z'$  reference frames in both bodies and ground are parallel, unit vectors used in definition of the joints are  $\mathbf{v}_x'^{ik} = \mathbf{u}_z$ ,  $\mathbf{v}_y'^{ik} = \mathbf{u}_y$ , and  $\mathbf{v}_z'^{ik} = \mathbf{u}_x$ ,  $i = 0, 1, 2$  and  $k = 1$  and  $2$ . For the revolute joint,  $\mathbf{s}_1^{i1} = \mathbf{0}$ ,  $\mathbf{s}_0^{i1} = [-1 \ 0.1 \ 0.12]^T$ ,  $R = 0.1$ ,  $\mu_s = 0.1$ , and  $\mu_d = 0.08$ . For the translational joint,  $\mathbf{s}_2^{i2} = \mathbf{s}_0^{i2} = \mathbf{0}$ ,  $\mu_s = 0$ , and  $\mu_d = 0$ . For the distance constraint,  $\mathbf{s}_1^{i3} = 0.08\mathbf{u}_z$ ,  $c_2 = 0.02$ ,  $\mathbf{s}_2^{i3} = c_2\mathbf{u}_y$ , and  $d^3 = 0.4$ . Masses and moments of inertia are  $m_1 = 0.5$  and  $J_1 = 0.2\mathbf{I}_3$  for body one and  $m_2 = 5$  and  $J_2 = 0.2\mathbf{I}_3$  for body two. Finally, initial conditions are  $\mathbf{r}_1^0 = [0, 0.1, 0.12]^T$ ,  $\mathbf{r}_2^0 = [0.3429, 0, 0]^T$ ,  $\mathbf{p}_1^0 = \mathbf{p}_2^0 = [1, 0, 0, 0]^T$ ,  $\dot{\mathbf{r}}_1^0 = [0, 0, 0]^T$ ,  $\dot{\mathbf{r}}_2^0 = [2.799, 0, 0]^T$ , and initial angular velocity of body 1 about the  $x$  axis is 120 rad/sec, so  $\dot{\mathbf{p}}_1^0 = 0.5\mathbf{E}(\mathbf{p}_1^0)^T 120\mathbf{u}_x$  and  $\dot{\mathbf{p}}_2^0 = [0, 0, 0, 0]^T$ . The values of  $z_2$  and  $\dot{z}_2$  are estimated and initial simulations run for  $tf = 0.002$ , adjusting the values until the variable arrays  $\mathbf{Q}$  and  $\mathbf{Qd}$  are consistent with initial conditions. Data for simulation using Code 8.9 are presented in the *AppData Function* of Fig. 8.10.6.

```

55 if app==2 %Spatial Slider-Crank
56
57 nb=2; %Number of bodies
58 ngc=7*nb; %number of generalized coordinates
59 NTSDA=0; %Number of TSDA force elements
60
61 %SJDT(28,nh): Joint Data Table
62 %SJDT(:,k)=[t;i;j;si(pr;sj(pr;d;vxipr;vzipr;vxjpr;vzjpr;a;b or R;mus;mud;mc;nm];
63 %k=joint No.; t=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
64 %6=Univ, 7=Strut, 8=Rev-Sph); i&j=bodies conn.,i>0;
65 %si&jpr=vectors to Pi&j; d=dist.; vxipr, vzipr, vxjpr, vzjpr;
66 %a,b or R=joint dimensions; mus,mud=coefficients of stat. &dyn. frict.
67 %ms=address of first Lag. Mult.;nm= No. of Constr. Eq.
68 SJDT(:,1)=[4;1;0;z3;0.1*uy+0.12*uz;0;uz;ux;uz;ux;0.1;0.1;0.1;0.08;1;5]; %Rev-Body1 to Ground
69 SJDT(:,2)=[5;2;0;z3;z3;0;uz;ux;uz;ux;0.1;0.1;0;0;6;5]; %Tran-Body2 to Ground
70 SJDT(:,3)=[1;1;2;0.08*uz;0.02*uz;0.3;z3;z3;z3;0.1;0.1;0.1;0.08;11;1];%Dist-Body 1 to 2

```

```

71
72 nh=3; %Number of holonomic constraints
73 nhc=11; %Number of holonomic constraint equations
74 nc=nhc+nb; %Number of constraint equations
75 nv=ngc-nc;
76 nu=nc;
77
78 %SMDT(4,nb): Mass Data Table (Centroidal with diagonal inertia matrix)
79 %SMDT=[[m1;J11;J12,J13],...,[mnb;Jnb1;Jnb2;Jnb3]]
80 SMDT=[[0.5;0.2;0.2;0.2],[5;0.2;0.2;0.2]];
81
82 %STSDAT(12,1): TSDA Data Table
83 if NTSDA==0
84 STSDAT=zeros(12,NTSDA);
85 end
86 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
87 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
88 %C=damping coefficient; el0=spring free length; F=const. force
89
90 %Initial generalized coordinates
91 r10=[0;0.1;0.12];
92 p10=[1;0;0;0];
93 r20=[0.2182;0;0];
94 p20=[1;0;0;0];
95 q0=[r10;p10;r20;p20];
96 r1d0=[0;0;0];
97 p1d0=0.5*EEval(p10)'*120*ux;
98 r2d0=[4.406;0;0];
99 p2d0=[0;0;0;0];
100 qd0=[r1d0;p1d0;r2d0;p2d0];
101
102 end

```

Figure 8.10.6 AppData Function, Spatial Slider-Crank

Numerical results for angular velocity  $\omega_{1x}$  and angular acceleration  $\omega_{1xd}$  of the crank for the data set in Fig. 8.10.6 with  $d^3 = 0.3$  (with  $x_2^0 = 0.2182$  and  $\dot{x}_2^0 = 4.406$ ) are shown in the bottom plots of Fig. 8.10.7. The effect of the relatively massive slider on angular velocity and acceleration is clear. Results for  $d^3 = 0.23$  (with  $x_2^0 = 0.1025$  and  $\dot{x}_2^0 = 9.369$ ) in the top plots show more extreme angular velocities and especially angular accelerations of the crank. This is due to their approach to a singular configuration at  $d^3 = 0.22$ . These results are similar to those of Fig. 5.10.10 of Section 5.10 for the same model without friction. The effects of friction are seen in the significant reduction in angular velocity and phase differences in both angular velocity and angular acceleration response.

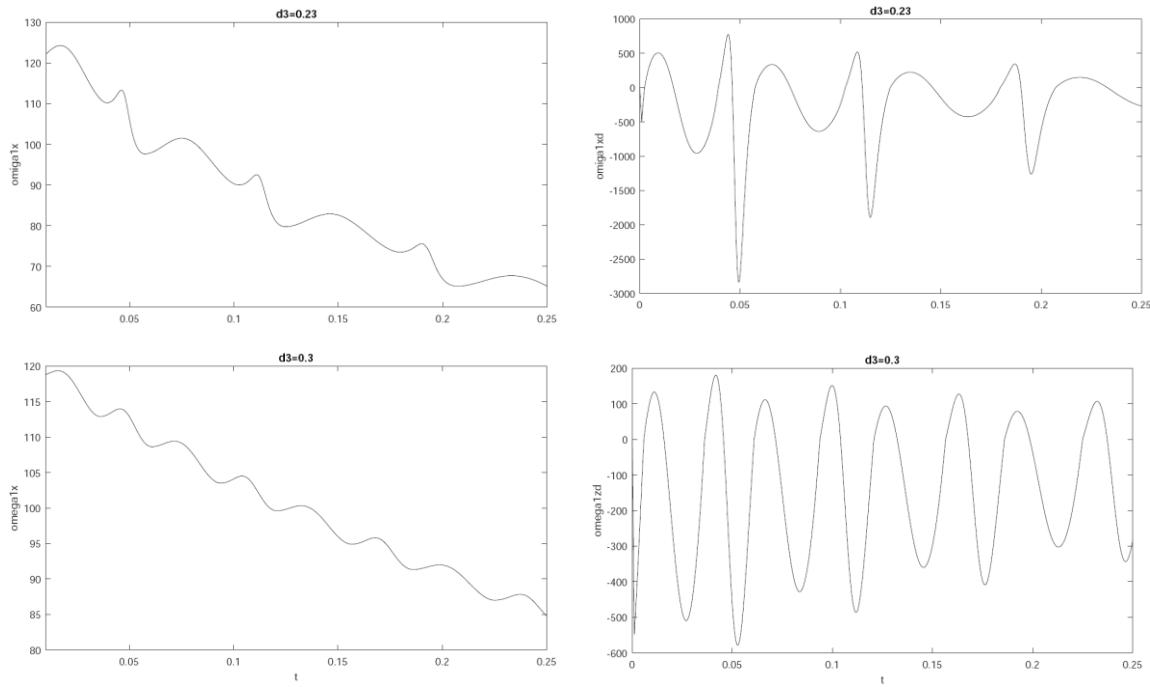


Figure 8.10.7  $\omega_{1x}$  and  $\omega_{1xd}$  vs  $t$  for Spatial Slider-Crank

As a result of the *near singular configuration* with  $d^3 = 0.23$ , constraint and friction forces are large, leading to *significant friction forces* and reduction in total energy. While increased numbers of iterations were required in this near singular simulation, which led to  $14 \times 14$  integration Jacobians with condition numbers in the range 10 to 100 and an average step size of  $5e-5$  sec, integration with the implicit trapezoidal method and Index 0 DAE formulation yielded reliable results. With an error tolerance in numerical integration of  $e-6$ , maximum constraint error norms over the simulation periods in position, velocity, and acceleration were  $5e-11$ ,  $3.5e-8$ , and  $2.5e-6$ , respectively. Even in this near singular situation only 9 *reparameterizations* were required in 5011 time steps (557 time steps per reparameterization).

Both successful simulations reported above with the *trapezoidal integrator* fail when run with the explicit integrators. This failure may occur due to the extreme nature of the dynamics, or as a result of stiff equations of motion. Since the trapezoidal integrator is not well suited to treat stiff systems, the system is *likely not stiff*, but a definitive judgment cannot be made.

### 8.10.3 Four Body Slider

A third example of friction and stiction simulation is the four mass spatial system shown in Fig. 8.10.8 that was treated by ad-hoc derivation and coding in Section 8.3.2. As in the planar three mass model problem, stiction will yield redundancy in this model. To see this, note that if the velocity of mass one is zero, so will be the velocity of mass two, due to the distance constraint between them. In turn, the velocity of mass 3 will be zero, due to the distance constraint between it and mass two. This yields a *third order stiction redundancy*.

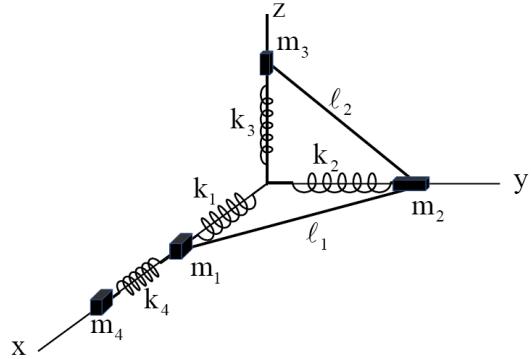


Figure 8.10.8 Four Mass Spatial Model Problem

Problem data are  $\ell_1 = 5 \text{ m}$ ,  $\ell_2 = 7 \text{ m}$ ,  $m_1 = m_2 = 2 \text{ kg}$ ,  $m_3 = m_4 = 6 \text{ kg}$ ,  $g = 9.8 \text{ m/sec}^2$ ,  $k_1 = k_2 = k_3 = 2 \text{ N/m}$ ,  $k_4 = 10 \text{ N/m}$ ,  $\mu_d = 0.13$ ,  $\mu_s = 0.2$ , and  $v_t = 10 \text{ h}$ . Initial conditions are  $\mathbf{q}^0 = [4 \ 0 \ 0 \ 0 \ 3 \ 0 \ 0 \ 6.32 \ 5]^T$  and  $\dot{\mathbf{q}}^0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1]^T$ . Translational joint dimensions for all four joints are  $a = b = 0.1 \text{ m}$ . Integration tolerances in MATLAB Code 8.9 of Appendix 8.A are  $\text{utol} = \text{Btol} = \text{e-8}$ ,  $\text{intol} = \text{e-6}$ , and  $\text{Maxv} = 0.7$ . The *AppData function* for the model is as shown in Fig. 8.10.9.

```

104 if app==3 %Four Body Slider
105
106 nb=4; %Number of bodies
107 nge=7*nb; %number of generalized coordinates
108 NTSDA=4; %Number of TSDA force elements
109
110 %SJDT(28,nh): Joint Data Table
111 %SJTd(:,k)=[t;i;j;sipr;jpr;d;vxipr;vzipr;vxjpr;vzjpr;a;b or R;mus;mud;mc;nm];
112 %k=joint No.; t=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,
113 %6=Univ, 7=Strut, 8=Rev-Sph); i&j=bodies conn.,i>0;
114 %si&jpr=vectors to Pi&j; d=dist.; vxipr, vzipr, vxjpr, vzjpr;
115 %a,b or R=joint dimensions; mus,mud=coeffidients of stat. &dyn. frict.
116 %ms=address of firsrt Lag. Mult.;nm= No. of Constr. Eq.
117 SJDT(:,1)=[5;1;0;z3;z3;0;uz;ux;uz;ux;0.1;0.1;0.14;0.12;1;5]; %Tran. - Body1 to Ground
118 SJDT(:,2)=[5;2;0;z3;z3;0;ux;uy;ux;uy;0.1;0.1;0.14;0.12;6;5]; %Tran. - Body2 to Ground
119 SJDT(:,3)=[5;3;0;z3;z3;0;ux;uz;ux;uz;0.1;0.1;0.14;0.12;11;5]; %Tran. - Body3 to Ground
120 SJDT(:,4)=[5;4;0;z3;z3;0;uz;ux;uz;ux;0.1;0.1;0.14;0.12;16;5]; %Tran. - Body3 to Ground
121 SJDT(:,5)=[1;1;2;z3;z3;5;z3;z3;z3;z3;0;0;0;0;21;1]; %Dist.-Bod1to2
122 SJDT(:,6)=[1;2;3;z3;z3;7;z3;z3;z3;z3;0;0;0;0;22;1]; %Dist.-Bod1to2
123
124 nh=6; %Number of holonomic constraints
125 nhc=22; %Number of holonomic constraint equations
126 nc=nhc+nb; %Number of constraint equations
127 nv=nge-nc;
128 nu=nc;
129
130 %SMDT(4,nb): Mass Data Table (Centroidal with diagonal inertia matrix)
131 %SMDT=[[m1;J11;J12;J13],...,[mnb;Jnb1;Jnb2;Jnb3]]
```

```

132 SMDT=[[2;1;1;1],[2;1;1;1],[6;1;1;1],[6;1;1;1]];
133
134 %STSDAT(12,1): TSDA Data Table
135 if NTSDA==0
136 STSDAT=zeros(12,NTSDA);
137 end
138 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
139 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
140 %C=damping coefficient; el0=spring free length; F=const. force
141 STSDAT(:,1)=[1;0;z3;-10*ux;2;0;13;0];
142 STSDAT(:,2)=[2;0;z3;-10*uy;2;0;14;0];
143 STSDAT(:,3)=[3;0;z3;-10*uz;2;0;13;0];
144 STSDAT(:,4)=[1;4;z3;10*ux;10;0;11;0];
145 %Initial generalized coordinates
146 r10=[4;0;0];
147 p10=[1;0;0;0];
148 q10=[r10;p10];
149 r1d0=[0;0;0];
150 p1d0=[0;0;0;0];
151 q1d0=[r1d0;p1d0];
152 r20=[0;3;0];
153 p20=[1;0;0;0];
154 q20=[r20;p20];
155 r2d0=[0;0;0];
156 p2d0=[0;0;0;0];
157 q2d0=[r2d0;p2d0];
158 r30=[0;0;8.32];
159 p30=[1;0;0;0];
160 q30=[r30;p30];
161 r3d0=[0;0;0];
162 p3d0=[0;0;0;0];
163 q3d0=[r3d0;p3d0];
164 r40=[5;0;0];
165 p40=[1;0;0;0];
166 q40=[r40;p40];
167 r4d0=[-1;0;0];
168 p4d0=[0;0;0;0];
169 q4d0=[r4d0;p4d0];
170 q0=[q10;q20;q30;q40];
171 qd0=[q1d0;q2d0;q3d0;q4d0];

```

Figure 8.10.9 AppData Function, Four Body Slider

Results of a simulation using Code 8.9 and the implicit *trapezoidal* integrator with the foregoing data are presented in Fig. 8.10.10, including a substantial period of *stiction* of masses 1, 2, and 3. Even though the geometry of joints and data in the present model and that of Section 8.3.2 are somewhat different, results presented in Figs. 8.3.2 and 8.10.10 are quite similar. Reducing the coefficients of friction slightly led to the velocity plots of Fig. 8.10.11, with the delay in initiation of stiction that has been seen earlier. A single parameterization sufficed for each of these simulations, and the maximum norms of position, velocity, and acceleration constraint errors over the simulation interval were 3.5e-12, 1.6e-10, and 2e-8, respectively.

Essentially identical results were obtained using the explicit *Nystrom4* integrator, indicating that the equations of motion are *not stiff*.

As a final note, it is interesting to compare the ease of use of Code 8.9 to the complexity of ad-hoc derivation of equations of motion and coding in Section 8.3.2 and Code 8.3.2.

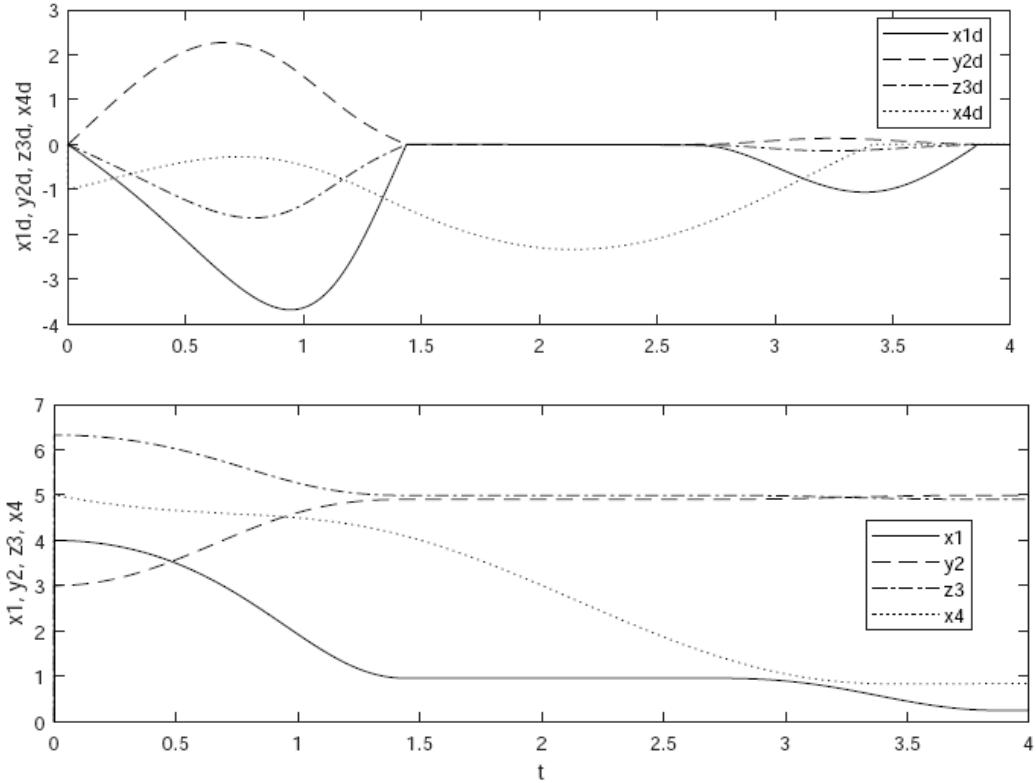


Figure 8.10.10 Positions and Velocities vs. Time,  $\mu_d = 0.12$ ,  $\mu_s = 0.14$

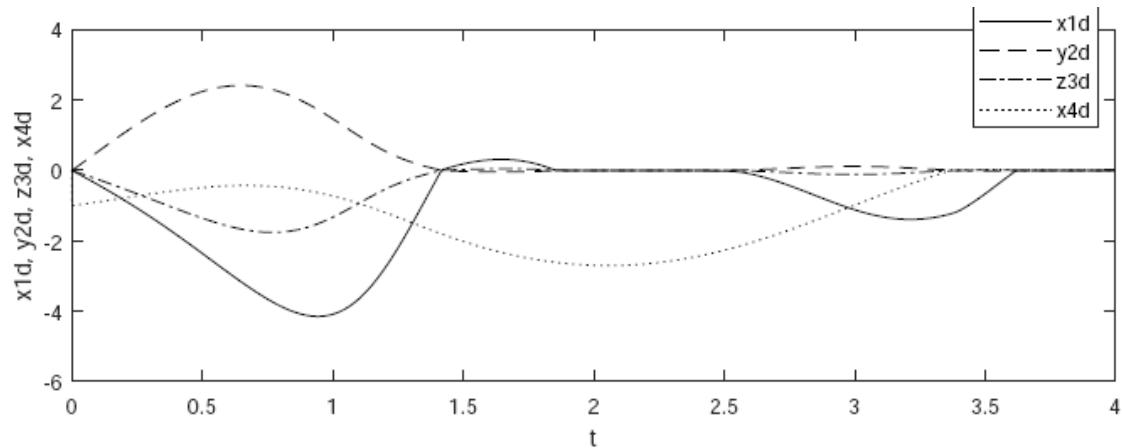


Figure 8.10.11 Velocities vs. Time,  $\mu_d = 0.1$ ,  $\mu_s = 0.12$

#### 8.10.4 Rotating Disk with Translating Body

The disk (body 1) in Fig. 8.10.12 rotates in a revolute joint relative to ground (body 0) about the global z axis. Body 2 translates in body 1 along a guide that is parallel to the body fixed  $y'_1$  axis, one unit to the right of the  $y'_1$  axis. A TSDA with spring constant  $K = 10 \text{ N/m}$ , damping constant  $C = 0$ , and free length  $\ell_0 = 10.1 \text{ m}$  acts between the origin of the  $x'_2-y'_2$  reference frame and a point  $\mathbf{u}_x^{1'} + 10\mathbf{u}_y^{1'}$  in body 1. Static and dynamic coefficients of friction are  $\mu_s = 0.05$  and  $\mu_d = 0.04$ , respectively and masses and moments of inertia of the bodies are  $m_1 = 10 \text{ kg}$ ,  $J_{aa} = 10 \text{ kg}\cdot\text{m}^2$ ,  $m_2 = 5 \text{ kg}$ , and  $J_{bb} = 5 \text{ kg}\cdot\text{m}^2$ . Dimensions of the joints are  $a = b = R = 0.1 \text{ m}$ . Initial conditions for the system are  $\mathbf{q}_1^0 = [0 \ 0 \ 0]^T$ ,  $\dot{\mathbf{q}}_1^0 = [0 \ 0 \ 0]^T$ ,  $\mathbf{q}_2^0 = [1 \ 0 \ 0]^T$ , and  $\dot{\mathbf{q}}_2^0 = [0 \ 0 \ 0]^T$ . To simulate motion in a horizontal plane, the acceleration due to gravity is set to zero; i.e.,  $g = 0$ .

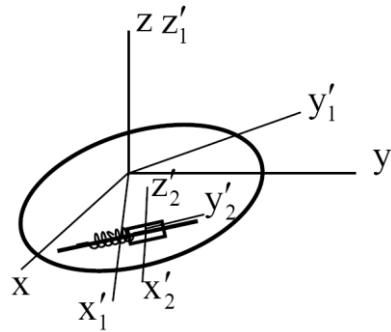


Figure 8.10.12 Rotating Disk with Translating Body

The *AppData Function* for simulation of this system using Code 8.9 is shown in Fig. 8.10.13. A simulation is carried out over a 10 sec interval, with plots of rotation theta1 of body 1 about the global z axis and translation of body 2 relative to body 1,  $y'_2 \equiv \text{dely2pr}$ , shown in Fig. 8.10.14. Identical results were obtained with all four integration formulations, indicating that the system is *not stiff*. Total energy is reduced by 96% over the simulation and the kinetic energy at the final time was  $7.3e-4 \text{ Nm}$ . A simulation with damping coefficients  $\mu_s = 0.08$  and  $\mu_d = 0.07$  lead to the more rapid damping shown in Fig. 8.10.15, a reduction of 99.9 % in total energy over the simulation and the kinetic energy at the final time was  $7.3e-4 \text{ Nm}$ . With larger coefficients of friction, the integrators failed when total energy approached zero.

```

175 if app==4 %Rotating Disk with Translating Body
176
177 nb=2; %Number of bodies
178 ngc=7*nb; %number of generalized coordinates
179
180 NTSDA=1; %Number of TSDA force elements
181
182 %SJDT(28,nh): Joint Data Table
183 %SJTd(:,k)=[t;i;j;sipr;sjpr;d;vxipr;vzipr;vxjpr;vzjpr;a;b or R;mus;mud;mc;nm];
184 %k=joint No.; t=joint type(1=Dist,2=Sph,3=Cyl, 4=Rev, 5=Tran,

```

```

185 %6=Univ, 7=Strut, 8=Rev-Sph); i&j=bodies conn.,i>0;
186 %si&jpr=vectors to Pi&j; d=dist.; vxipr, vzipr, vxjpr, vzjpr;
187 %a,b or R=joint dimensions; mus,mud=coeffidients of stat. &dyn. frict.
188 %ms=address of firstr Lag. Mult.;nm= No. of Constr. Eq.
189 SJDT(:,1)=[4;1;0;z3;0;ux;uz;ux;uz;0.1;0.1;0.08;0.07;1;5];%Rev - 1 to Ground
190 SJDT(:,2)=[5;1;2;ux;z3;0;ux;uy;ux;uy;0.1;0.1;0.08;0.07;6;5];%Tran. - 1 to 2
191
192 nh=2; %Number of holonomic constraints
193 nhc=10; %Number of holonomic constraint equations
194 nc=nhc+nb; %Number of constraint equations
195 nv=ngc-nc;
196 nu=nc;
197
198 %SMDT(4,nb): Mass Data Table (Centroidal with diagonal inertia matrix)
199 %SMDT=[[m1;J11;J12;J13],...,[mnb;Jnb1;Jnb2;Jnb3]]
200 SMDT=[[10;10;10;10],[5;5;5;5]];
201
202 %STSDAT(12,1): TSDA Data Table
203 if NTSDA==0
204 STSDAT=zeros(12,NTSDA);
205 end
206 %STSDAT(:,T)=[i;j;sipr;sjpr;K;C;el0;F];
207 %T=TSDA No.; i&j=bodies conn.;si&jpr=vectors to Pi&j; K=spring constant;
208 %C=damping coefficient; el0=spring free length; F=const. force
209 STSDAT(:,1)=[1;2;ux+10*uy;z3;10;0;10.1;0];
210
211 %Initial generalized coordinates
212 r10=[0;0;0];
213 p10=[1;0;0;0];
214 r20=[1;0;0];
215 p20=[1;0;0;0];
216 q0=[r10;p10;r20;p20];
217 r1d0=[0;0;0];
218 p1d0=[0;0;0;0];
219 r2d0=[0;0;0];
220 p2d0=[0;0;0;0];
221 qd0=[r1d0;p1d0;r2d0;p2d0];
222 end

```

Figure 8.10.13 AppData Function, Rotating Disk with Translating Body

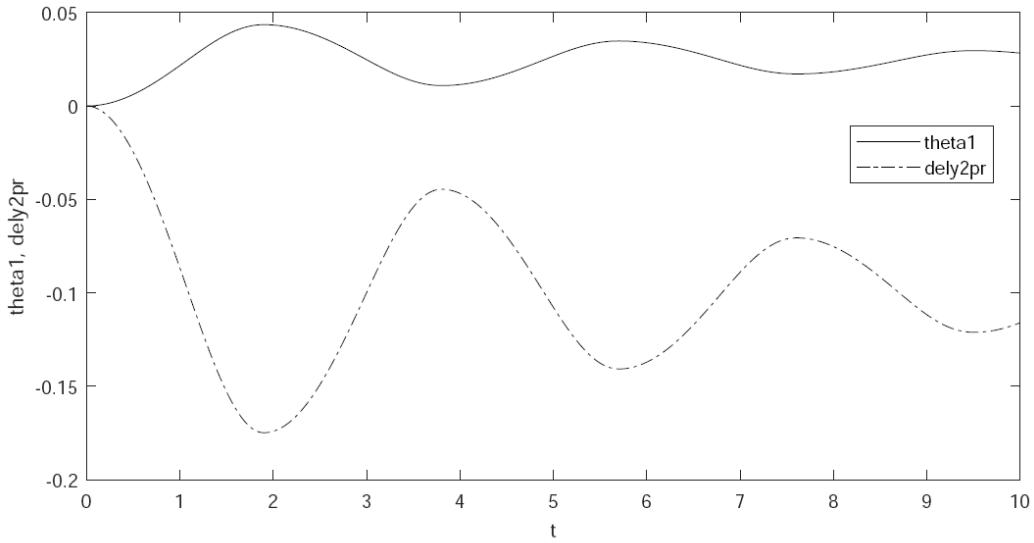


Figure 8.10.14  $\phi_1 = \text{phi1}$  and  $\delta y'_2 = \text{dely2pr}$ ,  $\mu_s = 0.05$  and  $\mu_d = 0.04$

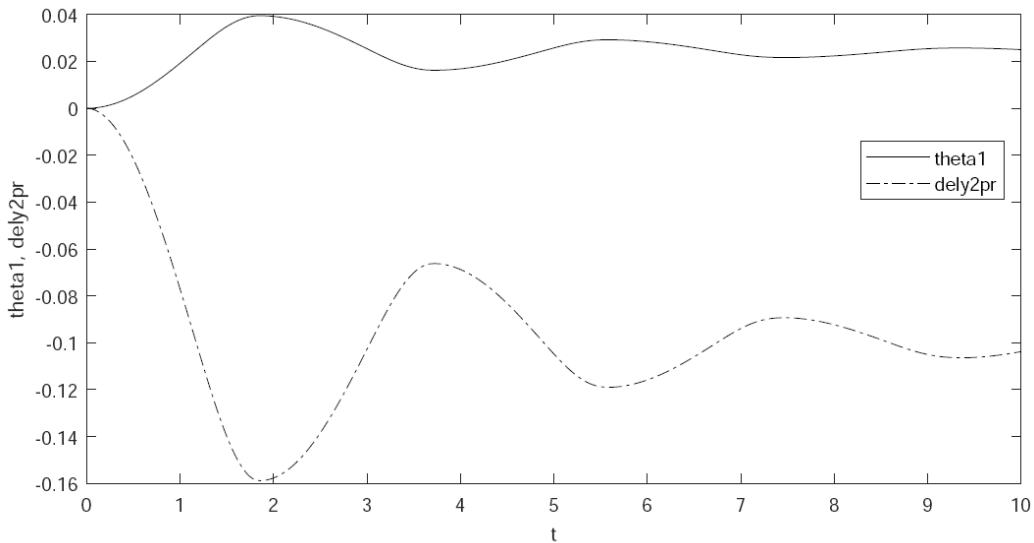


Figure 8.10.15  $\phi_1 = \text{phi1}$  and  $\delta y'_2 = \text{dely2pr}$ ,  $\mu_s = 0.08$  and  $\mu_d = 0.07$

Four spatial examples simulated using the Index 0 DAE formulation and Code 8.9, show that accurate results are obtained, including stiction that occurs in the first and third examples. The trapezoidal integration method provided reliable results in each of the examples. With careful selection of error control parameters, explicit integrators succeeded in each of the examples. This shows that multibody dynamics problems with friction are *not necessarily stiff*, contrary to what has been reported in the literature for Index 1 through 3 DAE formulations.

## **Appendix 8.A Tangent Space Index 0 Friction Simulation Code**

*Code 8.1 Three Particle, Minimal Coordinates*

*Code 8.2 Three Particle, Piecewise Coulomb*

*Code 8.3.1 Three Particle, Cartesian Coord*

*Code 8.3.2 Four Particle, Cartesian Coord*

*Code 8.7 Planar Multibody Simulation with Friction*

*Code 8.9 Spatial Multibody Simulation with Friction*

## Appendix 8.B Key Formulas, Chapter 8

$$F_f = -F_n(q, \lambda)S(v, \mu) \quad (8.1.9)$$

$$S(v, \mu) = \mu_d \tanh(4v/v_t) + (\mu_s - \mu_d)(v/v_t)/\left((v/2v_t)^2 + 3/4\right)^2 \quad (8.1.10)$$

$$M(q)\ddot{q} + \Phi_q(q)^T \lambda - Q^A(q, \dot{q}, \lambda) = 0 \quad (8.1.15)$$

$$\Phi(q) = 0 \quad (8.1.16)$$

$$\Phi_q \ddot{q} + \left(\Phi_q \hat{q}\right)_\sim \dot{q} \equiv \Phi_q \ddot{q} + \gamma = 0$$

## CHAPTER 9

### Manipulator Kinematics and Dynamics

This chapter deals with *manipulators* for which  $n$  *input coordinates* and  $m$  *output coordinates* are defined. *Forward kinematic mappings* from input to output coordinates and *inverse kinematic mappings* from output to input coordinates are analyzed, as are *differential equations of manipulator dynamics*. Part I of the chapter deals with *nonredundant manipulators* for which  $n = m$ . Part II deals with *redundant manipulators* for which  $n > m$ .

#### 9.0 Introduction to Part I, Nonredundant Manipulators

Using basic tools of *Euclidian space differential geometry*, *maximal singularity free components* of *regular manipulator configuration space* are defined, with conditions that establish the space as a *differentiable manifold*. This structure shows that the conventional categorization of nonredundant manipulators as either serial or parallel is incomplete and that four distinct categories of manipulator must be accounted for; (1) *serial manipulators* in which inputs globally determine outputs, (2) *explicit parallel manipulators* in which outputs globally determine inputs, (3) *implicit manipulators* in which there is no global input or output mapping, and (4) *compound manipulators* in which there is no global input or output mapping and in which generalized coordinates cannot be eliminated to form explicit input-output equations. Examples of each category of manipulator are presented in Section 9.1, followed by a summary of basic concepts of manipulator kinematics in Section 9.2 and analysis of each of the four categories in Sections 9.3 through 9.6. Results of *differential geometry* are used in Section 9.7 to show that *configuration space differentiable manifolds* in each manipulator category are partitioned into maximal, disjoint, path connected components in which the manipulator is singularity free and may be effectively controlled. This extends local analytical properties of manipulators that are used for analysis and control to global validity on maximal components of regular manipulator configuration space, providing explicit criteria for *avoidance of singular behavior* during system control. *Model manipulators* in each of the three categories are analyzed to illustrate application of the differentiable manifold structure, using only multivariable calculus and linear algebra.

Foundations for extending analytical kinematics to *dynamics of manipulators* are summarized in Section 9.8. Existence of forward and inverse configuration mappings throughout maximal singularity free manifold components is extended in Section 9.9, to obtain forward and inverse velocity mappings. Efficient computational algorithms for forward and inverse configuration and velocity analysis on a time grid are presented for each of the manipulator categories in Section 9.10. *Manifold parameterizations* derived in Sections 9.3 through 9.6 are used to transform variational equations of motion in manipulator coordinates to *second order ODE of manipulator dynamics* in Section 9.11, with both input and output coordinates as state variables, eliminating the need for ad-hoc derivation of equations of motion. Criteria that define manipulator differentiable manifolds are shown to guarantee that the *manipulator equations of motion are well posed* on maximal singularity free components of manipulator configuration space. It is shown that computation involved in evaluation of equations of manipulator kinematics and dynamics can be carried out in real-time on modern microprocessors, supporting in-line implementation of modern methods of manipulator control. This process is illustrated by presenting terms required for evaluation of equations of motion for four model manipulators in Section 9.12.

While analytical results are obtained for serial, explicit parallel, and implicit model manipulators of modes complexity, it is noted that complex industrial grade manipulators in each category may not yield easily to representations in the first three categories. It must be recalled that manipulators in these three categories are in fact special cases of compound manipulators. In case a complex manipulator that is technically in one of the first three categories cannot be practically treated analytically, it can be modelled as a compound manipulator and treated using tools available in that category. This reality is encountered in formulating ODE of manipulator dynamics, where the compound manipulator formulation is employed for all four categories in establishing ODE of dynamics. Attractive special features and computations in each of the first three categories can nevertheless be employed to enhance computational efficiency.

In Summary, the differentiable manifold structure established yields reliable algorithms for forward and inverse kinematics on singularity free maximal components of regular manipulator configuration space. Parameterizations of manifolds by input and output coordinates enable reduction of the manipulator variational equation of motion to well posed second order ODE, without ad-hoc derivation.

## 9.1 Characteristics of Nonredundant Manipulators

In multibody kinematics and dynamics treated thus far in the text, there has been no concept of input, output, or control. Nonredundant manipulators with *input coordinates*  $\mathbf{y} \in \mathbb{R}^k$  and *output coordinates*  $\mathbf{z} \in \mathbb{R}^k$  that are related in simple applications by equations of the form  $\mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0} \in \mathbb{R}^k$  require creation of a special differentiable manifold structure and associated parameterization to obtain ODE of dynamics. To avoid repetitive use of the qualifier *nonredundant*, it is suppressed and attention in Part I of the chapter is restricted to nonredundant manipulators with equal numbers of input and output coordinates. As illustrated schematically in Fig. 9.1.1, *manipulator kinematics* is based on *input coordinates* and *output coordinates* that require *forward kinematic mappings*  $\mathbf{z} = \mathbf{G}(\mathbf{y})$  and *inverse kinematic mappings*  $\mathbf{y} = \mathbf{H}(\mathbf{z})$ , in both analytical form and discretized on a time grid, and *differential equations of dynamics*  $\mathbf{M}(\mathbf{y})\ddot{\mathbf{y}} = \mathbf{Q}(\mathbf{y}, \dot{\mathbf{y}})$  and/or  $\mathbf{M}(\mathbf{z})\ddot{\mathbf{z}} = \mathbf{Q}(\mathbf{z}, \dot{\mathbf{z}})$  in support of *manipulator control*. It is shown that this additional structure defines a *differentiable manifold* in *manipulator configuration space*, with parameterizations by both input and output coordinates that enable systematic analytical and computational manipulator kinematic and dynamic analysis. This manifold is comprised of maximal, disjoint, path connected *components* that are singularity free; i.e., on which the manipulator is programmable and controllable. Analysis and computational methods that are made possible by this manifold structure are presented in this chapter for *real-time manipulator kinematics and dynamics*, as the basis for *in-line manipulator control* (Siciliano, Sciavicco, Villani, and Oriolo, 2010). Examples that exhibit properties of manipulators are presented in this section, to provide a basis for broadly applicable formulations and analysis methods presented in subsequent sections.

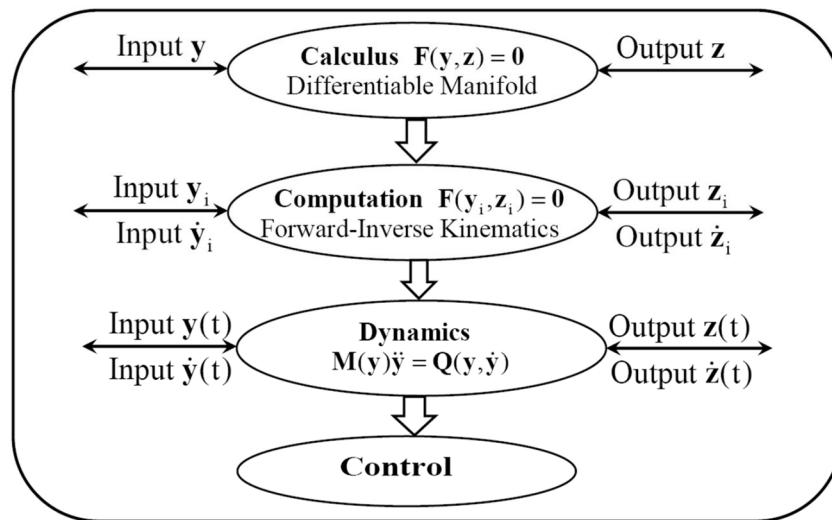


Figure 9.1.1 The Flow of Manipulator Kinematics, Dynamics, and Control

Rather than developing a manipulator differentiable manifold theory in the abstract and then applying it to specific manipulators, basic concepts are first established using only multivariable calculus for model manipulators and subsequently extended to differentiable

manifold structures in each of four distinct manipulator categories that are addressed. Treated in detail in Sections 9.3 through 9.6, these categories are (1) *serial manipulators* with *globally differentiable forward kinematics* and possibly singular inverse kinematics, (2) *explicit parallel manipulators* with *globally differentiable inverse kinematics* and possibly singular forward kinematics, (3) *implicit manipulators* that can be formulated in terms of only input and output coordinates and are neither serial nor explicit parallel, and (4) *compound manipulators* that are in none of the previous categories and require use of generalized coordinates in the kinematics formulation. Planar model manipulators are used to illustrate the formulation and results derived, avoiding the complexities of spatial kinematics.

### 9.1.1 Serial Manipulators

The *spatial serial manipulator* shown in Fig. 9.1.2 has seven actuated joints, with relative coordinates in joints between bodies that serve as inputs, denoted  $y_i = q_i$ ,  $i = 1, \dots, 7$ , including a gripper involving bodies 8 and 9. Six input coordinates,  $y_1$  through  $y_6$ , are to be chosen to control the position and orientation of the *end effector* (body 7) to which the gripper is attached. A simplified model *planar serial manipulator* with three actuated joints and input coordinates  $y_1$ ,  $y_2$ , and  $y_3$  is shown in Fig. 9.1.3. The three input coordinates are to be chosen to control the position and orientation of the outboard body, or *end-effector*. These manipulators enable explicit expressions for position and orientation of body  $i + 1$ , relative to body  $i$ , as a function of  $y_i$ , hence the designation serial. Such configurations are called *serial manipulators* in the literature and provide explicit analytical representation of output coordinates as functions of input coordinates,  $\mathbf{z} = \mathbf{G}(\mathbf{y})$ , called the *forward kinematic mapping*. There is, however, generally no global explicit representation of input coordinates as functions of output coordinates; i.e., *globally valid inverse kinematics*, for serial manipulators.

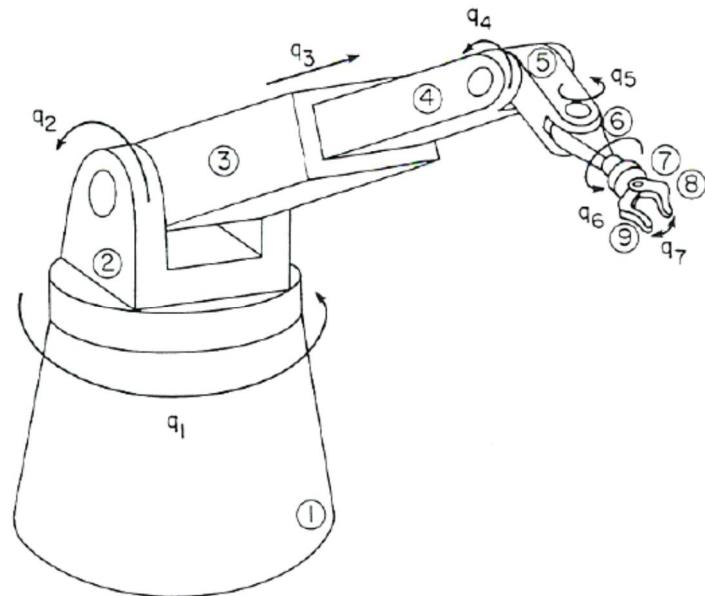


Figure 9.1.2 Spatial Serial Manipulator

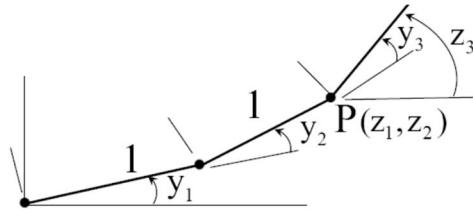


Figure 9.1.3 Planar Serial Manipulator

Equations that relate input and output coordinates for this planar serial manipulator are

$$\mathbf{F}(\mathbf{y}, \mathbf{z}) = \begin{bmatrix} \cos y_1 + \cos(y_1 + y_2) - z_1 \\ \sin y_1 + \sin(y_1 + y_2) - z_2 \\ y_1 + y_2 + y_3 - z_3 \end{bmatrix} = \mathbf{0} \quad (9.1.1)$$

This is an algebraic *input-output model* that analytically determines output coordinates as functions of input coordinates; i.e., the *forward kinematic mapping*

$$\mathbf{z} = \begin{bmatrix} \cos y_1 + \cos(y_1 + y_2) \\ \sin y_1 + \sin(y_1 + y_2) \\ y_1 + y_2 + y_3 \end{bmatrix} \equiv \mathbf{G}(\mathbf{y}) \quad (9.1.2)$$

The majority of manipulators treated in the robotics literature are serial in nature and yield to explicit representations of Eqs. (9.1.1) and (9.1.2). More realistic serial manipulators, such as that shown in Fig. 9.1.2 cannot be easily represented analytically in this simplified form and must be treated computationally.

### 9.1.2 Explicit Parallel Manipulators

The *driving simulator* spatial motion control system shown in Fig. 9.1.4, called a *Stewart platform* after its inventor (Merlot, 2006), uses six hydraulic actuators to control position and orientation of the *manipulator platform*, inside which a vehicle cab and computer graphics interact with the driver. In contrast with the serial nature of the manipulators shown in Figs. 9.1.2 and 9.1.3, there is not a sequence of bodies and associated relative coordinates in joints that control motion of the platform relative to ground. This is, therefore, not a serial manipulator. Such a configuration, with multiple serial chains of actuators that act in parallel between the fixed base and platform, is called a *parallel manipulator* (Merlet, 2006). In the case of the Stewart platform, each of the six parallel serial chains that acts between the base and platform consists of a single hydraulic actuator. The platform has six degrees of freedom in space, which are controlled by six *hydraulic actuators*.



Figure 9.1.4 Spatial Stewart Platform in Driving Simulator Application

Among the reasons that parallel manipulators such as the Stewart platform are used in applications, in spite of their analytical complexity, is their ability to maintain precision control of the geometry of the system and to support extreme loads that occur during motion of massive components, such as the driving simulator platform.

A much simpler planar parallel manipulator with input and output coordinates is the *planar platform* of Fig. 9.1.5 that is constrained so that its operating point O slides in a vertical slot in the global frame. This yields two kinematic degrees of freedom of the platform and a vector  $\mathbf{y} = [y_1 \ y_2]^T \in \mathbb{R}^2$  of *input coordinates*  $y_i > 0$  that control distances between points  $A_1$  and  $A_2$  in ground and  $B_1$  and  $B_2$  on the platform, with *output coordinates*  $\mathbf{z} = [z_1 \ z_2]^T \in \mathbb{R}^2$ .

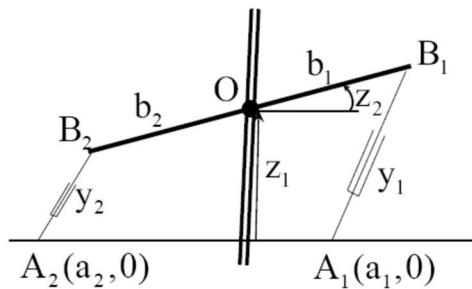


Figure 9.1.5 Two Degree of Freedom Planar Parallel Manipulator

Equations that relate input  $\mathbf{y}$  and output  $\mathbf{z}$  are

$$\mathbf{F}(\mathbf{y}, \mathbf{z}) = \begin{bmatrix} ((b_1 \cos z_2 - a_1)^2 + (z_1 + b_1 \sin z_2)^2 - y_1^2)/2 \\ ((-b_2 \cos z_2 - a_2)^2 + (z_1 - b_2 \sin z_2)^2 - y_2^2)/2 \end{bmatrix} = \mathbf{0} \quad (9.1.3)$$

This is an algebraic *input-output model* that analytically determines input coordinates as functions of output coordinates; i.e., the *globally valid inverse kinematic mapping*

$$\mathbf{y} = \begin{bmatrix} \sqrt{(b_1 \cos z_2 - a_1)^2 + (z_1 + b_1 \sin z_2)^2} \\ \sqrt{(-b_2 \cos z_2 - a_2)^2 + (z_1 - b_2 \sin z_2)^2} \end{bmatrix} \equiv \mathbf{H}(\mathbf{z}) \quad (9.1.4)$$

There is, however, no global analytical solution for output as a function of input; i.e., no global *forward kinematic mapping*.

These are called *explicit parallel manipulators*, since they possess explicit inverse kinematic mappings.

### 9.1.3 Implicit Manipulators with no Global Forward or Inverse Kinematic Mapping

The planar *parallel manipulator* shown in Fig. 9.1.6 consists of a platform with ends  $B_1$  and  $B_2$  and three degrees of freedom moving in the plane. A *serial chain* on the left with a single degree of freedom strut of length  $y_3 > 0$  as input and a serial chain on the right comprised of two bars with a pair of angles  $y_1$  and  $y_2$  in revolute joints as inputs support the platform.

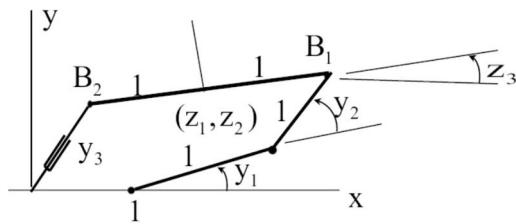


Figure 9.1.6 Three Input-Three Output Parallel Manipulator

To see geometrically that this is not an explicit parallel manipulator, note that the pair of configurations of the manipulator shown in Fig. 9.1.7 have the same output, but two distinct inputs. Thus, there is no globally valid inverse kinematic mapping, so this is not an explicit parallel manipulator.

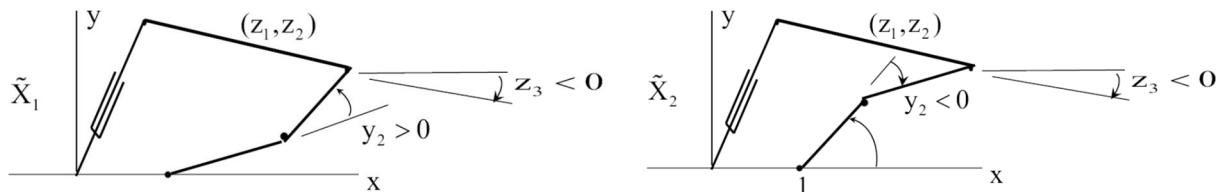


Figure 9.1.7 Parallel Manipulator with Multiple inputs for the Same Output

There are numerous parallel manipulators that fail to have explicit inverse kinematic mappings and hence are more challenging to control than their explicit counterparts. They form the category called *implicit manipulators*.

### 9.1.4 Compound Manipulators

A large class of manipulator applications that involve heavy loads include *material handling equipment* such as the *front-end loader* shown in Fig. 9.1.8 and the *excavator* in Fig. 9.1.9. Such systems consist of multiple moving components, whose configuration is controlled by *hydraulic actuators*. Both of these systems have *closed kinematic loop linkages* that support extreme loads on their buckets as they cut through earth during the loading operation.

*Manipulator kinematic generalized coordinates* are required to characterize these closed loop structures and cannot be analytically eliminated from the model. These are therefore neither serial, explicit parallel, nor implicit manipulators.



Figure 9.1.8 Construction Front End Loader



Figure 9.1.9 Construction Excavator

To see the effect of closed kinematic loops more clearly, consider the simplified *front end loader* of Fig. 9.1.10, which is used for scooping material on the ground as the chassis moves forward, tilting the bucket in an upward orientation to hold the material, raising the bucket, and subsequently rotating the bucket downward to dump the material into a container or vehicle. The forward and reverse motion of the chassis is suppressed in this example. The mechanism has three bodies that move relative to body 0 (chassis), whose positions and orientations are defined in the chassis-fixed x-y frame by three generalized coordinates shown in Fig. 9.1.10,

$$\mathbf{q} = [q_1 \quad q_2 \quad q_3]^T \in \mathbb{R}^3. \text{ Hydraulic actuator inputs } \mathbf{y} = [y_1 \quad y_2]^T \in \mathbb{R}^2 \text{ control motion of the}$$

system, and outputs are the orientation of the bucket and elevation of the tip of the bucket,  $\mathbf{z} = [q_1 + q_3 \quad y_{D_2}]^T \in \mathbb{R}^2$ . Body 2 slides along the axis of body 1 (boom) and is connected to point  $D_1$  on body 3 (bucket) by a bar of fixed length to control orientation of the bucket.

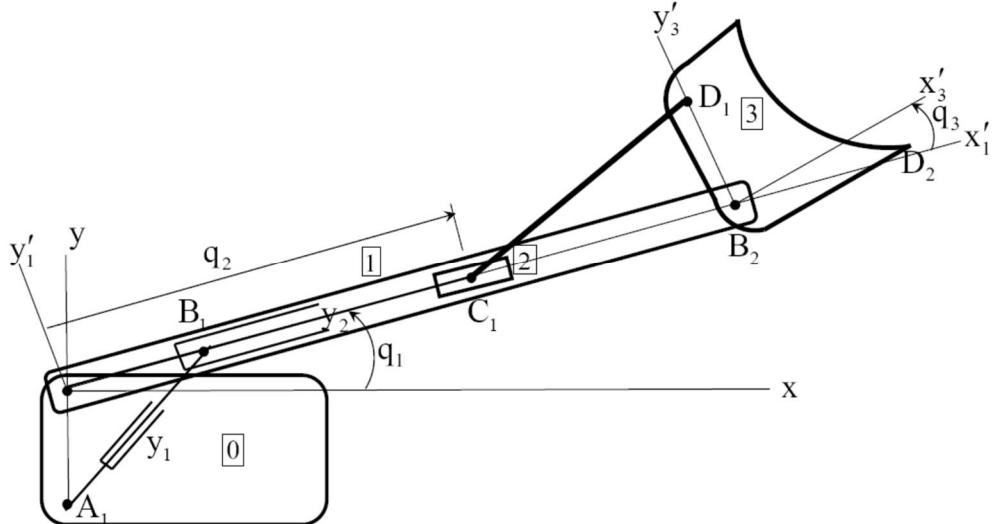


Figure 9.1.10 Front-End Loader

Actuators  $y_1$  and  $y_2$  act in parallel in this manipulator, so it is not a serial manipulator. It differs fundamentally from the manipulators of Sections 9.1.2 and 9.1.3, in that there is a *kinematic closed loop* involving the boom, slider, bucket, and bar that connects the bucket and slider. It thus fails to be an explicit parallel manipulator, which is typical of manipulators found in heavy-duty material handling, construction, and earth moving applications. Due to *kinematic closed loops* in such systems, they cannot be globally represented by an analytical input-output model of the kind encountered in Section 9.1.4.

Substantial complexity associated with this class of applications includes the requirement that the kinematics formulation must include mechanism generalized coordinates, in addition to input and output coordinates. This reality has been presented in a recent paper (Haug, 2021b) and is further developed in Section 9.6. This category is called *compound manipulators*.

### 9.1.5 Relation Between the Present Formulation and the Manipulator Literature

In a survey of parallel manipulators, Merlet, Gosselin, and Huang (2016) identify numerous open issues regarding manipulator analysis, among them the following:

- (1) Formulations need to account for *singularities* that input-output models fail to identify.
- (2) *Singularity free components of manipulator workspaces* need to be determined and criteria established for movement from one component to another without encountering singularities.

Relative to issue (1), Conconi and Carricato (2009) present an extensive analysis of *parallel manipulator singularities*. They show that manipulator formulations require inclusion of mechanism generalized coordinates and associated constraints, in order to account for *singularities* that are inherent in the underlying mechanism and have nothing to do with input or output coordinates. They expand classifications of singularities that are presented in the

literature, further reinforcing the need for inclusion of *mechanism generalized coordinates* and associated constraints in parallel manipulator models that cannot be represented in input-output form. While their approach extends the conventional input-output parallel manipulator formulation, it is limited to manipulators that are comprised of a single platform whose motion is controlled by parallel mechanisms that connect the platform to a fixed base. This is a generalization of leg actuated systems, such as the Stewart platform of Section 9.1.2, but retains the concept of a single platform whose motion is controlled by parallel chains of input actuators.

Four *nonredundant manipulator kinematics formulations* are introduced in Section 9.2 that provide analytical and computational structures for treating the distinctly different character of manipulators analyzed in this section, including one that is based on mechanism generalized coordinates and associated *kinematic constraint equations*, in addition to input and output coordinates and their defining equations. Attention herein is restricted to *nonredundant manipulators*, for which the dimension of input and output coordinates is equal to the number of *degrees of freedom* of the underlying mechanism, generally smaller than the number of generalized coordinates. Jacobian (derivative) matrices are defined at the configuration level that must be nonsingular for local existence and uniqueness of forward and inverse kinematics. This formulation accounts for singular behavior that cannot be dealt with in input-output formulations, either because the underlying mechanism has *kinematic singularities* or because complexity of the mechanism precludes analytical reduction to input-output form. It is emphasized that this chapter is not another in a long list of treatments of manipulator singularities; its purpose is to define and characterize *singularity free domains of functionality* in which manipulators can be reliably programmed and controlled.

Manipulators that are ideally suited for motion control under the influence of large forces are contrasted with classical serial manipulators that form the foundation of the field of robotics. The focus of the chapter is on the differential geometry of manipulators and manipulator input and output mappings that yield a rich manifold theoretic structure for kinematics and dynamics of manipulators.

## 9.2 Manipulator Kinematics

### 9.2.1 Manipulator Coordinates and Kinematic Equations

The mechanical component of a manipulator is a *mechanism*, whose configuration is defined by *mechanism generalized coordinates*  $\mathbf{q} \in \mathbb{R}^n$  that are subject to  $m = n - k > 0$  *holonomic constraints*,

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (9.2.1)$$

where  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is continuously differentiable. There are many choices for generalized coordinates that specify the configuration of a *manipulator mechanism*, independent of any concept of input or output, any of which may be employed in the present formulation. As commonly used in the mechanical systems literature, such generalized coordinates are not independent; i.e., they must satisfy holonomic constraints of Eq. (9.2.1).

*Input coordinates*  $\mathbf{y} \in \mathbb{R}^k$  are related to mechanism generalized coordinates  $\mathbf{q} \in \mathbb{R}^n$  through *input equations*,

$$\Psi(\mathbf{q}, \mathbf{y}) = \mathbf{0} \quad (9.2.2)$$

where  $\Psi : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}^k$  is continuously differentiable. In case an input coordinate  $y_i$  is a generalized coordinate  $q_j$ , Eq. (9.2.2) contains the condition  $q_j - y_i = 0$ . Equations (9.2.1) and (9.2.2) may be combined as  $n$  equations in  $n + k$  variables,

$$\Omega(\mathbf{q}, \mathbf{y}) \equiv \begin{bmatrix} \Phi(\mathbf{q}) \\ \Psi(\mathbf{q}, \mathbf{y}) \end{bmatrix} = \mathbf{0} \quad (9.2.3)$$

which is a necessary but not sufficient condition to determine  $\mathbf{q}$  as a function of  $\mathbf{y}$ .

*Output coordinates*  $\mathbf{z} \in \mathbb{R}^k$  define *manipulator working capability* and are related to mechanism generalized coordinates through *output equations*,

$$\Gamma(\mathbf{q}, \mathbf{z}) = \mathbf{0} \quad (9.2.4)$$

where  $\Gamma : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}^k$  is continuously differentiable. In case an output coordinate  $z_i$  is a generalized coordinate  $q_j$ , Eq. (9.2.4) contains the condition  $q_j - z_i = 0$ . Equations (9.2.1) and (9.2.4) may be combined as  $n$  equations in  $n + k$  variables,

$$\Lambda(\mathbf{q}, \mathbf{z}) \equiv \begin{bmatrix} \Phi(\mathbf{q}) \\ \Gamma(\mathbf{q}, \mathbf{z}) \end{bmatrix} = \mathbf{0} \quad (9.2.5)$$

which is a necessary but not sufficient condition to determine  $\mathbf{q}$  as a function of  $\mathbf{z}$ .

The *manipulator configuration space* is defined as the set of all *manipulator coordinates*  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{q}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^k \times \mathbb{R}^n \times \mathbb{R}^k = \mathbb{R}^{n+2k}$  that satisfy Eqs. (9.2.1), (9.2.2), and (9.2.4); i.e.,

$$X \equiv \left\{ \mathbf{x} : \Phi(\mathbf{q}) = \mathbf{0}, \Psi(\mathbf{q}, \mathbf{y}) = \mathbf{0}, \text{ and } \Gamma(\mathbf{q}, \mathbf{z}) = \mathbf{0} \right\} \subset \mathbb{R}^{n+2k} \quad (9.2.6)$$

Endowed with open sets of the form  $V \cap X \subset X$ , where  $V$  is open in  $R^{n+2k}$ ,  $X$  is a *topological subspace* of  $R^{n+2k}$  (Mendelson, 1962). The manipulator configuration space of Eq. (9.2.6) is very different from the *kinematic configuration space*  $C = \{\mathbf{q} \in R^n : \Phi(\mathbf{q}) = \mathbf{0}\}$  that is called the *c-space* of the mechanism in the manipulator literature.

### 9.2.2 Forward and Inverse Kinematics

At a manipulator configuration  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{q}}^T \quad \bar{\mathbf{z}}^T]^T \in X$ , conditions are required that assure Eqs. (9.2.3) and (9.2.4) uniquely determine  $\mathbf{q}$  and  $\mathbf{z}$  as differentiable functions of input  $\mathbf{y}$  in a neighborhood of  $\bar{\mathbf{x}}$ ; i.e., *forward kinematics*. Conversely, conditions are required that assure Eqs. (9.2.2) and (9.2.5) uniquely determine  $\mathbf{q}$  and  $\mathbf{y}$  as differentiable functions of output  $\mathbf{z}$  in a neighborhood of  $\bar{\mathbf{x}}$ ; i.e., *inverse kinematics*. Such conditions, in terms of the functions defined in Eqs. (9.2.2) through (9.2.5), are provided by the implicit function theorem as

$$|\Psi_y(\bar{\mathbf{q}}, \bar{\mathbf{y}})| \neq 0 \quad (9.2.7)$$

$$|\Gamma_z(\bar{\mathbf{q}}, \bar{\mathbf{z}})| \neq 0 \quad (9.2.8)$$

$$|\Omega_q(\bar{\mathbf{q}}, \bar{\mathbf{y}})| = \begin{bmatrix} \Phi_q(\bar{\mathbf{q}}) \\ \Psi_q(\bar{\mathbf{q}}, \bar{\mathbf{y}}) \end{bmatrix} \neq 0 \quad (9.2.9)$$

$$|\Lambda_q(\bar{\mathbf{q}}, \bar{\mathbf{z}})| = \begin{bmatrix} \Phi_q(\bar{\mathbf{q}}) \\ \Gamma_q(\bar{\mathbf{q}}, \bar{\mathbf{z}}) \end{bmatrix} \neq 0 \quad (9.2.10)$$

To account for singular points at which  $\Phi_q(\mathbf{q})$  fails to have *full rank*, it has been suggested that an additional criterion is needed, namely,  $\text{rank}(\Phi_q(\bar{\mathbf{q}})) = n - k$ ; i.e.,  $\Phi_q(\bar{\mathbf{q}})$  has full rank. Since full rank of  $\Phi_q(\bar{\mathbf{q}})$  is implied by Eqs. (9.2.9) and (9.2.10), it need not be redundantly imposed.

It is important to note that equality in any of the four conditions of Eqs. (9.2.7) through (9.2.10) defines a *singular configuration*; i.e., either forward or inverse kinematics fails. In particular, the present formulation accounts for singularities due to rank deficiency of  $\Phi_q(\mathbf{q})$  that appears in Eqs. (9.2.9) and (9.2.10), as well as other forms of *singularity*. Since the manipulator configuration space  $X$  of Eq. (9.2.6) may include singular configurations in which forward and/or inverse kinematics fail, it is too large a space for reliable manipulator operation. The four conditions of Eqs. (9.2.7) through (9.2.10) will be shown to eliminate such negative forms of behavior. This suggests defining the *regular manipulator configuration space* as

$$\tilde{X} = \left\{ \mathbf{x} = [\mathbf{y}^T \quad \mathbf{q}^T \quad \mathbf{z}^T]^T \in X : \begin{array}{l} |\Omega_q(\mathbf{q}, \mathbf{y})| \neq 0, \quad |\Psi_y(\mathbf{q}, \mathbf{y})| \neq 0, \\ |\Lambda_q(\mathbf{q}, \mathbf{z})| \neq 0, \text{ and } |\Gamma_z(\mathbf{q}, \mathbf{z})| \neq 0 \end{array} \right\} \quad (9.2.11)$$

#### 9.2.2.1 Forward Kinematics

The implicit function theorem of Section 2.2.5 provides a sufficient condition that an input  $\mathbf{y}$  uniquely determines the configuration  $\mathbf{q}$  of a manipulator in a neighborhood of  $(\bar{\mathbf{q}}, \bar{\mathbf{y}})$ , for  $\bar{\mathbf{x}} \in \tilde{X}$ , that satisfies Eq. (9.2.3), namely the condition of Eq. (9.2.9); i.e., there is a unique continuously differentiable solution of Eq. (9.2.3) in a neighborhood  $Y$  of  $\bar{\mathbf{y}}$ ,

$$\mathbf{q} = \mathbf{f}(\mathbf{y}) \quad (9.2.12)$$

A sufficient condition that  $\mathbf{q}$  determines a unique output  $\mathbf{z}$  that satisfies Eq. (9.2.4) in a neighborhood of  $(\bar{\mathbf{q}}, \bar{\mathbf{z}}) = (\mathbf{f}(\bar{\mathbf{y}}), \bar{\mathbf{z}})$  is Eq. (9.2.8); i.e., there is a unique continuously differentiable solution of Eq. (9.2.4) in a neighborhood  $W$  of  $\bar{\mathbf{q}}$ ,

$$\mathbf{z} = \mathbf{h}(\mathbf{q}) \quad (9.2.13)$$

If the conditions of Eqs. (9.2.8) and (9.2.9) hold, then Eqs. (9.2.12) and (9.2.13) uniquely determine an output for an input in a neighborhood of  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{q}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}$ ; i.e.,

$$\mathbf{z} = \mathbf{h}(\mathbf{f}(\mathbf{y})) \quad (9.2.14)$$

The *forward kinematic mapping* can fail for two reasons. First, even if  $\Phi_q(\bar{\mathbf{q}})$  has full rank, Eq. (9.2.9) may fail to hold, in which case Eq. (9.2.3) fails to determine a unique  $\mathbf{q}$ . Second, if the condition of Eq. (9.2.8) fails to hold, then Eq. (9.2.4) fails to determine a unique output  $\mathbf{z}$  for an admissible configuration  $\mathbf{q}$  of the mechanism. In either case, forward kinematics fail and the configuration  $(\bar{\mathbf{y}}, \bar{\mathbf{q}}, \bar{\mathbf{z}})$  is singular.

### 9.2.2.2 Inverse Kinematics

In order for an output  $\mathbf{z}$  to uniquely determine the configuration  $\mathbf{q}$  of a manipulator in a neighborhood of  $(\bar{\mathbf{q}}, \bar{\mathbf{z}})$ , for  $\bar{\mathbf{x}} \in \tilde{X}$ , that satisfies Eq. (9.2.5), Eq. (9.2.10) implies that there is a unique continuously differentiable solution of Eq. (9.2.5) in a neighborhood  $Z$  of  $\bar{\mathbf{z}}$ ,

$$\mathbf{q} = \mathbf{e}(\mathbf{z}) \quad (9.2.15)$$

Finally, Eq. (9.2.7) implies that there exists a unique continuously differentiable solution of Eq. (9.2.2) in a neighborhood  $W$  of  $\bar{\mathbf{q}}$ , where  $(\bar{\mathbf{q}}, \bar{\mathbf{y}})$  satisfies Eq. (9.2.2),

$$\mathbf{y} = \mathbf{g}(\mathbf{q}) \quad (9.2.16)$$

If Eqs. (9.2.7) and (9.2.10) hold, then Eqs. (9.2.15) and (9.2.16) uniquely determine an input to achieve the desired output in a neighborhood of  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{q}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}$ ; i.e.,

$$\mathbf{y} = \mathbf{g}(\mathbf{e}(\mathbf{z})) \quad (9.2.17)$$

The *inverse kinematic mapping* can fail for two reasons. First, even if  $\Phi_q(\bar{\mathbf{q}})$  has full rank, Eq. (9.2.10) may fail to hold, in which case Eq. (9.2.5) fails to determine a unique  $\mathbf{q}$ . Second, if the condition of Eq. (9.2.7) fails to hold, Eq. (9.2.2) fails to determine a unique input  $\mathbf{y}$  that leads to a kinematically admissible configuration of the mechanism. In either case, inverse kinematics fail and the configuration  $(\bar{\mathbf{y}}, \bar{\mathbf{q}}, \bar{\mathbf{z}})$  is singular.

### 9.2.3 Singular Configurations and Explicit Input-Output Equations

If any function  $e(z)$ ,  $f(y)$ ,  $g(q)$ , or  $h(q)$  does not exist in a neighborhood of  $\bar{x} \in X$ , the forward and/or inverse kinematic mapping fails and  $\bar{x} \notin \tilde{X}$  is called a *singular manipulator configuration*.

If  $f(y)$  and  $h(q)$  exist globally, Eq. (9.2.14) may be written as

$$z = h(f(y)) \equiv G(y) \quad (9.2.18)$$

which is of the form of Eq. (9.1.2) for a *serial manipulator*. As noted in Section 9.1.1, for serial manipulators, no globally valid inverse kinematic relation may exist. This occurs if either  $e(z)$  or  $g(q)$  fails to exist. An inverse argument follows for *explicit parallel manipulators*. In either category, singularities exist in the manipulator configuration space. For implicit and compound manipulators, forward and inverse kinematic mappings of Eqs. (9.2.14) and (9.2.17) exist only locally. In the case of *implicit manipulators*, it is assumed that generalized coordinates can be analytically eliminated to obtain input-output equations of the form  $F(y, z) = 0$ , which is possible only for very simple manipulators. For *compound manipulators*, there is no possibility of obtaining an input-output form of kinematic equations. In any case, both implicit and compound manipulators will generally have configuration spaces that contain extensive subsets of singular configurations.

The foregoing discussion fails to come to grips with the reality that practical manipulators with significant numbers of degrees of freedom are governed by complex kinematic equations. For even the serial manipulator of Fig. 9.1.2, the expectation that the forward kinematic relation  $z = h(f(y)) \equiv G(y)$  can be written analytically is unrealistic. The implicit function theorem implies the existence of  $h(q)$  and  $f(y)$ , which can be evaluated numerically, but it does not provide analytical forms of the functions. The principal tenant of this chapter is that, while attractive theoretical results make manipulator kinematics and dynamics possible, they must be implemented numerically in the real world. This is in contrast with an extensive literature on manipulator kinematics that is based on very simple kinematic models that can be expressed in analytical input-output form. Such models and the associated analytical results are helpful in gaining insight, but are seldom extendible to large scale industrial manipulators.

In the development that follows, if the explicit formulation of input-output relations for serial, explicit parallel, or implicit manipulators can be obtained, they are used as illustrations. In reality, even if such manipulators do not have closed kinematic loops, the compound manipulator formulation may be used for their kinematic and dynamic analysis. This is a practical reality for industrial scale spatial manipulators of each category. In fact, all is not lost if the practicing engineer finds it necessary to use the compound manipulator formulation in dealing with even serial manipulators that are not analytically tractable.

Manipulator kinematics are defined in terms of input, output, and generalized coordinates that are analytically related. Four Jacobian matrix determinants are defined that characterize a manipulator as belonging to one of the four categories. Local forward and inverse kinematic mappings are defined on the regular configuration space. For large scale industrial manipulators, it is observed that these mappings will generally not be analytically defined, but may be effectively computed.

### 9.3 Serial Manipulator Kinematics

The best-known category of manipulator is a *serial chain mechanism* that defines a *serial manipulator* of Eqs. (9.2.8) and (9.2.9) for all  $\mathbf{y} \in \mathbb{R}^k$ ; i.e., globally. Thus, for configurations that satisfy Eqs. (9.2.3), (9.2.4), (9.2.8) and (9.2.9), the forward kinematic mapping is

$$\mathbf{z} = \mathbf{h}(\mathbf{f}(\mathbf{y})) \equiv \mathbf{G}(\mathbf{y}) \quad (9.3.1)$$

which is written in terms of input and output coordinates.

#### 9.3.1 A Model Serial Manipulator

The *planar serial manipulator* shown in Fig. 9.3.1 consists of three bars of unit length that are pivoted at the origin of the x-y frame and at their intersections, the third bar being the *end-effector*. Generalized coordinates  $\mathbf{q} = [q_1 \ \dots \ q_7]^T \in \mathbb{R}^7$  shown in Fig. 9.3.1(a) define the configuration of all bodies in the mechanism. Input coordinates  $\mathbf{y} = [y_1 \ y_2 \ y_3]^T \in \mathbb{R}^3$  shown in Fig. 9.3.1(b) are employed to control the position of point P on the third bar and the orientation of that bar, defined by output coordinates  $\mathbf{z} = [z_1 \ z_2 \ z_3]^T \in \mathbb{R}^3$ .

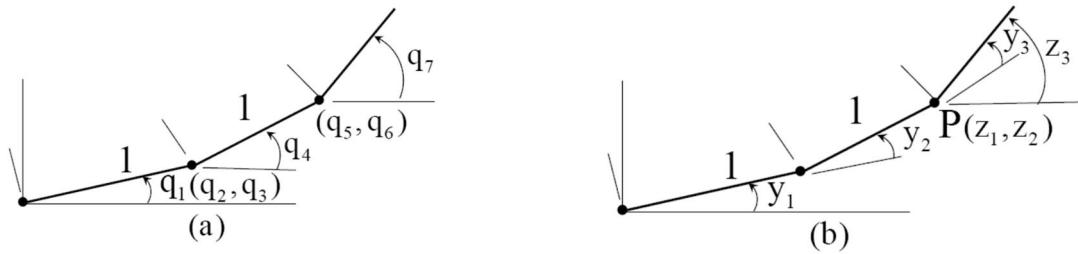


Figure 9.3.1 Planar Serial Manipulator

In the manipulator kinematics formulation of Section 9.2,

$$\begin{aligned} \Phi(\mathbf{q}) &= \begin{bmatrix} q_2 - \cos q_1 \\ q_3 - \sin q_1 \\ q_5 - q_2 - \cos q_4 \\ q_6 - q_3 - \sin q_4 \end{bmatrix} = 0 \\ \Psi(\mathbf{q}, \mathbf{y}) &= \begin{bmatrix} q_1 - y_1 \\ q_4 - y_2 - y_1 \\ q_7 - y_3 - y_2 - y_1 \end{bmatrix} = 0 \\ \Gamma(\mathbf{q}, \mathbf{z}) &= \begin{bmatrix} q_5 - z_1 \\ q_6 - z_2 \\ q_7 - z_3 \end{bmatrix} = 0 \end{aligned} \quad (9.3.2)$$

A direct calculation shows that  $|\Psi_y(\mathbf{q}, \mathbf{y})| = -1$  and  $|\Gamma_z(\mathbf{q}, \mathbf{z})| = -1$ , so the expressions

$\mathbf{y} = \mathbf{g}(\mathbf{q}) = [q_1 \quad q_4 - q_1 \quad q_7 - q_4 - q_1]^T$  and  $\mathbf{z} = \mathbf{h}(\mathbf{q}) = [q_5 \quad q_6 \quad q_7]^T$  of Eqs. (9.2.16) and (9.2.13) are globally valid. From the geometry of Fig. 9.3.1(b),

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \cos y_1 + \cos(y_1 + y_2) \\ \sin y_1 + \sin(y_1 + y_2) \\ y_1 + y_2 + y_3 \end{bmatrix} \equiv \mathbf{G}(\mathbf{y}) \quad (9.3.3)$$

is a *globally valid forward kinematic mapping* and, from the geometry of Figs. 9.3.1(a) and (b),  $\mathbf{q} = [y_1 \quad \cos y_1 \quad \sin y_1 \quad y_1 + y_2 \quad \cos y_1 + \cos(y_1 + y_2) \quad \sin y_1 + \sin(y_1 + y_2) \quad y_1 + y_2 + y_3]^T = \mathbf{f}(\mathbf{y})$

If  $y_2 = 0$ , bars 1 and 2 are colinear, so  $z_1 = 2\cos y_1$  and  $z_2 = 2\sin y_1$ . A perturbation of  $\mathbf{z}$  such that  $\delta z_1 = \varepsilon \cos y_1$  and  $\delta z_2 = \varepsilon \sin y_1$  yields no solution for  $\mathbf{y}$  if  $\varepsilon > 0$  and two solutions if  $\varepsilon < 0$ . Thus, inverse kinematics is not globally valid.

Serial manipulator generalized coordinates can often be efficiently eliminated (Lynch and Park, 2017) to create an input-output *forward kinematic mapping* of the form of Eq. (9.3.3). In such cases, *manipulator coordinates* for kinematic analysis may be defined as

$\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^6$ . The *serial manipulator kinematic configuration space* is thus

$$X^S = \{\mathbf{x} : \mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0}\} \subset \mathbb{R}^6 \quad (9.3.4)$$

To define conditions under which  $\mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0}$  locally determines  $\mathbf{y}$  as a function of  $\mathbf{z}$ ; i.e., there is an *inverse kinematic mapping*  $\mathbf{y} = \mathbf{H}(\mathbf{z})$ , the Jacobian of  $\mathbf{G}(\mathbf{y}) - \mathbf{z}$  with respect to  $\mathbf{y}$  is

$$\mathbf{G}'(\mathbf{y}) = \begin{bmatrix} -\sin y_1 - \sin(y_1 + y_2) & -\sin(y_1 + y_2) & 0 \\ \cos y_1 + \cos(y_1 + y_2) & \cos(y_1 + y_2) & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (9.3.5)$$

with determinant

$$\begin{aligned} |\mathbf{G}'(\mathbf{y})| &= -\sin y_1 \cos(y_1 + y_2) - \sin(y_1 + y_2) \cos(y_1 + y_2) \\ &\quad + \cos y_1 \sin(y_1 + y_2) + \cos(y_1 + y_2) \sin(y_1 + y_2) \\ &= -(\sin y_1 \cos(y_1 + y_2) - \cos y_1 \sin(y_1 + y_2)) \\ &= -\sin(y_1 - y_1 - y_2) = \sin y_2 \end{aligned} \quad (9.3.6)$$

To avoid singularities in  $X^S$  at which  $|\mathbf{G}'(\mathbf{y})| = 0$ , define the *regular serial manipulator configuration space* as the open subset of  $X^S$ ,

$$\tilde{X}^S = \{\mathbf{x} \in X^S : |\mathbf{G}'(\mathbf{y})| = \sin y_2 \neq 0\} \quad (9.3.7)$$

For any  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}^S$  such that  $|\mathbf{G}'(\bar{\mathbf{y}})| \neq 0$ , and  $\mathbf{G}(\bar{\mathbf{y}}) - \bar{\mathbf{z}} = \mathbf{0}$ , the *implicit function theorem* guarantees existence of a unique solution  $\mathbf{y} = \mathbf{H}(\mathbf{z})$  of  $\mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0}$  in a neighborhood

$Z$  of  $\bar{\mathbf{z}}$ ; i.e.,  $\mathbf{G}(\mathbf{H}(\mathbf{z})) - \mathbf{z} = \mathbf{0}$  for all  $\mathbf{z} \in Z$ . Maximal, singularity free, path connected components of  $\tilde{X}^S$  for this manipulator are three-dimensional disjoint subsets of  $\mathbb{R}^6$ ,

$$\begin{aligned}\tilde{X}_1^S &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{z} = \mathbf{G}(\mathbf{y}), -\pi \leq y_1 \leq \pi, 0 < y_2 < \pi, -\pi \leq y_3 \leq \pi \right\} \\ \tilde{X}_2^S &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{z} = \mathbf{G}(\mathbf{y}), -\pi \leq y_1 \leq \pi, -\pi < y_2 < 0, -\pi \leq y_3 \leq \pi \right\}\end{aligned}\quad (9.3.8)$$

The forms of manipulator configurations in these components are shown in Fig. 9.3.2. To avoid singular behavior, the control engineer may choose to work in either  $\tilde{X}_1^S$  or  $\tilde{X}_2^S$ . While it is not possible to draw pictures of these sets in  $\mathbb{R}^6$ , the control system engineer must function in the components of Eq. (9.3.8).

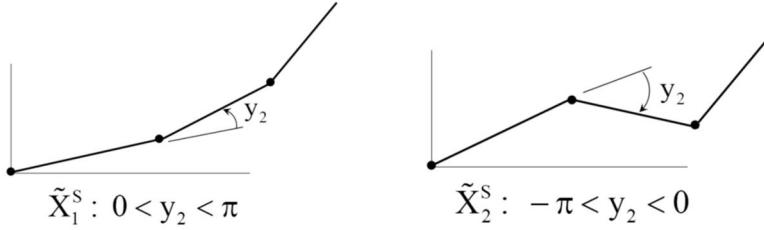


Figure 9.3.2 Manipulator Configurations in  $\tilde{X}_1^S$  and  $\tilde{X}_2^S$

Boundaries of the components of Eq. (9.3.8) are sets of *singular configurations*,

$$\begin{aligned}\tilde{X}^1 &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{z} = \mathbf{G}(\mathbf{y}), y_2 = 0, -\pi \leq y_1 \leq \pi \right\} \\ \tilde{X}^2 &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{z} = \mathbf{G}(\mathbf{y}), y_2 = \pi, -\pi \leq y_1 \leq \pi \right\} \\ \tilde{X}^3 &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{z} = \mathbf{G}(\mathbf{y}), y_2 = -\pi, -\pi \leq y_1 \leq \pi \right\}\end{aligned}\quad (9.3.9)$$

in which  $|\mathbf{G}'(\mathbf{y})| = 0$  and Eq. (9.3.3) has no unique solution  $\mathbf{y} = \mathbf{H}(\mathbf{z})$ ; i.e., inverse kinematics fail.

To see that configurations in different components of Eq. (9.3.8) cannot be connected by singularity free continuous paths, values of  $\mathbf{y}$  at the midpoints of ranges of  $y_i$  in Eq. (9.3.8) are tabulated, with associated values of  $|\mathbf{G}'(\mathbf{y})|$ , in Table 9.3.1. Since values of the determinant are different at these midpoints, they are different throughout components  $\tilde{X}_1^S$  and  $\tilde{X}_2^S$ . Thus, a continuous path between configurations in  $\tilde{X}_1^S$  and  $\tilde{X}_2^S$  must pass through a zero value of the determinant; i.e., a singular configuration.

Table 9.3.1 Values of  $|\mathbf{G}'(\mathbf{y})|$  in Disjoint Domains

$\tilde{X}_j^S$	Point	$y_1$	$y_2$	$y_3$	$ \mathbf{G}'(\mathbf{y}) $
$\tilde{X}_1^S$	(1)	0	/2	0	1
$\tilde{X}_2^S$	(2)	0	-/2	0	-1

### 9.3.2 Serial Manipulator Regular Configuration Manifold

For a *general serial manipulator* with input  $\mathbf{y} \in \mathbb{R}^k$ , output  $\mathbf{z} \in \mathbb{R}^k$ , and configuration  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^{2k}$ , kinematic *input-output equations* are of the form of Eq. (9.3.1); i.e.,  $\mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0}$ , where  $\mathbf{G}(\mathbf{y})$  is continuously differentiable. The explicit relation  $\mathbf{z} = \mathbf{G}(\mathbf{y})$  defines *forward kinematics* of the manipulator. The *serial manipulator configuration space* is

$$X^S = \{\mathbf{x} : \mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0}\} \subset \mathbb{R}^{2k} \quad (9.3.10)$$

To avoid singular configurations in  $X^S$ , the *regular serial manipulator configuration space* is defined as

$$\tilde{X}^S = \{\mathbf{x} \in X^S : |\mathbf{G}'(\mathbf{y})| \neq 0\} \quad (9.3.11)$$

This is an open subset of  $X^S$ , because its complement  $\{\mathbf{x} \in X^S : |\mathbf{G}'(\mathbf{y})| = 0\}$  is closed, so  $\tilde{X}^S$  is comprised of maximal, disjoint, singularity free, path connected *components*  $\tilde{X}_i^S$ . Without the need to numerically evaluate  $|\mathbf{G}'(\mathbf{y})|$  in such components, any path in configuration space between configurations in disjoint components must encounter a value of zero for the determinant  $|\mathbf{G}'(\mathbf{y})|$ , hence a singularity.

For any  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}^S$ , the implicit function theorem assures existence of a continuously differentiable function  $\mathbf{H}(\mathbf{z})$  and a neighborhood  $Z$  of  $\bar{\mathbf{z}}$  such that  $\mathbf{G}(\mathbf{H}(\mathbf{z})) - \mathbf{z} = \mathbf{0}$  for all  $\mathbf{z} \in Z$ ; i.e., a *local inverse kinematic mapping*. The relation  $\mathbf{y} = \mathbf{H}(\mathbf{z})$  thus defines inverse kinematics of the manipulator in  $Z$ . For any neighborhood  $Y$  of  $\bar{\mathbf{y}}$ , the subset  $U = Y \times Z \equiv \{\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T : \mathbf{y} \in Y, \mathbf{z} \in Z\} \subset \tilde{X}^S$  is an open neighborhood of  $\bar{\mathbf{x}} \in \tilde{X}^S$ . A countable collection of such neighborhoods  $U^i$  of  $\mathbf{x}^i \in \tilde{X}^S$  covers  $\tilde{X}^S$  (Mendelson, 1962).

For  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in U^i = Y^i \times Z^i$ , define the mapping  $\phi^i(\mathbf{x}) = \mathbf{y} \in Y^i$ . With  $\psi^i(\mathbf{y}) \equiv [\mathbf{y}^T \quad \mathbf{G}^T(\mathbf{y})]^T = \mathbf{x} \in \tilde{X}^S$  for  $\mathbf{y} \in Y^i$ ,  $\phi^i(\psi^i(\mathbf{y})) = \phi^i([\mathbf{y}^T \quad \mathbf{G}^T(\mathbf{y})]^T) = \mathbf{y}$ , for all  $\mathbf{y} \in Y^i$ . Thus,  $\psi^i = (\phi^i)^{-1}$  on  $Y^i$  and  $\tilde{X}^S$  is a  $k$ -dimensional *differentiable manifold* with *charts*  $(\phi^i, U^i)$ , parameterized with  $\mathbf{x} = \psi^i(\mathbf{y}) \equiv [\mathbf{y}^T \quad \mathbf{G}^T(\mathbf{y})]^T \in \tilde{X}^S$  by input  $\mathbf{y}$  (Guillemin and Pollack, 1974; Schlichtkrull, 2015).

Conversely, define  $\phi^i(\mathbf{x}) = \mathbf{z} \in Z^i$ . With  $\psi^i(\mathbf{z}) \equiv [\mathbf{H}^T(\mathbf{z}) \quad \mathbf{z}^T]^T = \mathbf{x} \in \tilde{X}^S$  for  $\mathbf{z} \in Z^i$ ,  $\phi^i(\psi^i(\mathbf{z})) = \phi^i([\mathbf{H}^T(\mathbf{z}) \quad \mathbf{z}^T]^T) = \mathbf{z}$ , so  $\psi^i = (\phi^i)^{-1}$  on  $Z^i$ . This again shows that  $\tilde{X}^S$  is a  $k$ -dimensional differentiable manifold, but with charts  $(\phi^i, U^i)$ , parameterized with  $\mathbf{x} = \psi^i(\mathbf{z}) \equiv [\mathbf{H}^T(\mathbf{z}) \quad \mathbf{z}^T]^T \in \tilde{X}^S$  by output  $\mathbf{z}$ . This natural *duality of parameterization* by input

and output coordinates for a manipulator is not shared by kinematics of general multibody systems (Haug, 2021a).

With either parameterization, the regular manipulator configuration space  $\tilde{X}^s$  for a serial manipulator is a  $k$ -dimensional *differentiable manifold*. Its maximal, singularity free, path connected *components*  $\tilde{X}_i^s$  are thus ideally suited for manipulator analysis and control.

Numerical methods for serial manipulator forward and inverse kinematic configuration and velocity analysis are presented in Section 9.10. Using the manifold parameterization  $\mathbf{x} = \psi^i(\mathbf{y})$ , singularity free ODE of manipulator dynamics in  $\tilde{X}_i^s$ , with input  $\mathbf{y}$  as the state variable, are derived in Section 9.11, without requiring ad-hoc derivation. Comparable results are obtained using the parameterization  $\mathbf{x} = \psi^i(\mathbf{z})$  with output  $\mathbf{z}$ .

A simple planar serial manipulator is shown to have a global forward kinematic solution, but to have a singularity in inverse kinematics. Two singularity free components are defined and it is shown that any trajectory between them must encounter a singularity. The regular configuration space for a general serial manipulator is shown to be a differentiable manifold, parameterized by either input or output coordinates.

## 9.4 Explicit Parallel Manipulator Kinematics

Many forms of *parallel manipulator* arise in applications (Merlet, 2006). A special form is based on Eqs.(9.2.2), (9.2.5), (9.2.7), and (9.2.10) that hold for all  $\mathbf{z} \in \mathbb{R}^k$ . As in Eqs. (9.2.15) and (9.2.16), the *global inverse kinematic mapping* is

$$\mathbf{y} = \mathbf{g}(\mathbf{e}(\mathbf{z})) \equiv \bar{\mathbf{H}}(\mathbf{z}) \quad (9.4.1)$$

### 9.4.1 A Model Explicit Parallel Manipulator

As an example of an *explicit parallel manipulator*, the *planar platform* of Fig. 9.4.1 has been studied by Peidro, Marin, Gil, and Reinoso (2015). This planar manipulator is analogous to the spatial Gough-Stewart platform of Fig. 9.1.4 that is extensively studied in the parallel manipulator literature (Merlet, 2006). The planar platform  $B_1$ -O- $B_2$  of Fig. 9.4.1 is constrained so that its operating point O slides in a vertical slot in the global frame. This yields a vector of *output coordinates*  $\mathbf{z} = [z_1 \ z_2]^T \in \mathbb{R}^2$  and a vector  $\mathbf{y} = [y_1 \ y_2]^T \in \mathbb{R}^2$  of *input coordinates* with  $y_i \geq 0$ , one in each of the serial chains that connect the base to points  $B_1$  and  $B_2$  in the platform. *Generalized coordinates* of the platform, the only moving body in the mechanism, are simply  $\mathbf{q} = [q_1 \ q_2]^T = [z_1 \ z_2]^T = \mathbf{z}$  and are subject to no kinematic constraints. Thus,

$$\Psi(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} (6\cos q_2 - 5)^2 + (q_1 + 6\sin q_2)^2 - y_1^2 \\ (-5\cos q_2 + 6)^2 + (q_1 - 5\sin q_2)^2 - y_2^2 \end{bmatrix} = \Omega(\mathbf{q}, \mathbf{y}) = 0 \quad (9.4.2)$$

$$\Gamma(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} q_1 - z_1 \\ q_2 - z_2 \end{bmatrix} = \Lambda(\mathbf{q}, \mathbf{z}) = 0 \quad (9.4.3)$$

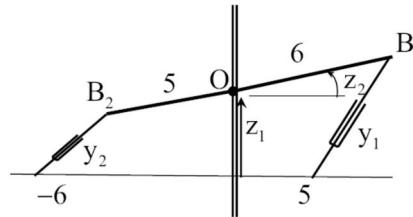


Figure 9.4.1 Planar Explicit Parallel Manipulator

With generalized coordinates eliminated, *manipulator coordinates* are

$\mathbf{x} = [\mathbf{y}^T \ \mathbf{z}^T]^T \subset \mathbb{R}^4$  and the equation that uniquely determines  $\mathbf{y}$  as a function of  $\mathbf{z}$ , for all  $\mathbf{z}$ ; i.e., *global inverse kinematics*, is

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \sqrt{(6\cos z_2 - 5)^2 + (z_1 + 6\sin z_2)^2} \\ \sqrt{(-5\cos z_2 + 6)^2 + (z_1 - 5\sin z_2)^2} \end{bmatrix} \equiv \bar{\mathbf{H}}(\mathbf{z}) \quad (9.4.4)$$

Input-output equations are thus of the form of Eq. (9.2.17); i.e.,  $\bar{\mathbf{H}}(\mathbf{z}) - \mathbf{y} = \mathbf{0}$ , and the *explicit parallel manipulator configuration space* is

$$X^{EP} = \left\{ \mathbf{x} = \begin{bmatrix} \mathbf{y}^T & \mathbf{z}^T \end{bmatrix}^T : \bar{\mathbf{H}}(\mathbf{z}) - \mathbf{y} = \mathbf{0} \right\} \subset \mathbb{R}^4 \quad (9.4.5)$$

As with serial manipulators, kinematic analysis of explicit parallel manipulators can be carried out without generalized coordinates, but the latter are required for manipulator dynamics.

Whereas serial manipulator forward kinematics is globally defined and inverse kinematics can have singularities, the character of explicit parallel manipulator kinematics is exactly the opposite. Explicit parallel manipulator inverse kinematics is globally defined, but in order to establish conditions under which Eq. (9.4.4) locally determines  $\mathbf{z}$  as a function of  $\mathbf{y}$ ; i.e., *forward kinematics*, the Jacobian of  $\bar{\mathbf{H}}(\mathbf{z}) - \mathbf{y}$  with respect to  $\mathbf{z}$  is

$$\bar{\mathbf{H}}'(\mathbf{z}) = \begin{bmatrix} (z_1 + 6\sin z_2)/y_1 & (30\sin z_2 + 6z_1 \cos z_2)/y_1 \\ (z_1 - 5\sin z_2)/y_2 & (30\sin z_2 - 5z_1 \cos z_2)/y_2 \end{bmatrix} \quad (9.4.6)$$

with determinant

$$|\bar{\mathbf{H}}'(\mathbf{z})| = (-11/y_1 y_2)(z_1^2 \cos z_2 - 30 \sin^2 z_2) \quad (9.4.7)$$

For  $y_i \neq 0$ , define the *regular explicit parallel manipulator configuration space* as

$$\tilde{X}^{EP} = \left\{ \mathbf{x} \in X^{EP} : |\bar{\mathbf{H}}'(\mathbf{z})| \neq 0 \right\} \quad (9.4.8)$$

For any  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}^{EP}$ ,  $|\bar{\mathbf{H}}'(\bar{\mathbf{z}})| \neq 0$ , so there exists a unique solution  $\mathbf{z} = \bar{\mathbf{G}}(\mathbf{y})$  of Eq. (9.4.4) in a neighborhood  $Y$  of  $\bar{\mathbf{y}}$ ; i.e.,  $\bar{\mathbf{H}}(\bar{\mathbf{G}}(\mathbf{y})) - \mathbf{y} = \mathbf{0}$  for all  $\mathbf{y} \in Y$ .

For  $z_2$  in the range  $-\pi/2 < z_2 < \pi/2$ ,  $\cos z_2 > 0$ . Thus, the condition

$z_1^2 \cos z_2 - 30 \sin^2 z_2 > 0$  is satisfied if  $z_1 > |\sin z_2| \sqrt{30/\cos z_2}$ , or  $z_1 < -|\sin z_2| \sqrt{30/\cos z_2}$ .

Similarly, the condition  $z_1^2 \cos z_2 - 30 \sin^2 z_2 < 0$  is satisfied if  $|z_1| < |\sin z_2| \sqrt{30/\cos z_2}$ . For  $z_2$  in the range  $\pi/2 < z_2 < 3\pi/2$ ,  $\cos z_2 < 0$ . Thus  $z_1^2 \cos z_2 - 30 \sin^2 z_2 \neq 0$ , except at the isolated values  $z_1 = 0$  and  $\sin z_2 = 0$ ; i.e.,  $\mathbf{z} = \mathbf{0}$  or  $\mathbf{z} = [0 \quad \pi]^T$ . Maximal, singularity free, path connected components of  $\tilde{X}^{EP}$  for this manipulator are the following two-dimensional subsets of  $\mathbb{R}^4$ :

$$\begin{aligned} \tilde{X}_1^{EP} &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{y} = \bar{\mathbf{H}}(\mathbf{z}), -\pi/2 < z_2 < \pi/2, z_1 > |\sin z_2| \sqrt{30/\cos z_2} \right\} \\ \tilde{X}_2^{EP} &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{y} = \bar{\mathbf{H}}(\mathbf{z}), -\pi/2 < z_2 < \pi/2, z_1 < -|\sin z_2| \sqrt{30/\cos z_2} \right\} \\ \tilde{X}_3^{EP} &= \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{y} = \bar{\mathbf{H}}(\mathbf{z}), -\pi/2 < z_2 < \pi/2, |z_1| < |\sin z_2| \sqrt{30/\cos z_2} \right\} \\ &\cup \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{y} = \bar{\mathbf{H}}(\mathbf{z}), \pi/2 < z_2 < 3\pi/2, -\infty < z_1 < \infty, \mathbf{z} \neq [0 \quad \pi]^T \right\} \end{aligned} \quad (9.4.9)$$

While it is not possible to draw pictures of these sets in  $\mathbb{R}^4$ , their projection onto the output space, shown in Fig. 9.4.2 as domains (1), (2), and (3), sheds light on the character of the components of Eq. (9.4.9). Nevertheless, the control engineer must operate in the full four-

dimensional components. Each of the components is singularity free and path connected, but component  $\tilde{X}_3^{\text{EP}}$  is *not simply connected*.

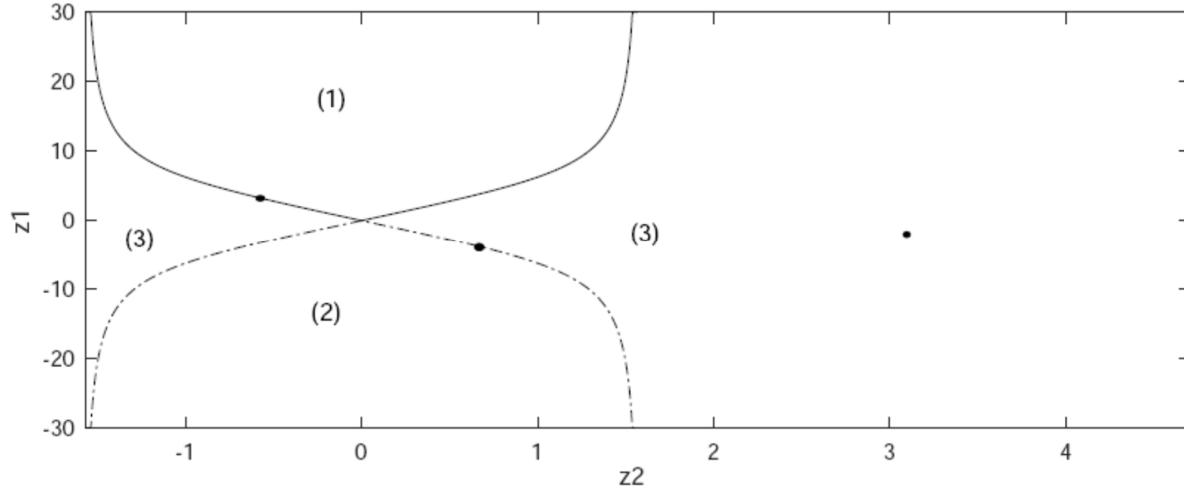


Figure 9.4.2 Components of  $\tilde{X}^{\text{EP}}$  in the  $z_1$ - $z_2$  Parameterization Plane

The case  $y_1 = 0$  yields, from Eq. (9.4.4),  $z_2 = \cos^{-1}(5/6) = \pm 0.58568$  and  $z_1 = 6\sin z_2 = \mp 3.3166$ . For these configurations,  $|\bar{H}'(\mathbf{z})| = 6\sin^2 z_2(6\cos z_2 - 5) = 0$ , so they are on boundaries between domains (1) and (3) and (2) and (3), shown as bold dots in Fig. 9.4.2.

*Singular configurations* of this manipulator are the isolated configuration

$\mathbf{x}^1 = [11 \ 11 \ 0 \ \pi]^T$ ; i.e., with  $\mathbf{z}^1 = [0 \ \pi]^T$ , and the one-dimensional boundaries of  $\tilde{X}_1^{\text{EP}}$  and  $\tilde{X}_2^{\text{EP}}$ ,

$$\tilde{X}^1 = \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{y} = \bar{H}(\mathbf{z}), z_1 = \sin z_2 \sqrt{30/\cos z_2}, -\pi/2 < z_2 < \pi/2 \right\}$$

$$\tilde{X}^2 = \left\{ \mathbf{x} \in \mathbb{R}^4 : \mathbf{y} = \bar{H}(\mathbf{z}), z_1 = -\sin z_2 \sqrt{30/\cos z_2}, -\pi/2 < z_2 < \pi/2 \right\}$$

Note that the intersection of  $\tilde{X}^1$  and  $\tilde{X}^2$  is the singular configuration with  $\mathbf{z}^0 = \mathbf{0}$ .

To see that configurations in the *disjoint components* of Eq. (9.4.9) cannot be connected by continuous paths that encounter no singularities, values of  $\mathbf{z}$  interior to each are presented in Table 9.4.1, with the associated values of  $|\bar{H}'(\mathbf{z})|$ . Since signs of the determinant  $|\bar{H}'(\mathbf{z})|$  in  $\tilde{X}_1^{\text{EP}}$  and  $\tilde{X}_2^{\text{EP}}$  and in  $\tilde{X}_2^{\text{EP}}$  and  $\tilde{X}_3^{\text{EP}}$  are different, any path between configurations in these pairs of components must encounter a value 0 of the determinant; i.e., a singularity. Two possibilities exist for paths between configurations in  $\tilde{X}_1^{\text{EP}}$  and  $\tilde{X}_2^{\text{EP}}$ . First, such a path may pass through component  $\tilde{X}_3^{\text{EP}}$ , hence encounter two singularities. Second, such a path may pass through the configuration with  $z_1 = z_2 = 0$  and  $y_1 = y_2 = 1$ , at which  $|\bar{H}'(\mathbf{z})| = 0$ ; i.e., a singular configuration. Thus, paths between configurations in any pair of disjoint components must pass through a singularity.

Table 9.4.1 Values of  $|\bar{\mathbf{H}}'(\mathbf{z})|$  in Disjoint Components

$\tilde{\mathbf{X}}_i^{\text{EP}}$	Point	$z_1$	$z_2$	$y_1$	$y_2$	$ \bar{\mathbf{H}}'(\mathbf{z}) $
$\tilde{\mathbf{X}}_1^{\text{EP}}$	(1)	1	0	$\sqrt{2}$	$\sqrt{2}$	-5.5
$\tilde{\mathbf{X}}_2^{\text{EP}}$	(2)	-1	0	$\sqrt{2}$	$\sqrt{2}$	-5.5
$\tilde{\mathbf{X}}_3^{\text{EP}}$	(3)	1	$\pi$	$\sqrt{122}$	$\sqrt{122}$	0.090

Each of the configurations in Fig. 9.4.3, where dashed lines are the actuators, that occur in components (1) through (3) of Fig. 9.4.2 are associated with a mode of functionality of the manipulator. In applications, only one of several disjoint components generally represents a desired mode of functionality. The engineer will wish to focus on that component and develop information needed to assure the manipulator can be effectively controlled therein.

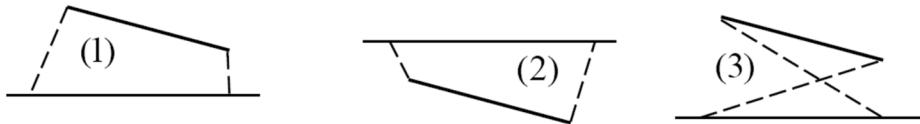


Figure 9.4.3 Manipulator Configurations in Components (1) through (3)

To confirm that the manipulator can operate in component (3) of Fig. 9.4.2 without encountering a singular configuration other than  $(0, \ )$ , a circle of radius  $r > 0$  and center at  $(0, \ )$ ,  $z_1^2 + (z_2 - \pi)^2 = r^2$ , is traversed using polar coordinate  $0 \leq \phi \leq 2\pi$ ; i.e.,  $z_1 = r \sin \phi$  and  $z_2 = \pi + r \cos \phi$ . Associated inputs  $y_1$  and  $y_2$  are calculated using Eq. (9.4.4). The path is traversed with the determinant  $|\bar{\mathbf{H}}'(\mathbf{z})|$  positive throughout, with values of  $0 < r < 3$ , so that the circle does not intersect the singular curves on the left of Fig. 9.4.2.

A form of behavior that has been extensively discussed in the literature concerns multiple manipulator configurations with the same input coordinates, called *assembly modes* (Peidro, Marin, Gil, and Reinoso, 2015; Innocenti and Parenti-Castelli, 1998). Assertions have been made that continuous paths in configuration space can connect assembly modes in different components without encountering singularities, contrary to the theory presented herein. Equating  $y_1$  and  $y_2$  in Eq. (9.4.4) and manipulating yields  $32z_1 \sin z_2 = 0$ . Solutions with  $\sin z_2 = 0$ ; i.e.,  $z_2 = 0$  or  $\pi$  yield (i)  $z_2 = 0$ ,  $\cos z_2 = 1$ , and  $z_1 = \pm\sqrt{y^2 - 1}$  and (ii)  $z_2 = \pi$ ,  $\cos z_2 = -1$ , and  $z_1 = \pm\sqrt{y^2 - 121}$ . With  $y = y_1 = y_2 = 11.2$ , this yields four configurations with the same input; i.e., *assembly modes*  $\mathbf{x} = [11.2 \ 11.2 \ 11.155 \ 0]^T$ ,  $[11.2 \ 11.2 \ -11.155 \ 0]^T$ ,  $[11.2 \ 11.2 \ 2.107 \ ]^T$ , and  $[11.2 \ 11.2 \ -2.107 \ ]^T$ . The first and second assembly modes define configurations in components (1) and (2) of Fig. 9.4.2 and the third and fourth are in component (3). There is therefore no nonsingular path between the first or second configuration and any of the other three. Since the third and fourth configurations are in component (3), which is a *path connected set*, there must be a continuous nonsingular trajectory between them. In fact, the circular trajectory  $z_1^2 + (z_2 - \pi)^2 = r^2$  with  $r = 2.107$  contains the third

and fourth assembly modes and resides entirely within component (3), providing a continuous nonsingular trajectory between these configurations. These results reinforce the fact that a path between configurations in distinct components of the regular configuration manifold necessarily encounters a singularity. Assembly modes that lie in the same component, however, may be connected by a path that lies inside that component and encounters no singularity.

#### 9.4.2 Explicit Parallel Manipulator Regular Configuration Manifold

A general explicit parallel manipulator has input  $\mathbf{y} \in \mathbb{R}^k$ , output  $\mathbf{z} \in \mathbb{R}^k$ , configuration  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^{2k}$ , and kinematic input-output equations of the form of Eq. (9.2.17); i.e.,  $\bar{\mathbf{H}}(\mathbf{z}) - \mathbf{y} = \mathbf{0}$ , where  $\bar{\mathbf{H}}(\mathbf{z})$  is continuously differentiable. The explicit relation  $\mathbf{y} = \bar{\mathbf{H}}(\mathbf{z})$  defines *global inverse kinematics* of the manipulator. The *explicit parallel configuration space* is thus

$$X^{EP} = \left\{ \mathbf{x} : \bar{\mathbf{H}}(\mathbf{z}) - \mathbf{y} = \mathbf{0} \right\} \subset \mathbb{R}^{2k} \quad (9.4.10)$$

To avoid singular configurations in  $X^{EP}$ , the *explicit parallel regular configuration space* is defined as

$$\tilde{X}^{EP} = \left\{ \mathbf{x} \in X^{EP} : |\bar{\mathbf{H}}'(\mathbf{z})| \neq 0 \right\} \quad (9.4.11)$$

Since this is an open subset of  $X^{EP}$ , it is comprised of maximal, disjoint, singularity free, path connected components. Without the need to numerically evaluate  $|\bar{\mathbf{H}}'(\mathbf{z})|$  in such components, any path between configurations in disjoint components must encounter a value of zero for the determinant, hence a singularity.

For any  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}^{EP}$ , the implicit function theorem assures existence of a continuously differentiable function  $\bar{\mathbf{G}}(\mathbf{y})$  and a neighborhood  $Y$  of  $\bar{\mathbf{y}}$  such that  $\bar{\mathbf{H}}(\bar{\mathbf{G}}(\mathbf{y})) - \mathbf{y} = \mathbf{0}$  for all  $\mathbf{y} \in Y$ ; i.e., the *local forward kinematic mapping*  $\mathbf{z} = \bar{\mathbf{G}}(\mathbf{y})$ . The relation  $\mathbf{y} = \bar{\mathbf{H}}(\mathbf{z})$  defines inverse kinematics of the manipulator in a neighborhood  $Z$  of  $\bar{\mathbf{z}}$ . For any neighborhood  $Y$  of  $\bar{\mathbf{y}}$ , the subset  $Y \times Z \equiv \left\{ \mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T : \mathbf{y} \in Y, \mathbf{z} \in Z \right\}$  is an open neighborhood of  $\bar{\mathbf{x}} \in X^{EP}$ . A countable collection of such neighborhoods  $U^i$  of configurations  $\mathbf{x}^i \in \tilde{X}^{EP}$  covers  $\tilde{X}^{EP}$  (Mendelson, 1962).

For  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in U^i = Y^i \times Z^i$ , define  $\phi^i(\mathbf{x}) = \mathbf{y} \in Y^i$ . With  $\psi^i(\mathbf{y}) \equiv [\mathbf{y}^T \quad \bar{\mathbf{G}}^T(\mathbf{y})]^T = \mathbf{x} \in \tilde{X}$  for  $\mathbf{y} \in Y^i$ ,  $\phi^i(\psi^i(\mathbf{y})) = \phi^i([\mathbf{y}^T \quad \bar{\mathbf{G}}^T(\mathbf{y})]^T) = \mathbf{y}$ , so  $\psi^i = (\phi^i)^{-1}$  on  $Y^i$ . This shows that  $\tilde{X}^{EP}$  is a  $k$ -dimensional differentiable manifold, with charts  $(\phi^i, U^i)$ , parameterized with  $\mathbf{x} = \psi^i(\mathbf{y}) \equiv [\mathbf{y}^T \quad \bar{\mathbf{G}}^T(\mathbf{y})]^T \in \tilde{X}^{EP}$  by input  $\mathbf{y}$ .

Conversely, define the mapping  $\phi^i(\mathbf{x}) = \mathbf{z} \in Z^i$ . With  $\psi^i(\mathbf{z}) \equiv [\bar{\mathbf{H}}^T(\mathbf{z}) \quad \mathbf{z}]^T = \mathbf{x} \in \tilde{X}^{EP}$  for  $\mathbf{z} \in Z^i$ ,  $\phi^i(\psi^i(\mathbf{z})) = \phi^i([\bar{\mathbf{H}}^T(\mathbf{z}) \quad \mathbf{z}]^T) = \mathbf{z}$ , for all  $\mathbf{z} \in Z^i$ . Thus,  $\psi^i = (\phi^i)^{-1}$  on  $Z^i$  and again  $\tilde{X}^{EP}$

is a  $k$ -dimensional differentiable manifold with charts  $(\phi^i, U^i)$ , parameterized with

$$\mathbf{x} = \psi^i(\mathbf{z}) \equiv [\bar{\mathbf{H}}^T(\mathbf{z}) \quad \mathbf{z}]^T \in \tilde{X}$$
 by output  $\mathbf{z}$ .

With either parameterization, the regular manipulator configuration space for a general explicit parallel manipulator is a  $k$ -dimensional differentiable manifold. Its disjoint maximal, singularity free, path connected components are thus ideally suited for manipulator analysis and control. As for serial manipulators, numerical methods for explicit parallel manipulator kinematics are presented in Section 9.10.

A simple planar explicit parallel manipulator is shown to have a global inverse kinematic solution, but to have singularities in forward kinematics. Three singularity free components are defined and it is shown that any trajectory between them must encounter a singularity. The regular configuration space for a general explicit parallel manipulator is shown to be a differentiable manifold, parameterized by either input or output coordinates.

## 9.5 Implicit Manipulator Kinematics

Numerous manipulators arise in which there is no global mapping from output to input coordinates or input to output coordinates. Such manipulators are neither serial nor explicit parallel, with input-output equations of the form  $\mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}$ , and are called *implicit manipulators*.

### 9.5.1 Model Implicit Manipulators

The *planar parallel manipulator* shown in Fig. 9.5.1 consists of a platform with ends  $B_1$  and  $B_2$  and three degrees of freedom moving in the plane. A serial chain on the left with a single degree of freedom strut of length  $y_3 > 0$  as input and a serial chain on the right comprised of two bars with a pair of rotations  $y_1$  and  $y_2$  as inputs support the platform. Since generalized coordinates of the platform are  $\mathbf{q} = \mathbf{z}$ , they need not be incorporated in the manipulator kinematic equations. *Input-output equations* are thus

$$\mathbf{F}(\mathbf{y}, \mathbf{z}) = \begin{bmatrix} 1 + \cos y_1 + \cos(y_1 + y_2) - z_1 - \cos z_3 \\ \sin y_1 + \sin(y_1 + y_2) - z_2 - \sin z_3 \\ (1/2)((y_3)^2 - (z_1 - \cos z_3)^2 - (z_2 - \sin z_3)^2) \end{bmatrix} = \mathbf{0} \quad (9.5.1)$$

As shown geometrically in Section 9.1.3, this is an *implicit parallel manipulator with input coordinates*  $\mathbf{y} = [y_1 \ y_2 \ y_3]^T \in \mathbb{R}^3$ , *output coordinates*  $\mathbf{z} = [z_1 \ z_2 \ z_3]^T \in \mathbb{R}^3$ , and *manipulator coordinates*  $\mathbf{x} = [\mathbf{y}^T \ \mathbf{z}^T]^T \in \mathbb{R}^6$ . The *implicit manipulator configuration space* is thus

$$X^I = \{\mathbf{x}: \mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}\} \subset \mathbb{R}^6 \quad (9.5.2)$$

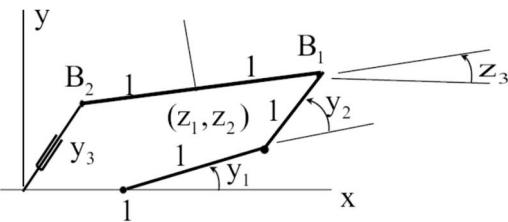


Figure 9.5.1 Implicit Parallel Manipulator

To avoid singular configurations in  $X^I$  and to assure local existence of forward and inverse kinematic mappings, Jacobians of  $\mathbf{F}(\mathbf{y}, \mathbf{z})$  with respect to  $\mathbf{y}$  and  $\mathbf{z}$  are

$$\mathbf{F}_y(\mathbf{y}, \mathbf{z}) = \begin{bmatrix} -\sin y_1 - \sin(y_1 + y_2) & -\sin(y_1 + y_2) & 0 \\ \cos y_1 + \cos(y_1 + y_2) & \cos(y_1 + y_2) & 0 \\ 0 & 0 & y_3 \end{bmatrix}$$

$$\mathbf{F}_z(\mathbf{y}, \mathbf{z}) = \begin{bmatrix} -1 & 0 & \sin z_3 \\ 0 & -1 & -\cos z_3 \\ -(z_1 - \cos z_3) & -(z_2 - \sin z_3) & -(z_1 \sin z_3 - z_2 \cos z_3) \end{bmatrix}$$

with the associated determinants

$$\begin{aligned} |\mathbf{F}_y(\mathbf{y}, \mathbf{z})| &= y_3 \sin y_2 \\ |\mathbf{F}_z(\mathbf{y}, \mathbf{z})| &= 2z_2 \cos z_3 - 2z_1 \sin z_3 \end{aligned} \quad (9.5.3)$$

Since  $y_3 > 0$ , the singularity free *implicit regular configuration space* is

$$\tilde{X}^I = \left\{ \mathbf{x} \in X^I : \sin y_2 \neq 0 \text{ and } z_2 \cos z_3 - z_1 \sin z_3 \neq 0 \right\}$$

with four *components*,

$$\begin{aligned} \tilde{X}_1^I &= \left\{ \mathbf{x} \in X^I : 0 < y_2 < \pi, z_2 \cos z_3 - z_1 \sin z_3 > 0 \right\} \\ \tilde{X}_2^I &= \left\{ \mathbf{x} \in X^I : -\pi < y_2 < 0, z_2 \cos z_3 - z_1 \sin z_3 > 0 \right\} \\ \tilde{X}_3^I &= \left\{ \mathbf{x} \in X^I : 0 < y_2 < \pi, z_2 \cos z_3 - z_1 \sin z_3 < 0 \right\} \\ \tilde{X}_4^I &= \left\{ \mathbf{x} \in X^I : -\pi < y_2 < 0, z_2 \cos z_3 - z_1 \sin z_3 < 0 \right\} \end{aligned} \quad (9.5.4)$$

that are maximal, disjoint, singularity free, path connected subsets of  $\tilde{X}^I$ , with associated configurations shown in Fig. 9.5.2. Since inequalities involving both  $\mathbf{y}$  and  $\mathbf{z}$  appear in definition of these components, they define three dimensional subsets of  $\mathbb{R}^6$  that involve both  $\mathbf{y}$  and  $\mathbf{z}$ . Boundaries of these components are *singular configurations* in which the bars in the chain on the right are colinear or the actuator on the left and the platform are colinear.

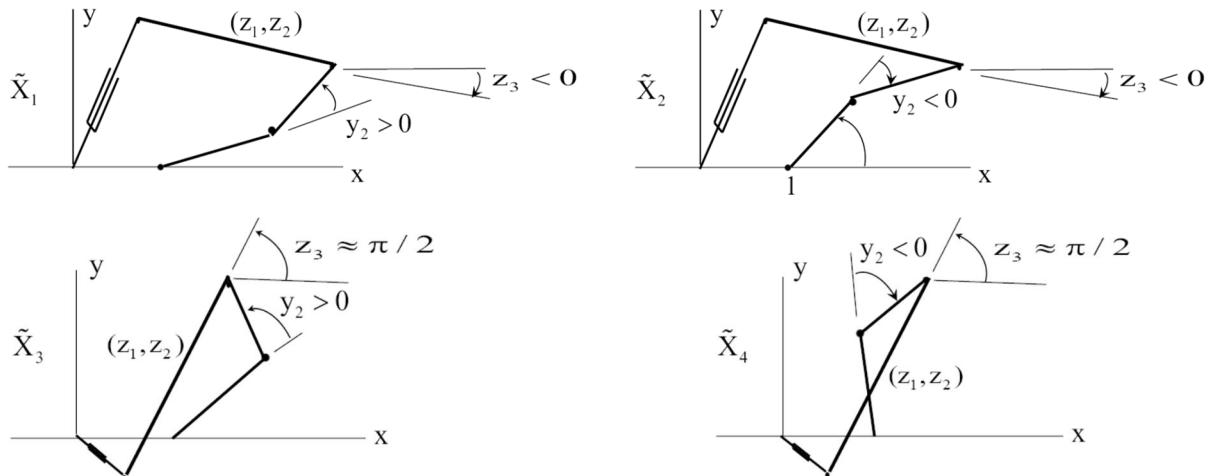


Figure 9.5.2 Manipulator Configurations in  $\tilde{X}_1$  Through  $\tilde{X}_4$

The components of  $\tilde{X}^I$  are subsets of  $\mathbb{R}^6$ , so drawing a picture of them is out of the question. Nevertheless, it is in these sets that the control engineer must operate. The fact that two

determinants must be nonzero in components for this example, in contrast to the requirement that just one determinant must be nonzero for serial and explicit parallel manipulators, suggests that this category of manipulator will generally have more components than the prior two categories; i.e., more disjoint sets in which effective control is possible.

A second example of an implicit manipulator with the same geometric structure as the planar serial manipulator of Fig. 9.3.1 is shown in Fig. 9.5.3. The rotation angle  $y_1$  in the serial manipulator is replaced by the length  $y_1$  of an actuator between the point  $(2,0)$  and the outboard end of body one. Since the length actuator is not integral to the first joint, this does not fit the definition of a serial manipulator.

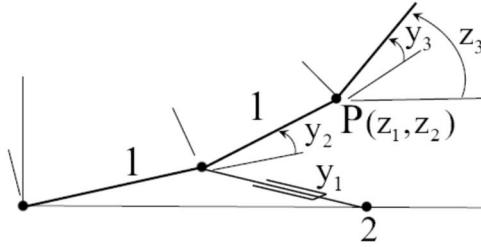


Figure 9.5.3 Implicit Manipulator with Serial Geometry

To see further that there is no global forward kinematic mapping for this manipulator, note that when  $y_1 = 1$ , the first bar and the actuator are colinear. In this configuration, it is not possible to decrease the actuator length from  $y_1 = 1$ . Further, an increase in actuator length from  $y_1 = 1$  yields two distinct configurations of the first body, one with the outboard end above the horizontal axis and one with the outboard end below. The configuration with  $y_1 = 1$  is thus *singular* and there is no globally valid forward kinematic mapping.

Many practical implicit manipulators with serial geometric structure, but with actuators that are not embedded in joints are used in applications. The excavator of Fig. 9.1.9 has actuators of this type, as do many other material handling and construction manipulators.

### 9.5.2 Implicit Manipulator Regular Configuration Manifold

For a general implicit manipulator with input coordinates  $\mathbf{y} \in \mathbb{R}^k$ , output coordinates  $\mathbf{z} \in \mathbb{R}^k$ , and configuration coordinates  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^{2k}$ , kinematic input-output equations are of the form

$$\mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0} \quad (9.5.5)$$

where  $\mathbf{F}(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^k$  is a continuously differentiable function of its arguments. The *implicit manipulator configuration space* is thus

$$X^I = \{\mathbf{x}: \mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}\} \subset \mathbb{R}^{2k} \quad (9.5.6)$$

To avoid singular configurations in  $X^I$ , the *implicit regular configuration space* is defined as

$$\tilde{X}^I = \left\{ \mathbf{x} \in X^I : |\mathbf{F}_y(\mathbf{y}, \mathbf{z})| \neq \mathbf{0} \text{ and } |\mathbf{F}_z(\mathbf{y}, \mathbf{z})| \neq \mathbf{0} \right\} \quad (9.5.7)$$

For any  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}^I$ , the implicit function theorem assures existence of a continuously differentiable function  $\mathbf{H}(\mathbf{z})$  and a neighborhood  $Z$  of  $\bar{\mathbf{z}}$  such that  $\mathbf{F}(\mathbf{H}(\mathbf{z}), \mathbf{z}) = \mathbf{0}$  for all  $\mathbf{z} \in Z$ ; i.e., the relation  $\mathbf{y} = \mathbf{H}(\mathbf{z})$  defines inverse kinematics of the manipulator in a neighborhood  $Z$  of  $\bar{\mathbf{z}}$ . Similarly, the implicit function theorem assures existence of a continuously differentiable function  $\mathbf{G}(\mathbf{y})$  and a neighborhood  $Y$  of  $\bar{\mathbf{y}}$  such that  $\mathbf{F}(\mathbf{y}, \mathbf{G}(\mathbf{y})) = \mathbf{0}$  for all  $\mathbf{y} \in Y$ ; i.e., the relation  $\mathbf{z} = \mathbf{G}(\mathbf{y})$  defines forward kinematics of the manipulator in a neighborhood  $Y$  of  $\bar{\mathbf{y}}$ . For neighborhoods  $Y$  of  $\bar{\mathbf{y}}$  and  $Z$  of  $\bar{\mathbf{z}}$ , the set

$Y \times Z \equiv \left\{ \mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T : \mathbf{y} \in Y, \mathbf{z} \in Z \right\}$  is an open neighborhood of  $\bar{\mathbf{x}} \in \tilde{X}^I$ . A countable collection of such neighborhoods  $U^i$  of configurations  $\mathbf{x}^i \in \tilde{X}^I$  covers  $\tilde{X}^I$  (Mendelson, 1962).

For  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in U^i = Y^i \times Z^i$ , define  $\phi^i(\mathbf{x}) = \mathbf{y} \in Y^i$ . Defining  $\psi^i(\mathbf{y}) = [\mathbf{y}^T \quad \mathbf{G}^T(\mathbf{y})]^T = \mathbf{x} \in \tilde{X}^I$  for  $\mathbf{y} \in Y^i$ ,  $\phi^i(\psi^i(\mathbf{y})) = \phi^i([\mathbf{y}^T \quad \mathbf{G}^T(\mathbf{y})]^T) = \mathbf{y}$ , so  $\psi^i = (\phi^i)^{-1}$  in  $Y^i$ . This shows that  $\tilde{X}^I$  is a k-dimensional *differentiable manifold* with *charts*  $(\phi^i, U^i)$ , parameterized by  $\mathbf{x} = \psi^i(\mathbf{y}) = [\mathbf{y}^T \quad \mathbf{G}^T(\mathbf{y})]^T \in \tilde{X}^I$  as functions of input  $\mathbf{y}$ .

Conversely, define the mapping  $\hat{\phi}^i(\mathbf{x}) = \mathbf{z} \in Z^i$ . With  $\hat{\psi}^i(\mathbf{z}) = [\mathbf{H}^T(\mathbf{z}) \quad \mathbf{z}]^T = \mathbf{x} \in \tilde{X}^I$  for  $\mathbf{z} \in Z^i$ ,  $\hat{\phi}^i(\hat{\psi}^i(\mathbf{z})) = \hat{\phi}^i([\mathbf{H}^T(\mathbf{z}) \quad \mathbf{z}^T]^T) = \mathbf{z}$ , for all  $\mathbf{z} \in Z^i$ . Thus,  $\hat{\psi}^i = (\hat{\phi}^i)^{-1}$  in  $Z^i$  and again  $\tilde{X}^I$  is a k-dimensional differentiable manifold with charts  $(\hat{\phi}^i, U^i)$ , parameterized by  $\mathbf{x} = \hat{\psi}^i(\mathbf{z}) = [\mathbf{H}^T(\mathbf{z}) \quad \mathbf{z}]^T \in \tilde{X}^I$  as functions of output  $\mathbf{z}$ .

In either case, the regular manipulator configuration space for an implicit manipulator is a k-dimensional differentiable manifold. Its maximal, singularity free, path connected components are thus ideally suited for manipulator analysis and control.

In contrast to serial and explicit parallel manipulators whose regular configuration space is defined by just one determinant condition, for implicit manipulators two determinant conditions must be satisfied. This increases the complexity of conditions to be satisfied and the number of sets of singularities that must be avoided to remain in singularity free components of the regular configuration manifold during system control.

## 9.6 Compound Manipulator Kinematics

Numerous heavy duty *material handling and construction manipulators* are based on *mechanisms with closed kinematic loops* that support extremely high loads. In such applications, the kinematic model necessarily involves generalized coordinates, in addition to input and output coordinates, that cannot be eliminated to create analytical input-output equations of the form of Eq. (9.2.14), (9.2.17), or (9.5.5). Such manipulators are not serial, explicit parallel, or implicit and are called *compound manipulators*.

### 9.6.1 A Model Compound Manipulator with Closed Kinematic Loop

The *front-end loader* shown in Fig. 9.6.1 is typical of high load capacity manipulators used in construction. The mechanism upon which this manipulator is based has three bodies that move relative to body 0 (chassis), whose positions and orientations are defined in the chassis-fixed x-y frame by three *generalized coordinates* shown in Fig. 9.6.1,  $\mathbf{q} = [q_1 \ q_2 \ q_3]^T \in \mathbb{R}^3$ . In this model, motion of the chassis is suppressed. *Hydraulic actuator inputs*  $y_i > 0$ ,  $\mathbf{y} = [y_1 \ y_2]^T \in \mathbb{R}^2$ , control motion of the system, and outputs are orientation of the bucket and elevation of its tip,  $\mathbf{z} = [z_1 \ z_2]^T = [q_1 + q_3 \ y_{D_2}]^T \in \mathbb{R}^2$ . The configuration of the manipulator is thus defined by *configuration coordinates*  $\mathbf{x} = [\mathbf{y}^T \ \mathbf{q}^T \ \mathbf{z}^T]^T \in \mathbb{R}^7$ . Body 2 slides along the axis of body 1 (boom) and is connected to point  $D_1$  on body 3 (bucket) by a bar of length 2m, to control orientation of the bucket. The manipulator is not serial, since the mechanism contains a *kinematic closed loop*,  $C_1-B_2-D_1-C_1$ . It is not parallel, since there are no serial chains that support a moving platform. As will be shown, its generalized coordinates cannot be analytically eliminated to obtain an input-output model, so it is a *compound manipulator*.

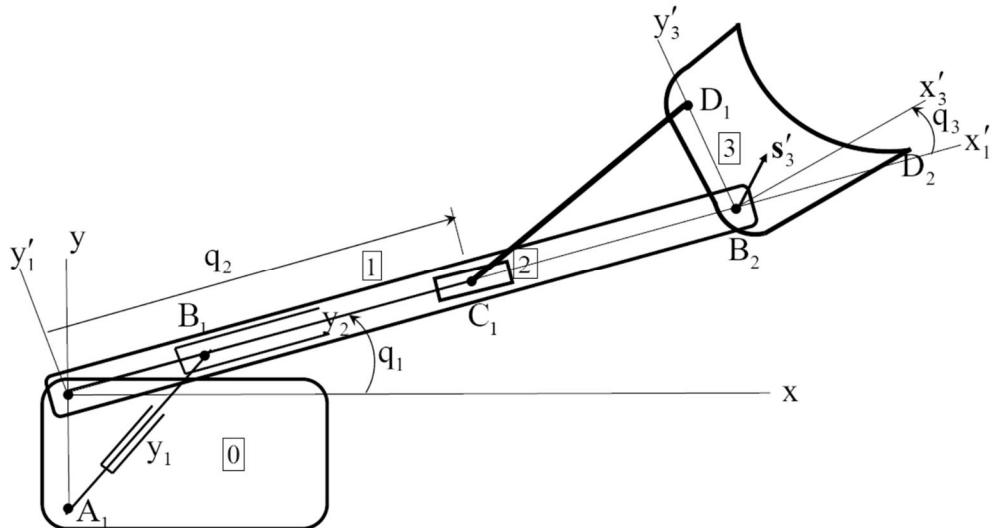


Figure 9.6.1 Front-End Loader

Body-fixed vectors that locate points of connection and the tip  $D_2$  of the bucket are

$$\mathbf{r}^{A_1} = \mathbf{r}'^{A_1} = -1.5\mathbf{u}_y, \mathbf{r}'^{B_1} = \mathbf{u}_x, \mathbf{r}'^{B_2} = 4\mathbf{u}_x, \mathbf{r}'^{C_1} = q_2\mathbf{u}_x, \mathbf{r}'^{D_1} = \mathbf{u}_y, \text{ and } \mathbf{r}'^{D_2} = \mathbf{u}_x - 0.5\mathbf{u}_y, \text{ where}$$

$$\mathbf{u}_x = [1 \ 0]^T \text{ and } \mathbf{u}_y = [0 \ 1]^T. \text{ Orientation transformation matrices for bodies 1 and 3 are}$$

$$\mathbf{A}_1 = \begin{bmatrix} \cos q_1 & -\sin q_1 \\ \sin q_1 & \cos q_1 \end{bmatrix} \text{ and } \mathbf{A}_3 = \begin{bmatrix} \cos(q_1 + q_3) & -\sin(q_1 + q_3) \\ \sin(q_1 + q_3) & \cos(q_1 + q_3) \end{bmatrix}. \text{ Vectors that define key points}$$

$$\text{on the loader in the } x\text{-}y \text{ plane are } \mathbf{r}^{B_1} = \mathbf{A}_1\mathbf{u}_x, \mathbf{r}^{B_2} = 4\mathbf{A}_1\mathbf{u}_x, \mathbf{r}^{C_1} = q_2\mathbf{A}_1\mathbf{u}_x,$$

$$\mathbf{r}^{D_1} = \mathbf{r}^{B_2} + \mathbf{A}_3\mathbf{u}_y = 4\mathbf{A}_1\mathbf{u}_x + \mathbf{A}_3\mathbf{u}_y, \text{ and } \mathbf{r}^{D_2} = \mathbf{r}^{B_2} + \mathbf{A}_3(\mathbf{u}_x - 0.5\mathbf{u}_y) = 4\mathbf{A}_1\mathbf{u}_x + \mathbf{A}_3(\mathbf{u}_x - 0.5\mathbf{u}_y). \text{ The vector } \mathbf{s}'_3 \text{ locates the centroid of the bucket in its body-fixed } x'_3\text{-}y'_3 \text{ reference frame.}$$

The kinematic constraint due to the 2m bar that connects points  $C_1$  and  $D_1$  is

$$\begin{aligned} (\mathbf{r}^{D_1} - \mathbf{r}^{C_1})^T(\mathbf{r}^{D_1} - \mathbf{r}^{C_1}) - 4 &= ((4 - q_2)\mathbf{u}_x^T \mathbf{A}_1^T + \mathbf{u}_y^T \mathbf{A}_3^T)((4 - q_2)\mathbf{A}_1\mathbf{u}_x + \mathbf{A}_3\mathbf{u}_y) - 4 \\ &= (4 - q_2)^2 - 2(4 - q_2)\sin q_3 - 3 = 0 \end{aligned}$$

Using the quadratic formula, this is equivalent to  $(4 - q_2) = \sin q_3 \pm \sqrt{\sin^2 q_3 + 3}$ . From the geometry of Fig. 9.6.1,  $1 \leq q_2 \leq 3$ , so  $3 \geq 4 - q_2 \geq 1$ . This requires selection of the plus sign, so the *kinematic constraint* may be written as

$$(\mathbf{q}) = q_2 - \left(4 - \sin q_3 - \sqrt{\sin^2 q_3 + 3}\right) = 0 \quad (9.6.1)$$

*Input equations* that relate  $\mathbf{y}$  and  $\mathbf{q}$  are

$$\Psi(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} (\mathbf{r}^{B_1} - \mathbf{r}^{A_1})^T(\mathbf{r}^{B_1} - \mathbf{r}^{A_1}) - y_1^2 \\ q_2 - y_2 \end{bmatrix} = \begin{bmatrix} 3\sin q_1 + 3.25 - y_1^2 \\ q_2 - y_2 \end{bmatrix} = \mathbf{0} \quad (9.6.2)$$

where  $y_1$  and  $y_2$  are positive, and *output equations* that relate  $\mathbf{q}$  and  $\mathbf{z}$  are

$$\Gamma(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} q_1 + q_3 - z_1 \\ \mathbf{u}_y^T \mathbf{r}^{D_2} - z_2 \end{bmatrix} = \begin{bmatrix} q_1 + q_3 - z_1 \\ 4\sin q_1 + \sin(q_1 + q_3) - 0.5\cos(q_1 + q_3) - z_2 \end{bmatrix} = \mathbf{0} \quad (9.6.3)$$

Due to nonlinearity of these equations,  $\mathbf{q}$  cannot be analytically eliminated from the formulation.

Rather than carrying out an analysis of this model manipulator, prior to presenting a general formulation that is essentially identical for manifolds of compound manipulators, the general formulation is first presented.

### 9.6.2 Compound Manipulator Regular Configuration Manifold

The regular manipulator configuration space  $\tilde{X}^C$  of Eq. (9.2.11), parameterized by the kinematic relations of Section 9.2, is a differentiable manifold. To prove that this is the case, for any configuration  $\mathbf{x}^i \in \tilde{X}^C$ , there is a neighborhood  $W^i = Y^i \times Q^i \times Z^i \subset \tilde{X}^C$  of  $\mathbf{x}^i$  in which one may define the mapping  $\phi^i(\mathbf{x}) = \mathbf{y}$ . With the continuously differentiable functions

$\mathbf{f}^i : R^k \rightarrow R^n$  and  $\mathbf{h}^i : R^n \rightarrow R^k$  defined in Eqs. (9.2.12) and (9.2.13), the mapping  $\psi^i : Y^i \rightarrow \tilde{X}^C$ ,

$\Psi^i(\mathbf{y}) = \begin{bmatrix} \mathbf{y}^T & \mathbf{f}^i(\mathbf{y})^T & \mathbf{h}^i(\mathbf{f}^i(\mathbf{y}))^T \end{bmatrix}^T = \mathbf{x} \in \tilde{X}^C$ , is differentiable since functions  $\mathbf{f}^i$  and  $\mathbf{h}^i$  are differentiable. Since  $\Psi^i(\phi^i(\mathbf{x})) = \begin{bmatrix} \mathbf{y}^T & \mathbf{f}^i(\mathbf{y})^T & \mathbf{h}^i(\mathbf{f}^i(\mathbf{y}))^T \end{bmatrix}^T = \mathbf{x}$  for all  $\mathbf{x} \in W^i$ ,  $\phi^i = (\Psi^i)^{-1}$ . The homeomorphisms  $\Psi^i$  and  $\phi^i$  on  $W^i$ , called charts  $(\phi^i, W^i)$  for a countable number of neighborhoods  $W^i$  that cover  $\tilde{X}^C$ ; i.e.,  $\tilde{X}^C = \bigcup_{i=1}^{\infty} W^i$ , define  $\tilde{X}^C$  as a differentiable manifold that is parameterized with  $\mathbf{x} = \Psi^i(\mathbf{y}) = \begin{bmatrix} \mathbf{y}^T & \mathbf{f}^i(\mathbf{y})^T & \mathbf{h}^i(\mathbf{f}^i(\mathbf{y}))^T \end{bmatrix}^T \in \tilde{X}^C$  by  $\mathbf{y}$ .

Conversely, for  $\mathbf{x}^i \in W^i$ , one may define  $\phi^i(\mathbf{x}) = \mathbf{z}$ . With continuously differentiable functions  $\mathbf{e}^i : R^k \rightarrow R^n$  and  $\mathbf{g}^i : R^n \rightarrow R^k$  defined in Eqs. (9.2.15) and (9.2.16), the mapping  $\Psi^i(\mathbf{z}) = \begin{bmatrix} \mathbf{g}^i(\mathbf{e}^i(\mathbf{z}))^T & \mathbf{e}^i(\mathbf{z})^T & \mathbf{z}^T \end{bmatrix}^T = \mathbf{x} \in \tilde{X}^C$  satisfies  $\Psi^i(\phi^i(\mathbf{x})) = \begin{bmatrix} \mathbf{g}^i(\mathbf{e}^i(\mathbf{z}))^T & \mathbf{e}^i(\mathbf{z})^T & \mathbf{z}^T \end{bmatrix}^T = \mathbf{x}$  for all  $\mathbf{x} \in W^i$ , so  $\phi^i = \Psi^{i(-1)}$ . This again shows that  $\tilde{X}^C$  is a differentiable manifold, but parameterized with  $\mathbf{x} = \Psi^i(\mathbf{z}) = \begin{bmatrix} \mathbf{g}^i(\mathbf{e}^i(\mathbf{z}))^T & \mathbf{e}^i(\mathbf{z})^T & \mathbf{z}^T \end{bmatrix}^T \in \tilde{X}^C$  by output  $\mathbf{z}$ . With requirements for both forward and inverse kinematics embedded in the conditions of Eq. (9.2.11), there is a duality between input and output coordinates.

With either parameterization, the *regular manipulator configuration space* for a *general compound manipulator* is a  $k$ -dimensional *differentiable manifold*. Its disjoint maximal, singularity free, path connected components are thus ideally suited for manipulator analysis and control. As for serial and explicit parallel manipulators, numerical methods for compound manipulator kinematics are presented in Section 9.10.6.

### 9.6.3 Singularity Free Manifold Components for the Front-End Loader

Jacobians of Eqs. (9.6.1) through (9.6.3) for the *front-end loader* are

$$q(\mathbf{q}) = \begin{bmatrix} 0 & 1 & (1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \cos q_3 \end{bmatrix}$$

$$\Psi_y(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} -2y_1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\Psi_q(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} 3 \cos q_1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\Gamma_z(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\Gamma_q(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} 1 & 0 & 1 \\ 4 \cos q_1 + \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) & 0 & \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) \end{bmatrix}$$

Since  $y_1 > 0$ ,  $|\Psi_y(\mathbf{q}, \mathbf{y})| = 2y_1 \neq 0$  and  $|\Gamma_z(\mathbf{q}, \mathbf{z})| = 1 \neq 0$ . Additional determinants needed are

$$|\Omega_q(\mathbf{q}, \mathbf{y})| = \begin{vmatrix} 0 & 1 & (1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \cos q_3 \\ 3 \cos q_1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix} = 3(1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \cos q_1 \cos q_3 \quad (9.6.4)$$

where  $(1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \neq 0$ , and

$$|\Lambda_q(\mathbf{q}, \mathbf{z})| = \begin{vmatrix} 0 & 1 & (1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \cos q_3 \\ 1 & 0 & 1 \\ 4 \cos q_1 + \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) & 0 & \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) \end{vmatrix} = 4 \cos q_1 \quad (9.6.5)$$

From Eqs. (9.2.11), (9.6.4), and (9.6.5), disjoint components of  $\tilde{X}^C$  are

$$\begin{aligned} \tilde{X}_1^C &= \{\mathbf{x} : \cos q_1 > 0 \text{ and } \cos q_3 > 0\} = \{\mathbf{x} : -\pi/2 < q_1 < \pi/2, -\pi/2 < q_3 < \pi/2\} \\ \tilde{X}_2^C &= \{\mathbf{x} : \cos q_1 > 0 \text{ and } \cos q_3 < 0\} = \{\mathbf{x} : -\pi/2 < q_1 < \pi/2, \pi/2 < q_3 < 3\pi/2\} \\ \tilde{X}_3^C &= \{\mathbf{x} : \cos q_1 < 0 \text{ and } \cos q_3 > 0\} = \{\mathbf{x} : \pi/2 < q_1 < 3\pi/2, -\pi/2 < q_3 < \pi/2\} \\ \tilde{X}_4^C &= \{\mathbf{x} : \cos q_1 < 0 \text{ and } \cos q_3 < 0\} = \{\mathbf{x} : \pi/2 < q_1 < 3\pi/2, \pi/2 < q_3 < 3\pi/2\} \end{aligned} \quad (9.6.6)$$

where

$$\begin{aligned} \mathbf{x} &= [y_1, y_2, q_1, q_2, q_3, z_1, z_2]^T \\ &= \left[ \sqrt{3.25 + 3 \sin q_1}, 4 - \sin q_3 - \sqrt{\sin^2 q_3 + 3}, q_1, 4 - \sin q_3 - \sqrt{\sin^2 q_3 + 3}, q_3, q_1 + q_3, 4 \sin q_1 + \sin(q_1 + q_3) - 0.5 \cos(q_1 + q_3) \right]^T \end{aligned} \quad (9.6.7)$$

The four maximal, singularity free, path connected *components* of Eq. (9.6.6) are two-dimensional subsets of  $\tilde{X}^C \subset \mathbb{R}^7$ , parameterized by  $q_1$  and  $q_3$ , so drawing their pictures is not possible. Nevertheless, the control engineer must function in these components, which have been evaluated without defining sets of singularities; i.e., their boundaries in  $X^C$ .

To assist in visualization, projections of the components of Eq. (9.6.6) onto the  $z_1$ - $z_2$  plane are presented in Figs. 9.6.2 and 9.6.3. Components in Fig. 9.6.2 are associated with  $q_1$  in the range  $-\pi/2 < q_1 < \pi/2$  and those in Fig. 9.6.3 are associated with  $q_1$  in the range

$\pi/2 < q_1 < 3\pi/2$ . In  $\tilde{X}_1^C$  and  $\tilde{X}_2^C$ , the loader is in the intended configuration shown in Fig. 9.6.1, whereas  $\tilde{X}_3^C$  and  $\tilde{X}_4^C$  consist of unwanted configurations with the boom oriented rearward. Only in  $\tilde{X}_1^C$  can the loader carry out its scooping action near ground level, as the chassis moves forward and the bucket is loaded in a near horizontal orientation; i.e., with  $z_1 \approx 0$ . The bucket is

then oriented upward ( $z_1 > 0$ ) to contain the load and elevated as the chassis moves toward the target hauler. The bucket is then oriented downward ( $z_1 < 0$ ) to dump the load into the hauler. Thus, only one of the four components represents the intended functionality of the loader.

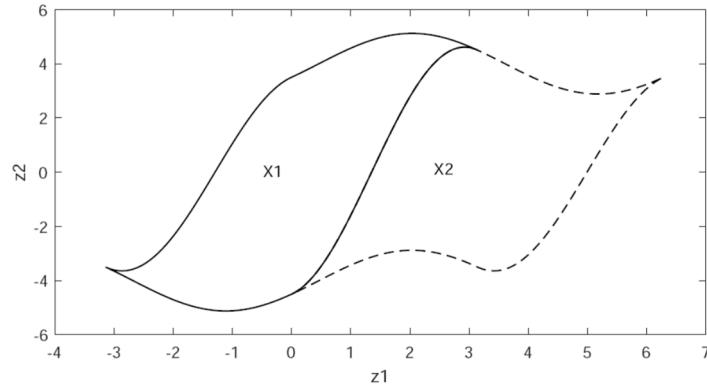


Figure 9.6.2 Loader Components  $X1 = \tilde{X}_1^C$  and  $X2 = \tilde{X}_2^C$

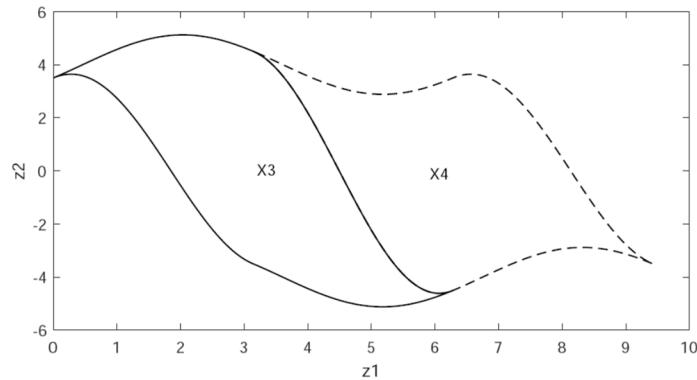


Figure 9.6.3 Loader Components  $X3 = \tilde{X}_3^C$  and  $X4 = \tilde{X}_4^C$

*Boundaries of components* shown in Figs. 9.6.2 and 9.6.3 are singularities. To determine whether it is possible to have nonsingular trajectories that connect configurations in these components, midpoints in the ranges of  $q_1$  and  $q_3$  that parameterize the components in Eq. (9.6.6) are presented in Table 9.6.1, with associated values of determinants in Eqs. (9.6.4) and (9.6.5). Since no pair of midpoints in Table 9.6.1 yields the same signs of  $|\Omega_q(\mathbf{q}, \mathbf{y})|$  and  $|\Lambda_q(\mathbf{q}, \mathbf{z})|$ , which is true for all configurations in the components, no points in the associated components can be connected by a continuous trajectory in configuration space without encountering a singularity, consistent with the theory.

Table 9.6.1  $|\Omega_q(\mathbf{q}, \mathbf{y})|$  and  $|\Lambda_q(\mathbf{q}, \mathbf{z})|$

$\tilde{\mathbf{X}}_i^C$	Point	$q_1$	$q_3$	$ \Omega_q(\mathbf{q}, \mathbf{y}) $	$ \Lambda_q(\mathbf{q}, \mathbf{z}) $
$\tilde{\mathbf{X}}_1^C$	(1)	0	0	3	4
$\tilde{\mathbf{X}}_2^C$	(2)	0		-3	4
$\tilde{\mathbf{X}}_3^C$	(3)		0	-3	-4
$\tilde{\mathbf{X}}_4^C$	(4)			3	-4

In contrast with prior categories, compound manipulators necessarily include generalized, input, and output coordinates in their kinematic representation. This and the fact that the full compliment of four determinant conditions must be satisfied in definition of the regular configuration space generally yields a higher number of components and associated singularities. Nevertheless, the regular configuration space is shown to be a differentiable manifold, with parameterizations that are computed in Section 9.10.6 and are ideally suited for defining ODE of manipulator dynamics.

## 9.7 Singularity Free Manifold Components and Manipulator Singularities

It is important to note that local existence of manipulator forward and inverse kinematics for each of the four categories of manipulator dictates that the associated regular configuration space  $\tilde{X}$  is a differentiable manifold. An abstract mathematical development was not required to obtain this result. In using manifold theory, however, it is important to note that a set of configurations is a differentiable manifold only if charts  $(\phi^i, W^i)$  are defined; i.e., sets without charts are not necessarily differentiable manifolds. Numerical methods for forward and inverse kinematics and for systematic formulation of ODE of dynamics on components  $\tilde{X}_i$ , based on these parameterizations, are presented in Sections 9.10 and 9.11, using the differentiable manifold structure.

A distinction between the approach presented herein and much of the literature on manipulator kinematics is the treatment of *singular configurations*. The focus of this development is on *components of regular manipulator configuration space*, which are disjoint, maximal, singularity free, path connected sets in which the manipulator can be effectively controlled. As a biproduct, boundaries of closures of these components are sets of singularities. This approach is similar to a modest literature (Chablat and Wenger, 1998) that defines singularity free domains for a restricted category of manipulators as *aspects*. In contrast, there is a massive literature on manipulator singularities; e.g., (Bohigas, Montserrat, and Ros 2017; Muller and Zlatanov, 2019) and references cited therein. A tenant of the manipulator singularity literature is that a knowledge of sets of singularities enables their avoidance in manipulator operation. Even if successful, this *singularity avoidance approach* fails to characterize the *differentiable manifold structure* presented herein, particularly parameterizations that are based on the manifold formulation and enable quantitative kinematic analysis and formulation of ODE of dynamics that are needed for control system design and implementation (Siciliano, Sciavicco, Villani, and Oriolo, 2010).

The analysis and examples in the previous sections verify theoretical predictions regarding properties of *differentiable manifold components*, but suggests difficulties in implementation using only *singularity information*. The intricate process of computing sets of singular configurations and piecing them together in  $(n + 2k)$ -dimensional configuration space to define boundaries of compound manipulator components  $\tilde{X}_i^C$  is problematic, for even the modest front-end loader manipulator. With evaluation of a large number of  $(n + 2k - 1)$ -dimensional sets of singular configurations in  $X^C \subset R^{n+2k}$ , it is not clear how to identify and characterize the likewise large number of singularity free components  $\tilde{X}_i^C$  in which the manipulator will function reliably. An example of an elementary spatial compound manipulator with  $n + 2k = 7$ , 16 disjoint components, and 32 sets of singularities is presented in (Haug, 2021b). Piecing these sets of singularities together to define manifold components is akin to solving a jigsaw puzzle in 7-dimensional space, a process that is unlikely to succeed. Even if successful, such a process defines only boundaries of manifold components, not their structure nor parameterizations that are key to manipulator analysis and control. In spite of the potential offered by singularity free manifold components and deficiencies associated with relying only on a knowledge of singularities, Web of Science literature searches using keywords “singularity free

manipulator configuration spaceö yielded 61 papers, ömanipulator working modes and aspectsö yielded 21 papers, and ömanipulator singularitiesö yielded 2,610 papers. Achieving the potential for definition and computation of singularity free manipulator regular configuration space components for use in manipulator design and control would appear to justify a higher research priority.

The manipulator differentiable manifold formulation presented in Sections 9.3 through 9.6 is used in Section 9.10 to create computational algorithms that function reliably on manipulator components in carrying out forward and inverse kinematics for each of the four categories of manipulator defined. Results of these computations and manifold parameterizations are used to create ODE of manipulator dynamics in Section 9.11 that can be implemented in real-time on manifold components, without requiring ad-hoc derivation. These developments, comprising the second and third functions identified in Fig. 9.1.1, provide a rational foundation for implementing modern methods of manipulator control.

The differentiable manifold structure presented for serial, explicit parallel, implicit, and compound manipulator kinematics defines and characterizes disjoint, maximal, singularity free, path connected components of regular configuration space for a broad spectrum of manipulators. Differential geometry shows that continuous paths in configuration space between these components must encounter singularities, but that continuous paths between all pairs of configurations in the same component exist. These components are in fact maximal singularity free domains of manipulator configuration space in which system control can be confidently carried out. While manifold components are analytically constructed for model manipulators, computational methods for their construction in more realistic applications remain an open research topic.

The differentiable manifold structure established for all four categories of manipulator provides a sound basis for kinematic and dynamic analysis that is not realized in a study of only singular manipulator configurations.

## 9.8 Foundations of Manipulator Dynamics

Formulations presented in the literature for manipulator dynamics tend to focus on serial and explicit parallel manipulators. Even for these ideal manipulators, equations of dynamics are derived using ad-hoc methods (Merlet, 2006; Siciliano, Sciavicco, Villani, and Oriolo, 2010; Lynch, and Park, 2017), or recursive dynamics methods that yield ordinary differential equations (ODE) of motion for serial manipulators (Featherstone, 2008). For more general forms of manipulator, methods for forward and inverse kinematics and dynamics are almost universally ad-hoc, leading to case-by-case analyses and failing to provide a mathematical and computational structure that is suitable for use in state-of-the-art control methods, such as those presented in (Siciliano, Sciavicco, Villani, and Oriolo, 2010). The purpose of the remainder of the chapter is to build upon the differentiable manifold formulation presented in Sections 9.3 through 9.6 to create systematic kinematics and dynamics computational algorithms for analysis and control of a broad spectrum of manipulators.

Analytical forward and inverse kinematic configuration mappings and a differentiable manifold structure are presented in Sections 9.3 through 9.6 for four categories of manipulator; (1) serial, (2) explicit parallel, (3) implicit, and (4) compound. To support manipulator dynamics for each of these categories, analytical formulations of forward and inverse velocity analysis are presented in Section 9.9. Algorithms based on the manipulator differentiable manifold structure and these analytical results are presented in Section 9.10 for real-time computation of forward and inverse configuration and velocity analysis on a time grid, in each of the manipulator categories. These algorithms are designed to be embedded in modern motion control formulations that rely on both *forward and inverse kinematics computation* and ODE of manipulator dynamics.

In carrying out forward and inverse kinematic analysis on a time grid,  $t_i$ ,  $i = 1, 2, \dots$ , care is taken to ensure that manipulator coordinates that are calculated remain in a *singularity free differentiable manifold component*  $\tilde{X}_\ell$ . In *forward kinematics computation*, inputs  $y_i$  and  $\dot{y}_i$  are specified and outputs  $z_i$  and  $\dot{z}_i$  are computed on the time grid. For each  $t_i$ , checks are made to confirm that the configuration obtained is in  $\tilde{X}_\ell$ . If  $x_i \notin \tilde{X}_\ell$  for some  $t_i$ , feasible outputs for the specified input cannot be obtained and forward kinematics fails. Comparable checks are made during inverse kinematics computation.

Second order *ODE of dynamics*, in terms of both input and output coordinates as state variables, that are singularity free in maximal disjoint manifold component  $\tilde{X}_\ell$  are presented for each manipulator category in Section 9.11. Using the *d'Alembert variational formulation* of multibody dynamics of Chapter 4 and parameterizations defined in Chapter 5, ODE of dynamics are embedded in each manifold component  $\tilde{X}_\ell$ . Data required for systematic evaluation of the ODE of dynamics are obtained using results of computational algorithms presented in Section 9.10, without the need for ad-hoc derivation of equations of motion. High fidelity ODE of manipulator dynamics are obtained in a form that enables their use in modern motion control algorithms. With real-time kinematics algorithms presented, these ODE can be implemented in in-line manipulator control systems.

Quantities required to evaluate ODE of dynamics for each of the four model manipulators studied in Sections 9.3 through 9.6 are presented in Section 9.12, using only matrix computation and requiring no ad-hoc derivation.

Historically, kinematic and dynamic analysis approaches used in manipulators has been based on the kinematic structure of each manipulator. With the exception of recursive dynamics that is based on forward kinematics of serial manipulators, dynamics formulations have been d-hoc in nature. The purpose of the remainder of this chapter is to develop systematic computational kinematics and dynamics formulations that can be uniformly applied to broad classes of manipulators.

## 9.8 Foundations of Manipulator Dynamics

Formulations presented in the literature for manipulator dynamics tend to focus on serial and explicit parallel manipulators. Even for these ideal manipulators, equations of dynamics are derived using ad-hoc methods (Merlet, 2006; Siciliano, Sciavicco, Villani, and Oriolo, 2010; Lynch, and Park, 2017), or *recursive dynamics methods* that yield ordinary differential equations (ODE) of motion for serial manipulators (Featherstone, 2008). For more general forms of manipulator, methods for forward and inverse kinematics and dynamics are almost universally ad-hoc, leading to case-by-case analyses and failing to provide a mathematical and computational structure that is suitable for use in state-of-the-art control methods. The purpose of the remainder of the chapter is to build upon the differentiable manifold formulation presented in Sections 9.3 through 9.6 to create systematic kinematics and dynamics computational algorithms for analysis and control of a broad spectrum of manipulators.

Analytical forward and inverse kinematic configuration mappings and a differentiable manifold structure are presented in Sections 9.3 through 9.6 for four categories of manipulator; (1) serial, (2) explicit parallel, (3) implicit, and (4) compound. To support manipulator dynamics for each of these categories, analytical formulations of forward and inverse velocity analysis are presented in Section 9.9. Algorithms based on the manipulator differentiable manifold structure and these analytical results are presented in Section 9.10 for real-time computation of forward and inverse configuration and velocity analysis on a time grid, in each of the manipulator categories. These algorithms are designed to be embedded in modern motion control formulations that rely on both forward and inverse kinematics computation and ODE of manipulator dynamics.

In carrying out forward and inverse kinematic analysis on a time grid,  $t_i$ ,  $i = 1, 2, \dots$ , care is taken to ensure that manipulator coordinates that are calculated remain in a singularity free differentiable manifold component  $\tilde{X}_\ell$ . In forward kinematics computation, inputs  $\mathbf{y}_i$  and  $\dot{\mathbf{y}}_i$  are specified and outputs  $\mathbf{z}_i$  and  $\dot{\mathbf{z}}_i$  are computed on the time grid. For each  $t_i$ , checks are made to confirm that the configuration obtained is in  $\tilde{X}_\ell$ . If  $\mathbf{x}_i \notin \tilde{X}_\ell$  for some  $t_i$ , feasible outputs for the specified input cannot be obtained and forward kinematics fails. Comparable checks are made during inverse kinematics computation.

Second order ODE of dynamics, in terms of both input and output coordinates as state variables, that are singularity free in maximal disjoint manifold component  $\tilde{X}_\ell$  are presented for each manipulator category in Section 9.11. Using the d'Alembert variational formulation of multibody dynamics of Chapter 4 and parameterizations defined in Chapter 5, ODE of dynamics are embedded in each manifold component  $\tilde{X}_\ell$ . Data required for systematic evaluation of the ODE of dynamics are obtained using results of computational algorithms presented in Section 9.10, without the need for ad-hoc derivation of equations of motion. High fidelity ODE of manipulator dynamics are obtained in a form that enables their use in modern motion control algorithms. With *real-time kinematics algorithms* presented, these ODE can be implemented in in-line manipulator control systems.

Quantities required to evaluate ODE of dynamics for each of the four model manipulators studied in Sections 9.3 through 9.6 are presented in Section 9.12, using only matrix computation and requiring no ad-hoc derivation.

Historically, kinematic and dynamic analysis approaches used in manipulators has been based on the kinematic structure of each manipulator. With the exception of recursive dynamics that is based on forward kinematics of serial manipulators, dynamics formulations have been d-hoc in nature. The purpose of the remainder of this chapter is to develop systematic computational kinematics and dynamics formulations that can be uniformly applied to broad classes of manipulators.

## 9.9 Forward and Inverse Velocity Mappings

Analytical *forward and inverse velocity mappings* are obtained by treating variables in kinematic configuration equations derived in Section 9.2 as time dependent and taking derivatives of the configuration equations with respect to time. The vector-matrix form of multivariable calculus used herein and basics of differential geometry required are presented in Sections 3.5 and 3.6.

For a general manipulator defined in Section 9.2, with *generalized coordinates*  $\mathbf{q} \in \mathbb{R}^n$ , *input coordinates*  $\mathbf{y} \in \mathbb{R}^k$ , and *output coordinates*  $\mathbf{z} \in \mathbb{R}^k$ , there are four *input-generalized coordinate-output equations*, Eqs. (9.2.2) through (9.2.5) of Section 9.2, repeated here as

$$\Psi(\mathbf{q}, \mathbf{y}) = \mathbf{0} \quad (9.9.1)$$

$$\Omega(\mathbf{q}, \mathbf{y}) \equiv \begin{bmatrix} \Phi(\mathbf{q}) \\ \Psi(\mathbf{q}, \mathbf{y}) \end{bmatrix} = \mathbf{0} \quad (9.9.2)$$

$$\Gamma(\mathbf{q}, \mathbf{z}) = \mathbf{0} \quad (9.9.3)$$

$$\Lambda(\mathbf{q}, \mathbf{z}) \equiv \begin{bmatrix} \Phi(\mathbf{q}) \\ \Gamma(\mathbf{q}, \mathbf{z}) \end{bmatrix} = \mathbf{0} \quad (9.9.4)$$

and the regular configuration space is

$$\tilde{\mathbf{X}} = \left\{ \mathbf{x} = \begin{bmatrix} \mathbf{y}^T & \mathbf{q}^T & \mathbf{z}^T \end{bmatrix}^T : \Phi(\mathbf{q}) = \mathbf{0}, \Psi(\mathbf{q}, \mathbf{y}) = \mathbf{0}, \Gamma(\mathbf{q}, \mathbf{z}) = \mathbf{0}, |\Psi_y(\mathbf{q}, \mathbf{y})| \neq 0, \right. \\ \left. |\Omega_q(\mathbf{q}, \mathbf{y})| \neq 0, |\Gamma_z(\mathbf{q}, \mathbf{z})| \neq 0, \text{ and } |\Lambda_q(\mathbf{q}, \mathbf{z})| \neq 0 \right\} \subset \mathbb{R}^{n+2k} \quad (9.9.5)$$

It is shown in Section 9.2.2 that, in  $\tilde{\mathbf{X}}$ , there exist locally unique *forward and inverse configuration mappings*

$$\mathbf{z} = \mathbf{h}(\mathbf{f}(\mathbf{y})) \quad (9.9.6)$$

$$\mathbf{y} = \mathbf{g}(\mathbf{e}(\mathbf{z})) \quad (9.9.7)$$

where  $\mathbf{q} = \mathbf{f}(\mathbf{y})$  is a solution of Eq. (9.9.2),  $\mathbf{z} = \mathbf{h}(\mathbf{q})$  is a solution of Eq. (9.9.3),  $\mathbf{q} = \mathbf{e}(\mathbf{z})$  is a solution of Eq. (9.9.4), and  $\mathbf{y} = \mathbf{g}(\mathbf{q})$  is a solution of Eq. (9.9.1).

### 9.9.1 Serial Manipulator Velocity Mappings

For *serial manipulators* defined in Section 9.3, the *forward configuration mapping* is given by Eq. (9.9.6) as

$$\mathbf{z} = \mathbf{h}(\mathbf{f}(\mathbf{y})) \equiv \mathbf{G}(\mathbf{y}) \quad (9.9.8)$$

Using the chain rule of differentiation,

$$\mathbf{G}'(\mathbf{y}) = \mathbf{h}'(\mathbf{f}(\mathbf{y}))\mathbf{f}'(\mathbf{y}) \quad (9.9.9)$$

Since  $\Gamma(\mathbf{q}, \mathbf{h}(\mathbf{q})) = \mathbf{0}$  is a local identity in  $\mathbf{q}$ ,  $\Gamma_z(\mathbf{q}, \mathbf{h}(\mathbf{q}))\mathbf{h}'(\mathbf{q}) + \Gamma_q(\mathbf{q}, \mathbf{h}(\mathbf{q})) = \mathbf{0}$  and, because  $|\Gamma_z(\mathbf{q}, \mathbf{z})| \neq 0$  in  $\tilde{X}$ ,  $\mathbf{h}'(\mathbf{q}) = -\Gamma_z^{-1}(\mathbf{q}, \mathbf{h}(\mathbf{q}))\Gamma_q(\mathbf{q}, \mathbf{h}(\mathbf{q}))$ . Similarly,  $\Omega(\mathbf{f}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$  is a local identity in  $\mathbf{y}$ , so  $\Omega_q(\mathbf{f}(\mathbf{y}), \mathbf{y})\mathbf{f}'(\mathbf{y}) + \Omega_y(\mathbf{f}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$  and  $\mathbf{f}'(\mathbf{y}) = -\Omega_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\Omega_y(\mathbf{f}(\mathbf{y}), \mathbf{y})$ . Substituting these results into Eq. (9.9.9),

$$\mathbf{G}'(\mathbf{y}) = \Gamma_z^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{h}(\mathbf{f}(\mathbf{y})))\Gamma_q(\mathbf{f}(\mathbf{y}), \mathbf{h}(\mathbf{f}(\mathbf{y})))\Omega_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\Omega_y(\mathbf{f}(\mathbf{y}), \mathbf{y}) \quad (9.9.10)$$

The term *local* used herein for a property means that property holds in a neighborhood of the configuration space.

In the *serial manipulator regular configuration space*

$\tilde{X}^S = \left\{ \mathbf{x} = \begin{bmatrix} \mathbf{y}^T & \mathbf{z}^T \end{bmatrix}^T : \mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0} \text{ and } |\mathbf{G}'(\mathbf{y})| \neq 0 \right\}$ , it is shown in Section 9.3.2 that there exists a locally unique *inverse configuration mapping*  $\mathbf{y} = \mathbf{H}(\mathbf{z})$  such that  $\mathbf{G}(\mathbf{H}(\mathbf{z})) - \mathbf{z} = \mathbf{0}$  is an identity in  $\mathbf{z}$ .

### 9.9.1.1 Serial Manipulator Forward Velocity Mapping

With input and output coordinates as functions of time, differentiating Eq. (9.9.8) with respect to time, the *serial manipulator forward velocity mapping* is

$$\dot{\mathbf{z}} = \mathbf{G}'(\mathbf{y})\dot{\mathbf{y}} \quad (9.9.11)$$

### 9.9.1.2 Serial Manipulator Inverse Velocity Mapping

Differentiating the local identity  $\mathbf{G}(\mathbf{H}(\mathbf{z})) - \mathbf{z} = \mathbf{0}$  with respect to  $\mathbf{z}$  and using the chain rule of differentiation,  $\mathbf{G}'(\mathbf{H}(\mathbf{z}))\mathbf{H}'(\mathbf{z}) = \mathbf{I}$  and, since  $|\mathbf{G}'(\mathbf{y})| \neq 0$  in  $\tilde{X}^S$ ,  $\mathbf{H}'(\mathbf{z}) = \mathbf{G}'^{(-1)}(\mathbf{H}(\mathbf{z}))$ . With Eq. (9.9.11), the *serial manipulator inverse velocity mapping* is

$$\dot{\mathbf{y}} = \mathbf{G}'^{(-1)}(\mathbf{H}(\mathbf{z}))\dot{\mathbf{z}} = \mathbf{H}'(\mathbf{z})\dot{\mathbf{z}} \quad (9.9.12)$$

### 9.9.2 Explicit Parallel Manipulator Velocity Mappings

For explicit parallel manipulators defined in Section 9.4, the *inverse configuration mapping* is given by Eq. (9.9.7) as

$$\mathbf{y} = \mathbf{g}(\mathbf{e}(\mathbf{z})) \equiv \bar{\mathbf{H}}(\mathbf{z}) \quad (9.9.13)$$

Differentiating with respect to  $\mathbf{z}$ ,

$$\bar{\mathbf{H}}'(\mathbf{z}) = \mathbf{g}'(\mathbf{e}(\mathbf{z}))\mathbf{e}'(\mathbf{z}) \quad (9.9.14)$$

Since  $\Psi(\mathbf{g}(\mathbf{q}), \mathbf{q}) = 0$  is a local identity in  $\mathbf{q}$ ,  $\Psi_y(\mathbf{g}(\mathbf{q}), \mathbf{q})\mathbf{g}'(\mathbf{q}) + \Psi_q(\mathbf{g}(\mathbf{q}), \mathbf{q}) = 0$  and, because  $|\Psi_y(\mathbf{g}(\mathbf{q}), \mathbf{q})| \neq 0$  in  $\tilde{X}$ ,  $\mathbf{g}'(\mathbf{q}) = -\Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q})\Psi_q(\mathbf{g}(\mathbf{q}), \mathbf{q})$ . Similarly,  $\Lambda(\mathbf{e}(\mathbf{z}), \mathbf{z}) = 0$  is a local identity in  $\mathbf{z}$ , so  $\Lambda_q(\mathbf{e}(\mathbf{z}), \mathbf{z})\mathbf{e}'(\mathbf{z}) + \Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z}) = 0$  and  $\mathbf{e}'(\mathbf{z}) = -\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})$ . Substituting these results into Eq. (9.9.14),

$$\bar{\mathbf{H}}'(\mathbf{z}) = \Psi_y^{-1}(\mathbf{g}(\mathbf{e}(\mathbf{z})), \mathbf{e}(\mathbf{z}))\Psi_q(\mathbf{g}(\mathbf{e}(\mathbf{z})), \mathbf{e}(\mathbf{z}))\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z}) \quad (9.9.15)$$

In the *explicit parallel manipulator regular configuration space*  $\tilde{X}^{EP} = \left\{ \mathbf{x} = \begin{bmatrix} \mathbf{y}^T & \mathbf{z}^T \end{bmatrix}^T : \bar{\mathbf{H}}(\mathbf{z}) - \mathbf{y} = \mathbf{0} \text{ and } |\bar{\mathbf{H}}'(\mathbf{z})| \neq 0 \right\}$ , it is shown in Section 9.4.2 that there exists a locally unique forward configuration mapping  $\mathbf{z} = \bar{\mathbf{G}}(\mathbf{y})$  such that  $\bar{\mathbf{H}}(\bar{\mathbf{G}}(\mathbf{y})) - \mathbf{y} = \mathbf{0}$  is an identity in  $\mathbf{y}$ .

### 9.9.2.1 Explicit Parallel Manipulator Inverse Velocity Mapping

Differentiating  $\mathbf{y} = \bar{\mathbf{H}}(\mathbf{z})$  with respect to time, the *explicit parallel manipulator inverse velocity mapping* is simply

$$\dot{\mathbf{y}} = \bar{\mathbf{H}}'(\mathbf{z})\dot{\mathbf{z}} \quad (9.9.16)$$

### 9.9.2.2 Explicit Parallel Manipulator Forward Velocity Mapping

Differentiating the identity  $\bar{\mathbf{H}}(\bar{\mathbf{G}}(\mathbf{y})) - \mathbf{y} = \mathbf{0}$  with respect to  $\mathbf{y}$ ,  $\bar{\mathbf{H}}'(\bar{\mathbf{G}}(\mathbf{y}))\bar{\mathbf{G}}'(\mathbf{y}) = \mathbf{I}$ . Since  $\bar{\mathbf{H}}'(\bar{\mathbf{G}}(\mathbf{y}))$  is nonsingular in  $\tilde{X}^{EP}$ ,  $\bar{\mathbf{G}}'(\mathbf{y}) = \bar{\mathbf{H}}'^{(-1)}(\bar{\mathbf{G}}(\mathbf{y}))$ . With Eq. (9.9.16), the explicit parallel manipulator forward velocity mapping is

$$\dot{\mathbf{z}} = \bar{\mathbf{H}}'^{(-1)}(\bar{\mathbf{G}}(\mathbf{y}))\dot{\mathbf{y}} = \bar{\mathbf{G}}'(\mathbf{y})\dot{\mathbf{y}} \quad (9.9.17)$$

## 9.9.3 Implicit Manipulator Velocity Mappings

With generalized coordinates eliminated and the input-output equation  $\mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}$  of Eq. (9.5.5), forward and inverse velocity relations on the regular configuration space  $\tilde{X}^1$  of Eq. (9.5.7) may be obtained through direct differentiation with respect to time,

$$\mathbf{F}_y(\mathbf{y}, \mathbf{z})\dot{\mathbf{y}} + \mathbf{F}_z(\mathbf{y}, \mathbf{z})\dot{\mathbf{z}} = 0 \quad (9.9.18)$$

### 9.9.3.1 Implicit Manipulator Forward Velocity Mapping

Since  $|\mathbf{F}_z(\mathbf{y}, \mathbf{z})| \neq 0$  in  $\tilde{X}^1$ , the *implicit manipulator forward velocity mapping*, from Eq. (9.9.18) is

$$\dot{\mathbf{z}} = -\mathbf{F}_z^{-1}(\mathbf{y}, \mathbf{z})\mathbf{F}_y(\mathbf{y}, \mathbf{z})\dot{\mathbf{y}} \quad (9.9.19)$$

### 9.9.3.1 Implicit Manipulator inverse Velocity Mapping

Since  $|\mathbf{F}_y(\mathbf{y}, \mathbf{z})| \neq 0$  in  $\tilde{X}^1$ , the *implicit manipulator inverse velocity mapping*, from Eq. (9.9.18) is

$$\dot{\mathbf{y}} = -\mathbf{F}_y^{-1}(\mathbf{y}, \mathbf{z})\mathbf{F}_z(\mathbf{y}, \mathbf{z})\dot{\mathbf{z}} \quad (9.9.20)$$

## 9.9.4 Compound Manipulator Velocity Mappings

### 9.9.4.1 Compound Manipulator Froward Velocity Mapping

For compound manipulators that are based on the general model of Eqs. (9.9.1) through (9.9.7), differentiating Eq. (9.9.6) with respect to time and using the chain rule of differentiation yields  $\dot{\mathbf{z}} = \mathbf{h}'(\mathbf{f}(\mathbf{y}))\mathbf{f}'(\mathbf{y})\dot{\mathbf{y}}$ . Since  $\mathbf{q} = \mathbf{f}(\mathbf{y})$  is a local solution of Eq. (9.9.2),  $\Omega(\mathbf{f}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$  is an identity in  $\mathbf{y}$ . Differentiating this identity with respect to  $\mathbf{y}$  yields

$\Omega_q(\mathbf{f}(\mathbf{y}), \mathbf{y})\mathbf{f}'(\mathbf{y}) + \Omega_y(\mathbf{f}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$  and, since  $\Omega_q(\mathbf{f}(\mathbf{y}), \mathbf{y})$  is nonsingular in  $\tilde{\mathbf{X}}^C$  of Eq.(9.9.5),  $\mathbf{f}'(\mathbf{y}) = -\Omega_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\Omega_y(\mathbf{f}(\mathbf{y}), \mathbf{y})$ . Similarly, since  $\mathbf{z} = \mathbf{h}(\mathbf{q})$  is a local solution of Eq. (9.9.3),  $\Gamma(\mathbf{q}, \mathbf{h}(\mathbf{q})) = \mathbf{0}$  is an identity in  $\mathbf{q}$ . Differentiating this identity with respect to  $\mathbf{q}$ ,  $\Gamma_q(\mathbf{q}, \mathbf{h}(\mathbf{q})) + \Gamma_z(\mathbf{q}, \mathbf{h}(\mathbf{q}))\mathbf{h}'(\mathbf{q}) = \mathbf{0}$  and, since  $\Gamma_z(\mathbf{q}, \mathbf{h}(\mathbf{q}))$  is nonsingular in  $\tilde{\mathbf{X}}^C$ ,  $\mathbf{h}'(\mathbf{q}) = -\Gamma_z^{-1}(\mathbf{q}, \mathbf{h}(\mathbf{q}))\Gamma_q(\mathbf{q}, \mathbf{h}(\mathbf{q}))$ . With these results, the *compound manipulator forward velocity mapping* is

$$\dot{\mathbf{z}} = \mathbf{h}'(\mathbf{f}(\mathbf{y}))\mathbf{f}'(\mathbf{y})\dot{\mathbf{y}} = \Gamma_z^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{h}(\mathbf{f}(\mathbf{y})))\Gamma_q(\mathbf{f}(\mathbf{y}), \mathbf{h}(\mathbf{f}(\mathbf{y})))\Omega_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\Omega_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\dot{\mathbf{y}} \quad (9.9.21)$$

#### 9.9.4.2 Compound Manipulator Inverse Velocity Mapping

Differentiating Eq. (9.9.7) with respect to time, using the chain rule of differentiation,  $\dot{\mathbf{y}} = \mathbf{g}'(\mathbf{e}(\mathbf{z}))\mathbf{e}'(\mathbf{z})\dot{\mathbf{z}}$ . Since  $\mathbf{y} = \mathbf{g}(\mathbf{q})$  is a local solution of Eq. (9.9.1),  $\Psi(\mathbf{q}, \mathbf{g}(\mathbf{q})) = \mathbf{0}$  is an identity in  $\mathbf{q}$ . Differentiating this identity with respect to  $\mathbf{q}$  yields  $\Psi_q(\mathbf{q}, \mathbf{g}(\mathbf{q})) + \Psi_y(\mathbf{q}, \mathbf{g}(\mathbf{q}))\mathbf{g}'(\mathbf{q}) = \mathbf{0}$  and, since  $\Psi_y(\mathbf{q}, \mathbf{g}(\mathbf{q}))$  is nonsingular in  $\tilde{\mathbf{X}}^C$ ,  $\mathbf{g}'(\mathbf{q}) = -\Psi_y^{-1}(\mathbf{q}, \mathbf{g}(\mathbf{q}))\Psi_q(\mathbf{q}, \mathbf{g}(\mathbf{q}))$ . Similarly, since  $\mathbf{q} = \mathbf{e}(\mathbf{z})$  is a local solution of Eq. (9.9.4),  $\Lambda(\mathbf{e}(\mathbf{z}), \mathbf{z}) = \mathbf{0}$  is an identity in  $\mathbf{z}$ . Differentiating this identity with respect to  $\mathbf{z}$ ,  $\Lambda_q(\mathbf{e}(\mathbf{z}), \mathbf{z})\mathbf{e}'(\mathbf{z}) + \Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z}) = \mathbf{0}$  and, since  $\Lambda_q(\mathbf{e}(\mathbf{z}), \mathbf{z})$  is nonsingular in  $\tilde{\mathbf{X}}^C$ ,  $\mathbf{e}'(\mathbf{z}) = -\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})$ . With these results, the *compound manipulator inverse velocity mapping* is

$$\dot{\mathbf{y}} = \mathbf{g}'(\mathbf{e}(\mathbf{z}))\mathbf{e}'(\mathbf{z})\dot{\mathbf{z}} = \Psi_y^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{g}(\mathbf{e}(\mathbf{z})))\Psi_q(\mathbf{e}(\mathbf{z}), \mathbf{g}(\mathbf{e}(\mathbf{z})))\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})\dot{\mathbf{z}} \quad (9.9.22)$$

Forward and inverse velocity mappings for all four manipulator categories are obtained by differentiating kinematic configuration equations of Section 9.3 through 9.6 with respect to time.

## 9.10 Computation of Forward and Inverse Kinematics on a Time Grid

While velocity analysis in Section 9.9 is carried out with manipulator coordinates as differentiable functions of time, computation of results is necessarily carried out on a discrete time grid  $t_1 < t_2 < t_3 < \dots$ . In most cases a small fixed time-step  $h$  is employed, such that

$t_{i+1} = t_i + h$ ,  $i = 1, 2, \dots$ . A *matrix version of the Newton-Raphson algorithm* is used to evaluate inverse matrices required in velocity analysis and to compute solutions of nonlinear manipulator kinematic equations on the time grid.

### 9.10.1 Serial Manipulator Forward Kinematics

In *forward kinematics for serial manipulators* of Section 9.3, with input-output equations  $\mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0}$  and inputs  $\mathbf{y}_i$  and  $\dot{\mathbf{y}}_i$  specified on the time grid, outputs are explicitly evaluated as  $\mathbf{z}_i = \mathbf{G}(\mathbf{y}_i)$  and  $\dot{\mathbf{z}}_i = \mathbf{G}'(\mathbf{y}_i)\dot{\mathbf{y}}_i$ . With  $\mathbf{x}_i = [\mathbf{y}_i^T \quad \mathbf{z}_i^T]^T \in \tilde{X}_\ell^S$ , to assure that  $\mathbf{x}_i = [\mathbf{y}_i^T \quad \mathbf{z}_i^T]^T$ ,  $i = 2, 3, \dots$ , stay in  $\tilde{X}_\ell^S$ , the condition  $|\mathbf{G}'(\mathbf{y}_i)| \neq 0$  is verified at each time-step and the sign of  $|\mathbf{G}'(\mathbf{y}_i)|$  is the same as  $|\mathbf{G}'(\mathbf{y}_1)|$ .

### 9.10.2 Explicit Parallel Manipulator Inverse Kinematics

In *inverse kinematics for explicit parallel manipulators* of Section 9.4, with input-output equations  $\mathbf{H}(\mathbf{z}) - \mathbf{y} = \mathbf{0}$  and outputs  $\mathbf{z}_i$  and  $\dot{\mathbf{z}}_i$  specified on the time grid, inputs are explicitly evaluated as  $\mathbf{y}_i = \mathbf{H}(\mathbf{z}_i)$  and  $\dot{\mathbf{y}}_i = \mathbf{H}'(\mathbf{z}_i)\dot{\mathbf{z}}_i$ . With  $\mathbf{x}_i = [\mathbf{y}_i^T \quad \mathbf{z}_i^T]^T \in \tilde{X}_\ell^S$ , to assure that  $\mathbf{x}_i = [\mathbf{y}_i^T \quad \mathbf{z}_i^T]^T$ ,  $i = 2, 3, \dots$ , stay in  $\tilde{X}_\ell^S$ , the condition  $|\mathbf{H}'(\mathbf{y}_i)| \neq 0$  has the same sign as at  $\mathbf{y}_1$  is verified at each time-step.

For serial manipulator inverse kinematics and explicit parallel manipulator forward kinematics, as in all other manipulator categories, nonlinear equations must be solved for configurations and the result used in evaluation of velocities. Iterative computations are carried out in each such case, using the Newton-Raphson method of Section 2.2.6, denoted N-R, for evaluation of configuration coordinates and inverse matrices that support both kinematic configuration and velocity analysis and formulation of ODE of dynamics. Real-time issues associated with these computations are addressed in Section 9.10.7.

### 9.10.3 Implicit Manipulator Kinematics

*Computational algorithms for implicit manipulator kinematics* are first derived, since they are the most general encountered. Algorithms for all other manipulator categories are obtained through application of these results.

#### 9.10.3.1 Implicit Manipulator Forward Kinematics

For the input-output equation  $\mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}$  of Eq. (9.5.5) and input coordinates and velocities  $\mathbf{y}_i$  and  $\dot{\mathbf{y}}_i$  specified on a time grid  $t_1 < t_2 < \dots$ , the objective in forward kinematics is to evaluate solutions  $\mathbf{z}_i = \mathbf{G}(\mathbf{y}_i)$  that are shown to exist in the regular configuration space in

Section 9.5.2 and to evaluate the solution Jacobian  $\mathbf{G}'(\mathbf{y}_i) = -\mathbf{F}_z^{-1}(\mathbf{y}_i, \mathbf{G}(\mathbf{y}_i))\mathbf{F}_y(\mathbf{y}_i, \mathbf{G}(\mathbf{y}_i))$  of Eq. (9.9.19), which yields  $\dot{\mathbf{z}}_i = \mathbf{G}'(\mathbf{y}_i)\dot{\mathbf{y}}_i$ . A *matrix form of the N-R method* is used for this purpose.

Given an estimate  $\mathbf{z}_1^{(1)}$  of the solution of  $\mathbf{F}(\mathbf{y}_1, \mathbf{z}) = \mathbf{0}$  at  $t_1$  for  $\mathbf{z}$ , where a superscript index in parentheses is the iteration number, an increment  $\Delta\mathbf{z}_1^{(1)}$  is computed using the linear approximation of the input-output equations,  $\mathbf{F}_z(\mathbf{y}_1, \mathbf{z}_1^{(1)})\Delta\mathbf{z}_1^{(1)} = -\mathbf{F}(\mathbf{y}_1, \mathbf{z}_1^{(1)})$ , and the approximate solution update is  $\mathbf{z}_1^{(2)} = \mathbf{z}_1^{(1)} + \Delta\mathbf{z}_1^{(1)}$ , where the linear equation is solved numerically. This N-R iteration is carried out as  $\mathbf{F}_z(\mathbf{y}_1, \mathbf{z}_1^{(j)})\Delta\mathbf{z}_1^{(j)} = -\mathbf{F}(\mathbf{y}_1, \mathbf{z}_1^{(j)})$ ;  $\mathbf{z}_1^{(j+1)} = \mathbf{z}_1^{(j)} + \Delta\mathbf{z}_1^{(j)}$ ,  $j = 2, 3, \dots$  until  $\|\mathbf{F}(\mathbf{y}_1, \mathbf{z}_1^{(j+1)})\| < \text{Tol}_1$ , in which case the approximate solution  $\mathbf{z}_1 = \mathbf{z}_1^{(j+1)}$  is accepted. The *manifold component*  $\tilde{\mathbf{X}}_\ell^1$  that contains  $\mathbf{x}_1 = [\mathbf{y}_1^T \quad \mathbf{z}_1^T]^T$  is identified by values of determinants in

$$\tilde{\mathbf{X}}^1 = \left\{ \mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T : \mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}, |\mathbf{F}_y(\mathbf{y}, \mathbf{z})| \neq 0, \text{ and } |\mathbf{F}_z(\mathbf{y}, \mathbf{z})| \neq 0 \right\}$$

For a good estimate  $\mathbf{z}_1^{(1)}$  of the solution, the N-R method is known to *converge quadratically*; i.e., there exists a constant  $a > 0$  such that  $\|\mathbf{F}(\mathbf{y}_1, \mathbf{z}_1^{(j+1)})\| < a \|\mathbf{F}(\mathbf{y}_1, \mathbf{z}_1^{(j)})\|^2$  for  $j$  sufficiently large (Atkinson, 1989). If a poor estimate is used, however, the iterative process may diverge and the process must be restarted with a new estimate. Assuming that N-R iteration with the equation  $\mathbf{F}(\mathbf{y}_1, \mathbf{z}) = \mathbf{0}$  yields a solution  $\mathbf{z}_1$  at  $t_1$ , the inverse  $\mathbf{F}_z^{-1}(\mathbf{y}_1, \mathbf{z}_1)$  is numerically evaluated. With this result,  $\mathbf{G}'(\mathbf{y}_1) = -\mathbf{F}_z^{-1}(\mathbf{y}_1, \mathbf{z}_1)\mathbf{F}_y(\mathbf{y}_1, \mathbf{z}_1)$  and  $\dot{\mathbf{z}}_1 = \mathbf{G}'(\mathbf{y}_1)\dot{\mathbf{y}}_1$ .

At  $t_k$ ,  $k \geq 2$ , a good estimate for the solution  $\mathbf{z}$  of  $\mathbf{F}(\mathbf{y}_k, \mathbf{z}) = \mathbf{0}$  is  $\mathbf{z}_k^{(1)} = \mathbf{z}_{k-1} + h\mathbf{G}'(\mathbf{y}_{k-1})\dot{\mathbf{y}}_{k-1}$ , and N-R iteration is carried out with the liner equation  $\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k^{(1)})\Delta\mathbf{z}_k^{(1)} = -\mathbf{F}(\mathbf{y}_k, \mathbf{z}_k^{(1)})$ . Since  $\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k^{(1)}) \approx \mathbf{F}_z^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1})$  and the N-R algorithm does not require an exact Jacobian for convergence (Atkinson, 1989),  $\Delta\mathbf{z}_k^{(1)} = -\mathbf{F}_z^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1})\mathbf{F}(\mathbf{y}_k, \mathbf{z}_k^{(1)})$  and  $\mathbf{z}_k^{(2)} = \mathbf{z}_k^{(1)} - \mathbf{F}_z^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1})\mathbf{F}(\mathbf{y}_k, \mathbf{z}_k^{(1)})$ . For iteration  $j$ ,  $\mathbf{z}_k^{(j+1)} = \mathbf{z}_k^{(j)} - \mathbf{F}_z^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1})\mathbf{F}(\mathbf{y}_k, \mathbf{z}_k^{(j)})$ ,  $j = 2, 3, \dots$ , until  $\|\mathbf{F}(\mathbf{y}_k, \mathbf{z}_k^{(j+1)})\| < \text{Tol}_1$  and the result is accepted as  $\mathbf{z}_k = \mathbf{z}_k^{(j+1)} \equiv \mathbf{G}(\mathbf{y}_k)$ . To assure that  $\mathbf{x}_k = [\mathbf{y}_k^T \quad \mathbf{z}_k^T]^T$  is in  $\tilde{\mathbf{X}}_\ell^1$ ,  $|\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)| \neq 0$  and  $|\mathbf{F}_y(\mathbf{y}_k, \mathbf{z}_k)| \neq 0$  are verified to have the same signs as at  $\mathbf{x}_1 = [\mathbf{y}_1^T \quad \mathbf{z}_1^T]^T$ .

Rather than numerically inverting  $\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)$  for  $k = 2, 3, \dots$ , a more efficient *matrix Newton-Raphson algorithm* is used, based on the matrix equation  $\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I} = \mathbf{0}$ . With the estimate  $\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) = \mathbf{F}_z^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1})$ , the equation  $\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I} = \mathbf{0}$  is solved for  $\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k)$ , using N-R iteration with the linearized equation  $\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\Delta\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) = -(\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I})$ . Since  $\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k) \approx \mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k)$ ,  $\Delta\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) = -\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k)(\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I})$  and  $\mathbf{F}_z^{-1(2)}(\mathbf{y}_k, \mathbf{z}_k) = 2\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k)$ . For iteration  $j$ ,

$\mathbf{F}_z^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) = 2\mathbf{F}_z^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{F}_z^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k)$ ,  $j = 2, 1$ , until  
 $\|\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I}\| < \text{Tol}_2$ , in which case  $\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k) = \mathbf{F}_z^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k)$  is accepted.

The forgoing computations comprise a *three-step algorithm* for *implicit manipulator forward kinematics* is as follows:

- (1) At time-step  $t_1$ , with estimate  $\mathbf{z}_1^{(1)}$ , solve  $\mathbf{F}(\mathbf{y}_1, \mathbf{z}) = \mathbf{0}$  for  $\mathbf{z}_1 = \mathbf{G}(\mathbf{y}_1)$  and numerically evaluate  $\mathbf{F}_z^{-1}(\mathbf{y}_1, \mathbf{z}_1)$  and  $\mathbf{G}'(\mathbf{y}_1)$ , as follows:

$$\begin{aligned} \mathbf{F}_z(\mathbf{y}_1, \mathbf{z}_1^{(j)})\Delta\mathbf{z}_1^{(j)} &= -\mathbf{F}(\mathbf{y}_1, \mathbf{z}_1^{(j)}); \quad \mathbf{z}_1^{(j+1)} = \mathbf{z}_1^{(j)} + \Delta\mathbf{z}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\mathbf{F}(\mathbf{y}_1, \mathbf{z}_1^{(j+1)})\| < \text{Tol}_1 \\ \mathbf{z}_1 &= \mathbf{z}_1^{(j+1)} = \mathbf{G}(\mathbf{y}_1) \\ \mathbf{F}_z^{-1}(\mathbf{y}_1, \mathbf{z}_1) &= (\mathbf{F}_z(\mathbf{y}_1, \mathbf{z}_1))^{-1} \\ \mathbf{G}'(\mathbf{y}_1) &= -\mathbf{F}_z^{-1}(\mathbf{y}_1, \mathbf{z}_1)\mathbf{F}_y(\mathbf{y}_1, \mathbf{z}_1) \end{aligned} \tag{9.10.1}$$

- (2) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics equation  $\mathbf{F}(\mathbf{y}_k, \mathbf{z}) = \mathbf{0}$  for  $\mathbf{z}_k = \mathbf{G}(\mathbf{y}_k)$ , as follows:

$$\begin{aligned} \mathbf{z}_k^{(1)} &= \mathbf{z}_{k-1} + h\mathbf{G}'(\mathbf{y}_{k-1})\dot{\mathbf{y}}_{k-1} \\ \mathbf{z}_k^{(j+1)} &= \mathbf{z}_k^{(j)} - \mathbf{F}_z^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1})\mathbf{F}(\mathbf{y}_k, \mathbf{z}_k^{(j)}), \quad j = 1, 2, \dots \text{ until } \|\mathbf{F}(\mathbf{y}_k, \mathbf{z}_k^{(j+1)})\| < \text{Tol}_1 \\ \mathbf{z}_k &= \mathbf{z}_k^{(j+1)} = \mathbf{G}(\mathbf{y}_k) \end{aligned} \tag{9.10.2}$$

confirm  $|\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)| \neq 0$  and  $|\mathbf{F}_y(\mathbf{y}_k, \mathbf{z}_k)| \neq 0$  and signs are as at  $\mathbf{x}_1 = [\mathbf{y}_1^T \quad \mathbf{z}_1^T]^T$

- (3) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics matrix equation

- $\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I} = \mathbf{0}$  for  $\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k)$  and evaluate the Jacobian  $\mathbf{G}'(\mathbf{y}_k)$  and output velocity  $\dot{\mathbf{z}}_k$ , as follows:

$$\begin{aligned} \mathbf{F}_z^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) &= \mathbf{F}_z^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1}) \\ \mathbf{F}_z^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) &= 2\mathbf{F}_z^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{F}_z^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k), \\ j &= 1, 2, \dots, \text{until } \|\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I}\| < \text{Tol}_2 \\ \mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k) &= \mathbf{F}_z^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) \\ \mathbf{G}'(\mathbf{y}_k) &= -\mathbf{F}_z^{-1}(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_y(\mathbf{y}_k, \mathbf{z}_k) \\ \dot{\mathbf{z}}_k &= \mathbf{G}'(\mathbf{y}_k)\dot{\mathbf{y}}_k \end{aligned} \tag{9.10.3}$$

This three-step algorithm for numerically solving a nonlinear equation and evaluating Jacobians on a time grid is used repeatedly in forward and inverse kinematic configuration and velocity analysis, for each of the four manipulator categories. The results are employed in Section 9.11 as a principal ingredient in constructing ODE of manipulator dynamics on a time grid. Rather than repeating derivation of the algorithm for each nonlinear equation encountered, reference will be made to results presented in this section.

### 9.10.3.2 Implicit Manipulator Inverse Kinematics

For the input-output equation  $\mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}$  and output coordinates  $\mathbf{z}_i$  and  $\dot{\mathbf{z}}_i$  specified on a time grid  $t_1 < t_2 < \dots$ , the objective is to evaluate  $\mathbf{y}_i = \mathbf{H}(\mathbf{z}_i)$  that is shown to exist in Section 9.5.2 and to evaluate the Jacobian  $\mathbf{H}'(\mathbf{z}_i) = -\mathbf{F}_y^{-1}(\mathbf{y}_i, \mathbf{z}_i)\mathbf{F}_z(\mathbf{y}_i, \mathbf{z}_i)$  and  $\dot{\mathbf{y}}_i = \mathbf{H}'(\mathbf{z}_i)\dot{\mathbf{z}}_i$ . Reversing the roles of  $\mathbf{y}$  and  $\mathbf{z}$  and of  $\mathbf{F}_z^{-1}(\mathbf{y}_i, \mathbf{z}_i)$  and  $\mathbf{F}_y^{-1}(\mathbf{y}_i, \mathbf{z}_i)$ , respectively, for inverse kinematics, the *three-step algorithm for implicit manipulator inverse kinematics* of Eqs. (9.10.1) through (9.10.3) is as follows:

(1) At time-step  $t_1$ , with estimate  $\mathbf{y}_1^{(1)}$ , solve  $\mathbf{F}(\mathbf{y}, \mathbf{z}_1) = \mathbf{0}$  for  $\mathbf{y}_1 = \mathbf{H}(\mathbf{z}_1)$  and numerically evaluate  $\mathbf{F}_y^{-1}(\mathbf{y}_1, \mathbf{z}_1)$  and  $\mathbf{H}'(\mathbf{z}_1)$ , as follows:

$$\begin{aligned} \mathbf{F}_y(\mathbf{y}_1^{(j)}, \mathbf{z}_1)\Delta\mathbf{y}_1^{(j)} &= -\mathbf{F}(\mathbf{y}_1^{(j)}, \mathbf{z}_1); \quad \mathbf{y}_1^{(j+1)} = \mathbf{y}_1^{(j)} + \Delta\mathbf{y}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\mathbf{F}(\mathbf{y}_1^{(j+1)}, \mathbf{z}_1)\| < \text{Tol}_1 \\ \mathbf{y}_1 &= \mathbf{y}_1^{(j+1)} = \mathbf{H}(\mathbf{z}_1) \\ \mathbf{F}_y^{-1}(\mathbf{y}_1, \mathbf{z}_1) &= (\mathbf{F}_y(\mathbf{y}_1, \mathbf{z}_1))^{-1} \\ \mathbf{H}'(\mathbf{z}_1) &= -\mathbf{F}_y^{-1}(\mathbf{y}_1, \mathbf{z}_1)\mathbf{F}_z(\mathbf{y}_1, \mathbf{z}_1) \end{aligned} \tag{9.10.4}$$

(2) At time-step  $t_k$ ,  $k \geq 2$ , solve the inverse kinematics equation  $\mathbf{F}(\mathbf{y}, \mathbf{z}_k) = \mathbf{0}$  for  $\mathbf{y}_k = \mathbf{H}(\mathbf{z}_k)$ , as follows:

$$\begin{aligned} \mathbf{y}_k^{(1)} &= \mathbf{y}_{k-1} + h\mathbf{H}'(\mathbf{z}_{k-1})\dot{\mathbf{z}}_{k-1} \\ \mathbf{y}_k^{(j+1)} &= \mathbf{y}_k^{(j)} - \mathbf{F}_y^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1})\mathbf{F}(\mathbf{y}_k^{(j)}, \mathbf{z}_k), \quad j = 1, 2, \dots, \text{until } \|\mathbf{F}(\mathbf{y}_k^{(j+1)}, \mathbf{z}_k)\| < \text{Tol}_1 \\ \mathbf{y}_k &= \mathbf{y}_k^{(j+1)} = \mathbf{H}(\mathbf{z}_k) \end{aligned} \tag{9.10.5}$$

confirm  $|\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k)| \neq 0$  and  $|\mathbf{F}_y(\mathbf{y}_k, \mathbf{z}_k)| \neq 0$  and signs are as at  $\mathbf{x}_1 = [\mathbf{y}_1^T \quad \mathbf{z}_1^T]^T$

(3) At time-step  $t_k$ ,  $k \geq 2$ , solve the inverse kinematics matrix equation

$\mathbf{F}_y(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_y^{-1}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I} = \mathbf{0}$  for  $\mathbf{F}_y^{-1}(\mathbf{y}_k, \mathbf{z}_k)$  and evaluate the Jacobian  $\mathbf{H}'(\mathbf{z}_k)$  and input velocity  $\dot{\mathbf{y}}_k$ , as follows:

$$\begin{aligned} \mathbf{F}_y^{-1(1)}(\mathbf{y}_k, \mathbf{z}_k) &= \mathbf{F}_y^{-1}(\mathbf{y}_{k-1}, \mathbf{z}_{k-1}) \\ \mathbf{F}_y^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) &= 2\mathbf{F}_y^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{F}_y^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_y(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_y^{-1(j)}(\mathbf{y}_k, \mathbf{z}_k), \\ j &= 1, 2, \dots, \text{until } \|\mathbf{F}_y(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_y^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) - \mathbf{I}\| < \text{Tol}_2 \\ \mathbf{F}_y^{-1}(\mathbf{y}_k, \mathbf{z}_k) &= \mathbf{F}_y^{-1(j+1)}(\mathbf{y}_k, \mathbf{z}_k) \\ \mathbf{H}'(\mathbf{z}_k) &= -\mathbf{F}_y^{-1}(\mathbf{y}_k, \mathbf{z}_k)\mathbf{F}_z(\mathbf{y}_k, \mathbf{z}_k) \\ \dot{\mathbf{y}}_k &= \mathbf{H}'(\mathbf{z}_k)\dot{\mathbf{z}}_k \end{aligned} \tag{9.10.6}$$

### 9.10.4 Serial Manipulator Inverse Kinematics

For the serial manipulator input-output equation  $\mathbf{G}(\mathbf{y}) - \mathbf{z} \equiv \mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{0}$  and output coordinates  $\mathbf{z}_i$  and  $\dot{\mathbf{z}}_i$  specified on a time grid  $t_1 < t_2 < \dots$ , the objective in inverse kinematics is to evaluate solutions  $\mathbf{y}_i = \mathbf{H}(\mathbf{z}_i)$  that are shown to exist in Section 9.3.2 and, from Section 9.10.3.2, to evaluate the Jacobian  $\mathbf{H}'(\mathbf{z}_i) = -\mathbf{F}_y^{-1}(\mathbf{y}_i, \mathbf{z}_i)\mathbf{F}_z(\mathbf{y}_i, \mathbf{z}_i) = \mathbf{G}'^{(-1)}(\mathbf{y}_i)$ , since  $\mathbf{F}_y(\mathbf{y}, \mathbf{z}) = \mathbf{G}'(\mathbf{y})$  and  $\mathbf{F}_z(\mathbf{y}, \mathbf{z}) = -\mathbf{I}$  and  $\dot{\mathbf{y}}_i = \mathbf{H}'(\mathbf{z}_i)\dot{\mathbf{z}}_i$ . Using these relations, the *three-step algorithm for serial manipulator inverse kinematics* of Section 9.10.2.2, Eqs. (9.10.4) through (9.10.6), is as follows:

- (1) At time-step  $t_1$ , with estimate  $\mathbf{y}_1^{(1)}$ , solve  $\mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0}$  for  $\mathbf{y}_1 = \mathbf{H}(\mathbf{z}_1)$  and numerically evaluate  $\mathbf{F}_y^{-1}(\mathbf{y}_1, \mathbf{z}_1) = \mathbf{G}'^{(-1)}(\mathbf{y}_1)$  and  $\mathbf{H}'(\mathbf{z}_1)$ , as follows:

$$\begin{aligned} \mathbf{G}'(\mathbf{y}_1^{(j)})\Delta\mathbf{y}_1^{(j)} &= -(\mathbf{G}(\mathbf{y}_1^{(j)}) - \mathbf{z}_1); \quad \mathbf{y}_1^{(j+1)} = \mathbf{y}_1^{(j)} + \Delta\mathbf{y}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\mathbf{G}(\mathbf{y}_1^{(j+1)}) - \mathbf{z}_1\| < \text{Tol}_1 \\ \mathbf{y}_1 &= \mathbf{y}_1^{(j+1)} = \mathbf{H}(\mathbf{z}_1) \\ \mathbf{G}'^{(-1)}(\mathbf{y}_1) &= (\mathbf{G}'(\mathbf{y}_1))^{-1} \\ \mathbf{H}'(\mathbf{z}_1) &= \mathbf{G}'^{(-1)}(\mathbf{y}_1, \mathbf{z}_1) \end{aligned} \tag{9.10.7}$$

- (2) At time-step  $t_k$ ,  $k \geq 2$ , solve the inverse kinematics equation  $\mathbf{G}(\mathbf{y}) - \mathbf{z}_k = \mathbf{0}$  for  $\mathbf{y}_k = \mathbf{H}(\mathbf{z}_k)$ , as follows:

$$\begin{aligned} \mathbf{y}_k^{(1)} &= \mathbf{y}_{k-1} + h\mathbf{H}'(\mathbf{z}_{k-1})\dot{\mathbf{z}}_{k-1} \\ \mathbf{y}_k^{(j+1)} &= \mathbf{y}_k^{(j)} - \mathbf{G}'^{(-1)}(\mathbf{y}_{k-1})(\mathbf{G}(\mathbf{y}_k^{(j)}) - \mathbf{z}_k), \quad j = 1, 2, \dots \text{ until } \|\mathbf{G}(\mathbf{y}_k^{(j+1)}) - \mathbf{z}_k\| < \text{Tol}_1 \\ \mathbf{y}_k &= \mathbf{y}_k^{(j+1)} = \mathbf{H}(\mathbf{z}_k) \\ \text{confirm } |\mathbf{G}'(\mathbf{y}_k)| &\neq 0 \text{ and the sign is as at } \mathbf{y}_1 \end{aligned} \tag{9.10.8}$$

- (3) At time-step  $t_k$ ,  $k \geq 2$ , solve the inverse kinematics matrix equation  $\mathbf{G}'(\mathbf{y}_k)\mathbf{G}'^{(-1)}(\mathbf{y}_k) - \mathbf{I} = \mathbf{0}$  for  $\mathbf{G}'^{(-1)}(\mathbf{y}_k)$  and evaluate the Jacobian  $\mathbf{H}'(\mathbf{z}_k)$  and input velocity  $\dot{\mathbf{y}}_k$ , as follows:

$$\begin{aligned} \mathbf{G}'^{(-1)(1)}(\mathbf{y}_k) &= \mathbf{G}'^{-1}(\mathbf{y}_{k-1}) \\ \mathbf{G}'^{(-1)(j+1)}(\mathbf{y}_k) &= 2\mathbf{G}'^{(-1)(j)}(\mathbf{y}_k) - \mathbf{G}'^{(-1)(j)}(\mathbf{y}_k)\mathbf{G}'(\mathbf{y}_k)\mathbf{G}'^{(-1)(j)}(\mathbf{y}_k), \\ j &= 1, 2, \dots, \text{until } \|\mathbf{G}'(\mathbf{y}_k)\mathbf{G}'^{(-1)(j+1)}(\mathbf{y}_k) - \mathbf{I}\| < \text{Tol}_2 \\ \mathbf{G}'^{(-1)}(\mathbf{y}_k) &= \mathbf{G}'^{(-1)(j+1)}(\mathbf{y}_k) \\ \mathbf{H}'(\mathbf{z}_k) &= \mathbf{G}'^{(-1)}(\mathbf{y}_k) \\ \dot{\mathbf{y}}_k &= \mathbf{H}'(\mathbf{z}_k)\dot{\mathbf{z}}_k \end{aligned} \tag{9.10.9}$$

### 9.10.5 Explicit Parallel Manipulator Forward Kinematics

The algorithm of Section 9.10.2.1 may be applied to solve the forward kinematic equation  $\mathbf{F}(\mathbf{y}, \mathbf{z}) = \mathbf{H}(\mathbf{z}) - \mathbf{y} = \mathbf{0}$  on a time grid  $t_1 < t_2 < \dots$  for outputs  $\mathbf{z}_i = \mathbf{G}(\mathbf{y}_i)$  that are shown to exist in Section 9.4.2 and for evaluation of solution Jacobians

$\mathbf{G}'(\mathbf{y}_i) = -\mathbf{F}_z^{-1}(\mathbf{y}_i, \mathbf{z}_i)\mathbf{F}_y(\mathbf{y}_i, \mathbf{z}_i) = \mathbf{H}'^{(-1)}(\mathbf{z})$  and output velocities  $\dot{\mathbf{z}}_i = \mathbf{G}'(\mathbf{y}_i)\dot{\mathbf{y}}_i$ . The *three-step algorithm for explicit parallel manipulator forward kinematics* of Eqs. (9.10.1) through (9.10.3) is as follows:

(1) At time-step  $t_1$ , with estimate  $\mathbf{z}_1^{(1)}$ , solve  $\mathbf{H}(\mathbf{z}) - \mathbf{y}_1 = \mathbf{0}$  for  $\mathbf{z}_1 = \mathbf{G}(\mathbf{y}_1)$  and numerically evaluate  $\mathbf{H}'^{(-1)}(\mathbf{z}_1)$  and  $\mathbf{G}'(\mathbf{y}_1)$ , as follows:

$$\begin{aligned} \mathbf{H}'(\mathbf{z}_1^{(j)})\Delta\mathbf{z}_1^{(j)} &= -(\mathbf{H}(\mathbf{z}_1^{(j)}) - \mathbf{y}_1); \quad \mathbf{z}_1^{(j+1)} = \mathbf{z}_1^{(j)} + \Delta\mathbf{z}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\mathbf{H}(\mathbf{z}_1^{(j+1)}) - \mathbf{y}_1\| < \text{Tol}_1 \end{aligned} \quad (9.10.10)$$

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{z}_1^{(j+1)} = \mathbf{G}(\mathbf{y}_1) \\ \mathbf{H}'^{(-1)}(\mathbf{z}_1) &= (\mathbf{H}'(\mathbf{z}_1))^{-1} \\ \mathbf{G}'(\mathbf{y}_1) &= \mathbf{H}'^{(-1)}(\mathbf{z}_1) \end{aligned}$$

(2) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics equation  $\mathbf{H}(\mathbf{z}) - \mathbf{y}_k = \mathbf{0}$  for  $\mathbf{z}_k = \mathbf{G}(\mathbf{y}_k)$ , as follows:

$$\begin{aligned} \mathbf{z}_k^{(1)} &= \mathbf{z}_{k-1} + h\mathbf{G}'(\mathbf{y}_{k-1})\dot{\mathbf{y}}_{k-1} \\ \mathbf{z}_k^{(j+1)} &= \mathbf{z}_k^{(j)} - \mathbf{H}'^{(-1)}(\mathbf{z}_{k-1})(\mathbf{H}(\mathbf{z}_k^{(j)}) - \mathbf{y}_k), \quad i = 1, 2, \dots, \text{until } \|\mathbf{H}(\mathbf{z}_k^{(j+1)}) - \mathbf{y}_k\| < \text{Tol}_1 \\ \mathbf{z}_k &= \mathbf{z}_k^{(j+1)} = \mathbf{G}(\mathbf{y}_k) \end{aligned} \quad (9.10.11)$$

confirm  $|\mathbf{H}'(\mathbf{z}_k)| \neq 0$  and the signs is as at  $\mathbf{z}_1$

(3) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics matrix equation

$\mathbf{H}'(\mathbf{z}_k)\mathbf{H}'^{(-1)}(\mathbf{z}_k) - \mathbf{I} = \mathbf{0}$  for  $\mathbf{H}'^{(-1)}(\mathbf{z}_k)$  and evaluate the Jacobian  $\mathbf{G}'(\mathbf{y}_k)$  and output velocity  $\dot{\mathbf{z}}_k$ , as follows:

$$\begin{aligned} \mathbf{H}'^{(-1)(1)}(\mathbf{z}_k) &= \mathbf{H}'^{(-1)}(\mathbf{z}_{k-1}) \\ \mathbf{H}'^{(-1)(j+1)}(\mathbf{z}_k) &= 2\mathbf{H}'^{(-1)(j)}(\mathbf{z}_k) - \mathbf{H}'^{(-1)(j)}(\mathbf{z}_k)\mathbf{H}'(\mathbf{z}_k)\mathbf{H}'^{(-1)(j)}(\mathbf{z}_k), \quad j = 1, 2, \dots, \\ &\quad \text{until } \|\mathbf{H}'(\mathbf{z}_k)\mathbf{H}'^{(-1)(j+1)}(\mathbf{z}_k) - \mathbf{I}\| < \text{Tol}_2 \\ \mathbf{H}'^{(-1)}(\mathbf{z}_k) &= \mathbf{H}'^{(-1)(j+1)}(\mathbf{z}_k) \\ \mathbf{G}'(\mathbf{z}_k) &= \mathbf{H}'^{(-1)}(\mathbf{z}_k) \\ \dot{\mathbf{z}}_k &= \mathbf{G}'(\mathbf{y}_k)\dot{\mathbf{y}}_k \end{aligned} \quad (9.10.12)$$

### 9.10.6 Compound Manipulator Kinematics

*Forward and inverse kinematics for compound manipulators* each involve solution of two nonlinear equations and two matrix inverse equations on a time grid. This requires repeated implementation of the three-step algorithms of Sections 9.10.2.1 and 9.10.2.2 at each time-step.

#### 9.10.6.1 Compound Manipulator Forward Kinematics

For forward kinematics of a compound manipulator, with input coordinates and velocities  $\mathbf{y}_i$  and  $\dot{\mathbf{y}}_i$  specified on a time grid  $t_1 < t_2 < \dots$ , a sequence of two equation solutions is required at each time-step  $t_k$ . First, Eq. (9.2.3) in the form  $\boldsymbol{\Omega}(\mathbf{q}, \mathbf{y}_k) = \mathbf{0}$  is solved for  $\mathbf{q}_k = \mathbf{f}(\mathbf{y}_k)$ , followed by evaluation of  $\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}_k), \mathbf{y}_k)$  and the solution Jacobian  $\mathbf{f}'(\mathbf{y}_k) = -\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}_k), \mathbf{y}_k)\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}_k), \mathbf{y}_k)$ . With this task completed, Eq. (9.2.4), in the form  $\boldsymbol{\Gamma}(\mathbf{q}_k, \mathbf{z}) = \mathbf{0}$ , is solved for  $\mathbf{z}_k = \mathbf{h}(\mathbf{q}_k)$ , followed by evaluation of  $\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_k, \mathbf{h}(\mathbf{q}_k))$  and the solution Jacobian  $\mathbf{h}'(\mathbf{q}_k) = -\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_k, \mathbf{h}(\mathbf{q}_k))\boldsymbol{\Gamma}_q(\mathbf{q}_k, \mathbf{h}(\mathbf{q}_k))$ . With both tasks completed, the forward velocity mapping of Eq. (9.9.21) may be evaluated.

In summary, compound manipulator forward kinematic analysis is carried out in a sequence of two applications of the three-step algorithm of Section 9.10.2.1; i.e., a *six-step algorithm for compound manipulator forward kinematics* is as follows:

- (1) At time-step  $t_1$ , with estimate  $\mathbf{q}_1^{(1)}$ , solve  $\boldsymbol{\Omega}(\mathbf{q}, \mathbf{y}_1) = \mathbf{0}$  for  $\mathbf{q}_1 = \mathbf{f}(\mathbf{y}_1)$  and numerically evaluate  $\boldsymbol{\Omega}_q^{-1}(\mathbf{q}_1, \mathbf{y}_1)$  and  $\mathbf{f}'(\mathbf{y}_1)$ , as follows:

$$\begin{aligned} \boldsymbol{\Omega}_q(\mathbf{q}_1^{(j)}, \mathbf{y}_1)\Delta\mathbf{q}_1^{(j)} &= -\boldsymbol{\Omega}(\mathbf{q}_1^{(j)}, \mathbf{y}_1); \quad \mathbf{q}_1^{(j+1)} = \mathbf{q}_1^{(j)} + \Delta\mathbf{q}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\boldsymbol{\Omega}(\mathbf{q}_1^{(j+1)}, \mathbf{y}_1)\| < \text{Tol}_1 \\ \mathbf{q}_1 &= \mathbf{q}_1^{(j+1)} = \mathbf{f}(\mathbf{y}_1) \\ \boldsymbol{\Omega}_q^{-1}(\mathbf{q}_1, \mathbf{y}_1) &= (\boldsymbol{\Omega}_q(\mathbf{q}_1, \mathbf{y}_1))^{-1} \\ \mathbf{f}'(\mathbf{y}_1) &= -\boldsymbol{\Omega}_q^{-1}(\mathbf{q}_1, \mathbf{y}_1)\boldsymbol{\Omega}_y(\mathbf{q}_1, \mathbf{y}_1) \end{aligned} \tag{9.10.13}$$

- (2) At time-step  $t_1$ , with estimate  $\mathbf{z}_1^{(1)}$ , solve  $\boldsymbol{\Gamma}(\mathbf{q}_1, \mathbf{z}) = \mathbf{0}$  for  $\mathbf{z}_1 = \mathbf{h}(\mathbf{q}_1)$  and numerically evaluate  $\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_1, \mathbf{z}_1)$  and  $\mathbf{h}'(\mathbf{z}_1)$ , as follows:

$$\begin{aligned} \boldsymbol{\Gamma}_z(\mathbf{q}_1, \mathbf{z}_1^{(j)})\Delta\mathbf{z}_1^{(j)} &= -\boldsymbol{\Gamma}(\mathbf{q}_1, \mathbf{z}_1^{(j)}); \quad \mathbf{z}_1^{(j+1)} = \mathbf{z}_1^{(j)} + \Delta\mathbf{z}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\boldsymbol{\Gamma}(\mathbf{q}_1, \mathbf{z}_1^{(j+1)})\| < \text{Tol}_1 \\ \mathbf{z}_1 &= \mathbf{z}_1^{(j+1)} = \mathbf{h}(\mathbf{q}_1) \\ \boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_1, \mathbf{z}_1) &= (\boldsymbol{\Gamma}_z(\mathbf{q}_1, \mathbf{z}_1))^{-1} \\ \mathbf{h}'(\mathbf{q}_1) &= -\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_1, \mathbf{z}_1)\boldsymbol{\Gamma}_q(\mathbf{q}_1, \mathbf{z}_1) \end{aligned} \tag{9.10.14}$$

- (3) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics equation  $\boldsymbol{\Omega}(\mathbf{q}, \mathbf{y}_k) = \mathbf{0}$  for  $\mathbf{q}_k = \mathbf{f}(\mathbf{y}_k)$ , as follows:

$$\begin{aligned}
\mathbf{q}_k^{(1)} &= \mathbf{q}_{k-1} + h\mathbf{f}'(\mathbf{y}_{k-1})\dot{\mathbf{y}}_{k-1} \\
\mathbf{q}_k^{(j+1)} &= \mathbf{q}_k^{(j)} - \boldsymbol{\Omega}_q^{-1}(\mathbf{q}_{k-1}, \mathbf{y}_{k-1})\boldsymbol{\Omega}(\mathbf{q}_k^{(j)}, \mathbf{y}_k), j = 1, 2, \dots \text{ until } \|\boldsymbol{\Omega}(\mathbf{q}_k^{(j+1)}, \mathbf{y}_k)\| < \text{Tol}_1 \\
\mathbf{q}_k &= \mathbf{q}_k^{(j+1)} = \mathbf{f}(\mathbf{y}_k) \\
\text{confirm } |\Psi_y(\mathbf{q}_k, \mathbf{y}_k)| &\neq 0 \text{ and } |\boldsymbol{\Omega}_y(\mathbf{q}_k, \mathbf{y}_k)| \neq 0 \text{ and signs are as with } (\mathbf{q}_1, \mathbf{y}_1)
\end{aligned} \tag{9.10.15}$$

(4) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics matrix equation

$\boldsymbol{\Omega}_q(\mathbf{q}_k, \mathbf{y}_k)\boldsymbol{\Omega}_q^{-1}(\mathbf{q}_k, \mathbf{y}_k) - \mathbf{I} = \mathbf{0}$  for  $\boldsymbol{\Omega}_q^{-1}(\mathbf{q}_k, \mathbf{y}_k)$  and evaluate the Jacobian  $\mathbf{f}'(\mathbf{y}_k)$  and generalized coordinate velocity  $\dot{\mathbf{q}}_k$ , as follows:

$$\begin{aligned}
\boldsymbol{\Omega}_q^{-1(1)}(\mathbf{q}_k, \mathbf{y}_k) &= \boldsymbol{\Omega}_q^{-1}(\mathbf{q}_{k-1}, \mathbf{y}_{k-1}) \\
\boldsymbol{\Omega}_q^{-1(j+1)}(\mathbf{q}_k, \mathbf{y}_k) &= 2\boldsymbol{\Omega}_q^{-1(j)}(\mathbf{q}_k, \mathbf{y}_k) - \boldsymbol{\Omega}_q^{-1(j)}(\mathbf{q}_k, \mathbf{y}_k)\boldsymbol{\Omega}_q(\mathbf{q}_k, \mathbf{y}_k)\boldsymbol{\Omega}_q^{-1(j)}(\mathbf{q}_k, \mathbf{y}_k), \\
&\quad j = 1, 2, \dots, \text{ until } \|\boldsymbol{\Omega}_q(\mathbf{q}_k, \mathbf{y}_k)\boldsymbol{\Omega}_q^{-1(j+1)}(\mathbf{q}_k, \mathbf{y}_k) - \mathbf{I}\| < \text{Tol}_2 \\
\boldsymbol{\Omega}_q^{-1}(\mathbf{q}_k, \mathbf{y}_k) &= \boldsymbol{\Omega}_q^{-1(j+1)}(\mathbf{q}_k, \mathbf{y}_k) \\
\mathbf{f}'(\mathbf{y}_k) &= -\boldsymbol{\Omega}_q^{-1}(\mathbf{q}_k, \mathbf{y}_k)\boldsymbol{\Omega}_y(\mathbf{q}_k, \mathbf{y}_k) \\
\dot{\mathbf{q}}_k &= \mathbf{f}'(\mathbf{y}_k)\dot{\mathbf{y}}_k
\end{aligned} \tag{9.10.16}$$

(5) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics equation  $\boldsymbol{\Gamma}(\mathbf{q}_k, \mathbf{z}) = \mathbf{0}$  for  $\mathbf{z}_k = \mathbf{h}(\mathbf{q}_k)$ , as follows:

$$\begin{aligned}
\mathbf{z}_k^{(1)} &= \mathbf{z}_{k-1} + h\mathbf{h}'(\mathbf{q}_{k-1})\dot{\mathbf{q}}_{k-1} \\
\mathbf{z}_k^{(j+1)} &= \mathbf{z}_k^{(j)} - \boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_{k-1}, \mathbf{z}_{k-1})\boldsymbol{\Gamma}(\mathbf{q}_k, \mathbf{z}_k^{(j)}), j = 1, 2, \dots \text{ until } \|\boldsymbol{\Gamma}(\mathbf{q}_k, \mathbf{z}_k^{(j+1)})\| < \text{Tol}_1 \\
\mathbf{z}_k &= \mathbf{z}_k^{(j+1)} = \mathbf{h}(\mathbf{q}_k) \\
\text{confirm } |\boldsymbol{\Gamma}_z(\mathbf{q}_k, \mathbf{z}_k)| &\neq 0 \text{ and } |\boldsymbol{\Lambda}_q(\mathbf{q}_k, \mathbf{z}_k)| \neq 0 \text{ and signs are as with } (\mathbf{q}_1, \mathbf{z}_1)
\end{aligned} \tag{9.10.17}$$

(6) At time-step  $t_k$ ,  $k \geq 2$ , solve the forward kinematics matrix equation

$\boldsymbol{\Gamma}_z(\mathbf{q}_k, \mathbf{z}_k)\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_k, \mathbf{z}_k) - \mathbf{I} = \mathbf{0}$  for  $\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_k, \mathbf{z}_k)$  and evaluate the Jacobian  $\mathbf{h}'(\mathbf{q}_k)$  and output velocity  $\dot{\mathbf{z}}_k$ , as follows:

$$\begin{aligned}
\boldsymbol{\Gamma}_z^{-1(1)}(\mathbf{q}_k, \mathbf{z}_k) &= \boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_{k-1}, \mathbf{z}_{k-1}) \\
\boldsymbol{\Gamma}_z^{-1(j+1)}(\mathbf{q}_k, \mathbf{z}_k) &= 2\boldsymbol{\Gamma}_z^{-1(j)}(\mathbf{q}_k, \mathbf{z}_k) - \boldsymbol{\Gamma}_z^{-1(j)}(\mathbf{q}_k, \mathbf{z}_k)\boldsymbol{\Gamma}_z(\mathbf{q}_k, \mathbf{z}_k)\boldsymbol{\Gamma}_z^{-1(j)}(\mathbf{q}_k, \mathbf{z}_k), \\
&\quad j = 1, 2, \dots, \text{ until } \|\boldsymbol{\Gamma}_z(\mathbf{q}_k, \mathbf{z}_k)\boldsymbol{\Gamma}_z^{-1(j+1)}(\mathbf{q}_k, \mathbf{z}_k) - \mathbf{I}\| < \text{Tol}_2 \\
\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_k, \mathbf{z}_k) &= \boldsymbol{\Gamma}_z^{-1(j+1)}(\mathbf{q}_k, \mathbf{z}_k) \\
\mathbf{h}'(\mathbf{q}_k) &= -\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}_k, \mathbf{z}_k)\boldsymbol{\Gamma}_q(\mathbf{q}_k, \mathbf{z}_k) \\
\dot{\mathbf{z}}_k &= \mathbf{h}'(\mathbf{q}_k)\dot{\mathbf{q}}_k
\end{aligned} \tag{9.10.18}$$

### 9.10.6.2 Compound Manipulator Inverse Kinematics

For inverse kinematics of a compound manipulator, with output coordinates and velocities  $\mathbf{z}_i$  and  $\dot{\mathbf{z}}_i$  specified on a time grid  $t_1 < t_2 < \dots$ , a sequence of two three-step algorithms is required at each time-step. First, Eq. (9.2.5), in the form  $\Lambda(\mathbf{q}, \mathbf{z}_k) = \mathbf{0}$ , is solved for  $\mathbf{q}_k = \mathbf{e}(\mathbf{z}_k)$ , followed by solution of Eq. (9.2.2), in the form  $\Psi(\mathbf{q}_k, \mathbf{y}) = \mathbf{0}$ , for  $\mathbf{y}_k = \mathbf{g}(\mathbf{q}_k)$ . This is followed by evaluation of solution Jacobians  $\mathbf{e}'(\mathbf{z}_k) = -\Lambda_q^{-1}(\mathbf{q}_k, \mathbf{z}_k)\Lambda_z(\mathbf{q}_k, \mathbf{z}_k)$  and  $\mathbf{g}'(\mathbf{q}_k) = -\Psi_y^{-1}(\mathbf{q}_k, \mathbf{y}_k)\Psi_q(\mathbf{q}_k, \mathbf{y}_k)$ , which yields the inverse velocity mapping of Eq. (9.9.22).

In summary, compound manipulator inverse kinematic analysis is carried out in a sequence of two applications of the three-step algorithm of Section 9.10.2.2; i.e., a *six-step algorithm for compound manipulator inverse kinematics* is as follows:

- (1) At time-step  $t_1$ , with estimate  $\mathbf{q}_1^{(1)}$ , solve  $\Lambda(\mathbf{q}, \mathbf{z}_1) = \mathbf{0}$  for  $\mathbf{q}_1 = \mathbf{e}(\mathbf{z}_1)$  and numerically evaluate  $\Lambda_q^{-1}(\mathbf{q}_1, \mathbf{z}_1)$  and  $\mathbf{e}'(\mathbf{z}_1)$ , as follows:

$$\begin{aligned} \Lambda_q(\mathbf{q}_1^{(j)}, \mathbf{z}_1)\Delta\mathbf{q}_1^{(j)} &= -\Lambda(\mathbf{q}_1^{(j)}, \mathbf{z}_1); \quad \mathbf{q}_1^{(j+1)} = \mathbf{q}_1^{(j)} + \Delta\mathbf{q}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\Lambda(\mathbf{q}_1^{(j+1)}, \mathbf{z}_1)\| < \text{Tol}_1 \\ \mathbf{q}_1 &= \mathbf{q}_1^{(j+1)} = \mathbf{e}(\mathbf{z}_1) \end{aligned} \quad (9.10.19)$$

$$\begin{aligned} \Lambda_q^{-1}(\mathbf{q}_1, \mathbf{z}_1) &= (\Lambda_q(\mathbf{q}_1, \mathbf{z}_1))^{-1} \\ \mathbf{e}'(\mathbf{z}_1) &= -\Lambda_q^{-1}(\mathbf{q}_1, \mathbf{z}_1)\Lambda_z(\mathbf{q}_1, \mathbf{z}_1) \end{aligned}$$

- (2) At time-step  $t_1$ , with estimate  $\mathbf{y}_1^{(1)}$ , solve  $\Psi(\mathbf{q}_1, \mathbf{y}) = \mathbf{0}$  for  $\mathbf{y}_1 = \mathbf{g}(\mathbf{q}_1)$  and numerically evaluate  $\Psi_y^{-1}(\mathbf{q}_1, \mathbf{y}_1)$  and  $\mathbf{g}'(\mathbf{q}_1)$ , as follows:

$$\begin{aligned} \Psi_y(\mathbf{q}_1, \mathbf{y}_1^{(j)})\Delta\mathbf{y}_1^{(j)} &= -\Psi(\mathbf{q}_1, \mathbf{y}_1^{(j)}); \quad \mathbf{y}_1^{(j+1)} = \mathbf{y}_1^{(j)} + \Delta\mathbf{y}_1^{(j)}, \\ j &= 1, 2, \dots, \text{until } \|\Psi(\mathbf{q}_1, \mathbf{y}_1^{(j+1)})\| < \text{Tol}_1 \\ \mathbf{y}_1 &= \mathbf{y}_1^{(j+1)} = \mathbf{g}(\mathbf{q}_1) \end{aligned} \quad (9.10.20)$$

$$\begin{aligned} \Psi_y^{-1}(\mathbf{q}_1, \mathbf{y}_1) &= (\Psi_y(\mathbf{q}_1, \mathbf{y}_1))^{-1} \\ \mathbf{g}'(\mathbf{q}_1) &= -\Psi_y^{-1}(\mathbf{q}_1, \mathbf{y}_1)\Psi_q(\mathbf{q}_1, \mathbf{y}_1) \end{aligned}$$

- (3) At time-step  $t_k$ ,  $k \geq 2$ , solve the inverse kinematics equation  $\Lambda(\mathbf{q}, \mathbf{z}_k) = \mathbf{0}$  for  $\mathbf{q}_k = \mathbf{e}(\mathbf{z}_k)$ , as follows:

$$\begin{aligned} \mathbf{q}_k^{(1)} &= \mathbf{q}_{k-1} + h\mathbf{e}'(\mathbf{z}_{k-1})\dot{\mathbf{z}}_{k-1} \\ \mathbf{q}_k^{(j+1)} &= \mathbf{q}_k^{(j)} - \Lambda_q^{-1}(\mathbf{q}_{k-1}, \mathbf{z}_{k-1})\Lambda(\mathbf{q}_k^{(j)}, \mathbf{z}_k), \quad j = 1, 2, \dots \text{ until } \|\Lambda(\mathbf{q}_k^{(j+1)}, \mathbf{z}_k)\| < \text{Tol}_1 \\ \mathbf{q}_k &= \mathbf{q}_k^{(j+1)} = \mathbf{e}(\mathbf{z}_k) \end{aligned} \quad (9.10.21)$$

confirm  $|\Gamma_z(\mathbf{q}_k, \mathbf{z}_k)| \neq 0$  and  $|\Lambda_q(\mathbf{q}_k, \mathbf{z}_k)| \neq 0$  and signs are as with  $(\mathbf{q}_1, \mathbf{z}_1)$

(4) At time-step  $t_k$ ,  $k \geq 2$ , solve the inverse kinematics matrix equation  $\Lambda_q(\mathbf{q}_k, \mathbf{z}_k) \Lambda_q^{-1}(\mathbf{q}_k, \mathbf{z}_k) - \mathbf{I} = \mathbf{0}$  for  $\Lambda_q^{-1}(\mathbf{q}_k, \mathbf{z}_k)$  and evaluate the Jacobian  $\mathbf{e}'(\mathbf{z}_k)$  and generalized coordinate velocity  $\dot{\mathbf{q}}_k$ , as follows:

$$\begin{aligned}\Lambda_q^{-1(1)}(\mathbf{q}_k, \mathbf{z}_k) &= \Lambda_q^{-1}(\mathbf{q}_{k-1}, \mathbf{z}_{k-1}) \\ \Lambda_q^{-1(j+1)}(\mathbf{q}_k, \mathbf{z}_k) &= 2\Lambda_q^{-1(j)}(\mathbf{q}_k, \mathbf{z}_k) - \Lambda_q^{-1(j)}(\mathbf{q}_k, \mathbf{z}_k)\Lambda_q(\mathbf{q}_k, \mathbf{z}_k)\Lambda_q^{-1(j)}(\mathbf{q}_k, \mathbf{z}_k), \\ j &= 1, 2, \dots, \text{until } \|\Lambda_q(\mathbf{q}_k, \mathbf{z}_k)\Lambda_q^{-1(j+1)}(\mathbf{q}_k, \mathbf{z}_k) - \mathbf{I}\| < \text{Tol}_2 \quad (9.10.22) \\ \Lambda_q^{-1}(\mathbf{q}_k, \mathbf{z}_k) &= \Lambda_q^{-1(j+1)}(\mathbf{q}_k, \mathbf{z}_k) \\ \mathbf{e}'(\mathbf{z}_k) &= -\Lambda_q^{-1}(\mathbf{q}_k, \mathbf{z}_k)\Lambda_z(\mathbf{q}_k, \mathbf{z}_k) \\ \dot{\mathbf{q}}_k &= \mathbf{e}'(\mathbf{z}_k)\dot{\mathbf{z}}_k\end{aligned}$$

(5) At time-step  $t_k$ ,  $k \geq 2$ , with  $\mathbf{q}_k$  and  $\dot{\mathbf{q}}_k$  known, solve the inverse kinematics equation  $\Psi(\mathbf{q}_k, \mathbf{y}) = \mathbf{0}$  for  $\mathbf{y}_k = \mathbf{g}(\mathbf{q}_k)$ , as follows:

$$\begin{aligned}\mathbf{y}_k^{(1)} &= \mathbf{y}_{k-1} + h\mathbf{g}'(\mathbf{q}_{k-1})\dot{\mathbf{q}}_{k-1} \\ \mathbf{y}_k^{(j+1)} &= \mathbf{y}_k^{(j)} - \Psi_y^{-1}(\mathbf{q}_{k-1}, \mathbf{y}_{k-1})\Psi(\mathbf{q}_k, \mathbf{y}_k^{(j)}), \quad j = 1, 2, \dots, \text{until } \|\Psi(\mathbf{q}_k, \mathbf{y}_k^{(j+1)})\| < \text{Tol}_1 \quad (9.10.23) \\ \mathbf{y}_k &= \mathbf{y}_k^{(j+1)} = \mathbf{g}(\mathbf{q}_k) \\ \text{confirm } |\Psi_y(\mathbf{q}_k, \mathbf{y}_k)| &\neq 0 \text{ and } |\Omega_q(\mathbf{q}_k, \mathbf{y}_k)| \neq 0 \text{ and signs are as with } (\mathbf{q}_1, \mathbf{y}_1)\end{aligned}$$

(6) At time-step  $t_k$ ,  $k \geq 2$ , solve the inverse kinematics matrix equation

$\Psi_y(\mathbf{q}_k, \mathbf{y}_k)\Psi_y^{-1}(\mathbf{q}_k, \mathbf{y}_k) - \mathbf{I} = \mathbf{0}$  for  $\Psi_y^{-1}(\mathbf{q}_k, \mathbf{y}_k)$  and evaluate the Jacobian  $\mathbf{g}'(\mathbf{q}_k)$  and input velocity  $\dot{\mathbf{y}}_k$ , as follows:

$$\begin{aligned}\Psi_y^{-1(1)}(\mathbf{q}_k, \mathbf{y}_k) &= \Psi_y^{-1}(\mathbf{q}_{k-1}, \mathbf{y}_{k-1}) \\ \Psi_y^{-1(j+1)}(\mathbf{q}_k, \mathbf{y}_k) &= 2\Psi_y^{-1(j)}(\mathbf{q}_k, \mathbf{y}_k) - \Psi_y^{-1(j)}(\mathbf{q}_k, \mathbf{y}_k)\Psi_y(\mathbf{q}_k, \mathbf{y}_k)\Psi_y^{-1(j)}(\mathbf{q}_k, \mathbf{y}_k), \\ j &= 1, 2, \dots, \text{until } \|\Psi_y(\mathbf{q}_k, \mathbf{y}_k)\Psi_y^{-1(j+1)}(\mathbf{q}_k, \mathbf{y}_k) - \mathbf{I}\| < \text{Tol}_2 \quad (9.10.24) \\ \Psi_y^{-1}(\mathbf{q}_k, \mathbf{y}_k) &= \Psi_y^{-1(j+1)}(\mathbf{q}_k, \mathbf{y}_k) \\ \mathbf{g}'(\mathbf{q}_k) &= -\Psi_y^{-1}(\mathbf{q}_k, \mathbf{y}_k)\Psi_q(\mathbf{q}_k, \mathbf{y}_k) \\ \dot{\mathbf{y}}_k &= \mathbf{g}'(\mathbf{q}_k)\dot{\mathbf{q}}_k\end{aligned}$$

### 9.10.7 Real-Time Computation of Forward and Inverse Kinematics

One might question the wisdom of *computing inverse matrices* in forward and inverse kinematics, whereas matrix factorization might be more efficient. Two factors are presented in justification of this approach. First, extensive experience in formulating and solving ODE of multibody dynamics on differentiable manifolds in Chapter 5 has shown that there is minimal, if any, penalty in using this approach versus numerical solution of matrix equations on a time grid. Second, and much more important from a manipulator control point of view, the inverse matrices are used in Section 9.11 for efficient evaluation of *ODE of manipulator dynamics* in both input

and output coordinates. These results are directly applicable in advanced control algorithms, especially inverse dynamics control, robust control, and adaptive control. The real-time implementation analysis that follows, provided by Prof. Dan Negrut of the University of Wisconsin, strengthens the rationale for this approach.

For most manipulator applications, the *dimension of the configuration space* is 10 to 50, in which case matrices encountered will fit into the L1 or L2 cache of modern CPUs. The typical L2 cache size is between 256 and 512 KB per core, which means that it is likely that all data needed to carry out a kinematic analysis will fit in the L2 cache memory of one core. As such, any matrix-vector multiplication will benefit from data latencies of the order of 10 to 20 ns and very high bandwidths. Moreover, Intel AVX512 technologies can vectorize these matrix-vector multiplications to the point where in one clock cycle one can perform simultaneously, 8 in double precision or 16 in single precision, fused multiply-add operations; i.e., 16 and 32 operations, respectively, per clock cycle, with a clock cycle typically in the vicinity of 0.5 ns. Thus, when the dimension of the configuration space is 10 to 50, one may expect to carry out forward and inverse kinematic analysis in real-time on a 0.001 sec time grid, with a modern microprocessor.

Computational algorithms that use a matrix version of the Newton-Raphson algorithm are employed to evaluate kinematic Jacobian inverses in each of the four categories of manipulator. These matrices enable iterative evaluation of nonlinear forward and inverse configuration mappings of Sections 9.3 through 9.6 and evaluation of forward and inverse velocity mappings of Section 9.9. Even though these algorithms are intricate, an analysis is provided that supports the potential of implementing them in real-time on modern microprocessors for use in in-line control of a broad class of manipulators.

## 9.11 ODE of Manipulator Dynamics

Using results of Chapter 4, the *d'Alembert variational equation of motion* for a manipulator is written in terms of Cartesian generalized coordinates and input and output coordinates as

$$\delta \mathbf{q}^T (\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^q(\mathbf{q}, \dot{\mathbf{q}}, t)) - \delta \mathbf{y}^T \mathbf{F}^y(\mathbf{y}, \dot{\mathbf{y}}, t) - \delta \mathbf{z}^T \mathbf{Q}^z(\mathbf{z}, \dot{\mathbf{z}}, t) = 0 \quad (9.11.1)$$

which must hold for all virtual displacements  $\delta \mathbf{q}$ , input variations  $\delta \mathbf{y}$ , and output variations  $\delta \mathbf{z}$  that satisfy linearized forms of manipulator kinematic equations. In Eq. (9.11.1),  $\mathbf{M}(\mathbf{q})$  is the *mechanism mass matrix*,  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})$  is a vector of *mechanism velocity coupling terms* (sometimes called Coriolis forces),  $\mathbf{Q}^q(\mathbf{q}, \dot{\mathbf{q}}, t)$  is a vector of *mechanism generalized forces* due to gravity and internal force elements such as springs and dampers,  $\mathbf{F}^y(\mathbf{y}, \dot{\mathbf{y}}, t)$  is a vector of *input generalized forces* that are intended to control motion of the manipulator, and  $\mathbf{Q}^z(\mathbf{z}, \dot{\mathbf{z}}, t)$  is a vector of generalized forces that act on the manipulator end-effector. If manipulator generalized coordinates are not Cartesian, then the first term in Eq. (9.11.1) must be transformed from Cartesian coordinate form to manipulator generalized coordinate form. While this entails some effort, it enables use of generalized coordinates in manipulator kinematic analysis that are most natural for the application, often of minimal dimension.

Using manipulator configuration relations that are embedded in manipulator configuration manifolds defined in Sections 9.3 through 9.6, velocity mappings of Section 9.9, solution variables and Jacobians computed on a time grid in Section 9.10, and acceleration relations to be established, the variational equation of motion of Eq. (9.11.1) is reduced to *second order ODE of manipulator dynamics*, with input or output coordinates as state variables. This reduction is possible, without ad-hoc derivation, due to parameterizations of the manipulator differentiable manifold presented in Sections 9.3 through 9.6. Since they explicitly account for generalized coordinates that dominate the equations of motion of Eq. (9.11.1), ODE of compound manipulators with input and output coordinates as state variables are presented in Sections 9.11.1.1 and 9.11.1.2, respectively.

Analytical features of serial, explicit parallel, and implicit manipulators that accrue due to elimination of generalized coordinates are attractive in kinematic analysis, but disqualify them from use in dynamics. Since each of these categories is a special case of the compound manipulator, the *compound manipulator ODE formulation* is used in their dynamic analysis, taking advantage of special features of each category when possible. Due to their inherent computational efficiency, ODE in input coordinates for serial manipulators and ODE in output coordinates for explicit parallel manipulators are presented in Sections 9.11.2.1 and 9.11.3.2, using corresponding algorithms for compound manipulators. Finally, ODE in output coordinates for serial manipulators, ODE in input coordinates for explicit parallel manipulators, and ODE in input and output coordinates for implicit manipulators are presented in Sections 9.11.2.2, 9.11.3.1, and 9.11.4 using algorithms for compound manipulators.

### 9.11.1 Compound Manipulator ODE

Even though compound manipulator kinematics are more intricate than for other manipulator categories, their dynamics benefit from the fact that generalized coordinates  $\mathbf{q}$  that are required for dynamics are embedded in their formulation. With inverses of Jacobian matrices  $\boldsymbol{\Omega}_q^{-1}$ ,  $\Gamma_z^{-1}$ ,  $\Lambda_q^{-1}$ , and  $\Psi_y^{-1}$  and forward and inverse configuration and velocity solutions computed in Section 9.10.6, ODE for compound manipulator dynamics in terms of both input and output coordinates as state variables are surprisingly easy to obtain, using only matrix operations. Substantial computation is required in algorithms that evaluate kinematic quantities but, as outlined in Section 9.10.7, it can be carried out in real-time on modern microprocessors.

#### 9.11.1.1 Compound Manipulator ODE of Dynamics in Input Coordinates

With kinematics defined in Section 9.9,  $\mathbf{q} = \mathbf{f}(\mathbf{y})$ . From Section 9.9.3.1,  $\dot{\mathbf{q}} = \mathbf{f}'(\mathbf{y})\dot{\mathbf{y}}$ , where  $\mathbf{f}'(\mathbf{y}) = -\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})$ , so  $\delta\mathbf{q} = \mathbf{f}'(\mathbf{y})\delta\mathbf{y}$  and

$$\begin{aligned}\ddot{\mathbf{q}} &= \mathbf{f}'(\mathbf{y})\ddot{\mathbf{y}} - \left( \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\ddot{\mathbf{y}} \right)_y \dot{\mathbf{y}} \\ &= \mathbf{f}'(\mathbf{y})\ddot{\mathbf{y}} - \left( \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\ddot{\mathbf{b}} \right)_y \dot{\mathbf{y}}_{|b=\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\dot{\mathbf{y}}} \\ &\quad - \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y}) \left\{ \left( \boldsymbol{\Omega}_y(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{y}} \right)_q \mathbf{f}'(\mathbf{y})\dot{\mathbf{y}}_{|q=f(y)} + \left( \boldsymbol{\Omega}_y(\mathbf{f}(\dot{\mathbf{y}}), \mathbf{y})\ddot{\mathbf{y}} \right)_y \dot{\mathbf{y}} \right\}\end{aligned}$$

With  $\mathbf{a}$  constant, differentiating the identity  $\boldsymbol{\Omega}_q(\mathbf{f}(\mathbf{y}), \mathbf{y})\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\mathbf{a} = \mathbf{a}$  with respect to  $\mathbf{y}$ ,

$$\left\{ \left( \boldsymbol{\Omega}_q(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{c}} \right)_q \mathbf{f}'(\mathbf{y})_{|q=f(y)} + \left( \boldsymbol{\Omega}_q(\mathbf{f}(\dot{\mathbf{y}}), \mathbf{y})\ddot{\mathbf{c}} \right)_y \right\}_{|c=\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\mathbf{a}} + \boldsymbol{\Omega}_q(\mathbf{f}(\mathbf{y}), \mathbf{y}) \left( \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\ddot{\mathbf{a}} \right)_y = \mathbf{0},$$

where  $\mathbf{c} = \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\mathbf{a}$ . With  $\mathbf{a} = \boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\dot{\mathbf{y}}$ ,  $\mathbf{c} = \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\dot{\mathbf{y}}$  and

$$\left( \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\ddot{\mathbf{a}} \right)_y = -\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y}) \left\{ \left( \boldsymbol{\Omega}_q(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{c}} \right)_q \mathbf{f}'(\mathbf{y})_{|q=f(y)} + \left( \boldsymbol{\Omega}_q(\mathbf{f}(\dot{\mathbf{y}}), \mathbf{y})\ddot{\mathbf{c}} \right)_y \right\}_{|c=\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\dot{\mathbf{y}}}.$$

This yields

$$\begin{aligned}\ddot{\mathbf{q}} &= \mathbf{f}'(\mathbf{y})\ddot{\mathbf{y}} + \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y}) \left\{ \left( \boldsymbol{\Omega}_q(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{c}} \right)_q \mathbf{f}'(\mathbf{y}) - \left( \boldsymbol{\Omega}_y(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{y}} \right)_q \mathbf{f}'(\mathbf{y}) \right. \\ &\quad \left. + \left( \boldsymbol{\Omega}_q(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{c}} \right)_y - \left( \boldsymbol{\Omega}_y(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{y}} \right)_y \right\} \dot{\mathbf{y}}_{|q=f(y)}_{|c=\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\dot{\mathbf{y}}} \\ &= \mathbf{f}'(\mathbf{y})\ddot{\mathbf{y}} + \mathbf{J}(\mathbf{y}, \dot{\mathbf{y}})\end{aligned}$$

$$\text{where } \mathbf{J}(\mathbf{y}, \dot{\mathbf{y}}) \equiv \boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y}) \left\{ \left( \boldsymbol{\Omega}_q(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{c}} \right)_q \mathbf{f}'(\mathbf{y}) - \left( \boldsymbol{\Omega}_y(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{y}} \right)_q \mathbf{f}'(\mathbf{y}) \right. \\ \left. + \left( \boldsymbol{\Omega}_q(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{c}} \right)_y - \left( \boldsymbol{\Omega}_y(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{y}} \right)_y \right\} \dot{\mathbf{y}}_{|q=f(y)}_{|c=\boldsymbol{\Omega}_q^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{y})\boldsymbol{\Omega}_y(\mathbf{f}(\mathbf{y}), \mathbf{y})\dot{\mathbf{y}}}$$

From Section 9.9,  $\mathbf{z} = \mathbf{h}(\mathbf{q})$  and  $\mathbf{q} = \mathbf{f}(\mathbf{y})$ , so  $\mathbf{z} = \mathbf{h}(\mathbf{f}(\mathbf{y}))$ ,  $\dot{\mathbf{z}} = \mathbf{h}'(\mathbf{f}(\mathbf{y}))\mathbf{f}'(\mathbf{y})\dot{\mathbf{y}}$ ,  $\delta\mathbf{z} = \mathbf{h}'(\mathbf{f}(\mathbf{y}))\mathbf{f}'(\mathbf{y})\delta\mathbf{y}$ , and  $\delta\mathbf{q} = \mathbf{f}'(\mathbf{y})\delta\mathbf{y}$ . From Section 9.9.3.1,  $\mathbf{h}'(\mathbf{q}) = -\boldsymbol{\Gamma}_z^{-1}(\mathbf{q}, \mathbf{h}(\mathbf{q}))\boldsymbol{\Gamma}_q(\mathbf{q}, \mathbf{h}(\mathbf{q}))$ . As a function of  $\mathbf{y}$ , this is  $\mathbf{h}'(\mathbf{f}(\mathbf{y})) = -\boldsymbol{\Gamma}_z^{-1}(\mathbf{f}(\mathbf{y}), \mathbf{h}(\mathbf{f}(\mathbf{y})))\boldsymbol{\Gamma}_q(\mathbf{f}(\mathbf{y}), \mathbf{h}(\mathbf{f}(\mathbf{y})))$ . Inverse matrices required for evaluation of the preceding expressions are computed in Section 9.10.6.1.

Substituting the foregoing into Eq. (9.11.1),

$$\begin{aligned} \delta y^T f'^T(y) & \left\{ \begin{array}{l} M(f(y))f'(y)\ddot{y} + M(f(y))J(y, \dot{y}) \\ -S(f(y), f'(y)\dot{y}) - Q^q(f(y), f'(y)\dot{y}, t) \end{array} \right\} \\ -\delta y^T F^y(y, \dot{y}, t) - \delta y^T f'^T(y) h'^T(f(y))Q^z(h(f(y)), h'(f(y))f'(y)\dot{y}, t) & = 0 \end{aligned}$$

Since  $\delta y$  is arbitrary, its coefficient must be zero, yielding second order *ODE of manipulator dynamics in input coordinates*,

$$\begin{aligned} & f'^T(y)M(f(y))f'(y)\ddot{y} \\ & + f'^T(y)\left\{ M(f(y))J(y, \dot{y}) - S(f(y), f'(y)\dot{y}) - Q^q(f(y), f'(y)\dot{y}, t) \right\} \\ & - F^y(y, \dot{y}, t) - f'^T(y)h'^T(f(y))Q^z(h(f(y)), h'(f(y))f'(y)\dot{y}, t) = \mathbf{0} \end{aligned} \quad (9.11.2)$$

Note the importance of the checks  $|\Omega_q(q, y)| \neq 0 \neq |\Gamma_z(q, y)|$  in Section 9.10.6.1, which assure boundedness of  $f'(y)$  and  $h'(f(y))$  in  $\tilde{X}_\ell^C$  that appear in Eq. (9.11.2).

Kinetic energy of the system is  $KE = \dot{q}^T M(q) \dot{q} = \dot{y}^T f'^T(y) M(f(y)) f'(y) \dot{y}$ . For physically meaningful systems,  $f'(y)$  has full rank and kinetic energy is positive for all nonzero input velocities  $\dot{y}$ , so as shown in Section 4.6.3 the matrix  $f'^T(y)M(f(y))f'(y)$  is positive definite, hence nonsingular. This shows that the coefficient matrix of  $\ddot{y}$  in the ODE of Eq. (9.11.2) is nonsingular. With physically meaningful initial conditions on  $y$  and  $\dot{y}$  and moderate regularity assumptions on terms in the equations of motion of Eq. (9.11.2), the theory of ODE employed in Section 5.11 shows that the initial-value problem of Eq. (9.11.2) and these initial conditions is *well posed*; i.e., it has a unique solution that depends continuously on data that appear in the equations. The same reasoning is valid for each of the ODE presented in this Section.

### 9.11.1.2 Compound Manipulator ODE of Dynamics in Output Coordinates

With kinematics defined in Section 9.9,  $q = e(z)$ . From Section 9.9.3.2,

$\dot{q} = e'(z)\dot{z} = -\Lambda_q^{-1}(e(z), z)\Lambda_z(e(z), z)\dot{z}$ , where  $e'(z) = -\Lambda_q^{-1}(e(z), z)\Lambda_z(e(z), z)$ , so  $\delta q = e'(z)\delta z$  and

$$\begin{aligned} \ddot{q} &= e'(z)\ddot{z} - \left( \Lambda_q^{-1}(e(z), z)\Lambda_z(e(z), z)\ddot{z} \right)_z \dot{z} \\ &= e'(z)\ddot{z} - \left( \Lambda_q^{-1}(e(z), z)\ddot{b} \right)_z \dot{z} \Big|_{b=\Lambda_z(e(z), z)\dot{z}} \\ &\quad - \Lambda_q^{-1}(e(z), z) \left\{ \left( \Lambda_z(q, \ddot{z}) \right)_q e'(z) \dot{z} \Big|_{q=e(z)} + \left( \Lambda_z(e(\ddot{z}), z) \right)_z \dot{z} \right\} \end{aligned}$$

With  $a$  constant, differentiating the identity  $\Lambda_q(e(z), z)\Lambda_q^{-1}(e(z), z)a = a$  with respect to  $z$ ,

$$\left\{ \left( \Lambda_q(q, \ddot{z}) \right)_q e'(z) \Big|_{q=e(z)} + \left( \Lambda_q(e(\ddot{z}), z) \right)_z \ddot{c} \right\} \Big|_{c=\Lambda_q^{-1}(e(z), z)a} + \Lambda_q(e(z), z) \left( \Lambda_q^{-1}(e(z), z) \ddot{a} \right)_z = \mathbf{0}$$

where  $\mathbf{c} = \Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\mathbf{a}$ . With  $\mathbf{a} = \Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})\dot{\mathbf{z}}$ ,  $\mathbf{c} = \Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})\dot{\mathbf{z}}$  and  $(\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\ddot{\mathbf{a}})_z = -\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\left\{(\Lambda_q(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{c}})_q \mathbf{e}'(\mathbf{z}) + (\Lambda_q(\mathbf{e}(\dot{\mathbf{z}}), \mathbf{z})\ddot{\mathbf{c}})_z\right\}_{\substack{\mathbf{q}=\mathbf{e}(\mathbf{z}) \\ \mathbf{c}=\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})\dot{\mathbf{z}}}}$

This yields

$$\ddot{\mathbf{q}} = \mathbf{e}'(\mathbf{z})\ddot{\mathbf{z}} + \Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\left\{\begin{array}{l} (\Lambda_q(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{c}})_q \mathbf{e}'(\mathbf{z}) - (\Lambda_z(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{z}})_q \mathbf{e}'(\mathbf{z}) \\ + (\Lambda_q(\dot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{c}})_z - (\Lambda_z(\dot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{z}})_z \end{array}\right\}_{\substack{\mathbf{q}=\mathbf{e}(\mathbf{z}) \\ \mathbf{c}=\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})\dot{\mathbf{z}}}} \dot{\mathbf{y}}$$

$$= \mathbf{e}'(\mathbf{z})\ddot{\mathbf{z}} + \bar{\mathbf{J}}(\mathbf{z}, \dot{\mathbf{z}})$$

$$\text{where } \bar{\mathbf{J}}(\mathbf{z}, \dot{\mathbf{z}}) \equiv \Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\left\{\begin{array}{l} (\Lambda_q(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{c}})_q \mathbf{e}'(\mathbf{z}) - (\Lambda_z(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{z}})_q \mathbf{e}'(\mathbf{z}) \\ + (\Lambda_q(\dot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{c}})_z - (\Lambda_z(\dot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{z}})_z \end{array}\right\}_{\substack{\mathbf{q}=\mathbf{e}(\mathbf{z}) \\ \mathbf{c}=\Lambda_q^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{z})\Lambda_z(\mathbf{e}(\mathbf{z}), \mathbf{z})\dot{\mathbf{z}}}}.$$

From Section 9.9,  $\mathbf{y} = \mathbf{g}(\mathbf{q})$  and  $\mathbf{q} = \mathbf{e}(\mathbf{z})$ , so  $\mathbf{y} = \mathbf{g}(\mathbf{e}(\mathbf{z}))$ ,  $\dot{\mathbf{y}} = \mathbf{g}'(\mathbf{e}(\mathbf{z}))\mathbf{e}'(\mathbf{z})\dot{\mathbf{z}}$ ,

$\delta\mathbf{y} = \mathbf{g}'(\mathbf{e}(\mathbf{z}))\mathbf{e}'(\mathbf{z})\delta\mathbf{z}$ , and  $\delta\mathbf{q} = \mathbf{e}'(\mathbf{z})\delta\mathbf{z}$ . In addition, from Section 9.9.3.2,

$\mathbf{g}'(\mathbf{q}) = -\Psi_y^{-1}(\mathbf{q}, \mathbf{g}(\mathbf{q}))\Psi_q(\mathbf{q}, \mathbf{g}(\mathbf{q}))$ . As a function of  $\mathbf{z}$ , this is

$\mathbf{g}'(\mathbf{e}(\mathbf{z})) = -\Psi_y^{-1}(\mathbf{e}(\mathbf{z}), \mathbf{g}(\mathbf{e}(\mathbf{z})))\Psi_q(\mathbf{e}(\mathbf{z}), \mathbf{g}(\mathbf{e}(\mathbf{z})))$ . Inverse matrices required for evaluation of the preceding expressions are computed in Section 9.10.6.2.

Substituting the foregoing into Eq. (9.11.1),

$$\begin{aligned} \delta\mathbf{z}^T \mathbf{e}'^T(\mathbf{z}) &\left\{ \begin{array}{l} \mathbf{M}(\mathbf{e}(\mathbf{z}))\mathbf{e}'(\mathbf{z})\ddot{\mathbf{z}} \\ + \mathbf{M}(\mathbf{e}(\mathbf{z}))\bar{\mathbf{J}}(\mathbf{z}, \dot{\mathbf{z}}) - \mathbf{S}(\mathbf{e}(\mathbf{z}), \mathbf{e}'(\mathbf{z})\dot{\mathbf{z}}) \\ - \mathbf{Q}^q(\mathbf{e}(\mathbf{z}), \mathbf{e}'(\mathbf{z})\dot{\mathbf{z}}, t) \end{array} \right\} \\ &- \delta\mathbf{z}^T \mathbf{e}'^T(\mathbf{z}) \mathbf{g}'^T(\mathbf{e}(\mathbf{z})) \mathbf{F}^y(\mathbf{g}(\mathbf{e}(\mathbf{z})), \mathbf{g}'(\mathbf{e}(\mathbf{z}))\dot{\mathbf{z}}, t) - \delta\mathbf{z}^T \mathbf{Q}^z(\mathbf{z}, \dot{\mathbf{z}}, t) = 0 \end{aligned}$$

Since  $\delta\mathbf{z}$  is arbitrary, its coefficient must be zero, yielding second order *ODE of manipulator dynamics in output coordinates*,

$$\begin{aligned} &\mathbf{e}'^T(\mathbf{z}) \mathbf{M}(\mathbf{e}(\mathbf{z})) \mathbf{e}'(\mathbf{z}) \ddot{\mathbf{z}} \\ &+ \mathbf{e}'^T(\mathbf{z}) \left\{ \begin{array}{l} \mathbf{M}(\mathbf{e}(\mathbf{z})) \bar{\mathbf{J}}(\mathbf{z}, \dot{\mathbf{z}}) - \mathbf{S}(\mathbf{e}(\mathbf{z}), \mathbf{e}'(\mathbf{z})\dot{\mathbf{z}}) \\ - \mathbf{Q}^q(\mathbf{e}(\mathbf{z}), \mathbf{e}'(\mathbf{z})\dot{\mathbf{z}}, t) \end{array} \right\} \\ &- \mathbf{e}'^T(\mathbf{z}) \mathbf{g}'^T(\mathbf{e}(\mathbf{z})) \mathbf{F}^y(\mathbf{g}(\mathbf{e}(\mathbf{z})), \mathbf{g}'(\mathbf{e}(\mathbf{z}))\dot{\mathbf{z}}, t) - \mathbf{Q}^z(\mathbf{z}, \dot{\mathbf{z}}, t) = \mathbf{0} \end{aligned} \quad (9.11.3)$$

Note the importance of the checks  $|\Lambda_q(\mathbf{q}, \mathbf{z})| \neq 0 \neq |\Psi_y(\mathbf{q}, \mathbf{y})|$  in Section 9.10.6.2, which assure boundedness of  $\mathbf{e}'(\mathbf{z})$  and  $\mathbf{g}'(\mathbf{e}(\mathbf{z}))$  in  $\tilde{X}_\ell^C$  that appear in Eq. (9.11.3). As in Eq. (9.11.2), the coefficient matrix of  $\ddot{\mathbf{z}}$  is positive definite and an initial-value problem with Eq. (9.11.3) is well posed.

## **9.11.2 Serial Manipulator ODE**

### **9.11.2.1 Serial Manipulator ODE of Dynamics in Input Coordinates**

The algorithm of Section 9.11.1.1, implemented with efficiencies provided by forward kinematic computations of Section 9.10.1, yields *ODE of dynamics for serial manipulators* with inputs as state variables. The gain in efficiency by taking advantage of state-of-the-art forward kinematics for serial manipulators may be significant.

### **9.11.2.2 Serial Manipulator ODE of Dynamics in Output Coordinates**

Same as Section 9.11.1.2, taking advantage of efficiencies in numerical computation in Section 9.10.5.

## **9.11.3 Explicit Parallel Manipulator ODE**

### **9.11.3.1 Explicit Parallel Manipulator ODE of Dynamics in Input Coordinates**

Same as Section 9.11.1.1, taking advantage of efficiencies in numerical computation in Section 9.10.5.

### **9.11.3.2 Explicit Parallel Manipulator ODE of Dynamics in Output Coordinates**

The algorithm of Section 9.11.1.2, implemented with efficiencies provided by inverse kinematic computations of Section 9.10.5, yields *ODE of dynamics for explicit parallel manipulators* with output coordinates as state variables. The gain in efficiency by taking advantage of state-of-the-art inverse kinematics for explicit parallel manipulators may be significant.

## **9.11.4 Implicit Manipulator ODE**

### **9.11.4.1 Implicit Manipulator ODE of Dynamics in Input Coordinates**

The algorithm of Section 9.11.1.1, implemented with efficiencies provided by forward kinematic computations of Section 9.10.3.1, yields *ODE of dynamics for implicit manipulators* with inputs as state variables.

### **9.11.4.2 Implicit Manipulator ODE of Dynamics in Output Coordinates**

The algorithm of Section 9.11.1.2, implemented with efficiencies provided by inverse kinematic computations of Section 9.10.3.2, yields ODE of dynamics for implicit manipulators with outputs as state variables.

Using the full generality of the compound manipulator formulation, the variational equation of motion for all four manipulator categories is systematically reduced to ODE in terms of both input and output coordinates as state variables. This minimizes the need for ad-hoc derivation of ODDE of dynamics and enables exploitation of efficiencies made possible by special characteristics of serial and explicit parallel manipulators. Finally, as shown, there is potential for real-time implementation of manipulator equations of motion, using modern high speed microprocessors as an in-line component of modern control systems.

## 9.12 Equations of Dynamics for Model Manipulators

A variational equation of motion in terms of manipulator generalized, input, and output coordinates, based on results of Chapter 4, is the starting point for obtaining ODE of dynamics for manipulators in Section 9.11. These results, with vectors and matrices computed on a time grid in Section 9.10, reduce the problem of evaluating the ODE of dynamics in Section 9.11 to one of linear algebra, with no ad-hoc derivation required. Since each of the model manipulators treated in Sections 9.3 through 9.6 is based on a planar mechanism, the formulation of planar system variational equations of motion is first summarized in Section 9.12.1, followed by application to the four model manipulators in Sections 9.12.2 through 9.12.5. A derivation of ODE of dynamics for the spatial six-leg Gaugh-Stewart platform, using a differential geometric formulation similar to that employed herein, is presented in (Haug, 2021c)

### 9.12.1 Variational Equation of Motion for Planar Systems

For body  $i$  in a planar system, shown in Fig. 9.1.21, Cartesian generalized coordinates are  $\mathbf{q}_i = [\mathbf{r}_i^T \quad \phi_i]^T$ ,  $i = 1, \dots, nb$ , where  $nb$  is the number of bodies in a system that is subject to holonomic constraints. The composite vector of Cartesian coordinates for the system is

$$\mathbf{q} = [\mathbf{q}_1^T \quad \dots \quad \mathbf{q}_{nb}^T]^T \in \mathbb{R}^{3nb}. \text{ The orientation matrix for body } i \text{ is } \mathbf{A}_i = \mathbf{A}(\phi_i) = \begin{bmatrix} \cos \phi_i & -\sin \phi_i \\ \sin \phi_i & \cos \phi_i \end{bmatrix},$$

with properties  $\mathbf{A}_i^T \mathbf{A}_i = \mathbf{I}_2$ ,  $\mathbf{P} \equiv \mathbf{A}(\pi/2)$ ,  $\mathbf{P}^T = -\mathbf{P}$ ,  $\mathbf{P}\mathbf{P} = -\mathbf{I}_2$ , and  $\mathbf{A}'(\phi) = (d/d\phi)\mathbf{A}(\phi) = \mathbf{PA}(\phi)$ . From Section 4.2, the variational equation of motion of a system of  $nb$  bodies, in terms of Cartesian generalized coordinates, is

$$\sum_{i=1}^{nb} \delta \mathbf{q}_i^T (\mathbf{M}_i(\mathbf{q}_i) \ddot{\mathbf{q}}_i - \mathbf{S}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) - \mathbf{Q}_i^{q_i}(\mathbf{q}_i, \dot{\mathbf{q}}_i, t)) - \delta \mathbf{y}^T \mathbf{F}^y(\mathbf{y}, \dot{\mathbf{y}}, t) - \delta \mathbf{z}^T \mathbf{Q}^z(\mathbf{z}, \dot{\mathbf{z}}, t) \\ = \delta \mathbf{q}^T (\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} - \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^q(\mathbf{q}, \dot{\mathbf{q}}, t)) - \delta \mathbf{y}^T \mathbf{F}^y(\mathbf{y}, \dot{\mathbf{y}}, t) - \delta \mathbf{z}^T \mathbf{Q}^z(\mathbf{z}, \dot{\mathbf{z}}, t) = 0 \quad (9.12.1)$$

where, for body  $i$ ,  $\mathbf{M}_i(\mathbf{q}_i) = \begin{bmatrix} m_i \mathbf{I}_2 & m_i \mathbf{PA}(\phi_i) \mathbf{s}'_i \\ m_i \mathbf{s}'_i^T \mathbf{A}^T(\phi_i) \mathbf{P}^T & J_i \end{bmatrix}$ ,  $\mathbf{S}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) = \begin{bmatrix} m_i \dot{\phi}_i^2 \mathbf{A}(\phi_i) \mathbf{s}'_i \\ 0 \end{bmatrix}$ ,

$\mathbf{Q}_i^{q_i}(\mathbf{q}_i, \dot{\mathbf{q}}_i, t) = [\mathbf{F}_i^T(\mathbf{q}_i, \dot{\mathbf{q}}_i, t) \quad n'_i(\mathbf{q}_i, \dot{\mathbf{q}}_i, t)]^T$ , and  $\mathbf{s}'_i$  is a body-fixed vector that locates the centroid of the body, relative to the body-fixed  $x'_i$ - $y'_i$  reference frame. For a multibody system,  $\mathbf{M}(\mathbf{q}) = \text{diag}(\mathbf{M}_1(\mathbf{q}_1) \quad \dots \quad \mathbf{M}_{nb}(\mathbf{q}_{nb}))$ ,  $\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) = [\mathbf{S}_1^T(\mathbf{q}_1, \dot{\mathbf{q}}_1) \quad \dots \quad \mathbf{S}_{nb}^T(\mathbf{q}_{nb}, \dot{\mathbf{q}}_{nb})]^T$ ,  $\mathbf{Q}^q(\mathbf{q}, \dot{\mathbf{q}}, t) = [\mathbf{Q}_1^{q_1 T}(\mathbf{q}_1, \dot{\mathbf{q}}_1, t) \quad \dots \quad \mathbf{Q}_{nb}^{q_{nb} T}(\mathbf{q}_{nb}, \dot{\mathbf{q}}_{nb}, t)]^T$ ,  $\mathbf{F}^y(\mathbf{y}, \dot{\mathbf{y}}, t)$  is input generalized force that is intended to control the manipulator, and  $\mathbf{Q}^z(\mathbf{z}, \dot{\mathbf{z}}, t)$  is generalized force that acts on the end effector. The variational equation of Eq. (9.12.1) holds for all  $\delta \mathbf{q}$ ,  $\delta \mathbf{y}$ , and  $\delta \mathbf{z}$  that satisfy the linearized form of manipulator constraints of Eqs. (9.2.1), (9.2.2), and (9.2.4).

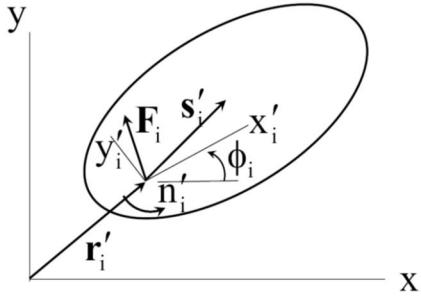


Figure 9.12.1 Planar Body i

In case generalized coordinates used in a manipulator application are not Cartesian, a transformation must be derived that defines Cartesian coordinates as a function of the manipulator generalized coordinates used. In such a case, let  $\bar{\mathbf{q}}$  be Cartesian coordinates and the transformation required is

$$\bar{\mathbf{q}} = \mathbf{T}(\mathbf{q}) \quad (9.12.2)$$

Derivatives and differential are thus

$$\begin{aligned}\dot{\bar{\mathbf{q}}} &= \mathbf{T}'(\mathbf{q})\dot{\mathbf{q}} \\ \ddot{\bar{\mathbf{q}}} &= \mathbf{T}'(\mathbf{q})\ddot{\mathbf{q}} + \left( \mathbf{T}'(\mathbf{q})\ddot{\mathbf{q}} \right)_{\mathbf{q}} \dot{\mathbf{q}} \\ \delta\bar{\mathbf{q}} &= \mathbf{T}'(\mathbf{q})\delta\mathbf{q}\end{aligned} \quad (9.12.3)$$

The first term in Eq. (9.12.1) is thus transformed as follows:

$$\delta\bar{\mathbf{q}}^T \left( \mathbf{M}(\bar{\mathbf{q}})\ddot{\bar{\mathbf{q}}} - \mathbf{S}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) - \mathbf{Q}^{\bar{\mathbf{q}}}(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t) \right) = \delta\mathbf{q}^T \mathbf{T}'^T(\mathbf{q}) \begin{pmatrix} \mathbf{M}(\mathbf{T}(\mathbf{q}))\ddot{\mathbf{q}} + \mathbf{M}(\mathbf{T}(\mathbf{q}))\left( \mathbf{T}'(\mathbf{q})\ddot{\mathbf{q}} \right)_{\mathbf{q}} \dot{\mathbf{q}} \\ -\mathbf{S}(\mathbf{T}(\mathbf{q}), \mathbf{T}'(\mathbf{q})\dot{\mathbf{q}}) - \mathbf{Q}^{\bar{\mathbf{q}}}(\mathbf{T}(\mathbf{q}), \mathbf{T}'(\mathbf{q})\dot{\mathbf{q}}, t) \end{pmatrix} \quad (9.12.4)$$

The variational equation of motion of Eq. (9.12.1), hence Eq. (9.11.1), may thus be transformed to manipulator generalized coordinates. While this requires some effort, it enables the engineer/analyst to employ reduced dimension generalized coordinates that may greatly enhance efficiency of computation carried out in Section 9.10.

The ODE of manipulator dynamics are derived for each of the four manipulator categories in Section 9.11. All matrices that are required to implement these ODE for the model problems studied in Sections 9.3 through 9.6 are presented in Sections 9.12.2 through 9.12.5. With these data, ODE of manipulator dynamics may be implemented using only matrix operations, without ad-hoc derivation.

## 9.12.2 Model Serial Manipulator

With noncentroidal body-fixed reference frames located at the left end of each body in Fig. 9.3.1(a), generalized coordinates are Cartesian, where  $\mathbf{r}_1 = \mathbf{0}_2 \equiv [0 \ 0]^T$ ,  $\phi_1 = y_1$ ,  $\mathbf{r}_2 = \mathbf{A}(y_1)\mathbf{u}_x$ ,  $\phi_2 = y_1 + y_2$ ,  $\mathbf{r}_3 = (\mathbf{A}(y_1) + \mathbf{A}(y_1 + y_2))\mathbf{u}_x$ , and  $\phi_3 = y_1 + y_2 + y_3$ . Kinetic terms are  $J'_i = m_i / 3$ ,  $s'_i = (1/2)\mathbf{u}_x$ ,  $\mathbf{F}_i = m_i g \mathbf{u}_y$ , and  $n'_i = 0$ ,  $i = 1, 2$ , and  $3$ . Generalized, input, and output

coordinates are  $\mathbf{q} \in \mathbb{R}^7$ ,  $\mathbf{y} \in \mathbb{R}^3$ , and  $\mathbf{z} \in \mathbb{R}^3$ . Jacobians of manipulator kinematic equations of Eq. (9.3.2) are the sparse matrices

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} \sin q_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\cos q_1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & \sin q_4 & 1 & 0 & 0 \\ 0 & 0 & -1 & -\cos q_4 & 0 & 1 & 0 \end{bmatrix}$$

$$\Psi_q(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Gamma_q(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Psi_y(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \Gamma_z(\mathbf{q}, \mathbf{z}) = -\mathbf{I}_3$$

With these Jacobians,

$$\Omega_q(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} \Phi_q(\mathbf{q}) \\ \Psi_q(\mathbf{q}, \mathbf{y}) \end{bmatrix} \quad \Omega_y(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} 0 \\ \Psi_q(\mathbf{q}, \mathbf{y}) \end{bmatrix} \quad \Lambda_q(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} \Phi_q(\mathbf{q}) \\ \Gamma_q(\mathbf{q}, \mathbf{z}) \end{bmatrix} \quad \Lambda_z(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} 0 \\ \Gamma_q(\mathbf{q}, \mathbf{z}) \end{bmatrix}$$

and second order derivatives required are

$$(\Phi_q(\mathbf{q})\ddot{\mathbf{c}})_q = \begin{bmatrix} c_1 \cos q_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 \sin q_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_4 \cos q_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_4 \sin q_4 & 0 & 0 & 0 \end{bmatrix} \quad (\Omega_q(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{c}})_q = \begin{bmatrix} (\Phi_q(\mathbf{q})\ddot{\mathbf{c}})_q \\ 0 \end{bmatrix}$$

$$(\Omega_q(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{y}})_y = 0 \quad (\Omega_y(\mathbf{q}, \ddot{\mathbf{y}})\ddot{\mathbf{y}})_q = 0 \quad (\Omega_y(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{c}})_y = 0$$

$$(\Lambda_q(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{c}})_q = \begin{bmatrix} (\Phi_q(\mathbf{q})\ddot{\mathbf{c}})_q \\ 0 \end{bmatrix} \quad (\Lambda_q(\ddot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{c}})_z = 0 \quad (\Lambda_z(\mathbf{q}, \ddot{\mathbf{z}})\ddot{\mathbf{z}})_q = 0 \quad (\Lambda_z(\ddot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{z}})_z = 0$$

All other terms required for the ODE of dynamics in Section 9.11.2 are available in Section 9.3, the kinematics algorithms of Section 9.10.2, and Section 9.12.1.

### 9.12.3 Model Explicit Parallel Manipulator

For the single moving body of the explicit parallel manipulator platform in Fig. 9.4.1, the origin O is the centroid, so Cartesian generalized coordinates are  $\mathbf{r} = z_1 \mathbf{u}_y$  and  $\phi = z_2$ . Kinetic terms for the equations of motion are  $\mathbf{s}' = \mathbf{0}$ ,  $J' = 25\text{m}/3$ ,  $n'(\mathbf{v}, \dot{\mathbf{v}}, t) = 0$ , and  $\mathbf{F}(\mathbf{v}, \dot{\mathbf{v}}, t) = -mg\mathbf{u}_y$ . Generalized, input, and output coordinates are  $\mathbf{q} \in \mathbb{R}^2$ ,  $\mathbf{y} \in \mathbb{R}^2$ , and  $\mathbf{z} \in \mathbb{R}^2$ . Jacobians of input and output equations of Eqs. (9.4.2) and (9.4.3) are

$$\begin{aligned}\Omega_q(\mathbf{q}, \mathbf{y}) &= 2 \begin{bmatrix} q_1 + 6 \sin q_2 & 30 \sin q_2 + 6 q_1 \cos q_2 \\ q_1 - 5 \sin q_2 & 30 \sin q_2 - 5 q_1 \cos q_2 \end{bmatrix} \quad \Omega_y(\mathbf{q}, \mathbf{y}) = -2 \begin{bmatrix} y_1 & 0 \\ 0 & y_2 \end{bmatrix} \\ \Lambda_q(\mathbf{q}, \mathbf{z}) &= \mathbf{I}_2 \quad \Lambda_z(\mathbf{q}, \mathbf{z}) = -\mathbf{I}_2\end{aligned}$$

With these Jacobians, second derivative terms required are

$$\begin{aligned}(\Omega_q(\mathbf{q}, \mathbf{y}) \ddot{\mathbf{c}})_q &= 2 \begin{bmatrix} c_1 + 6c_2 \cos q_2 & (6c_1 + 30c_2) \cos q_2 - 6c_2 q_1 \sin q_2 \\ c_1 - 5c_2 \cos q_2 & (-5c_1 + 30c_2) \cos q_2 + 5c_2 q_1 \sin q_2 \end{bmatrix} \\ (\Omega_q(\mathbf{q}, \mathbf{y}) \ddot{\mathbf{c}})_y &= 0 \quad (\Omega_y(\mathbf{q}, \mathbf{y}) \ddot{\mathbf{y}})_q = 0 \quad (\Omega_y(\mathbf{q}, \mathbf{y}) \ddot{\mathbf{y}})_y = -2 \begin{bmatrix} \dot{y}_1 & 0 \\ 0 & \dot{y}_2 \end{bmatrix} \\ (\Lambda_q(\mathbf{q}, \mathbf{z}) \ddot{\mathbf{c}})_q &= (\Lambda_q(\mathbf{q}, \mathbf{z}) \ddot{\mathbf{c}})_z = (\Lambda_z(\mathbf{q}, \mathbf{z}) \ddot{\mathbf{z}})_q = (\Lambda_z(\mathbf{q}, \mathbf{z}) \ddot{\mathbf{z}})_z = 0\end{aligned}$$

All other terms required for the ODE of dynamics in Section 9.11.3 are available in Section 9.4, the kinematics algorithms of Section 9.10, and Section 9.12.1.

#### 9.12.4 Model Implicit Manipulator

With  $\mathbf{q} = \mathbf{z}$  in Section 9.5.1, there is no holonomic constraint  $\Phi(\mathbf{q}) = 0$  on Cartesian generalized coordinate  $\mathbf{q}$ , so Eq. (9.5.1) may be written as

$$\Psi(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} 1 + \cos y_1 + \cos(y_1 + y_2) - q_1 - \cos q_3 \\ \sin y_1 + \sin(y_1 + y_2) - q_2 - \sin q_3 \\ (1/2)((y_3)^2 - (q_1 - \cos q_3)^2 - (q_2 - \sin q_3)^2) \end{bmatrix} = \mathbf{0}$$

and

$$\Gamma(\mathbf{q}, \mathbf{z}) = \mathbf{q} - \mathbf{z} = \mathbf{0}$$

Thus,  $\Omega(\mathbf{q}, \mathbf{y}) = \Psi(\mathbf{q}, \mathbf{y})$  and  $\Lambda(\mathbf{q}, \mathbf{z}) = \Gamma(\mathbf{q}, \mathbf{z})$ , so

$$\begin{aligned}\Omega_q(\mathbf{q}, \mathbf{y}) &= \begin{bmatrix} -1 & 0 & \sin q_3 \\ 0 & -1 & -\cos q_3 \\ -(q_1 - \cos q_3) & -(q_2 - \sin q_3) & -q_1 \sin q_3 + q_2 \cos q_3 \end{bmatrix} \\ \Omega_y(\mathbf{q}, \mathbf{y}) &= \begin{bmatrix} -\sin y_1 - \sin(y_1 + y_2) & -\sin(y_1 + y_2) & 0 \\ \cos y_1 + \cos(y_1 + y_2) & \cos(y_1 + y_2) & 0 \\ 0 & 0 & y_3 \end{bmatrix} \\ \Lambda_q(\mathbf{q}, \mathbf{z}) &= \mathbf{I}_3 \quad \Lambda_z(\mathbf{q}, \mathbf{z}) = -\mathbf{I}_3\end{aligned}$$

With these Jacobians, second derivative terms required are

$$\begin{aligned} \left( \Omega_q(\mathbf{q}, \dot{\mathbf{y}}) \ddot{\mathbf{c}} \right)_q &= \begin{bmatrix} 0 & 0 & c_3 \cos q_3 \\ 0 & 0 & c_3 \sin q_3 \\ -c_1 - c_3 \sin q_3 & -c_2 + c_3 \cos q_3 & -c_1 \sin q_3 + c_2 \cos q_3 - c_3 (q_1 \cos q_3 + q_2 \sin q_3) \end{bmatrix} \\ \left( \Omega_q(\dot{\mathbf{q}}, \mathbf{y}) \ddot{\mathbf{c}} \right)_y &= 0 \quad \left( \Omega_y(\mathbf{q}, \dot{\mathbf{y}}) \ddot{\mathbf{c}} \right)_q = 0 \\ \left( \Omega_y(\dot{\mathbf{q}}, \mathbf{y}) \ddot{\mathbf{c}} \right)_y &= \begin{bmatrix} -c_1 (\cos y_1 + \cos(y_1 + y_2)) - c_2 \cos(y_1 + y_2) & (c_1 + c_2) \cos(y_1 + y_2) & 0 \\ -c_1 (\sin y_1 + \sin(y_1 + y_2)) - c_2 \sin(y_1 + y_2) & -(c_1 + c_2) \sin(y_1 + y_2) & 0 \\ 0 & 0 & c_3 \end{bmatrix} \\ \left( \Lambda_q(\mathbf{q}, \dot{\mathbf{z}}) \ddot{\mathbf{c}} \right)_q &= \left( \Lambda_q(\dot{\mathbf{q}}, \mathbf{z}) \ddot{\mathbf{c}} \right)_z = \left( \Lambda_z(\mathbf{q}, \dot{\mathbf{z}}) \ddot{\mathbf{c}} \right)_q = \left( \Lambda_z(\dot{\mathbf{q}}, \mathbf{z}) \ddot{\mathbf{c}} \right)_z = 0 \end{aligned}$$

All other terms required for the ODE of dynamics in Section 9.11.4 are available in Section 9.5, the kinematics algorithms of Section 9.10.3, and Section 9.12.1.

### 9.12.5 Model Compound Manipulator

For the *front-end loader compound manipulator* in Fig. 9.6.1, Cartesian generalized coordinates are  $\mathbf{r}_1 = \mathbf{0}$ ,  $\phi_1 = q_1$ ,  $\mathbf{r}_2 = q_2 \mathbf{A}(q_1) \mathbf{u}_x$ ,  $\phi_2 = q_1$ ,  $\mathbf{r}_3 = 4\mathbf{A}(q_1) \mathbf{u}_x$ , and  $\phi_3 = q_1 + q_3$ . Thus, the transformation from manipulator generalized coordinates to Cartesian generalized coordinates of Eq. (9.12.2) is

$$\bar{\mathbf{q}} = \mathbf{T}(\mathbf{q}) = \begin{bmatrix} 0 & 0 & q_1 & (q_2 \mathbf{A}(q_1) \mathbf{u}_x)^T & q_1 & (4\mathbf{A}(q_1) \mathbf{u}_x)^T & (q_1 + q_3) \end{bmatrix}^T \quad (9.12.5)$$

which enables the transformation of Eq. (9.12.4) of the variational equation of motion from cartesian generalized coordinates to manipulator generalized coordinates. This simple transformation reduces computation in Sections 9.10 and 9.11 from dimension 9 in Cartesian coordinates to dimension 3 in manipulator generalized coordinates. The gain in computational efficiency is enormous.

Kinetic terms are  $J'_1 = m_1 / 3$ ,  $J'_2 = m_2 / 12$ ,  $J'_3 = m_3 / 2$ ,  $s'_1 = 4\mathbf{u}_x$ ,  $s'_2 = \mathbf{0}_2$ ,  $s'_3 = (1/2)(\mathbf{u}_x + \mathbf{u}_y)$ ,  $\mathbf{n}' = \mathbf{0}_{3 \times 1}$ , and  $\mathbf{F} = \begin{bmatrix} -m_1 g \mathbf{u}_y^T & -m_2 g \mathbf{u}_y^T & -m_3 g \mathbf{u}_y^T \end{bmatrix}^T$ . Generalized, input, and output coordinates are  $\mathbf{q} \in \mathbb{R}^3$ ,  $\mathbf{y} \in \mathbb{R}^2$ , and  $\mathbf{z} \in \mathbb{R}^2$ . The Jacobians of terms in Section 9.6.1 are

$$\begin{aligned} \mathbf{q}(\mathbf{q}) &= \begin{bmatrix} 0 & 1 & (1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \cos q_3 \end{bmatrix} \\ \Psi_y(\mathbf{q}, \mathbf{y}) &= \begin{bmatrix} -2y_1 & 0 \\ 0 & -1 \end{bmatrix} \quad \Psi_q(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} 3 \cos q_1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \Gamma_z(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \\ \Gamma_q(\mathbf{q}, \mathbf{z}) &= \begin{bmatrix} 1 & 0 & 1 \\ 4 \cos q_1 + \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) & 0 & \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) \end{bmatrix} \end{aligned}$$

yielding

$$\Omega_q(\mathbf{q}, \mathbf{y}) = \begin{bmatrix} 0 & 1 & (1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \cos q_3 \\ 3 \cos q_1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\Lambda_q(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} 0 & 1 & (1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3) \cos q_3 \\ 1 & 0 & 1 \\ 4 \cos q_1 + \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) & 0 & \cos(q_1 + q_3) + 0.5 \sin(q_1 + q_3) \end{bmatrix}$$

Required second derivative matrices are

$$(\Omega_q(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{c}})_q = \begin{bmatrix} 0 & 0 & c_3 a \\ -3c_1 \sin q_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where

$$a \equiv -\left(1 + (\sin^2 q_3 + 3)^{-1/2} \sin q_3\right) \sin q_3$$

$$+ \left((\sin^2 q_3 + 3)^{-1/2} \cos q_3 - \sin^2 q_3 \cos q_3 (\sin^2 q_3 + 3)^{-3/2}\right) \cos q_3$$

$$(\Omega_q(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{c}})_y = 0 \quad (\Omega_y(\mathbf{q}, \dot{\mathbf{y}})\ddot{\mathbf{y}})_q = 0 \quad (\Omega_y(\ddot{\mathbf{q}}, \mathbf{y})\ddot{\mathbf{y}})_y = \begin{bmatrix} 0 & 0 \\ -2c_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$(\Lambda_q(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{c}})_q = \begin{bmatrix} 0 & 0 & c_3 a \\ 0 & 0 & 0 \\ b & 0 & d \end{bmatrix}$$

where

$$b = -4c_1 \sin q_1 + (c_1 + c_3)(-\sin(q_1 + q_3) + 0.5 \cos(q_1 + q_3))$$

$$d = (c_1 + c_3)(-\sin(q_1 + q_3) + 0.5 \cos(q_1 + q_3))$$

and

$$(\Lambda_q(\ddot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{c}})_z = 0 \quad (\Lambda_z(\mathbf{q}, \dot{\mathbf{z}})\ddot{\mathbf{z}})_q = 0 \quad (\Lambda_z(\ddot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{z}})_z = 0$$

All other terms required for the ODE of dynamics in Section 9.11.1 are available in Section 9.6, the kinematics algorithms of Section 9.10.6, and Section 9.12.1.

### 9.12.6 Real-Time Computation of the ODE of Dynamics

As illustrated by computation of terms in equations of motion for model problems in Sections 9.12.2 through 9.12.5, only elementary matrix operations are required to numerically evaluate the ODE of dynamics. In addition, most matrices involved are sparse, enhancing numerical efficiency. In general, even for spatial systems, computations in Section 9.11 involve

only manipulation of matrices that are evaluated in Section 9.10. These computations may thus be implemented in real-time. The greater challenge to achieving real-time evaluation of the ODE of dynamics is computations in forward and inverse kinematics of Section 9.10. As shown in Section 9.10.7, these calculations can be implemented, for meaningful manipulator applications, in real-time on modern microprocessors. The conclusion is, therefore, that all computation required for evaluation of the ODE of dynamics for meaningful manipulators can be carried out in real-time on modern microprocessors. This enables evaluation of equations for forward and inverse kinematics and ODE of dynamics needed for real-time in-line control of manipulators using modern control methods, such as those presented in (Siciliano, Sciavicco, Villani, and Oriolo, 2010), practical in real-time.

Evaluation of quantities for each of the four model manipulators that are required for implementation of the ODE of motion of Section 9.11 is straightforward. No ad-hoc derivations are required to implement the ODE of motion for all four categories of manipulator. It is further shown that there is potential for real-time implementation of the equations of motion using modern high speed microprocessors in-line with modern control algorithms.

## 9.13 Introduction to Part II, Redundant Manipulators

Nonredundant manipulators have numbers of input and output coordinates equal to the number of degrees of freedom of the underlying mechanism. As shown in Part I, except at singular configurations, they have single valued forward and inverse kinematic mappings. In contrast, *kinematically redundant manipulators* (Chiaverine, Oriolo, and Maciejewski, 2016; Luces, Mills, and Benhabib, 2017) have a number of input coordinates equal to the number of degrees of freedom of the underlying mechanism, but have fewer output coordinates. This creates a situation in which inverse kinematic mappings are set valued, providing the control engineer with freedom to plan trajectories in configuration space that avoid obstacles, avoid singularities, or enhance performance of the manipulator. This opportunity comes with attendant analytical complexity associated with the fact that inverse kinematic mappings are set-valued; i.e., for a given output, there are an infinite number of inputs that yield the given output. As a result, the role of differentiable manifolds in kinematics of redundant manipulators is pervasive, whereas in the case of nonredundant manipulators, analysis on manipulator configuration manifolds reduces for the most part to multivariable calculus. A restricted class of redundant manipulators, called *redundant input manipulators*, has inputs greater in number than the number of mechanism degrees of freedom (Muller, A, and Hufnagel, T, 2012). This category of redundant manipulator is not addressed herein.

The treatment of nonredundant manipulator kinematics and dynamics on differentiable manifolds presented in Part I of this chapter is extended to *kinematically redundant manipulators* in Part II. Analysis at the configuration level shows that forward kinematics and dynamics of redundant manipulators are identical to that for nonredundant manipulators. Manifold-based inverse kinematics for redundant manipulators, in contrast, yield *parameterizations of set-valued inverse mappings* at the configuration level, where sharper results are obtained than those presented in the literature using velocity formulations. Explicit expressions are derived for *set-valued configuration inverse mappings* for both serial and non-serial composite kinematically redundant manipulators, as functions of vectors of free parameters. Parameterizations are presented for both manipulator *regular configuration manifolds* and *self-motion manifolds*, the latter comprised of sets of inputs that map into the same output. It is shown that kinematically redundant configuration and self-motion differentiable manifolds are distinctly different and play complementary roles in redundant manipulator kinematics. Computational methods are presented for evaluation of set-valued inverse kinematic mappings, without problem dependent ad-hoc analytical manipulations. Redundant serial and compound manipulator examples illustrate computation of set-valued inverse kinematic mappings and use of self-motion manifold mappings in obstacle avoidance applications. Differentiation of configuration level inverse mappings yields inverse velocity and acceleration mappings as functions of time dependent free parameters that play a central role in manipulator dynamics and control.

## 9.14 Characteristics of Redundant Manipulators

Most literature on kinematically redundant manipulators (Chiaverine, Oriolo, and Maciejewski, 2016; Luces, Mills, and Benhabib, 2017), except for early contributions (Burdick, 1989; Lee and Bejczy, 1991; Luck and Lee, 1995; DeMers and Kreutz-Delgato, 1996) and recent developments in 7-DOF serial manipulators (Shimizu et. al., 2008; Fari et. al., 2018; Wang et. al., 2021), deals with manipulator performance at the velocity level where equations are linear in velocities. This leads to a situation in which configurations must be accounted for using approximate numerical integration formulas that induce error and numerous other difficulties that are summarized in (Fari et. al., 2018). In contrast, the treatment presented herein obtains explicit representations for *set-valued inverse kinematic mappings* at the configuration level that can support obstacle avoidance and other manipulator performance requirements.

A substantial literature deals with seven degree of freedom (7-DOF) redundant serial manipulators that model the human arm and shoulder or that involve spray painting and manufacturing functions (Shimizu et. al., 2008; Fari et. al., 2018; Wang et. al., 2021). Following an early approach that introduced concepts of manifold parameterization (Lee and Bejczy, 1991), inverse kinematic mappings for 7-DOF manipulators are presented using one of the joint coordinates in the redundant serial manipulator as an independent parameter. In this approach, all other joint coordinates are derived as ad-hoc problem dependent expressions, often quite complex, in terms of the independent coordinate. While useful results are obtained for specific applications, this explicit parameterization approach is not valid for redundant manipulators whose configuration manifold cannot be globally parameterized by a single variable. The differentiable manifold approach presented herein does not rely on existence of a globally valid parameterization or on ad-hoc representation of dependent joint variables as functions of independent joint variables. Instead, it yields a broadly applicable computational formulation that can be implemented in real-time on modern high speed microprocessors, using only basic results of differential geometry on n-dimensional Euclidean space  $R^n$  presented in Sections 9.10.7 and 9.12.5.

In order to treat set-valued inverse kinematic mappings, except in the case of *redundant serial manipulators*, the manipulator kinematics formulation must account for mechanism generalized coordinates and associated kinematic constraints. This reality reduces the four categories of nonredundant manipulator treated in Part I to just *redundant serial manipulators* and *redundant compound manipulators*, the latter encompassing a broad spectrum of applications that include high load capacity construction, earth moving, and material handling manipulators. As defined herein, compound manipulators include classically defined *parallel manipulators* that are comprised of a single moving platform that is supported by parallel serial chains of actuated joints (Gosselin and Schreiber, 2018), as well as diverse other manipulator geometries. In short, redundant compound manipulators account for all redundant manipulators that are not serial. As in the case of nonredundant manipulators, regular configuration spaces for both serial and compound kinematically redundant manipulators are differentiable manifolds. Furthermore, ordinary differential equations (ODE) of dynamics in terms of input coordinates presented in Section 9.11 remain valid for kinematically redundant manipulators. However, *ODE in output coordinates for kinematically redundant manipulators are not possible*.

While set valued inverse kinematic mappings at the configuration level have recently been derived for specialized 7 DOF serial manipulators (Fari et. al., 2018; Wang et. al., 2021), no

general inverse configuration mapping approach has been presented for kinematically redundant manipulators. A differential geometric construct is employed herein that yields explicit characterizations of inverse kinematic mappings as functions of vectors of free parameters on charts of associated differentiable manifolds for serial and compound kinematically redundant manipulators in Sections 9.15 and 9.16, respectively. Analytical complexities associated with kinematically redundant manipulators include the necessity to characterize inverse kinematic mappings locally on *charts* that cover the *manipulator configuration manifold* (Guillemin and Pollack, 1974), since global parameterizations of redundant manipulator kinematics are generally not possible. Numerical methods are presented in Appendix A for evaluation of functions involved in analysis on charts, suitable for use in kinematics, dynamics, and control. Inverse kinematic velocity and acceleration mappings that are required for manipulator dynamics and control are presented for both serial and compound manipulators.

To make issues involved with kinematically redundant manipulators more precise, it is helpful to consider two elementary examples.

### 9.14.1 A Redundant Serial Manipulator

The *redundant serial manipulator* shown in Fig. 9.14.1 has one translational joint and two rotational joints. The mapping from input coordinates  $\mathbf{y} \in \mathbb{R}^3$  to output coordinates  $\mathbf{z} \in \mathbb{R}^2$  is

$$\mathbf{z} = \mathbf{G}(\mathbf{y}) = \begin{bmatrix} y_1 + \cos y_2 + 2 \cos(y_2 + y_3) \\ \sin y_2 + 2 \sin(y_2 + y_3) \end{bmatrix} \quad (9.14.1)$$

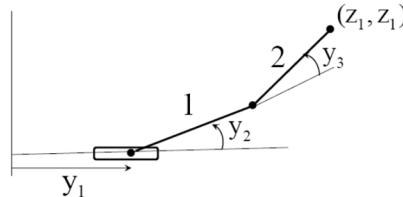


Figure 9.14.1 Redundant Serial Manipulator with One Translational and Two Rotational Joints

Since  $\mathbf{G}(\mathbf{y})$  is nonlinear in  $\mathbf{y}$ , it is not obvious how to obtain an inverse mapping. To gain insight, the *Jacobian* of  $\mathbf{G}(\mathbf{y})$  is

$$\mathbf{G}'(\mathbf{y}) \equiv \left[ \frac{\partial \mathbf{G}_i(\mathbf{y})}{\partial y_j} \right] = \begin{bmatrix} 1 & -\sin y_2 - 2 \sin(y_2 + y_3) & -2 \sin(y_2 + y_3) \\ 0 & \cos y_2 + 2 \cos(y_2 + y_3) & 2 \cos(y_2 + y_3) \end{bmatrix} \quad (9.14.2)$$

This Jacobian matrix has full row rank if there is a  $2 \times 2$  nonsingular submatrix, hence a *locally defined one parameter inverse kinematic mapping*. If the second row is zero, which occurs if and only if  $\cos y_2 = 0$  and  $\cos(y_2 + y_3) = 0$ , a one parameter inverse mapping fails to exist; i.e., there is no one parameter inverse mapping in a neighborhood of any  $\mathbf{y}$  for which  $\cos y_2 = 0$  and  $\cos(y_2 + y_3) = 0$ . Other than at this family of singularities, the implicit function theorem of Section 2.2.5 implies there is a locally valid one parameter inverse kinematic mapping.

With the *serial manipulator configuration* defined as  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^5$ , the *serial manipulator configuration space* is  $X^s = \{ \mathbf{x} \in \mathbb{R}^5 : \mathbf{G}(\mathbf{y}) - \mathbf{z} = \mathbf{0} \}$  and the *serial manipulator*

*regular configuration space*, in which the manipulator has locally a one parameter inverse kinematic mapping, is

$$\begin{aligned}
\tilde{X}^s &= \left\{ \mathbf{x} \in X^s : \cos y_2 \neq 0 \text{ or } \cos(y_2 + y_3) \neq 0 \right\} \\
&= \left\{ \mathbf{x} \in X^s : \cos y_2 \neq 0 \right\} \cup \left\{ \mathbf{x} \in X^s : \cos(y_2 + y_3) \neq 0 \right\} \\
&= \left\{ \mathbf{x} \in X^s : \cos y_2 > 0 \right\} \cup \left\{ \mathbf{x} \in X^s : \cos y_2 < 0 \right\} \\
&\quad \cup \left\{ \mathbf{x} \in X^s : \cos(y_2 + y_3) > 0 \right\} \cup \left\{ \mathbf{x} \in X^s : \cos(y_2 + y_3) < 0 \right\} \\
&\equiv \tilde{X}_1^s \cup \tilde{X}_2^s \cup \tilde{X}_3^s \cup \tilde{X}_4^s
\end{aligned} \tag{9.14.3}$$

Each  $\tilde{X}_i^s$  is path connected. They are not disjoint, however, as seen by configurations  $\mathbf{x}^1 = [y_1 \ 0 \ 0 \ y_1+3 \ 0]^T \in \tilde{X}_1^s \cap \tilde{X}_3^s$ ,  $\mathbf{x}^2 = [y_1 \ \pi \ 0 \ y_1-3 \ 0]^T \in \tilde{X}_2^s \cap \tilde{X}_4^s$ , and  $\mathbf{x}^3 = [y_1 \ \pi \ -\pi \ y_1+1 \ 0]^T \in \tilde{X}_2^s \cap \tilde{X}_3^s$ . Even though  $\tilde{X}_1^s$  and  $\tilde{X}_2^s$  are disjoint, they are connected by a path in  $\tilde{X}_3^s$  from  $\mathbf{x}^1 \in \tilde{X}_1^s \cap \tilde{X}_3^s$  to  $\mathbf{x}^3 \in \tilde{X}_2^s \cap \tilde{X}_3^s$ . Similarly, even though  $\tilde{X}_3^s$  and  $\tilde{X}_4^s$  are disjoint, they are connected by a path in  $\tilde{X}_2^s$  from  $\mathbf{x}^3 \in \tilde{X}_2^s \cap \tilde{X}_3^s$  to  $\mathbf{x}^2 \in \tilde{X}_2^s \cap \tilde{X}_4^s$ . The union of the four sets is thus path connected. Hence,  $\tilde{X}^s$  is comprised of a single *path connected, singularity free component*. Every pair of configurations  $\mathbf{x}^i = [\mathbf{y}^{iT} \ \mathbf{z}^{iT}]^T$ ,  $i = 1, 2$ , in  $\tilde{X}^s$  can thus be connected by one or more singularity free continuous trajectories  $\mathbf{x}(t) \in \tilde{X}^s$  such that  $\mathbf{x}(0) = \mathbf{x}^1$  and  $\mathbf{x}(1) = \mathbf{x}^2$ .

It is instructive to compare connectivity of the regular configuration space of the foregoing redundant serial manipulator to that of the nonredundant serial manipulator shown in Fig. 9.14.2 that is obtained by eliminating the second bar in the manipulator of Fig. 9.14.1. For this manipulator,  $\mathbf{y} = [y_1 \ y_2]^T \in \mathbb{R}^2$  and the forward kinematic map is

$$\mathbf{z} = \begin{bmatrix} y_1 + \cos y_2 \\ \sin y_2 \end{bmatrix} = \bar{\mathbf{G}}(\mathbf{y}) \tag{9.14.4}$$

with Jacobian  $\bar{\mathbf{G}}'(\mathbf{y}) = \begin{bmatrix} 1 & -\sin y_2 \\ 0 & \cos y_2 \end{bmatrix}$  and associated determinant  $|\bar{\mathbf{G}}'(\mathbf{y})| = \cos y_2$ .

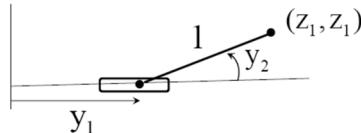


Figure 9.14.2 Nonredundant Serial Manipulator

This manipulator has a locally single valued continuously differentiable inverse kinematic mapping  $\mathbf{y} = \bar{\mathbf{G}}^{-1}(\mathbf{z})$  in  $X = \left\{ \mathbf{x} = [\mathbf{y}^T \ \mathbf{z}^T]^T \in \mathbb{R}^4 : \bar{\mathbf{G}}(\mathbf{y}) - \mathbf{z} = \mathbf{0} \right\}$ , provided  $\cos y_2 \neq 0$ . The *regular configuration space* is  $\tilde{X} = \left\{ \mathbf{x} \in X : \cos y_2 > 0 \right\} \cup \left\{ \mathbf{x} \in X : \cos y_2 < 0 \right\} \equiv \tilde{X}^1 \cup \tilde{X}^2$  where both  $\tilde{X}^1$  and  $\tilde{X}^2$  are path connected, singularity free sets. In the case of this nonredundant

manipulator, however,  $\tilde{X}^1$  and  $\tilde{X}^2$  are disjoint and it is impossible to find a singularity free continuous trajectory between configurations in  $\tilde{X}^1$  and  $\tilde{X}^2$ .

The distinction between the serial manipulators of Figs. 9.14.1 and 9.14.2 is profound. Adding a third link to the nonredundant manipulator of Fig. 9.14.2 yields the redundant manipulator of Fig. 9.14.1. Whereas the nonredundant manipulator has disjoint components whose configurations cannot be connected by continuous nonsingular trajectories, every pair of configurations in the regular configuration space of the redundant manipulator can be connected by one or more continuous nonsingular trajectories. This illustrates one of the positive aspects of redundant manipulators.

### 9.14.2 A Redundant Compound Manipulator

Two bodies shown in Fig. 9.14.3 move in a plane without rotation, connected by a unit length distance constraint. They are located in the plane by generalized coordinates,  $\mathbf{q} \in \mathbb{R}^3$ .

Input coordinates  $\mathbf{y} \in \mathbb{R}^2$  are intended to control the position of body 1 and the vertical coordinate of body 2, which is the output coordinate  $z = q_3 \in \mathbb{R}^1$ . The unit distance constraint between bodies 1 and 2 is represented as the *holonomic constraint equation*

$$\Phi(\mathbf{q}) = (q_1^2 + (q_3 - q_2)^2 - 1)/2 = 0 \quad (9.14.5)$$

Inputs are related to generalized coordinates by the *input equation*

$$\Psi(\mathbf{y}, \mathbf{q}) = [y_1 - q_1 \quad y_2 - q_2]^T = \mathbf{0} \quad (9.14.6)$$

and outputs are related to generalized coordinates by the *output equation*

$$\Gamma(\mathbf{q}, \mathbf{z}) = z - q_3 = 0 \quad (9.14.7)$$

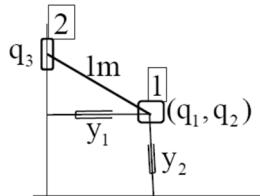


Figure 9.14.3 Compound Manipulator with Two Bodies and One Distance Constraint

The Jacobian of the combined constraints of Eqs. (9.14.5) and (9.14.6) with respect to  $\mathbf{q}$  is

$$\Omega_q(\mathbf{y}, \mathbf{q}) = \begin{bmatrix} \Phi(\mathbf{q}) \\ \Psi(\mathbf{y}, \mathbf{q}) \end{bmatrix}_q = \begin{bmatrix} \Phi_q(\mathbf{q}) \\ \Psi_q(\mathbf{y}, \mathbf{q}) \end{bmatrix} = \begin{bmatrix} q_1 & -(q_3 - q_2) & (q_3 - q_2) \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad (9.14.8)$$

with determinant  $|\Omega_q(\mathbf{y}, \mathbf{q})| = q_3 - q_2$ . Thus, in a neighborhood of any configuration with  $q_3 - q_2 = 0$ , there is no continuous forward kinematic mapping from  $\mathbf{y}$  to  $\mathbf{q}$ , hence none from  $\mathbf{y}$  to  $\mathbf{z}$ . The *compound manipulator configuration space*, comprised of configurations

$\mathbf{x} = [y^T \quad q^T \quad z]^T \in R^6$ , is  $X^C \equiv \{x \in R^6 : \Psi(y, q) = \mathbf{0}, \Phi(q) = 0, \text{ and } \Gamma(q, z) = \mathbf{0}\}$ . The compound regular manipulator configuration space is  $\tilde{X}^c \equiv \{x \in X^c : q_3 - q_2 \neq 0\}$ , which is partitioned into disjoint components  $\tilde{X}^+ \equiv \{x \in R^6 : \Psi(y, q) = \mathbf{0}, \Phi(q) = 0, \Gamma(q, z) = \mathbf{0}, \text{ and } (q_3 - q_2) > 0\}$  and  $\tilde{X}^- \equiv \{x \in R^6 : \Psi(y, q) = \mathbf{0}, \Phi(q) = 0, \Gamma(q, z) = \mathbf{0}, \text{ and } (q_3 - q_2) < 0\}$ . From Eq. (9.14.5),  $q_3 - q_2 = \pm\sqrt{1 - q_1^2} \neq 0$ . With  $q_1 \equiv v$ ,  $-1 < v < 1$ , the relation  $q_3 - q_2 = \pm\sqrt{1 - v^2}$  and Eqs. (9.14.6) and (9.14.7) yield one parameter inverse kinematic mappings in each of the components,  $y^\pm = [v \quad z - (\pm\sqrt{1 - v^2})]^T$  and  $q^\pm = [v \quad z - (\pm\sqrt{1 - v^2}) \quad z]^T$ . These mappings provide information that enable planning a trajectory  $x(t) = [y^T(t) \quad q^T(t) \quad \bar{z}(t)]^T$  for a specified  $\bar{z}(t)$  to avoid obstacles.

The foregoing situation, in which the regular manipulator configuration space is partitioned into disjoint components throughout which continuous forward kinematic mappings and set-valued inverse kinematic mappings exist, is typical of redundant compound manipulators. It leads to more intricate conditions for regular behavior of redundant compound manipulators than arise in the case of redundant serial manipulators.

As a simple illustration of *obstacle avoidance* using the free parameter  $v$  and the *set-valued inverse kinematic mapping*  $q^\pm = [v \quad z - (\pm\sqrt{1 - v^2}) \quad z]^T$ , a nominal configuration trajectory is defined in  $X^+$  by  $v = 0$ , with output  $\bar{z}(t) = q_3(t) = t$ ,  $0 \leq t \leq 1$ . Using the inverse mappings  $y^\pm$  and  $q^\pm$ , one may select  $y(t) = [0 \quad t-1]^T$  and  $q(t) = [0 \quad t-1 \quad t]^T$ . If an obstacle defined by the inequality  $q_2 \geq -1/2$  is imposed, the foregoing generalized coordinate trajectory  $q_2(t) = t-1$  penetrates the obstacle for  $0 \leq t < 1/2$ . Setting  $q_2(t) = t - (\sqrt{1 - v^2}) = -1/2$ , for  $0 \leq t < 1/2$ , to avoid penetrating the obstacle,  $v = \sqrt{1 - (t + 1/2)^2}$ . The value of  $v = 0$  is retained for  $1/2 \leq t \leq 1$ . The value of  $\bar{z}(t)$  over the entire period  $0 \leq t \leq 1$  is not changed by this so-called *self-motion transformation*, but new input and generalized coordinates are obtained from Eqs. (9.14.5) and (9.14.6),

$$\begin{aligned} \ddot{y}(t) &= \begin{bmatrix} \sqrt{1 - (t + 1/2)^2} \\ -1/2 \end{bmatrix}, 0 \leq t < 1/2; \quad \ddot{y}(t) = \begin{bmatrix} 0 \\ t-1 \end{bmatrix}, 1/2 \leq t \leq 1 \\ \ddot{q}(t) &= \begin{bmatrix} \sqrt{1 - (t + 1/2)^2} \\ -1/2 \\ t \end{bmatrix}, 0 \leq t < 1/2; \quad \ddot{q}(t) = \begin{bmatrix} 0 \\ t-1 \\ t \end{bmatrix}, 1/2 \leq t \leq 1 \end{aligned}$$

which avoid penetration of the obstacle. This selection of the free parameter in the inverse kinematic mapping yields inputs and generalized coordinates that retain the desired output trajectory  $\bar{z}(t) = t$ , while avoiding penetration of the obstacle. It is shown in Section 9.16 that

such explicit analytical computation is generally not possible in realistic applications, but algorithms based on differentiable manifold theory are presented to avoid obstacles.

## 9.15 Kinematically Redundant Serial Manipulators

A *serial manipulator* is defined as a *chain of bodies* connected by *single degree of freedom joints* in which *input coordinates define transformations from inboard bodies to adjacent outboard bodies* and determine the configurations of all bodies in the chain. The *vector of input coordinates* in joints of the manipulator is denoted  $\mathbf{y} \in \mathbb{R}^n$ . The last body in the chain is called the *end-effector*. Some or all configuration coordinates of the end-effector comprise the *vector of output coordinates*  $\mathbf{z} \in \mathbb{R}^m$ . If  $n > m$ , the manipulator is said to be a *kinematically redundant serial manipulator* and the *forward kinematic mapping* is the *input-output equation*

$$\mathbf{z} = \mathbf{G}(\mathbf{y}) \quad (9.15.1)$$

where  $\mathbf{G} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is  $k$ -times continuously differentiable,  $k \geq 2$ . The first task in redundant serial manipulator kinematics is to consider the *set-valued inverse mapping*  $\mathbf{G}^{-1} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ ,

$$\mathbf{G}^{-1}(\mathbf{z}) = \left\{ \mathbf{y} \in \mathbb{R}^n : \mathbf{G}(\mathbf{y}) = \mathbf{z} \right\} \quad (9.15.2)$$

### 9.15.1 Serial Manipulator Configuration Space

To characterize the inverse mapping  $\mathbf{G}^{-1}$ , it is first necessary to define the serial manipulator configuration space. With  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^{n+m}$  defined as *manipulator configuration coordinates*, the *serial manipulator configuration space* is the set  $X^s = \left\{ \mathbf{x} \in \mathbb{R}^{n+m} ; \mathbf{z} = \mathbf{G}(\mathbf{y}) \right\}$ . Provided the Jacobian  $\mathbf{G}'(\mathbf{y}) \equiv \left[ \frac{\partial \mathbf{G}_i(\mathbf{y})}{\partial \mathbf{y}_j} \right]_{m \times n}$  has full row rank at  $\bar{\mathbf{y}}$ , there is an  $m \times m$  *nonsingular submatrix* of  $\mathbf{G}'(\bar{\mathbf{y}})$ . Let  $\check{\mathbf{y}} \in \mathbb{R}^m$  be comprised of input coordinates corresponding to columns of this submatrix and  $\check{\mathbf{y}} \in \mathbb{R}^{n-m}$  be comprised of the remaining input coordinates. Provided  $\bar{\mathbf{x}} \in X^s$ , the *implicit function theorem* assures there is a *unique differentiable solution* of  $\mathbf{G}(\check{\mathbf{y}}, \check{\mathbf{y}}) = \mathbf{z}$ ,  $\check{\mathbf{y}} = \mathbf{w}(\mathbf{z}, \check{\mathbf{y}})$ , for all  $(\mathbf{z}, \check{\mathbf{y}})$  in a neighborhood of  $(\bar{\mathbf{z}}, \bar{\check{\mathbf{y}}})$ ; i.e., an  $n \times m$  parameter family of local solutions of  $\mathbf{G}(\mathbf{y}) = \mathbf{z}$  for fixed  $\mathbf{z}$ , characterized by  $\check{\mathbf{y}}$ . There is no assurance, however, that such a solution is valid over all of  $X^s$ . If  $\text{rank}(\mathbf{G}'(\bar{\mathbf{y}})) < m$ , there is no such family of solutions throughout any neighborhood of  $\bar{\mathbf{y}}$ , in which case  $\bar{\mathbf{x}}$  is called a *singular configuration*.

### 9.15.2 Regular Serial Manipulator Configuration Space

To avoid singular configurations, the *regular serial manipulator configuration space* is defined as  $\tilde{X}^s = \left\{ \mathbf{x} \in X^s : \text{rank}(\mathbf{G}'(\mathbf{y})) = m \right\} = \left\{ \mathbf{x} \in X^s : \left| \mathbf{G}'(\mathbf{y}) \mathbf{G}'^T(\mathbf{y}) \right| > 0 \right\}$ . This is an open subset of  $X^s$  in the *relative topology*, so it is comprised of a finite number of maximal, disjoint, path connected, *singularity free components*  $\tilde{X}_i^s$ , such that  $\tilde{X}_i^s \cap \tilde{X}_j^s = \emptyset$  if  $i \neq j$  and  $\cup_i \tilde{X}_i^s = \tilde{X}^s$ , where  $\emptyset$  is the empty set (Mendelson, 1962). With *local parameterizations*  $\check{\mathbf{y}} = \mathbf{w}(\mathbf{z}, \check{\mathbf{y}})$ ,  $\tilde{X}^s$  is a *differentiable manifold* and the  $\tilde{X}_i^s$  are *submanifolds*.

### 9.15.3 Serial Manipulator Inverse Kinematic Configuration Mapping

In early literature focused on characterizing the set-valued inverse kinematic mapping of Eq. (9.15.2) (Burdick, 1989; Luck and Lee, 1995; DeMers and Kreutz-Delgado, 1996), concepts of differential geometry were used to address the problem at the configuration level. Subsequently, an extensive literature has focused almost exclusively on redundant manipulator analysis at the velocity level, where equations are linear in velocities. While useful results have been obtained with velocity analysis, information at the configuration level is lost (Fari et. al., 2018). The purpose of this section is to employ differential geometry to analytically and computationally characterize set-valued inverse mappings of Eq. (9.15.2) at the configuration level.

As shown in Section 9.15.2, for  $\bar{\mathbf{x}} \in \tilde{X}^s$ , there exists an  $n - m$  parameter inverse configuration mapping in a neighborhood of  $\bar{\mathbf{x}}$ . To characterize such a mapping, define

$$\mathbf{U} = \mathbf{G}'^T(\bar{\mathbf{y}}) \quad (9.15.3)$$

and use *singular value decomposition* to evaluate a solution of

$$\mathbf{G}'(\bar{\mathbf{y}})\mathbf{V} = \mathbf{0} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I} \quad (9.15.4)$$

where the column rank of  $\mathbf{U}$  is  $m$  and that of  $\mathbf{V}$  is  $n - m$  (Atkinson, 1989). Thus, the columns of  $\mathbf{U}$  and  $\mathbf{V}$  span  $\mathbb{R}^n$  and any  $\mathbf{y} \in \mathbf{G}^{-1}(\mathbf{z}) \subset \mathbb{R}^n$  may be represented in the form

$$\mathbf{y} = \bar{\mathbf{y}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u} \quad (9.15.5)$$

Note that at  $\mathbf{y} = \bar{\mathbf{y}}$ ,  $\bar{\mathbf{v}} = \mathbf{0}$  and  $\bar{\mathbf{u}} = \mathbf{0}$ . The condition that  $\mathbf{y}$  of Eq. (9.15.5) is in  $\mathbf{G}^{-1}(\mathbf{z})$  is

$$\mathbf{G}(\bar{\mathbf{y}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}) - \mathbf{z} = \mathbf{0} \quad (9.15.6)$$

The Jacobian of the left side of Eq. (9.15.6) with respect to  $\mathbf{u}$ , at  $\bar{\mathbf{x}}$ ; i.e.,  $\mathbf{v} = \mathbf{0}$ ,  $\mathbf{u} = \mathbf{0}$ , and  $\mathbf{z} = \bar{\mathbf{z}}$ , is  $-\mathbf{G}'(\bar{\mathbf{y}})\mathbf{U} = -\mathbf{U}^T\mathbf{U}$ , which is nonsingular. Thus, the implicit function theorem implies Eq. (9.15.6) has a unique continuously differentiable solution  $\mathbf{u} = \mathbf{h}(\mathbf{v}, \mathbf{z})$  for  $\mathbf{u}$  as a function of  $\mathbf{v}$  and  $\mathbf{z}$  in a neighborhood of  $(\mathbf{v}, \mathbf{z}) = (\mathbf{0}, \bar{\mathbf{z}})$ ; i.e.,

$$\mathbf{y}(\mathbf{v}, \mathbf{z}) = \bar{\mathbf{y}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z}) \quad (9.15.7)$$

### 9.15.4 The Regular Serial Manipulator Configuration Differentiable Manifold

Equation (9.15.7) provides a *continuously differentiable parameterization* of  $\tilde{X}^s$  on a neighborhood  $N$  of  $(\mathbf{v}, \mathbf{z}) = (\mathbf{0}, \bar{\mathbf{z}})$ . This neighborhood and the mapping of Eq. (9.15.7) that are assured by the implicit function theorem define a *chart* on  $\tilde{X}^s$ . Creating a family of such charts that cover  $\tilde{X}^s$  provides an *atlas* that defines  $\tilde{X}^s$  as a *differentiable manifold* with disjoint, maximal, path connected, singularity free *components*  $\tilde{X}_i^s$  (Guillemin and Pollack, 1974).

Analytical relations that characterize differentiable manifolds are defined locally on open sets  $N^i$  as parameterizations  $\mathbf{y}(\mathbf{v}, \mathbf{z}) = \psi^i(\mathbf{v}, \mathbf{z}) \equiv \bar{\mathbf{y}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z})$  of Eq. (9.15.7). The set  $N^i$  and mapping  $\psi^i(\mathbf{v}, \mathbf{z})$  is called a *chart*, denoted  $(N^i, \psi^i(\mathbf{v}, \mathbf{z}))$ . A family of charts, called an *atlas*, is defined to cover  $\tilde{X}^s$ ; i.e.,  $\cup_i N^i = \tilde{X}^s$ , such that the mappings  $\psi^i$  are compatible (Guillemin and Pollack, 1974). Kinematic analysis on a component  $\tilde{X}_i^s$  of  $\tilde{X}^s$  must therefore be carried out on

individual charts and transitioned to adjacent charts as manipulator configurations progress along a trajectory in  $\tilde{X}_i^s$ , as shown schematically in Fig. 9.15.1.

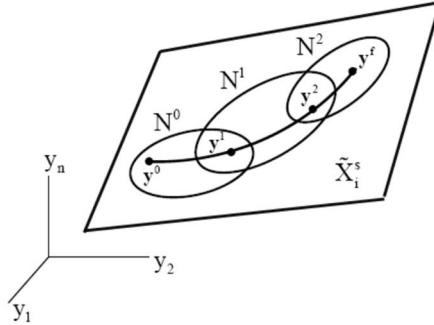


Figure 9.15.1 Trajectory Along Charts in  $\tilde{X}_i^s$

This *piecewise smooth form of analysis* will be illustrated in mapping components of serial and compound manipulator manifolds. Piecewise analysis shown in Fig. 9.15.1 is unavoidable, since in general *there is no globally valid parameterization*  $\Psi^i(\mathbf{v}, \mathbf{z})$  on a component  $\tilde{X}_i^s$  of regular configuration space. As elegantly explained by Hirsch (1976), this attribute of *differentiable manifold theory* that transforms local to global properties of sets and mappings is perhaps its greatest contribution. The unavoidable reality is that one must adopt *local manifold parameterizations*, since no global analytical representation generally exists.

For use in *kinematic analysis* and *trajectory planning* in a component  $\tilde{X}_i^s$ , partial derivatives of  $\mathbf{h}(\mathbf{v}, \mathbf{z})$  and  $\mathbf{y}(\mathbf{v}, \mathbf{z})$  with respect to  $\mathbf{v}$  and  $\mathbf{z}$  are needed. With  $\mathbf{u} = \mathbf{h}(\mathbf{v}, \mathbf{z})$ , Eq. (9.15.6) is the identity  $\mathbf{G}(\bar{\mathbf{y}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z})) - \mathbf{z} = \mathbf{0}$  in variables  $\mathbf{v}$  and  $\mathbf{z}$ . Taking partial derivatives of this identity with respect to  $\mathbf{v}$  and  $\mathbf{z}$ , evaluated at  $\mathbf{y} = \bar{\mathbf{y}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z})$ , yields

$$\begin{aligned}\mathbf{G}'(\mathbf{y})(\mathbf{V} - \mathbf{U}\mathbf{h}_v) &= \mathbf{0} \\ \mathbf{G}'(\mathbf{y})(-\mathbf{U}\mathbf{h}_z) - \mathbf{I} &= \mathbf{0}\end{aligned}\tag{9.15.8}$$

At  $\mathbf{y} = \bar{\mathbf{y}}$ , from Eq. (9.15.3),  $\mathbf{G}'(\bar{\mathbf{y}})\mathbf{U} = \mathbf{U}^T\mathbf{U}$  is nonsingular, so define  $\mathbf{B}(\bar{\mathbf{y}}) = (\mathbf{U}^T\mathbf{U})^{-1}$ . Since  $\mathbf{G}'(\mathbf{y})$  is a continuous function of  $\mathbf{y}$ ,

$$\mathbf{B}(\mathbf{y}) \equiv (\mathbf{G}'(\mathbf{y})\mathbf{U})^{-1}\tag{9.15.9}$$

is well defined and continuously differentiable in a neighborhood of  $\mathbf{y} = \bar{\mathbf{y}}$ . Thus, Eqs. (9.15.8) yield

$$\begin{aligned}\mathbf{h}_v(\mathbf{v}, \mathbf{z}) &= \mathbf{B}(\mathbf{y})\mathbf{G}'(\mathbf{y})\mathbf{V} \\ \mathbf{h}_z(\mathbf{v}, \mathbf{z}) &= -\mathbf{B}(\mathbf{y})\end{aligned}\tag{9.15.10}$$

Applying these results to  $\mathbf{y}(\mathbf{v}, \mathbf{z})$  of Eq. (9.15.7),

$$\begin{aligned}\mathbf{y}_v(\mathbf{v}, \mathbf{z}) &= \mathbf{V} - \mathbf{U}\mathbf{B}(\mathbf{y}(\mathbf{v}, \mathbf{z}))\mathbf{G}'(\mathbf{y}(\mathbf{v}, \mathbf{z}))\mathbf{V} \\ \mathbf{y}_z(\mathbf{v}, \mathbf{z}) &= \mathbf{U}\mathbf{B}(\mathbf{y}(\mathbf{v}, \mathbf{z}))\end{aligned}\tag{9.15.11}$$

The derivatives of Eqs. (9.15.10) and (9.15.11) enable redundant serial manipulator applications such as *obstacle avoidance*, *singularity avoidance*, and *dynamic performance optimization*.

### 9.15.5 The Serial Manipulator Self-Motion Manifold

For fixed  $\mathbf{z} = \bar{\mathbf{z}}$  with  $\bar{\mathbf{x}} \in \tilde{X}^s$ , the mapping of Eq. (9.15.7) reduces to

$$\mathbf{y}(\mathbf{v}, \bar{\mathbf{z}}) = \bar{\mathbf{y}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \bar{\mathbf{z}}) \quad (9.15.12)$$

for all  $\mathbf{v}$  in a neighborhood of  $\mathbf{0}$ . For a sequence  $\bar{\mathbf{y}}_j \in \mathbb{R}^n$ , this yields

$N^j(\bar{\mathbf{z}}) = \left\{ \mathbf{y} = \bar{\mathbf{y}}_j + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \bar{\mathbf{z}}), \text{ for all } \mathbf{v} \text{ in a neighborhood of } \mathbf{0} \right\}$ , whose union  $\cup_j N^j(\bar{\mathbf{z}})$  is a *differentiable manifold*  $\mathbf{Y}^s(\bar{\mathbf{z}})$  in the input space. This manifold is called the *self-motion manifold*, since it is a set of inputs, all of which have forward kinematic mappings onto the same output  $\bar{\mathbf{z}}$ . The *inverse kinematic mapping* of Eq. (9.15.12) is a *parameterization of the self-motion manifold*  $\mathbf{Y}^s(\bar{\mathbf{z}})$ . Thus,  $\mathbf{Y}^s(\bar{\mathbf{z}})$  is a set of inputs that all yield output  $\bar{\mathbf{z}}$ , some of which may have attractive properties such as avoiding obstacles and singularities. As with all differentiable manifolds,  $\mathbf{Y}^s(\bar{\mathbf{z}})$  may be partitioned into maximal disjoint components  $\mathbf{Y}_i^s(\bar{\mathbf{z}})$ ; i.e., it may not be a connected set.

If  $\bar{\mathbf{z}}_i$  and  $\bar{\mathbf{z}}_j$  are adjacent outputs in a neighborhood of  $\bar{\mathbf{z}}$ , one may select vastly different  $\mathbf{y}_i \in \mathbf{Y}_i^s(\bar{\mathbf{z}}_i)$  and  $\mathbf{y}_j \in \mathbf{Y}_j^s(\bar{\mathbf{z}}_j)$ , so the resulting  $\bar{\mathbf{x}}_i$  and  $\bar{\mathbf{x}}_j$  are distinctly different in  $\tilde{X}^s$ . If one wishes to maintain manipulator configuration continuity in the manifold  $\tilde{X}^s$ , uncorrelated large excursions in  $\mathbf{Y}^s(\bar{\mathbf{z}})$ , with  $\mathbf{z} = \bar{\mathbf{z}}$  fixed, must be avoided. In short, the *manipulator regular configuration manifold*  $\tilde{X}^s$  and the *self-motion manifold*  $\mathbf{Y}^s(\bar{\mathbf{z}})$  are fundamentally different.

### 9.15.6 Mapping One-Dimensional Serial Manipulator Self-Motion Manifold Components

To map a *self-motion manifold component*  $\mathbf{Y}_i^s(\bar{\mathbf{z}})$  in case  $n - m = 1$ , beginning at a configuration  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}_i^s$ , that satisfies Eq. (9.15.1). The output  $\bar{\mathbf{z}}$  is held fixed during the process of mapping  $\mathbf{Y}^s(\bar{\mathbf{z}})$ . At  $\mathbf{y} = \bar{\mathbf{y}}$ ,  $\mathbf{v}^0 = \mathbf{0}$ , Eqs. (9.15.3) and (9.15.4) are used to evaluate  $\mathbf{U}$  and  $\mathbf{V}$ , and  $\mathbf{B}^0 = \mathbf{B}(\bar{\mathbf{y}}) = (\mathbf{U}^T \mathbf{U})^{-1}$  is evaluated. The determinant  $|\mathbf{G}'(\mathbf{y})\mathbf{G}'(\mathbf{y})^T|$  must be positive throughout  $\mathbf{Y}_i^s(\bar{\mathbf{z}})$ , to avoid crossing a discontinuity and entering a different component  $\mathbf{Y}_j^s(\bar{\mathbf{z}})$ . For a step  $h > 0$  in  $\mathbf{v}$ ,  $\mathbf{v}^1 = h$  and  $\mathbf{h}(\mathbf{v}^1, \bar{\mathbf{z}})$  is obtained using the iterative process of Eq. (9.A.2), with  $\Lambda$  replaced by  $\mathbf{G} - \mathbf{z}$ , and  $\mathbf{B} = \mathbf{B}^0$  for  $\mathbf{u}^1 = \mathbf{h}(\mathbf{v}^1, \bar{\mathbf{z}})$ . Input  $\mathbf{y}^1$  is  $\mathbf{y}^1 = \bar{\mathbf{y}} + \mathbf{V}\mathbf{v}^1 - \mathbf{U}\mathbf{h}(\mathbf{v}^1, \bar{\mathbf{z}})$  from Eq. (9.15.7). The iterative process of Eq. (9.A.1), with  $\Lambda$  replaced by  $\mathbf{G} - \mathbf{z}$ , is then used to evaluate  $\mathbf{B}^1$ . At  $\mathbf{v}^k = kh$ ,  $k = 2, 3, \dots$ ,  $\mathbf{h}(\mathbf{v}^k, \bar{\mathbf{z}})$  is evaluated as  $\mathbf{u}^k = \mathbf{h}(\mathbf{v}^k, \bar{\mathbf{z}})$  from Eq. (9.A.2), with  $\Lambda$  replaced by  $\mathbf{G} - \mathbf{z}$ , and  $\mathbf{B} = \mathbf{B}^{k-1}$ ,  $\mathbf{y}^k = \bar{\mathbf{y}} + \mathbf{V}\mathbf{v}^k - \mathbf{U}\mathbf{h}(\mathbf{v}^k, \bar{\mathbf{z}})$  of Eq. (9.15.7), and  $\mathbf{B}^k$  is evaluated using the iterative process of Eq. (9.A.1), with  $\Lambda$  replaced by  $\mathbf{G} - \mathbf{z}$ . The determinant  $|\mathbf{G}'(\mathbf{y}^k)\mathbf{G}'(\mathbf{y}^k)^T|$  is confirmed to be positive to have  $\mathbf{y}^k \in \mathbf{Y}_i^s(\bar{\mathbf{z}})$  and the sign of  $|\mathbf{G}'(\mathbf{y}^k)\mathbf{U}|$  must be invariant in  $\mathbf{Y}_i^s(\bar{\mathbf{z}})$ .

If the condition number (see Section 2.2.7) of  $\mathbf{G}'(\mathbf{y}^k)\mathbf{U}$  is within tolerance, continue the mapping process. Otherwise, redefine  $\bar{\mathbf{y}} = \mathbf{y}^k$ , retain  $\mathbf{z} = \bar{\mathbf{z}}$ , use Eqs. (9.15.3) and (9.15.4) to evaluate new  $\mathbf{U}$  and  $\mathbf{V}$ , reset  $v^0 = 0$ , evaluate  $\mathbf{B}^0 = (\mathbf{U}^T \mathbf{U})^{-1}$ , index  $v^k = kh$ , and continue the mapping process. This process implements the *chart-to-chart continuation process* outlined in Section 2.4 and shown schematically in Fig. 9.15.1.

As an illustration of a self-motion manifold with disjoint components, consider the two-bar mechanism shown in Fig. 9.15.2. The vertical coordinate of point P is the output coordinate, so

$$z = \sin y_1 + \sin(y_1 + y_2) \equiv G(\mathbf{y}) \quad (9.15.13)$$

with Jacobian

$$G'(\mathbf{y}) = \begin{bmatrix} \cos y_1 + \cos(y_1 + y_2) & \cos(y_1 + y_2) \end{bmatrix} \quad (9.15.14)$$

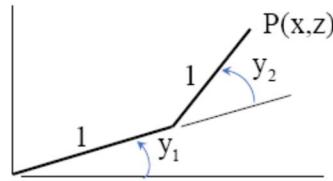


Figure 9.15.2 Two-Bar Manipulator

With  $\mathbf{x} = [y_1 \ y_2 \ z]^T$ , at configuration  $\bar{\mathbf{x}} = [0 \ 0 \ 0]^T$ ,  $G'(\bar{\mathbf{y}}) = [2 \ 1] \neq \mathbf{0}$ . For self-motion mapping, beginning at  $\bar{\mathbf{x}}$  and holding  $\bar{z} = 0$ ,  $\mathbf{U} = [2 \ 1]^T$  and  $\mathbf{V} = (1/\sqrt{5})[-1 \ 2]^T$ .

From Eq. (9.15.7),  $\mathbf{y} = [-v/\sqrt{5} - 2u \ 2v/\sqrt{5} - u]^T$ . Substituting this into Eq. (9.15.13),  $-\sin(v/\sqrt{5} + 2u) + \sin(v/\sqrt{5} - 3u) = 0$ , so (1)  $v/\sqrt{5} + 2u = v/\sqrt{5} - 3u$  and  $u = 0$  or (2), using the identity  $\sin \alpha = \sin(\pi - \alpha)$ ,  $v/\sqrt{5} + 2u = \pi - v/\sqrt{5} + 3u$  and  $u = 2v/\sqrt{5} - \pi$ . In case (1), the self-motion mapping is  $\mathbf{y} = (v/\sqrt{5})[-1 \ 2]^T$ , with  $G'(\mathbf{y}) = \cos(v/\sqrt{5})[2 \ 1]$ , which is zero if and only if  $v = \pm\sqrt{5}\pi/2$ . This self-motion mapping, in the range  $-\sqrt{5}\pi/2 < v < \sqrt{5}\pi/2$ , is the slanted line segment on the left of Fig. 6, which is a component of the self-motion manifold. In case (2), the self-motion mapping is  $\mathbf{y} = [-5v/\sqrt{5} + 2\pi \ \pi]^T$ , with  $G'(\mathbf{y}) = \cos(5v/\sqrt{5})[0 \ -1]$ , which is zero if and only if  $5v/\sqrt{5} = \pm\pi/2$ , or  $v = \pm\sqrt{5}\pi/10$ . The self-motion mapping in the range  $-\sqrt{5}\pi/10 < v < \sqrt{5}\pi/10$  has  $y_2 = \pi$  and  $-\pi/2 < y_1 < \pi/2$ . This is the upper horizontal line in Fig. 9.15.3, which is a component of the self-motion manifold.

At configuration  $\bar{\mathbf{x}} = [3\pi \ 0 \ 0]^T$ ,  $G'(\bar{\mathbf{y}}) = [-2 \ -1] \neq \mathbf{0}$ . For self-motion mapping, beginning at  $\bar{\mathbf{x}}$  and holding  $\bar{z} = 0$ ,  $\mathbf{U} = [-2 \ -1]^T$  and  $\mathbf{V} = (1/\sqrt{5})[1 \ -2]^T$ . From Eq. (9.15.7),  $\mathbf{y} = [3\pi + v/\sqrt{5} + 2u \ -2v/\sqrt{5} + u]^T$ . Substituting this into Eq. (9.15.13),  $\sin(3\pi + v/\sqrt{5} + 2u) + \sin(3\pi - v/\sqrt{5} + 3u) = 0$ . Using trigonometric identities, this reduces to

$\sin(v/\sqrt{5} + 2u) - \sin(v/\sqrt{5} - 3u) = 0$ , so (3)  $v/\sqrt{5} + 2u = v/\sqrt{5} - 3u$  and  $u = 0$  or (4), using the identity  $\sin \alpha = \sin(\pi - \alpha)$ ,  $v/\sqrt{5} + 2u = \pi - v/\sqrt{5} + 3u$  and  $u = 2v/\sqrt{5} - \pi$ . In case (3), the self-motion mapping is  $\mathbf{y} = [\pi + v/\sqrt{5} \quad -2v/\sqrt{5}]^T$ , with  $G'(\mathbf{y}) = \cos(v/\sqrt{5})[-2 \quad -1]$ , which is zero if and only if  $v = \pm\sqrt{5}\pi/2$ . This self-motion mapping, in the range  $-\sqrt{5}\pi/2 < v < \sqrt{5}\pi/2$ , is the slanted line segment on the right of Fig. 6, which is a component of the self-motion manifold. In case (4), the self-motion mapping is  $\mathbf{y} = [5v/\sqrt{5} + \pi \quad -\pi]^T$ , with  $G'(\mathbf{y}) = \cos(5v/\sqrt{5})[0 \quad 1]$ , which is zero if and only if  $5v/\sqrt{5} = \pm\pi/2$ , or  $v = \pm\sqrt{5}\pi/10$ . This self-motion mapping in the range  $-\sqrt{5}\pi/10 < v < \sqrt{5}\pi/10$  has  $y_2 = -\pi$  and  $\pi/2 < y_1 < 3\pi/2$ . This is the lower horizontal line in Fig. 6, which is a component of the self-motion manifold.

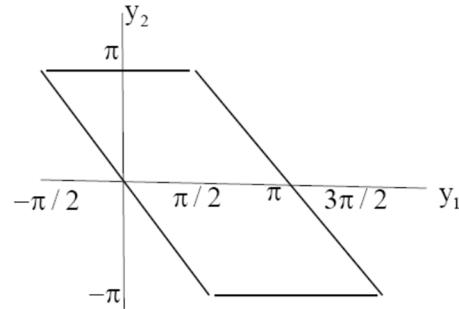


Figure 9.15.3 Disjoint Components of Self-Motion Manifold with  $\bar{z} = 0$

Self-motion mappings with  $\bar{z} = 0$  yield *self-motion manifolds with disjoint components*. With other values of  $\bar{z}$ , the self-motion manifold consists of a single component in the same range of  $y_1$ . For example, with  $\bar{x} = [0 \quad \pi/2 \quad 1]^T$ ,  $G'(\bar{y}) = [1 \quad 0] \neq \mathbf{0}$ ,  $\mathbf{U} = [1 \quad 0]^T$  and  $\mathbf{V} = [0 \quad 1]^T$ . From Eq. (9.15.7),  $\mathbf{y} = [-u \quad \pi/2 + v]^T$ . Substituting this into Eq. (9.15.13) with  $\bar{z} = 1$ ,  $\sin(-u) + \sin(-u + \pi/2 + v) = 1$ , or  $f(u, v) = -\sin(u) + \sin(-u + \pi/2 + v) - 1 = 0$ . Since  $u = 0$  is a solution of this equation and, at these values,  $f_u(0, 0) = -\cos 0 - \cos(\pi/2) = 1$ ,  $f(u, v) = 0$  can be numerically solved for  $u$  as a function of  $v$  on a grid of values of  $v$ . The result is substituted into  $\mathbf{y} = [-u \quad \pi/2 + v]^T$  and plotted as a single closed curve in the  $y$ -plane shown in Fig. 9.15.4.

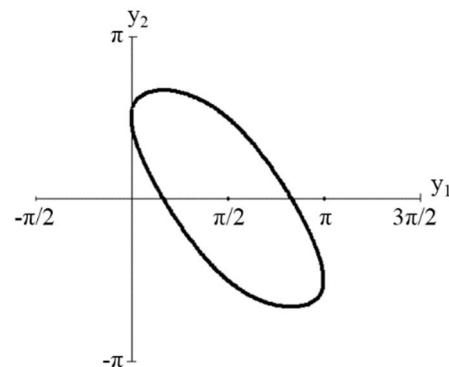


Figure 9.15.4 Single Component Self-Motion Manifold with  $\bar{z} = 1$

### 9.15.7 A Large Scale Redundant Serial Manipulator

To illustrate the foregoing formulation with a large scale manipulator, the inverse kinematic mapping of a *generic 7-DOF serial manipulator* with seven revolute joints, defined by its Denavit-Hartenberg (DH) parameters (Jazar, 2010) in Table 9.15.1, is computed. The DH parameter transformation between reference frames attached to links  $i$  and  $i-1$  is as follows:

- (a)  $z_0 \rightarrow z_6$  are the axes of the seven revolute joints
- (b)  $\theta_i$  is the rotation about axis  $z_{i-1}$  to make axis  $x_{i-1}$  parallel to axis  $x_i$
- (c)  $d_i$  is the translation along axis  $z_{i-1}$  to make axis  $x_{i-1}$  coincident with axis  $x_i$
- (d)  $a_i$  is the translation along axis  $x_i$  to make the origins of both frames coincident
- (e)  $\varphi_i$  is the rotation about axis  $x_i$  to make axis  $z_{i-1}$  coincident with axis  $z_i$

All joints are of revolute type, so angles  $\varphi_i$  are the input coordinates  $y_i$  with offsets shown in the second column of Table 1. The design parameters of this manipulator ( $d_i$ ,  $a_i$ , and  $\varphi_i$ ) are generic. This means that, unlike most industrial serial manipulators, the axes of successive joints of the manipulator are not subject to special relations (perpendicularity, parallelism, intersection, null link offset, etc) that simplify its kinematics. This *7R serial manipulator* is illustrated in Fig. 9.15.5 at its home configuration, with  $y_i = 0$ ,  $i = 1, \dots, 7$ .

Table 9.15.1. DH parameters of a Generic 7-DOF Revolute Serial Manipulator

Frame $i-1$ to $i$	$\varphi_i$ (rad)	$d_i$ (m)	$a_i$ (m)	$\theta_i$ (rad)
$0 \rightarrow 1$	$y_1 + 0.0779$	0.34720	0.28996	-2.1364
$1 \rightarrow 2$	$y_2 - 0.1052$	-0.31939	-0.252	-3.5813
$2 \rightarrow 3$	$y_3 - 0.1437$	0.3600	-0.4538	-1.1741
$3 \rightarrow 4$	$y_4 - 0.2941$	-0.3534	0.3260	0.6745
$4 \rightarrow 5$	$y_5 + 0.2321$	-0.25974	0.4102	-0.7619
$5 \rightarrow 6$	$y_6 - 0.1586$	0.21070	0.19289	2.6738
$6 \rightarrow 7$	$y_7 + 0.3050$	-0.24998	0.23925	0.9863

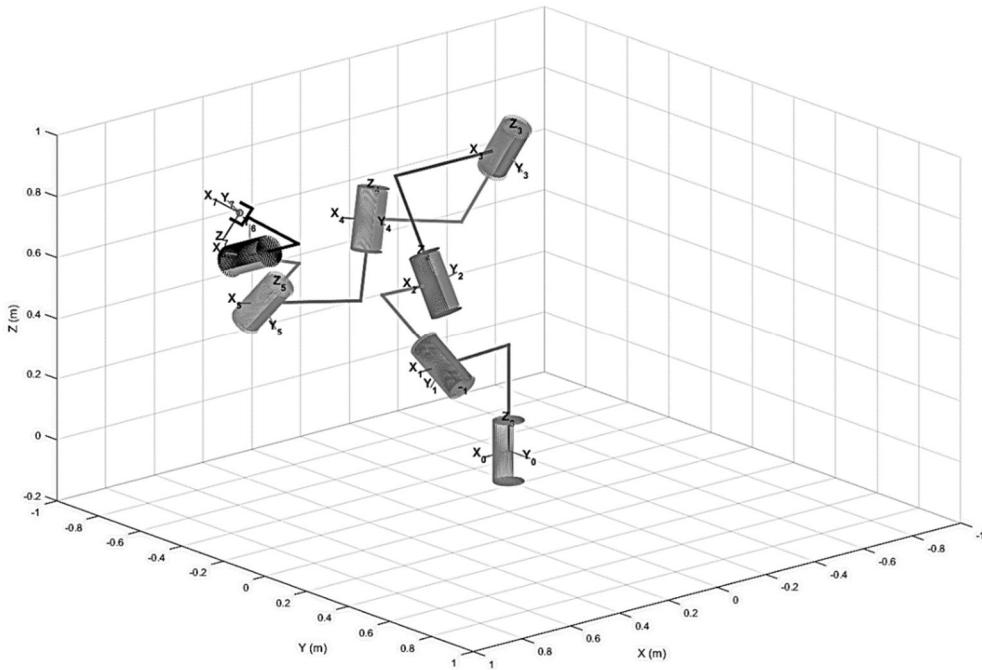


Figure 9.15.5 Schematic Representation of Generic 7R Robot in its Home Configuration

The forward kinematics of this manipulator are computed using the DH method (Jazar, 2010), which multiplies the following homogeneous transformation matrices from the base frame to the end-effector frame:

$${}^0\mathbf{T}_7(\mathbf{y}) = \prod_{i=1}^7 {}^{i-1}\mathbf{T}_i(y_i)$$

$${}^{i-1}\mathbf{T}_i(y_i) = \begin{bmatrix} \cos(\varphi_i) & -\cos(\varphi_i)\sin(\theta_i) & \sin(\varphi_i)\sin(\theta_i) & a_i \cos(\varphi_i) \\ \sin(\varphi_i) & \cos(\varphi_i)\cos(\theta_i) & -\sin(\varphi_i)\cos(\theta_i) & a_i \sin(\varphi_i) \\ 0 & \sin(\theta_i) & \cos(\theta_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9.15.15)$$

The result of this product is  ${}^0\mathbf{T}_7$ , which represents the position and orientation of the end-effector  $x_7-y_7-z_7$  frame relative to the fixed-base  $x_0-y_0-z_0$  frame, as a function of the input coordinates. The forward kinematic mapping required in Eq. (9.15.1) can be constructed from Eq. (9.15.15), as follows. Output coordinates  $\mathbf{z} = [z_1 \ z_2 \ z_3 \ z_4 \ z_5 \ z_6]^T$  are defined as position and orientation coordinates of the end-effector relative to the base frame. The position coordinates are extracted from entries (1,4), (2,4), and (3,4) of  ${}^0\mathbf{T}_7$ . In the following, the entry of  ${}^0\mathbf{T}_7$  at its  $i$ -th row and  $j$ -th column is denoted  $t_{ij}$ , in which case,

$$[z_1 \ z_2 \ z_3]^T = [t_{14} \ t_{24} \ t_{34}]^T \quad (9.15.16)$$

*Orientation output coordinates* are chosen as three *Euler angles*; e.g., three successive rotations about moving axes that rotate the base frame until its axes are parallel to those of the end-effector frame. The first rotation is defined as  $z_4$  about axis X, the second rotation as  $z_5$  about

axis Y, and the third rotation as  $z_6$  about axis Z. It can be confirmed that the rotation angles are obtained from  ${}^0T_7$  as

$$\begin{bmatrix} z_4 & z_5 & z_6 \end{bmatrix}^T = \left[ \arctan2\left(\frac{-t_{23}}{\sqrt{1-t_{13}^2}}, \frac{t_{33}}{\sqrt{1-t_{13}^2}}\right) \quad \arcsin(t_{13}) \quad \arctan2\left(\frac{-t_{12}}{\sqrt{1-t_{13}^2}}, \frac{t_{11}}{\sqrt{1-t_{13}^2}}\right) \right]^T \quad (9.15.17)$$

where  $\arctan2(y,x)$  is the two-argument inverse tangent that transforms y and x coordinates of a point to its polar angle in the correct quadrant. Equations (9.15.16) and (9.15.17) constitute the input-output equation of Eq. (9.15.1) for this manipulator. The self-motion manifold will be obtained for the following *end-effector pose*:

$$\bar{z} = [0.7507 \text{ m} \quad -0.4658 \text{ m} \quad 0.6662 \text{ m} \quad 2.8893 \text{ rad} \quad 0.1559 \text{ rad} \quad 0.2839 \text{ rad}]^T \quad (9.15.18)$$

which was obtained by setting  $y$  to the following random value  $\bar{y}^*$ , and using Eqs. (9.15.16) and (9.15.17):

$$\bar{y}^* = [-0.0007 \quad 0.1533 \quad -0.0770 \quad 0.0371 \quad -0.0226 \quad 0.1117 \quad -0.1089]^T \text{ rad}$$

For 7R redundant manipulators with simplified design, it is possible to construct an analytic parameterization of the inverse kinematic mapping and its self-motion manifold. This is the case for a 7R robot with SRS structure, where the axes of the first three revolute joints intersect at a point, effectively constituting a spherical (S) joint. This is also true for the last three joints of the SRS manipulator. An intermediate revolute joint (R) connects these two spherical joints. The self-motion manifolds of the SRS manipulator can be analytically parameterized by an “arm angle” (Shimizu et al., 2008), which is the angle between the plane that contains the arm and some reference plane. As shown therein, the input coordinates of the SRS can be analytically parameterized in terms of  $\theta$ , for a desired position and orientation  $\bar{z}$ . This is, however, not possible for the 7R arm with generic design.

Another possible method to evaluate the inverse kinematic mapping for a desired  $\bar{z}$  is to choose an input coordinate  $\dot{y}_j$  as the independent parameter. This effectively transforms the original redundant manipulator into a nonredundant one, so that the remaining 6 input coordinates  $y_i \neq \dot{y}_j$  can be analytically determined as functions of the independent coordinate and  $\bar{z}$ . Although this is feasible for relatively simple redundant manipulators (Peidro et. al., 2018), for more complex manipulators such as the one studied in this section, this is not practical, or even possible. Since, the 7R manipulator studied has a generic design, if any input coordinate is chosen as an independent parameter, the resulting *6R manipulator configuration* still has a generic design. It is known that a generic 6R manipulator admits up to 16 different real solutions for a desired  $\bar{z}$  (Lee and Liang, 1988); i.e., up to sixteen components of the regular configuration space and self-motion manifold. This means that, for a given value of the independent input coordinate  $\dot{y}_j$ , solving for the remaining six input coordinates requires a costly algebraic elimination procedure to reduce the kinematic equations to a 16-th degree polynomial. This is even worse if the independent input coordinate is varied on a grid to map the

self-motion manifold (Peidro et. al., 2018), since each node of the grid would require solving this 16-th degree polynomial with updated coefficients.

In summary, there is no practical analytical method to evaluate the inverse kinematic mapping of a general 7R serial manipulator. A formulation such as that presented in Sections 9.15.1 through 9.15.6 is required. The self-motion manifold defined by  $\bar{z}$  of Eq. (9.15.18) is mapped using the algorithm of Section 2.6. The mapping starts at  $\bar{y}^*$  and travels along a *one-dimensional self-motion manifold component* until returning again to  $\bar{y}^*$  (considering the wrapping of angles every  $2\pi$  rad). Computations are carried out using the following algorithmic parameters:

- (a) Step  $h$  used to map the self-motion manifold: 0.01
- (b) Tolerance used in the iterative computation of  $\mathbf{h}$  and  $\mathbf{B}$  according to Appendix 9.A, during mapping of the manifolds, according to Section 2.6: 0.0001.
- (c) A reset of  $v$ ,  $\mathbf{U}$ ,  $\mathbf{V}$ , and  $\mathbf{B}$  is performed whenever the distance between the current point  $\mathbf{y}^k$  generated on the manifold and the previous one  $\mathbf{y}^{k-1}$  exceeds  $0.1$  rad, i.e.:  $\|\mathbf{y}^k - \mathbf{y}^{k-1}\| > 0.1$ .

This is done to prevent the current parameterization ( $v, \mathbf{V}$ ) from becoming singular, since a sudden change of  $\mathbf{y}$  is a signal that a singularity of the current parameterization is being approached.

When run with MATLAB 2015a on an Intel Core i3-8130U CPU @ 2.20GHz with 8GB RAM, the complete mapping of the manifold took an average time of 17.7 seconds

Figures 9.15.6(a-g) show the evolution of the seven components of  $\mathbf{y}$  along the self-motion manifold, whereas Fig. 9.15.6(h) shows the projection of the manifold onto the subspace of angles  $(y_4, y_5, y_6)$ . In Figs. 9.15.6(a-g), the horizontal axis represents the index  $k$  of each  $\mathbf{y}^k$  generated along the manifold. As seen in Fig. 9.15.6, all input coordinates except  $y_2$  and  $y_3$  describe closed trajectories. Angles  $y_2$  and  $y_3$  suffer wrapping, such that the difference between their final and initial values is  $2\pi$ . Accordingly, the initial pose of each body of the robot coincides with its final pose, as adding an integer multiple of  $2\pi$  to any  $y_i$  does not alter the configuration of the robot. Considering wrapping, the self-motion manifold is a closed curve, which starts at  $\bar{y}^*$ , travels along the manifold in the direction of the dotted arrow of Fig. 9(h), and ends at  $\bar{y}^*$ .

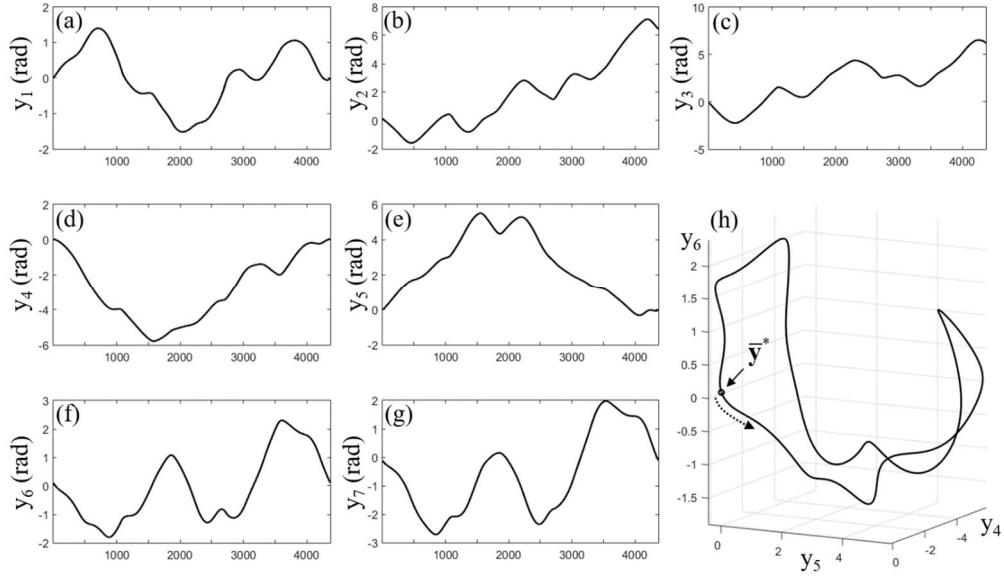


Figure 9.15.6 (a-g) Plots of the of  $y_i$  Along the Self-Motion Manifold Defined by Eq. (9.15.18);  
(h) Projection of the Self-Motion Manifold on the Subspace of  $(y_4 \ y_5 \ y_6)$

### 9.15.8 Serial Manipulator Inverse Dynamics

To extend the foregoing to the dynamics domain, differentiating Eq. (9.15.7) with respect to time and suppressing arguments,  $\dot{\mathbf{y}} = \mathbf{V}\dot{\mathbf{v}} - \mathbf{U}(\mathbf{h}_v\dot{\mathbf{v}} + \mathbf{h}_z\dot{\mathbf{z}})$ . Substituting from Eqs. (9.15.10),

$$\dot{\mathbf{y}} = \mathbf{V}\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{y})\mathbf{G}'\mathbf{V}\dot{\mathbf{v}} + \mathbf{U}\mathbf{B}(\mathbf{y})\dot{\mathbf{z}} = (\mathbf{D}(\mathbf{y})\mathbf{V}\dot{\mathbf{v}} + \mathbf{U}\mathbf{B}(\mathbf{y})\dot{\mathbf{z}})_{|_{\mathbf{y}=\bar{\mathbf{y}}+\mathbf{V}\mathbf{v}-\mathbf{U}\mathbf{h}(\mathbf{v},\mathbf{z})}} \quad (9.15.19)$$

Where  $\mathbf{D}(\mathbf{y}) \equiv (\mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{y})\mathbf{G}'(\mathbf{y}))_{|_{\mathbf{y}=\bar{\mathbf{y}}+\mathbf{V}\mathbf{v}-\mathbf{U}\mathbf{h}(\mathbf{v},\mathbf{z})}}$ . Equation **Error! Reference source not found.** is a  $2(n-m)$  parameter inverse kinematic velocity mapping, for arbitrary  $\mathbf{v}$  and  $\dot{\mathbf{v}}$ . Note that the redundant manipulator literature that works only at the velocity level misses half of these parameters; namely  $\mathbf{v}$ .

Differentiating Eq. **Error! Reference source not found.** with respect to time,

$$\ddot{\mathbf{y}} = \mathbf{D}(\mathbf{y})\mathbf{V}\ddot{\mathbf{v}} + \mathbf{U}\mathbf{B}(\mathbf{y})\ddot{\mathbf{z}} - \mathbf{U}(\mathbf{B}(\mathbf{y})\mathbf{G}'\mathbf{V}\ddot{\mathbf{v}})_{\mathbf{y}} \dot{\mathbf{y}} - \mathbf{U}\mathbf{B}(\mathbf{y})(\mathbf{G}'(\mathbf{y})\mathbf{V}\ddot{\mathbf{v}})_{\mathbf{y}} \dot{\mathbf{y}} + \mathbf{U}(\mathbf{B}(\mathbf{y})\ddot{\mathbf{z}})_{\mathbf{y}} \dot{\mathbf{y}} \quad (9.15.20)$$

To evaluate terms of the form  $(\mathbf{B}(\mathbf{y})\ddot{\mathbf{a}})_{\mathbf{y}}$ , Eq. (9.15.9) is written in the form  $\mathbf{G}'(\mathbf{y})\mathbf{U}\mathbf{B}(\mathbf{y}) = \mathbf{I}$  or, with  $\mathbf{a}$  constant,  $\mathbf{G}'(\mathbf{y})\mathbf{U}\mathbf{B}(\mathbf{y})\mathbf{a} = \mathbf{a}$ . Differentiating this identity with respect to  $\mathbf{y}$ ,

$$(\mathbf{G}'(\mathbf{y})\mathbf{U}\ddot{\mathbf{B}}\mathbf{a})_{\mathbf{y}} + \mathbf{G}'(\mathbf{y})\mathbf{U}(\mathbf{B}(\mathbf{y})\ddot{\mathbf{a}})_{\mathbf{y}} = \mathbf{0}.$$

Using Eq. (9.15.9),  $(\mathbf{B}(\mathbf{y})\ddot{\mathbf{a}})_{\mathbf{y}} = -\mathbf{B}(\mathbf{y})(\mathbf{G}'(\mathbf{y})\mathbf{U}\ddot{\mathbf{B}}\mathbf{a})_{\mathbf{y}}$ ,

which is computable. Thus, for all  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ , the inverse acceleration mapping of Eq.

**Error! Reference source not found.** is

$$\begin{aligned} \ddot{\mathbf{y}} &= (\mathbf{D}(\mathbf{y})\mathbf{V}\ddot{\mathbf{v}} + \mathbf{U}\mathbf{B}(\mathbf{y})\ddot{\mathbf{z}})_{|_{\mathbf{y}=\bar{\mathbf{y}}+\mathbf{V}\mathbf{v}-\mathbf{U}\mathbf{h}(\mathbf{v},\mathbf{z})}} \\ &\quad + \mathbf{U}\mathbf{B}(\mathbf{y}) \left[ (\mathbf{G}'(\mathbf{y})\mathbf{U}\ddot{\mathbf{B}}\mathbf{G}'\mathbf{V}\ddot{\mathbf{v}})_{\mathbf{y}} - (\mathbf{G}'(\mathbf{y})\mathbf{V}\ddot{\mathbf{v}})_{\mathbf{y}} - (\mathbf{G}'(\mathbf{y})\mathbf{U}\ddot{\mathbf{B}}\mathbf{z})_{\mathbf{y}} \right] \dot{\mathbf{y}}_{|_{\substack{\mathbf{y}=\bar{\mathbf{y}}+\mathbf{V}\mathbf{v}-\mathbf{U}\mathbf{h}(\mathbf{v},\mathbf{z}) \\ \dot{\mathbf{y}}=(\mathbf{D}(\mathbf{y})\mathbf{V}\dot{\mathbf{v}}-\mathbf{U}\mathbf{B}(\mathbf{y})\dot{\mathbf{z}})}}} \end{aligned} \quad (9.15.21)$$

The function  $\mathbf{h}(\mathbf{v}, \mathbf{z})$  and matrix  $\mathbf{B}(\mathbf{y})$  in the foregoing are evaluated in Appendix A. Provided the choice of  $\mathbf{v}$  assures  $[\mathbf{y}^T \quad \mathbf{z}^T]^T \in \tilde{\mathbf{X}}_i^s$ , the foregoing relations involve free parameters  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  that may be chosen to satisfy kinematic, dynamic, and control requirements.

## 9.16 Kinematically Redundant Compound Manipulators

Consider a manipulator with *input coordinates*  $\mathbf{y} \in \mathbb{R}^n$ , *generalized coordinates*  $\mathbf{q} \in \mathbb{R}^{ngc}$  of the underlying mechanism, and *output coordinates*  $\mathbf{z} \in \mathbb{R}^m$ , with  $m < n$ . The generalized coordinates are subject to  $n_{hc} = ngc - n$  *holonomic constraints*,

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (9.16.1)$$

with  $n_{hc} \times ngc$  *Jacobian*  $\Phi_q(\mathbf{q})$ . In neighborhoods of  $\mathbf{q}$  at which the constraint Jacobian has full row rank, the mechanism has  $n = ngc - n_{hc}$  *degrees of freedom*. Input and generalized coordinates are required to satisfy  $n$  *input equations*,

$$\Psi(\mathbf{y}, \mathbf{q}) = \mathbf{0} \quad (9.16.2)$$

that are intended to determine  $\mathbf{y}$  as a function of  $\mathbf{q}$  in *inverse kinematics*. Output and generalized coordinates are required to satisfy  $m$  *output equations*,

$$\Gamma(\mathbf{q}, \mathbf{z}) = \mathbf{0} \quad (9.16.3)$$

that are intended to determine  $\mathbf{z}$  as a function of  $\mathbf{q}$  in *forward kinematics*. A manipulator with this structure are called *compound kinematically redundant manipulators*.

### 9.16.1 Compound Manipulator Configuration Space

With *manipulator coordinates*  $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{q}^T \quad \mathbf{z}^T]^T \in \mathbb{R}^{n+ngc+m}$ , the *compound manipulator configuration space* is defined as  $X^c = \{\mathbf{x} : \Phi(\mathbf{q}) = \mathbf{0}, \Psi(\mathbf{y}, \mathbf{q}) = \mathbf{0}, \text{ and } \Gamma(\mathbf{q}, \mathbf{z}) = \mathbf{0}\}$ . While this space characterizes the geometry of the manipulator, it contains *singular configurations* that preclude desirable manipulator input and output relations.

The combined equations

$$\Omega(\mathbf{y}, \mathbf{q}) \equiv \begin{bmatrix} \Phi(\mathbf{q}) \\ \Psi(\mathbf{y}, \mathbf{q}) \end{bmatrix} = \mathbf{0} \quad (9.16.4)$$

comprise a system of  $n_{hc} + n = ngc$  equations that are intended to determine  $\mathbf{q}$  as a function of  $\mathbf{y}$  in *forward kinematics*. A sufficient condition that this is possible in a neighborhood of a configuration  $\bar{\mathbf{x}} \in X^c$  is

$$|\Omega_q(\bar{\mathbf{y}}, \bar{\mathbf{q}})| = \left\| \begin{bmatrix} \Phi_q(\bar{\mathbf{q}}) \\ \Psi_q(\bar{\mathbf{y}}, \bar{\mathbf{q}}) \end{bmatrix} \right\| \neq \mathbf{0} \quad (9.16.5)$$

In this case, the implicit function theorem implies there is a unique solution of Eq. (9.16.4),

$$\mathbf{q} = \mathbf{f}(\mathbf{y}) \quad (9.16.6)$$

in a neighborhood of  $\bar{\mathbf{y}}$ . The  $m$  output equations of Eq. (9.16.3) are intended to determine  $\mathbf{z} \in \mathbb{R}^m$  as a function of  $\mathbf{q}$  in forward kinematics. A sufficient condition for this is

$$|\Gamma_z(\bar{\mathbf{q}}, \bar{\mathbf{z}})| \neq 0 \quad (9.16.7)$$

which assures existence of a unique solution,

$$\mathbf{z} = \mathbf{e}(\mathbf{q}) \quad (9.16.8)$$

in a neighborhood of  $\bar{\mathbf{q}}$ . Under the foregoing sufficiency conditions, Eqs. (9.16.6) and (9.16.8) yield the *forward kinematic mapping* that is key to control of the manipulator,

$$\mathbf{z} = \mathbf{e}(\mathbf{f}(\mathbf{y})) \equiv \mathbf{G}(\mathbf{y}) \quad (9.16.9)$$

Since the dimension  $m$  of  $\mathbf{z}$  is less than the dimension  $n$  of  $\mathbf{y}$ , there is no prospect for a single valued inverse kinematic mapping from  $\mathbf{z}$  to  $\mathbf{y}$  that corresponds to Eq. (9.16.9). In fact, the combined conditions of Eqs. (9.16.1) and (9.16.3),

$$\Lambda(\mathbf{q}, \mathbf{z}) \equiv \begin{bmatrix} \Phi(\mathbf{q}) \\ \Gamma(\mathbf{q}, \mathbf{z}) \end{bmatrix} = \mathbf{0} \quad (9.16.10)$$

comprise  $n_{hc} + m = n_{gc} - (n - m) < n_{gc}$  equations in  $n_{gc}$  generalized coordinates  $\mathbf{q}$ . There cannot exist, therefore, a unique solution of Eq. (9.16.10) for  $\mathbf{q}$  as a function of  $\mathbf{z}$ . If the Jacobian  $\Lambda_q(\bar{\mathbf{q}}, \bar{\mathbf{z}})$  has full row rank  $n_{gc} - (n - m)$  for  $\bar{\mathbf{x}} \in \tilde{X}^c$ , Eq. (9.16.10) has an  $(n_{gc} - (n - m)) \times (n_{gc} - (n - m))$  nonsingular submatrix. The implicit function theorem implies that the associated  $n_{gc} - (n - m)$  elements of  $\mathbf{q} \in R^{n_{gc}}$  can be determined by Eq. (9.16.10) as functions of  $\mathbf{z}$  and the remaining  $n - m$  elements of  $\mathbf{q}$ ; i.e., an  $n - m$  parameter family of solutions of Eq. (9.16.10) for  $\mathbf{z}$  in a neighborhood of  $\bar{\mathbf{z}}$ . A sufficient condition that the  $n$  input equations of Eq. (9.16.2) determine  $\mathbf{y} \in R^n$  as a function of  $\mathbf{q}$  in inverse kinematics is

$$|\Psi_y(\bar{\mathbf{y}}, \bar{\mathbf{q}})| \neq 0 \quad (9.16.11)$$

which assures existence of a unique solution of Eq. (9.16.2) in a neighborhood of  $\bar{\mathbf{q}}$ ,

$$\mathbf{y} = \mathbf{g}(\mathbf{q}) \quad (9.16.12)$$

Thus, the  $n \times m$  parameter inverse kinematic mapping from  $\mathbf{z}$  to  $\mathbf{q}$  yields an  $n - m$  parameter *inverse mapping* from  $\mathbf{z}$  to  $\mathbf{y}$ , using Eq. (9.16.12).

### 9.16.2 Regular Compound Manipulator Configuration Space

To assure existence of the foregoing manipulator properties and to avoid *singular configurations* that occur in  $X^c$ , the *regular compound manipulator configuration space* is defined as  $\tilde{X}^c = \left\{ \mathbf{x} \in X^c : |\Omega_q(\mathbf{y}, \mathbf{q})| \neq 0, |\Psi_y(\mathbf{y}, \mathbf{q})| \neq 0, |\Lambda_q(\mathbf{q}, \mathbf{z})\Lambda_q^T(\mathbf{q}, \mathbf{z})| \neq 0, \text{ and } |\Gamma_z(\mathbf{q}, \mathbf{z})| \neq 0 \right\}$

Since this is an open subset of  $X^c$  in the *relative topology* of  $R^{n+n_{gc}+m}$ , it is comprised of a collection of maximal disjoint *components*  $\tilde{X}_i^c$  that are path connected, and singularity free (Mendelson, 1962). Further,  $\tilde{X}_i^c \cap \tilde{X}_j^c = \emptyset$  if  $i \neq j$  and  $\cup_i \tilde{X}_i^c = \tilde{X}^c$ . For  $\bar{\mathbf{x}} \in \tilde{X}^c$ , the functions  $\mathbf{f}(\mathbf{y})$  and  $\mathbf{e}(\mathbf{q})$  of Eqs. (9.16.6) and (9.16.8) are continuously differentiable and

$\Psi(\mathbf{y}) = \begin{bmatrix} \mathbf{y}^T & \mathbf{f}^T(\mathbf{y}) & \mathbf{e}^T(\mathbf{f}(\mathbf{y})) \end{bmatrix}^T = \mathbf{x} \in \tilde{X}^c$  is a differentiable mapping from open subsets of  $R^n$  onto open subsets of  $\tilde{X}^c$ , with inverse mapping  $\phi(\mathbf{x}) = \mathbf{y}$ . As shown in Part I,  $X^c$  and its components are *differentiable manifolds*, parameterized by  $\mathbf{y} \in R^n$ .

### 9.16.3 Compound Manipulator Inverse Kinematic Configuration Mapping

While regularity properties of functions involved in the definition of  $\tilde{X}^c$  assure existence of the *forward kinematic mapping* of Eq. (9.16.9), they do not assure existence of a *single valued inverse kinematic mapping* of  $\mathbf{z}$  to  $\mathbf{y}$ . In fact, since  $\mathbf{y} \in \mathbb{R}^n$ ,  $\mathbf{z} \in \mathbb{R}^m$ , and  $n > m$ , such an inverse mapping is necessarily *set-valued*; i.e., from Eq. (9.16.9)

$$\mathbf{G}^{-1}(\mathbf{z}) = \{\mathbf{y} : \mathbf{z} = \mathbf{G}(\mathbf{y})\} \subset \mathbb{R}^n \quad (9.16.13)$$

In early literature focused on characterizing the set-valued inverse kinematic mapping of Eq. (9.16.13) (Burdick, 1989; Luck, and Lee, 1995; DeMers and Kreutz-Delgado, 1996), concepts of differential geometry were used to address the problem at the configuration level. Subsequently, an extensive literature has focused almost exclusively on redundant manipulator analysis at the velocity level, where equations are linear in velocities. While useful results have been obtained with velocity analysis, information at the configuration level is lost (Fari et-al, 2018). This section follows the differential geometry approach to analytically and computationally characterize the set-valued inverse mapping of Eq. (9.16.13) at the configuration level.

The first step in *characterizing the set-valued inverse mapping* of Eq. (9.16.13) is to find all  $\mathbf{q}$  that satisfy Eq. (9.16.10) for a given  $\mathbf{z}$ . Since this is a system of  $n_{hc} + m$  equations in  $n_{gc}$  variables and  $\Lambda_q(\mathbf{q}, \mathbf{z})$  has row rank  $n_{gc} - (n - m)$  in  $\tilde{X}^c$ , there is an  $n - m$  parameter family of solutions of Eq. (9.16.10). To construct this family in a neighborhood of  $\bar{\mathbf{x}} \in \tilde{X}^c$ , define

$$\mathbf{U} = \Lambda_q^T(\bar{\mathbf{q}}, \bar{\mathbf{z}}) \quad (9.16.14)$$

that has column rank  $n_{gc} - (n - m)$ , and use *singular value decomposition*

$$\Lambda_q(\bar{\mathbf{q}}, \bar{\mathbf{z}})\mathbf{V} = \mathbf{0} \quad \mathbf{V}^T\mathbf{V} = \mathbf{I} \quad (9.16.15)$$

to obtain  $\mathbf{V}$  with column rank  $n - m$  (Atkinson, 1989). Since the columns of  $\mathbf{V}$  and  $\mathbf{U}$  span  $\mathbb{R}^{n_{gc}}$ , any solution  $\mathbf{q}$  of Eq. (9.16.10) may be represented as

$$\mathbf{q} = \bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u} \quad (9.16.16)$$

where  $\mathbf{v} \in \mathbb{R}^{n-m}$  and  $\mathbf{u} \in \mathbb{R}^{n_{gc}-(n-m)}$ . Note that at  $\mathbf{q} = \bar{\mathbf{q}}$ ,  $\mathbf{v} = \bar{\mathbf{v}} = \mathbf{0}$  and  $\mathbf{u} = \bar{\mathbf{u}} = \mathbf{0}$ . In order for  $\mathbf{q}$  of Eq. (9.16.16) to satisfy Eq. (9.16.10), for given values of  $\mathbf{v}$  and  $\mathbf{z}$ ,  $\mathbf{u}$  must satisfy

$$\Lambda(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}, \mathbf{z}) = \mathbf{0} \quad (9.16.17)$$

The Jacobian of the left side of this equation with respect to  $\mathbf{u}$ , evaluated at  $\bar{\mathbf{x}}$ , is

$\Lambda(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}, \bar{\mathbf{z}})_u = -\Lambda_q(\bar{\mathbf{q}}, \bar{\mathbf{z}})\mathbf{U} = -\mathbf{U}^T\mathbf{U}$ , which is nonsingular. Therefore, there exists a unique differentiable solution  $\mathbf{u} = \mathbf{h}(\mathbf{v}, \mathbf{z})$  of Eq. (9.16.17), for all  $(\mathbf{v}, \mathbf{z})$  in a neighborhood of  $(\mathbf{0}, \bar{\mathbf{z}})$ . The solution of Eq. (9.16.10) for  $\mathbf{q}$  is thus

$$\mathbf{q}(\mathbf{v}, \mathbf{z}) = \bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z}) \quad (9.16.18)$$

for any  $\mathbf{v} \in \mathbb{R}^{n-m}$  in a neighborhood of  $\mathbf{0}$  and  $\mathbf{z}$  in a neighborhood of  $\bar{\mathbf{z}}$ . Using Eq. (9.16.12),

$$\mathbf{y}(\mathbf{v}, \mathbf{z}) = \mathbf{g}(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z})) \quad (9.16.19)$$

for any  $\mathbf{v} \in \mathbb{R}^{n-m}$  in a neighborhood of  $\mathbf{0}$  and  $\mathbf{z}$  in a neighborhood of  $\bar{\mathbf{z}}$ .

#### 9.16.4 Parameterization of the Regular Compound Manipulator Configuration Space

Equation (9.16.19) provides a continuously differentiable parameterization of  $\tilde{X}^s$  on a neighborhood  $N$ ; i.e.,

$$\tilde{X}_N^c = \left\{ \mathbf{x} \in X^c : \mathbf{q} = \bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z}), \mathbf{y} = \mathbf{g}(\mathbf{q}), \begin{array}{l} \text{for all } \mathbf{v} \text{ in a neighborhood of } \mathbf{0} \\ \text{and } \mathbf{z} \text{ in a neighborhood of } \bar{\mathbf{z}} \end{array} \right\} \quad (9.16.20)$$

Neighborhoods assured by the implicit function theorem and the mappings of Eqs. (9.16.18) and (9.16.19) define a *chart* on  $\tilde{X}^c$ . Creating a family of such charts that cover  $\tilde{X}^c$  provides an *atlas* that defines  $\tilde{X}^c$  as a *differentiable manifold* with disjoint, maximal, path connected, singularity free components  $\tilde{X}_i^c$  (Guillemin and Pollack, 1974). The basis for this extension of local charts to the global manifold  $\tilde{X}^c$  is as outlined in Section 9.15.4.

For use in manipulator kinematic analysis and control, partial derivatives of  $\mathbf{h}(\mathbf{v}, \mathbf{z})$ ,  $\mathbf{q}(\mathbf{v}, \mathbf{z})$ , and  $\mathbf{y}(\mathbf{v}, \mathbf{z})$  with respect to  $\mathbf{v}$  and  $\mathbf{z}$  are needed. To obtain  $\mathbf{h}_v(\mathbf{v}, \mathbf{z})$  and  $\mathbf{h}_z(\mathbf{v}, \mathbf{z})$ , the partial derivatives of Eq. (9.16.17) with respect to  $\mathbf{v}$  and  $\mathbf{z}$ , evaluated at  $\mathbf{u} = \mathbf{h}(\mathbf{v}, \mathbf{z})$ , are

$$\begin{aligned} \Lambda_q (\mathbf{V} - \mathbf{U}\mathbf{h}_v) &= \mathbf{0} \\ \Lambda_q (-\mathbf{U}\mathbf{h}_z) + \Lambda_z &= \mathbf{0} \end{aligned} \quad (9.16.21)$$

Since  $\Lambda_q \mathbf{U} = \mathbf{U}^T \mathbf{U}$  is nonsingular at  $\bar{\mathbf{x}}$  and  $\Lambda_q(\mathbf{q}, \mathbf{z})$  is a differentiable function of  $\mathbf{q}$  and  $\mathbf{z}$ ,

$$\mathbf{B}(\mathbf{v}, \mathbf{z}) \equiv (\Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}) \mathbf{U})^{-1} \quad (9.16.22)$$

is nonsingular and differentiable in a neighborhood of  $\bar{\mathbf{x}}$ . Using Eq. (9.16.22) in Eqs. (9.16.21),

$$\begin{aligned} \mathbf{h}_v(\mathbf{v}, \mathbf{z}) &= \mathbf{B}(\mathbf{v}, \mathbf{z}) \Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}) \mathbf{V} \\ \mathbf{h}_z(\mathbf{v}, \mathbf{z}) &= \mathbf{B}(\mathbf{v}, \mathbf{z}) \Lambda_z(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}) \end{aligned} \quad (9.16.23)$$

Using these results with Eq. (9.16.18),

$$\begin{aligned} \mathbf{q}_v(\mathbf{v}, \mathbf{z}) &= \mathbf{V} - \mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z}) \Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}) \mathbf{V} \\ \mathbf{q}_z(\mathbf{v}, \mathbf{z}) &= -\mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z}) \Lambda_z(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}) \end{aligned} \quad (9.16.24)$$

Likewise, with Eq. (9.16.19),  $\mathbf{y}_v(\mathbf{v}, \mathbf{z}) = \mathbf{g}'(\mathbf{q}(\mathbf{v}, \mathbf{z})) \mathbf{q}_v(\mathbf{v}, \mathbf{z})$  and  $\mathbf{y}_z(\mathbf{v}, \mathbf{z}) = \mathbf{g}'(\mathbf{q}(\mathbf{v}, \mathbf{z})) \mathbf{q}_z(\mathbf{v}, \mathbf{z})$ .

Since  $\Psi(\mathbf{g}(\mathbf{q}), \mathbf{q}) = \mathbf{0}$  is an identity in  $\mathbf{q}$ ,  $\Psi_y(\mathbf{g}(\mathbf{q}), \mathbf{q}) \mathbf{g}'(\mathbf{q}) + \Psi_q(\mathbf{g}(\mathbf{q}), \mathbf{q}) = \mathbf{0}$  in  $\tilde{X}^c$ , where

$|\Psi_y(\mathbf{g}(\mathbf{q}), \mathbf{q})| \neq 0$ . Thus,

$$\mathbf{g}'(\mathbf{q}) = -\Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q}) \Psi_q(\mathbf{g}(\mathbf{q}), \mathbf{q}) \quad (9.16.25)$$

and, using Eq. (9.16.24),

$$\begin{aligned} \mathbf{y}_v(\mathbf{v}, \mathbf{z}) &= \mathbf{g}'(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z})) (\mathbf{V} - \mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z}) \Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}) \mathbf{V}) \\ \mathbf{y}_z(\mathbf{v}, \mathbf{z}) &= -\mathbf{g}'(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z})) \mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z}) \Lambda_z(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}) \end{aligned} \quad (9.16.26)$$

The derivatives of Eqs. (9.16.23), (9.16.24) and (9.16.26) enable redundant compound manipulator applications such as obstacle and singularity avoidance and dynamic performance optimization.

### 9.16.5 The Compound Manipulator Self-Motion Manifold

For fixed  $\mathbf{z} = \bar{\mathbf{z}}$  with  $\bar{\mathbf{x}} \in \tilde{X}_i^c$ , the mapping of Eq. (9.16.19) is

$$\mathbf{y}(\mathbf{v}, \bar{\mathbf{z}}) = \mathbf{g}(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \bar{\mathbf{z}})) \quad (9.16.27)$$

for all  $\mathbf{v}$  in a neighborhood of  $\mathbf{0}$ . This yields sets of inputs in neighborhoods  $N$ ,

$$Y_N^c(\bar{\mathbf{z}}) = \left\{ \mathbf{y} : \mathbf{y} = \mathbf{g}(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \bar{\mathbf{z}})), \text{ for all } \mathbf{v} \in R^{n-m} \text{ in a neighborhood of } \mathbf{0} \right\}, \text{ whose union}$$

$Y^c(\bar{\mathbf{z}})$  is a differentiable manifold in input space. This manifold is called the *self-motion manifold*, since it is a set of inputs, all of which have forward kinematic mappings onto the same output  $\bar{\mathbf{z}}$ . The inverse kinematic mapping of Eq. (9.16.27) is a parameterization of  $Y^c(\bar{\mathbf{z}})$ . Thus,  $Y^c(\bar{\mathbf{z}})$  is a set of inputs that all yield  $\bar{\mathbf{z}}$ , some of which may have attractive properties such as *avoiding obstacles and singularities*. As with all differentiable manifolds,  $Y^c(\bar{\mathbf{z}})$  may be partitioned into maximal disjoint submanifolds  $Y_i^c(\bar{\mathbf{z}})$ ; i.e., it may not be a connected set.

If  $\bar{\mathbf{z}}_i$  and  $\bar{\mathbf{z}}_j$  are adjacent outputs in a neighborhood of  $\bar{\mathbf{z}}$ , one may select vastly different  $\mathbf{y}_i \in Y^c(\bar{\mathbf{z}}_i)$  and  $\mathbf{y}_j \in Y^c(\bar{\mathbf{z}}_j)$ , so the resulting  $\bar{\mathbf{x}}_i$  and  $\bar{\mathbf{x}}_j$  are distinctly different in  $\tilde{X}^c$ . If one wishes to maintain manipulator configuration continuity in  $\tilde{X}^c$ , large uncorrelated excursions in  $Y^c(\bar{\mathbf{z}})$ , with  $\mathbf{z} = \bar{\mathbf{z}}$  fixed, must be avoided. In short, the manipulator regular configuration manifold  $\tilde{X}^c$  and the self-motion manifold  $Y^c(\bar{\mathbf{z}})$  are fundamentally different.

### 9.16.6 Mapping One-Dimensional Compound Manipulator Self-Motion Manifolds

A self-motion manifold component  $Y_i^c(\bar{\mathbf{z}})$  with  $n - m = 1$  is to be mapped, beginning at a configuration  $\bar{\mathbf{x}} = [\bar{\mathbf{y}}^T \quad \bar{\mathbf{q}}^T \quad \bar{\mathbf{z}}^T]^T \in \tilde{X}$ . The output  $\bar{\mathbf{z}}$  is held fixed during the process of mapping  $Y_i^c(\bar{\mathbf{z}})$ . At  $\mathbf{y} = \bar{\mathbf{y}}$ ,  $\mathbf{v}^0 = \mathbf{0}$ , Eqs. (9.16.14) and (9.16.15) are used to evaluate  $\mathbf{U}$  and  $\mathbf{V}$ , and  $\mathbf{B}^0 = \mathbf{B}(0, \bar{\mathbf{z}}) = (\mathbf{U}^T \mathbf{U})^{-1}$  is evaluated. For a small step  $h > 0$  in  $\mathbf{v}$ ,  $\mathbf{v}^1 = h$  and  $\mathbf{h}(\mathbf{v}^1, \bar{\mathbf{z}})$  is obtained, using the iterative process of Eq. (9.A.2) with  $\mathbf{B} = \mathbf{B}^0$  for  $\mathbf{u}^1 = \mathbf{h}(\mathbf{v}^1, \bar{\mathbf{z}})$ . Generalized coordinates  $\mathbf{q}^1(\mathbf{v}, \bar{\mathbf{z}}) = \bar{\mathbf{q}} + \mathbf{V}\mathbf{v}^1 - \mathbf{U}\mathbf{h}(\mathbf{v}^1, \bar{\mathbf{z}})$  are evaluated in Eq. (9.16.18) and  $\mathbf{y}^1$  is obtained by solving  $\Psi(\mathbf{y}, \mathbf{q}^1) = \mathbf{0}$  of Eq. (9.16.2), using Newton-Raphson iteration, beginning with estimate  $\mathbf{y}^{(1)} = \bar{\mathbf{y}}$ . The iterative process of Eq. (9.A.1) is then used to evaluate  $\mathbf{B}^1$ . At  $\mathbf{v}^k = kh$ ,  $k = 2, 1, \dots$ ,  $\mathbf{h}(\mathbf{v}^k, \bar{\mathbf{z}})$  is evaluated as  $\mathbf{u}^k = \mathbf{h}(\mathbf{v}^k, \bar{\mathbf{z}})$  from Eq. (9.A.2) with  $\mathbf{B} = \mathbf{B}^{k-1}$ ,  $\mathbf{q}^k(\mathbf{v}^k, \bar{\mathbf{z}}) = \bar{\mathbf{q}} + \mathbf{V}\mathbf{v}^k - \mathbf{U}\mathbf{h}(\mathbf{v}^k, \bar{\mathbf{z}})$  is evaluated from Eq. (9.16.18),  $\mathbf{y}^k$  is evaluated by iterative solution of  $\Psi(\mathbf{y}, \mathbf{q}^k) = \mathbf{0}$  in Eq. (9.16.2), beginning with estimate  $\mathbf{y}^{(k)} = \mathbf{y}^{k-1}$ , and  $\mathbf{B}^k$  is evaluated using the iterative process of Eq. (9.A.1). The determinant  $|\Lambda_q(\mathbf{q}^k, \bar{\mathbf{z}}) \Lambda_q^T(\mathbf{q}^k, \bar{\mathbf{z}})|$  must be confirmed to be positive to assure  $\mathbf{y}^k \in Y_i^c(\bar{\mathbf{z}})$ . The determinant  $|\Lambda_q(\mathbf{y}^k, \bar{\mathbf{z}}) \mathbf{U}|$  must also have the same sign throughout the mapping process.

If the condition number of  $\Lambda_q(\mathbf{y}^k, \bar{\mathbf{z}})\mathbf{U}$  is within tolerance, continue the mapping process. Otherwise, redefine  $\bar{\mathbf{y}} = \mathbf{y}^k$  and  $\bar{\mathbf{q}} = \mathbf{q}^k$ , retain  $\mathbf{z} = \bar{\mathbf{z}}$ , use Eqs. (9.16.14) and (9.16.15) to evaluate new matrices  $\mathbf{U}$  and  $\mathbf{V}$ , reset  $v^0 = 0$ , evaluate  $\mathbf{B}^0 = (\mathbf{U}^T \mathbf{U})^{-1}$ , index  $v^k = kh$ , and continue the mapping process. This process is as outlined on charts of the differentiable manifold in Section 9.15.4 and shown schematically in Fig. 9.15.1.

### 9.16.7 Model Kinematically Redundant Compound Manipulator Kinematics

The two-bar manipulator shown in Fig. 9.16.1 has a pin at the left end of body 2 that slides in a slot in body 1. The output point P is 2 m to the right of the origin of the body 1 reference frame. The model has 3 input coordinates, 5 generalized coordinates, and 2 output coordinates. The condition that the pin on body 2 slides in the slot in body 1 is the holonomic constraint

$$\Phi(\mathbf{q}) = q_1 \mathbf{u}_y + q_5 \mathbf{A}(q_2) \mathbf{u}_x - (q_3 \mathbf{u}_x - \mathbf{A}(q_4) \mathbf{u}_x) = \begin{bmatrix} q_5 \cos q_2 - q_3 + \cos q_4 \\ q_1 + q_5 \sin q_2 + \sin q_4 \end{bmatrix} = \mathbf{0} \quad (9.16.28)$$

Input and output equations are

$$\Psi(\mathbf{y}, \mathbf{q}) = [q_3 - y_1 \quad q_1 - y_2 \quad q_5 - y_3]^T = \mathbf{0} \quad (9.16.29)$$

$$\Gamma(\mathbf{q}, \mathbf{z}) = [2 \cos q_2 - z_1 \quad q_1 + 2 \sin q_2 - z_2]^T = \mathbf{0} \quad (9.16.30)$$

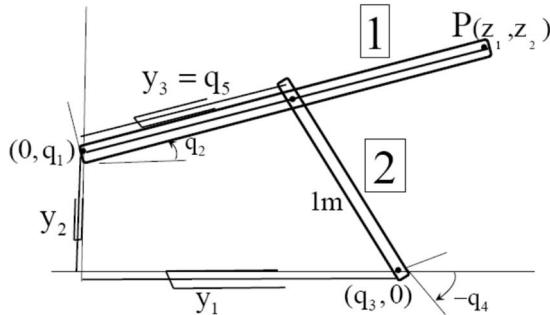


Figure 9.16.1 Model Compound Manipulator

Jacobians  $\Psi_y(\mathbf{y}, \mathbf{q})$  and  $\Gamma_z(\mathbf{q}, \mathbf{z})$  are nonsingular, leading to the solutions

$$\begin{aligned} \mathbf{y} &= [q_3 \quad q_1 \quad q_5]^T \\ \mathbf{z} &= [2 \cos q_2 \quad q_1 + 2 \sin q_2]^T \end{aligned} \quad (9.16.31)$$

The composite Jacobian

$$\Omega_q(\mathbf{y}, \mathbf{q}) = \begin{bmatrix} \Phi_q(\mathbf{q}) \\ \Psi_q(\mathbf{y}, \mathbf{q}) \end{bmatrix} = \begin{bmatrix} 0 & -q_5 \sin q_2 & -1 & -\sin q_4 & \cos q_2 \\ 1 & q_5 \cos q_2 & 0 & \cos q_4 & \sin q_2 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (9.16.32)$$

has determinant  $|\boldsymbol{\Omega}_q(\mathbf{y}, \mathbf{q})| = q_5 \sin(q_4 - q_2)$ . Thus, as long as  $q_5 = y_3 \neq 0$  and the bars are not colinear; i.e.,  $\sin(q_4 - q_2) \neq 0$ , there is a unique solution  $\mathbf{q} = \mathbf{f}(\mathbf{y})$  of  $\boldsymbol{\Omega}(\mathbf{y}, \mathbf{q}) = \mathbf{0}$  in each component of  $\tilde{X}^c$ . The composite Jacobian

$$\boldsymbol{\Lambda}_q(\mathbf{q}, \mathbf{z}) = \begin{bmatrix} \boldsymbol{\Phi}_q(\mathbf{q}) \\ \boldsymbol{\Gamma}_q(\mathbf{q}, \mathbf{z}) \end{bmatrix} = \begin{bmatrix} 0 & -q_5 \sin q_2 & -1 & -\sin q_4 & \cos q_4 \\ 1 & q_5 \cos q_2 & 0 & \cos q_4 & \sin q_4 \\ 0 & -2 \sin q_2 & 0 & 0 & 0 \\ 1 & 2 \cos q_2 & 0 & 0 & 0 \end{bmatrix} \quad (9.16.33)$$

should have full row rank. Setting  $\boldsymbol{\Lambda}_q^T(\mathbf{q}, \mathbf{z})\mathbf{a} = \mathbf{0}$  leads to the following equations:  $a_2 + a_4 = 0$ ,  $-q_5 \sin q_2 a_1 + q_5 \cos q_2 a_2 - 2 \sin q_2 a_3 + 2 \cos q_2 a_4 = 0$ ,  $a_1 = 0$ ,  $-\sin q_4 a_1 + \cos q_4 a_2 = 0$ , and  $\cos q_4 a_1 + \sin q_4 a_2 = 0$ . Thus,  $a_1 = 0$ ,  $\cos q_4 a_2 = 0 = \sin q_4 a_2 \Rightarrow a_2 = 0$ ,  $a_4 = 0$ , and  $-2 \sin q_2 a_3 = 0 \Rightarrow a_3 = 0$  if  $\sin q_2 \neq 0$ . This shows that the composite Jacobian  $\boldsymbol{\Lambda}_q(\mathbf{q}, \mathbf{z})$  has full row rank if  $\sin q_2 \neq 0$ . If  $\sin q_2 = 0$ , bar 1 is horizontal and  $\boldsymbol{\Lambda}_q(\mathbf{q}, \mathbf{z})$  fails to have full row rank. With the bar horizontal and to the right of the vertical axis, if  $\delta z_1 > 0$  there is no solution for  $\mathbf{q}$  and if  $\delta z_1 < 0$  there are two distinctly different solutions. This gives insight into the physical significance of the singularity  $\sin q_2 = 0$ .

The configuration space for this manipulator is  $X^c = \{\mathbf{x} \in \mathbb{R}^{10} : \boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0}, \boldsymbol{\Psi}(\mathbf{y}, \mathbf{q}) = \mathbf{0}, \text{ and } \boldsymbol{\Gamma}(\mathbf{q}, \mathbf{z}) = \mathbf{0}\}$  and the regular configuration space is  $\tilde{X}^c = \{\mathbf{x} \in X^c : q_5 \neq 0, \sin(q_4 - q_2) \neq 0, \text{ and } \sin q_2 \neq 0\}$ . Thus, there are eight disjoint, maximal, path connected, singularity free components,

$$\begin{aligned} \tilde{X}_1^c &= \{\mathbf{x} \in X^c : q_5 > 0, \sin(q_4 - q_2) > 0, \sin q_2 > 0\} \\ \tilde{X}_2^c &= \{\mathbf{x} \in X^c : q_5 > 0, \sin(q_4 - q_2) > 0, \sin q_2 < 0\} \\ \tilde{X}_3^c &= \{\mathbf{x} \in X^c : q_5 > 0, \sin(q_4 - q_2) < 0, \sin q_2 < 0\} \\ \tilde{X}_4^c &= \{\mathbf{x} \in X^c : q_5 > 0, \sin(q_4 - q_2) < 0, \sin q_2 > 0\} \\ \tilde{X}_5^c &= \{\mathbf{x} \in X^c : q_5 < 0, \sin(q_4 - q_2) > 0, \sin q_2 > 0\} \\ \tilde{X}_6^c &= \{\mathbf{x} \in X^c : q_5 < 0, \sin(q_4 - q_2) > 0, \sin q_2 < 0\} \\ \tilde{X}_7^c &= \{\mathbf{x} \in X^c : q_5 < 0, \sin(q_4 - q_2) < 0, \sin q_2 < 0\} \\ \tilde{X}_8^c &= \{\mathbf{x} \in X^c : q_5 < 0, \sin(q_4 - q_2) < 0, \sin q_2 > 0\} \end{aligned} \quad (9.16.34)$$

### 9.16.8 Compound Manipulator Obstacle Avoidance Using Self-Motion Manifolds

An application of the redundant manipulator formulation presented is obstacle avoidance through use of self-motion manifolds. This is illustrated with the compound manipulator of Section 9.16.7. Output point P is required to follow the trajectory  $\mathbf{z}_d(t) = [z_{1d}(t) \ z_{2d}(t)]^T$ ,  $0 < t < 0.7$ , as illustrated in Fig. 9.16.2(a), where

$\mathbf{z}_d(t) = [2\cos(t + \pi/20) \quad \sin(t + \pi/20) \quad -\sin(t - 2\pi/5)]^T$ ,  $0 \leq t < 0.7$ . This can be achieved with the *nominal* generalized coordinate and input trajectories

$\mathbf{q}_n(t) = [q_{1n}(t) \quad q_{2n}(t) \quad q_{3n}(t) \quad q_{4n}(t) \quad q_{5n}(t)]^T$  and  $\mathbf{y}_n(t) = [q_{3n}(t) \quad q_{1n}(t) \quad q_{5n}(t)]^T$ , where  $q_{1n}(t) = -\sin(t + \pi/20) - \sin(t - 2\pi/5)$ ,  $q_{2n}(t) = t + \pi/20$ ,  $q_{3n}(t) = \cos(t + \pi/20) + \cos(t - 2\pi/5)$ ,  $q_{4n}(t) = t - 2\pi/5$ , and  $q_{5n}(t) = 1$ , which is in component  $\tilde{\mathbf{X}}_4^c$  of Eq. (9.16.34). A circular obstacle with radius  $r = 0.2$  m, centered at (0.86m, 0.4m) in the plane of Fig. 9.16.2, is to be avoided by link 2 in achieving the desired output trajectory  $\mathbf{z}_d(t)$ .

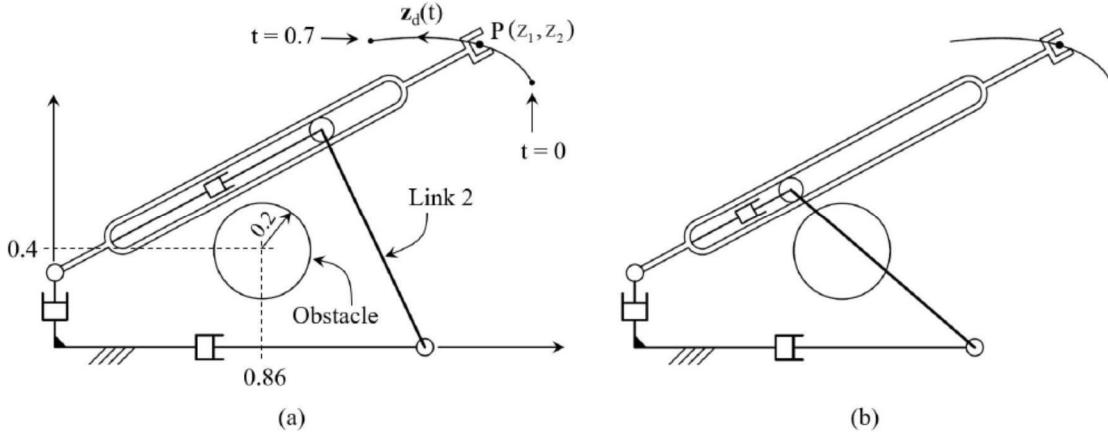


Figure 9.16.2 (a) Desired Output Trajectory  $\mathbf{z}_d(t)$  and Obstacle; (b) Penetration of Obstacle by Link 2

The following algorithm is used to achieve the desired output trajectory, while avoiding collisions between the obstacle and link 2:

- (1) On a grid  $t_i$ ,  $0 \leq t_i \leq 0.7$ , at  $t_1$  no collision exists, as shown in Fig. 9.16.2 (a), so  $\mathbf{q}_n(t_1)$  is a valid configuration. Therefore, set  $\mathbf{q}_l = \mathbf{q}_n(t_1)$ .
- (2) At step  $t_k$ ,  $k \times 2$ , determine if the nominal configuration  $\mathbf{q}_n(t_k)$  produces interference between the obstacle and link 2. When checking interference, link 2 is regarded as a line segment with zero thickness.
  - (2a) If no interference occurs, set  $\mathbf{q}_k = \mathbf{q}_n(t_k)$  and proceed to step  $t_{k+1}$ , repeating (2).
  - (2b) If interference occurs, as in Fig. 9.16.2 (b), proceed to step (3).
- (3) Since  $\mathbf{q}_n(t_k)$  produces interference, a new configuration on the self-motion manifold defined by  $\bar{\mathbf{z}} = \mathbf{z}_d(t_k)$  is to be determined that yields no penetration of the obstacle by link 2. To do so, the one-dimensional self-motion mapping algorithm of Section 9.16.6 is used, starting from  $\bar{\mathbf{q}} = \mathbf{q}_n(t_k)$ . There are two possible directions to map the self-motion manifold, starting from  $\bar{\mathbf{q}}$ , namely  $v$  and  $-v$ .
  - (3a) March along the self-motion manifold in direction  $v$  with step  $h$ , starting from  $\bar{\mathbf{q}}$ , generating  $\mathbf{q}^i(v^i, \bar{\mathbf{z}})$  until finding the first  $\mathbf{q}^j(v^j, \bar{\mathbf{z}})$  that does not produce interference. This means that the obstacle is just contacted for some configuration  $\mathbf{q}^*(v^*, \bar{\mathbf{z}})$  between  $\mathbf{q}^j(v^j, \bar{\mathbf{z}})$  and  $\mathbf{q}^{j-1}(v^{j-1}, \bar{\mathbf{z}})$ . The configuration  $\mathbf{q}^*(v^*, \bar{\mathbf{z}})$  is estimated by returning to  $\mathbf{q}^{j-1}(v^{j-1}, \bar{\mathbf{z}})$  and advancing toward  $\mathbf{q}^j(v^j, \bar{\mathbf{z}})$  using a finer step  $h/10$ , until interference does not occur.

**(3a1)** If index  $i$  exceeds a preset maximum number of steps ( $i_{\max}$ ) while marching along the manifold in the direction of  $v$  without finding an interference-free configuration, abort the mapping and proceed to **(3b)**.

**(3a2)** If  $\|\mathbf{q}_{k-1} - \mathbf{q}^*(v^*, \bar{\mathbf{z}})\| < \text{Tol}$ , with tolerance  $\text{Tol}$ , the  $\mathbf{q}$  trajectory is regarded as continuous.

In that case, set  $\mathbf{q}_k = \mathbf{q}^*(v^*, \bar{\mathbf{z}})$  and proceed to the next time-step  $t_{k+1}$ , repeating **(2)**. Otherwise, proceed to **(3b)**.

**(3b)** Proceed as in **(3a)**, but marching in direction  $-v$ . If this step generates a valid  $\mathbf{q}_k$ , proceed to step  $t_{k+1}$ , repeating step **(2)**. If this sequence of steps fails to yield a valid  $\mathbf{q}_k$ , the desired output trajectory  $\mathbf{z}_d(t)$  is infeasible, and the algorithm ends.

The foregoing algorithm generates a sequence of configurations  $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots\}$  that approximates a continuous trajectory in  $\mathbf{q}$  space, which generates the desired output trajectory  $\mathbf{z}_d(t)$ . The result of executing this algorithm to track the output trajectory  $\mathbf{z}_d(t)$  is shown in Fig. 9.16.3, as trajectories of  $q_i(t)$ . Trajectories shown as continuous lines are obtained by the proposed algorithm, while trajectories shown as dashed lines are the nominal  $q_i(t)$  that leads to penetration of the obstacle. Both trajectories coincide until approximately  $t = 0.45$ , after which collision occurs and link 2 remains in contact with the obstacle. Having the desired  $\mathbf{q}(t_i)$  trajectory, Eq. (9.16.29) is solved to obtain the desired input trajectory  $y(t_i)$ .

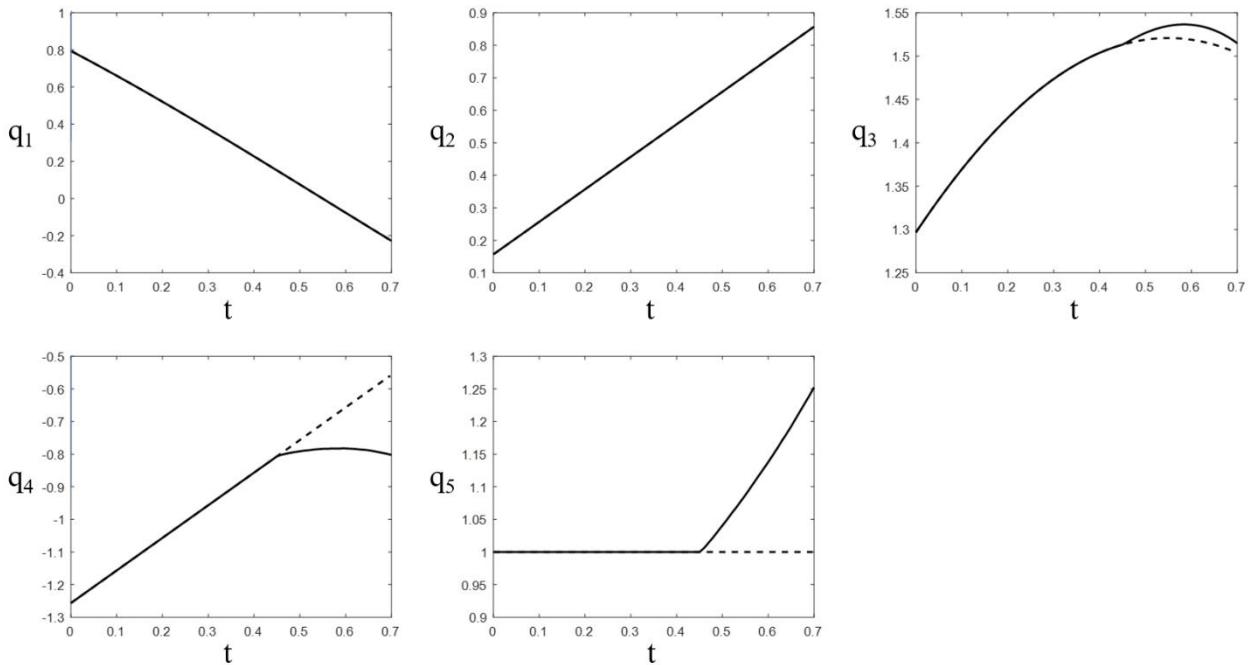


Figure 9.16.3 Evolution of  $\mathbf{q}$  for the Nominal Trajectory (dashed line) and the Collision-Free Trajectory Obtained by the Foregoing Algorithm (continuous line).

The average CPU time required to run the foregoing algorithm and obtain the collision-free trajectory was 0.1 sec, using the same computer used for the 7R manipulator of Section 2.7. The algorithm was run with the following parameters:

- (a) Maximum number  $i_{\max}$  of steps marching along self-motion manifold when trying to find an interference-free configuration: 50
- (b) Tolerance Tol to consider the sequence  $\mathbf{q}_k$  as a continuous trajectory: 0.05
- (c) Step size on the interval  $0 \leq t_i \leq 0.7 : 0.01$
- (d) Step h used to map the self-motion manifold: 0.01
- (e) Tolerance used in the iterative computation of  $\mathbf{h}$  and  $\mathbf{B}$  according to Appendix 9.A, during mapping of the manifolds, according to Section 3.6: 0.0001.
- (f) Maximum number of non-converging iterations to compute  $\mathbf{h}$  or  $\mathbf{B}$  before reevaluating  $\mathbf{U}$  and  $\mathbf{V}$  and resetting the parameterization of self-motion manifolds ( $v^0 = 0$ ) according to Section 9.16.6: 10 iterations.

### 9.16.9 Compound Manipulator Inverse Dynamics

To extend the foregoing to the dynamics domain, the time derivative of Eq. (9.16.18), suppressing arguments, is  $\dot{\mathbf{q}} = \mathbf{V}\dot{\mathbf{v}} - \mathbf{U}(\mathbf{h}, \dot{\mathbf{v}} + \mathbf{h}\dot{\mathbf{z}})$ . Substituting from Eqs. (9.16.23),

$$\dot{\mathbf{q}} = \mathbf{V}\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z})\left(\Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z})\mathbf{V}\dot{\mathbf{v}} + \Lambda_z(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z})\dot{\mathbf{z}}\right) = \mathbf{D}(\mathbf{v}, \mathbf{z})\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z})\Lambda_z(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z})\dot{\mathbf{z}} \quad (9.16.35)$$

where  $\mathbf{D}(\mathbf{v}, \mathbf{z}) \equiv (\mathbf{I} - \mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z})\Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z}))\mathbf{V}$ . From Eqs. (9.16.12), (9.16.18), and (9.16.35),  $\dot{\mathbf{y}} = \mathbf{g}'(\mathbf{q})\dot{\mathbf{q}} = \mathbf{g}'(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{h}(\mathbf{v}, \mathbf{z}))(\mathbf{D}(\mathbf{v}, \mathbf{z})\dot{\mathbf{v}} - \mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z})\Lambda_z(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z})\dot{\mathbf{z}})$ , where  $\mathbf{g}'(\mathbf{q})$  is given in Eq. (9.16.25). This is a  $2(n - m)$  parameter inverse kinematic velocity mapping, for arbitrary  $\mathbf{v}$  and  $\dot{\mathbf{v}}$  and associated  $\mathbf{z}$  and  $\dot{\mathbf{z}}$ . The redundant manipulator literature that works only at the velocity level misses half of these parameters; namely  $\mathbf{v}$ .

Differentiating Eq. (9.16.35) with respect to time,

$$\ddot{\mathbf{q}} = \mathbf{D}\ddot{\mathbf{v}} - \mathbf{U}\mathbf{B}\Lambda_z\ddot{\mathbf{z}} - \mathbf{U}\left(\mathbf{B}(\mathbf{v}, \ddot{\mathbf{z}})\left(\ddot{\Lambda}_q\mathbf{V}\ddot{\mathbf{v}} + \ddot{\Lambda}_z\ddot{\mathbf{z}}\right)\right)_v \dot{\mathbf{v}} - \mathbf{U}\left(\mathbf{B}(\ddot{\mathbf{v}}, \mathbf{z})\left(\ddot{\Lambda}_q\mathbf{V}\ddot{\mathbf{v}} + \ddot{\Lambda}_z\ddot{\mathbf{z}}\right)\right)_z \dot{\mathbf{z}} - \mathbf{U}\mathbf{B}\left(\Lambda_q(\mathbf{q}, \ddot{\mathbf{z}})\mathbf{V}\ddot{\mathbf{v}} + \Lambda_z(\mathbf{q}, \ddot{\mathbf{z}})\ddot{\mathbf{z}}\right)_q \dot{\mathbf{q}} - \mathbf{U}\mathbf{B}\left(\Lambda_q(\ddot{\mathbf{q}}, \mathbf{z})\mathbf{V}\ddot{\mathbf{v}} + \Lambda_z(\ddot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{z}}\right)_z \dot{\mathbf{z}} \quad (9.16.36)$$

To obtain computable expressions for terms of the form  $(\mathbf{B}(\mathbf{v}, \ddot{\mathbf{z}})\ddot{\mathbf{a}})_v$  and  $(\mathbf{B}(\ddot{\mathbf{v}}, \mathbf{z})\ddot{\mathbf{a}})_z$ , Eq. (9.16.22) is written as  $\Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z})\mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z}) = \mathbf{I}$  and multiplied on the right by a constant vector  $\mathbf{a}$ , yielding  $\Lambda_q(\mathbf{q}(\mathbf{v}, \mathbf{z}), \mathbf{z})\mathbf{U}\mathbf{B}(\mathbf{v}, \mathbf{z})\mathbf{a} = \mathbf{a}$ . Differentiating this equation with respect to  $\mathbf{v}$  and  $\mathbf{z}$ ,

$$(\Lambda_q(\mathbf{q}, \ddot{\mathbf{z}})\ddot{\mathbf{b}})_q \mathbf{q}_v + \Lambda_q \mathbf{U}(\mathbf{B}(\mathbf{v}, \ddot{\mathbf{z}})\ddot{\mathbf{a}})_v = \mathbf{0} \text{ and } (\Lambda_q(\mathbf{q}, \ddot{\mathbf{z}})\ddot{\mathbf{b}})_q \mathbf{q}_z + (\Lambda_q(\ddot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{b}})_z + \Lambda_q \mathbf{U}(\mathbf{B}(\ddot{\mathbf{v}}, \mathbf{z})\mathbf{a})_z = \mathbf{0},$$

where  $\mathbf{b} = \mathbf{U}\mathbf{B}\mathbf{a}$ . Using Eqs. (9.16.22), (9.16.18), and (9.16.23),

$$(\mathbf{B}(\mathbf{v}, \ddot{\mathbf{z}})\ddot{\mathbf{a}})_v = -\mathbf{B}(\Lambda_q(\mathbf{q}, \ddot{\mathbf{z}})\ddot{\mathbf{b}})_q (\mathbf{I} - \mathbf{U}\mathbf{B}\Lambda_q)\mathbf{V} \text{ and}$$

$$(\mathbf{B}(\ddot{\mathbf{v}}, \mathbf{z})\ddot{\mathbf{a}})_z = -\mathbf{B}\left\{-(\Lambda_q(\mathbf{q}, \ddot{\mathbf{z}})\ddot{\mathbf{b}})_q \mathbf{U}\mathbf{B}\Lambda_z + (\Lambda_q(\ddot{\mathbf{q}}, \mathbf{z})\ddot{\mathbf{b}})_z\right\}.$$

Substituting these results into Eq. (9.16.36) yields an intricate, but computable, expression for

$$\ddot{\mathbf{q}} = \mathbf{Q}_{dd}(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}, \mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}) \quad (9.16.37)$$

for arbitrary values of  $\mathbf{v}, \dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$ . Finally, differentiating  $\dot{\mathbf{y}} = \mathbf{g}'(\mathbf{q})\dot{\mathbf{q}}$  yields

$$\ddot{\mathbf{y}} = \mathbf{g}'(\mathbf{q})\ddot{\mathbf{q}} + \left( \mathbf{g}'(\mathbf{q})\ddot{\mathbf{q}} \right)_{\mathbf{q}} \dot{\mathbf{q}} \quad (9.16.38)$$

which, with  $\ddot{\mathbf{q}} = \mathbf{Q}_{dd}(\mathbf{v}, \dot{\mathbf{v}}, \mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}})$ , defines the input coordinate acceleration, with  $\mathbf{v}$ ,  $\dot{\mathbf{v}}$ , and  $\ddot{\mathbf{v}}$  arbitrary. Evaluation of the second term on the right of Eq. (9.16.38) requires some manipulation. From Eq. (9.16.25),  $\mathbf{g}'(\mathbf{q}) = -\Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q})\Psi_q(\mathbf{g}(\mathbf{q}), \mathbf{q})$  and

$$\left( \mathbf{g}'(\mathbf{q})\ddot{\mathbf{q}} \right)_{\mathbf{q}} = -\left( \Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q})\Psi_q\ddot{\mathbf{q}} \right)_{\mathbf{q}} - \Psi_y^{-1} \left[ \left( \Psi_q(\mathbf{y}, \dot{\mathbf{q}})\ddot{\mathbf{q}} \right)_y \mathbf{g}'(\mathbf{q}) + \left( \Psi_q(\mathbf{y}, \dot{\mathbf{q}})\ddot{\mathbf{q}} \right)_{\mathbf{q}} \right]_{\mathbf{y}=\mathbf{g}(\mathbf{q})} \quad (9.16.39)$$

Differentiating the identity  $\Psi_y(\mathbf{g}(\mathbf{q}), \mathbf{q})\Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q})\ddot{\mathbf{a}} = \ddot{\mathbf{a}}$  with respect to  $\mathbf{q}$ ,

$$\left[ \left( \Psi_y(\mathbf{y}, \dot{\mathbf{q}})\ddot{\mathbf{b}} \right)_y \mathbf{g}'(\mathbf{q}) + \left( \Psi_y(\mathbf{y}, \dot{\mathbf{q}})\ddot{\mathbf{b}} \right)_{\mathbf{q}} \right]_{\mathbf{y}=\mathbf{g}(\mathbf{q})} + \Psi_y(\mathbf{g}(\mathbf{q}), \mathbf{q}) \left( \Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q})\ddot{\mathbf{a}} \right)_{\mathbf{q}} = \mathbf{0} \quad (9.16.40)$$

where  $\mathbf{a} = \Psi_q(\mathbf{g}(\mathbf{q}), \mathbf{q})\dot{\mathbf{q}}$  and  $\mathbf{b} = \Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q})\mathbf{a}$ . Thus,

$$\left( \Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q})\Psi_q\ddot{\mathbf{q}} \right)_{\mathbf{q}} = -\Psi_y^{-1}(\mathbf{g}(\mathbf{q}), \mathbf{q}) \left[ \left( \Psi_y(\mathbf{y}, \dot{\mathbf{q}})\ddot{\mathbf{b}} \right)_y \mathbf{g}'(\mathbf{q}) + \left( \Psi_y(\mathbf{y}, \dot{\mathbf{q}})\ddot{\mathbf{b}} \right)_{\mathbf{q}} \right]_{\mathbf{y}=\mathbf{g}(\mathbf{q})} \quad (9.16.41)$$

and, with Eq. (9.16.39), the inverse acceleration mapping of Eq. (9.16.37) is evaluated

## Appendix 9.A: Computation of $\mathbf{h}(\mathbf{v}, \mathbf{z})$ and $\mathbf{B}(\mathbf{v}, \mathbf{z})$

While vector and matrix functions  $\mathbf{h}(\mathbf{v}, \mathbf{z})$  and  $\mathbf{B}(\mathbf{v}, \mathbf{z})$  for compound manipulators are shown to exist and be differentiable functions of  $\mathbf{v}$  and  $\mathbf{z}$ , the derivation does not show how to evaluate them. Since they are central to implementing inverse kinematic, velocity, and acceleration analysis in Section 9.16.9, numerical methods for their evaluation are needed.

At  $\bar{\mathbf{x}} \in \tilde{X}^c$ ,  $\mathbf{B}(\bar{\mathbf{v}}, \bar{\mathbf{z}}) = (\Lambda_q(\mathbf{q}(\bar{\mathbf{v}}, \bar{\mathbf{z}}), \bar{\mathbf{z}})\mathbf{U})^{-1} = (\mathbf{U}^T\mathbf{U})^{-1}$  is numerically evaluated. For  $(\mathbf{v}^i, \mathbf{z}^i)$  at time  $t^i$ ,  $\mathbf{B}(\mathbf{v}^i, \mathbf{z}^i)$  must satisfy Eq. (9.16.22), in the form

$\bar{\mathbf{R}} \equiv (\Lambda_q(\mathbf{q}(\mathbf{v}^i, \mathbf{z}^i), \mathbf{z}^i)\mathbf{U})\mathbf{B}(\mathbf{v}^i, \mathbf{z}^i) - \mathbf{I} = \mathbf{0}$ . With an approximation  $\mathbf{B}^{(1)} \approx \mathbf{B}(\mathbf{v}^{i-1}, \mathbf{z}^{i-1})$  of the solution and suppressing arguments  $(\mathbf{v}^i, \mathbf{z}^i)$  since they do not change in the iterative process for  $\mathbf{B}(\mathbf{v}^i, \mathbf{z}^i)$ , the *matrix version of Newton-Raphson iteration* is defined by

$(\Lambda_q\mathbf{U})\Delta\mathbf{B}^{(j)} = -\bar{\mathbf{R}}^{(j)} = -\Lambda_q\mathbf{U}\mathbf{B}^{(j)} + \mathbf{I}$ , where  $(j)$  denotes iteration number. Since the matrix  $\Lambda_q\mathbf{U}$  need not be inverted with great precision for use in the Newton-Raphson process (Atkinson, 1989) and  $\mathbf{B}^{(j)} \approx (\Lambda_q\mathbf{U})^{-1}$ ,  $\Delta\mathbf{B}^{(j)} = -\mathbf{B}^{(j)}\Lambda_q\mathbf{U}\mathbf{B}^{(j)} + \mathbf{B}^{(j)}$  and  $\mathbf{B}^{(j+1)} = \mathbf{B}^{(j)} + \Delta\mathbf{B}^{(j)}$ . This yields the iterative algorithm

$$\mathbf{B}^{(j+1)} = 2\mathbf{B}^{(j)} - \mathbf{B}^{(j)}\Lambda_q\mathbf{U}\mathbf{B}^{(j)}, \quad j=1, 2, \dots, \text{until } \|\Lambda_q\mathbf{U}\mathbf{B}^{(j+1)} - \mathbf{I}\| \leq \text{Btol} \quad (9.A.1)$$

where Btol is a specified error tolerance. This is an efficient computation, requiring only matrix multiplication.

While  $\mathbf{h}(\mathbf{v}^i, \mathbf{z}^i)$  cannot be analytically determined, it can be evaluated as accurately as desired using Newton-Raphson iteration to solve Eq. (9.16.17) for  $\mathbf{u} = \mathbf{h}(\mathbf{v}^i, \mathbf{z}^i)$ , with

$\Lambda_u\Delta\mathbf{u}^{(j)} = -\Lambda_q\mathbf{U}\Delta\mathbf{u}^{(j)} = -\mathbf{B}^{-1}\Delta\mathbf{u}^{(j)} = -\Lambda(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^{(j)}, \mathbf{z})$ . The solution is

$\Delta\mathbf{u}^{(j)} = \mathbf{B}\Lambda(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^{(j)}, \mathbf{z})$  and  $\mathbf{u}^{(j+1)} = \mathbf{u}^{(j)} + \Delta\mathbf{u}^{(j)}$ . This yields the iterative algorithm

$$\mathbf{u}^{(j+1)} = \mathbf{u}^{(j)} + \mathbf{B}\Lambda(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^{(j)}, \mathbf{z}), \quad j=1, 2, \dots, \text{until } \|\Lambda(\bar{\mathbf{q}} + \mathbf{V}\mathbf{v} - \mathbf{U}\mathbf{u}^{(j+1)}, \mathbf{z})\| \leq \text{utol} \quad (9.A.2)$$

where utol is a specified error tolerance. Since the Newton-Raphson method does not require an exact Jacobian, the matrix  $\mathbf{B}$  is held constant throughout the process. This is an efficient computation, requiring only matrix multiplication.

The foregoing computations may be adapted to the less complex case of kinematically redundant serial manipulators of Section 9.15 by changing  $\Lambda_q(\mathbf{q}, \mathbf{z})$  to  $\mathbf{G}'(\mathbf{y})$ .

## Bibliography

- Abraham, R, and Marsden, J.E., 1978, Foundations of Mechanics, Second Ed., Benjamin/Cummings, Reading, Mass.
- Adkins, F. A., and Haug, E. J., 1997, Operational Envelope of a Spatial Stewart Platform, Journal of Mechanical Design, Vol. 119, pp. 330-332.
- Anderson, K.S., and Hsu, Y., 2004, Order-(n+m) direct differentiation determination of design sensitivity for constrained multibody dynamic systems, Struct Multidisc Optim, vol. 26, pp 171-182.
- Arnold, M., 2017, DAE aspects of multibody system dynamics, In: A. Ilchmann, T. Reis (eds.): Surveys in Differential-Algebraic Equations IV, *Differential-Algebraic Equations Forum*, pp. 41-106. Springer International Publishing.
- Arnold, M., and Bruls, O., 2007, Convergence of the Generalized- $\alpha$  Scheme for Constrained Mechanical Systems, Multibody System Dynamics, Vol.18, No.2, pp.185-202.
- Arnold, V. I., 1978, Mathematical Methods of Classical Mechanics, Springer, New York.
- Ascher, U.M., and Petzold, L.R., 1998, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia.
- Atkinson, K.E., 1989, An Introduction to Numerical Analysis, Second Ed., Wiley, New York.
- Awrejcewicz, J. and Olejnik, p., 2005, Analysis of Dynamic Systems With Various Friction Laws, Applied Mechanics Reviews, Vol. 58, pp. 389-410.
- Banerjee, J., and McPhee, J, 2016, Graph-theoretic sensitivity analysis of multi-domain dynamic systems: theory and symbolic computer implementation, Nonlinear Dynamics, vol. 85, pp. 203-227.
- Banerjee, J., and McPhee, J, 2016, Graph-theoretic sensitivity analysis of multi-domain dynamic systems: theory and symbolic computer implementation, Nonlinear Dynamics, vol. 85, pp. 203-227.
- Bauchau, O.A., and Laulusa, A., 2008, Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems, Journal of Computational and Nonlinear Dynamics, Vol. 3, p. 0110005.
- Berger, E.J., 2002, Friction modeling for dynamic system simulation, Applied Mechanics Reviews, Vol. 55, No. 6, pp535-577.
- Bernstein, D.S., 2009, Matrix Mathematics, Second Ed., Princeton University Press, Princeton, NJ.
- Betsch, P., 2004, A Unified Approach to the Energy-Consistent Numerical Integration of Nonholonomic Mechanical Systems and Flexible Multibody Dynamics, GAMM-Mitt., Vol. 27, No. 1, p. 66.
- Bianchi, G., and Schiehlen, W. (eds.), 1986, Dynamics of Multibody Systems, Springer-Verlag, Berlin.
- Bloch, A.M., 2003, Nonholonomic Mechanics and Control, Springer, New York.

- Blumentals, A., Brogliato, B, and Bertails-Descoubes, F., 2016, The contact problem in Lagrangian systems subject to bilateral and unilateral constraints, with or without sliding Coulomb's friction: a tutorial, *Multibody System Dynamics*, Vol. 38, pp.43-76.
- Bohigas, O., Montserrat, M., and Ros, L, 2017, *Singularities of Robot Mechanisms*, Springer International Publishing, Switzerland.
- Borri, M., Bottasso, C., and Mantegazza, P., 1990, Equivalence of Kane's and Maggi's Equations, *Meccanica*, Vol. 25, No. 4, pp. 272-274.
- Borri, M., Bottasso, C., and Mantegazza, P., 1992, Acceleration Projection Method in Multibody Dynamics, *European Journal of Mechanics, A/Solids*, Vol. 11, No. 3.
- Brenan, K.E., Campbell, S.L., and Petzold, L.R., 1989, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North Holland, New York.
- Brown, P., and McPhee, J., 2016, A Continuous Velocity-Based Friction Model for Dynamics and Control with Physically Meaningful Parameters, *Journal of Computational and Nonlinear Dynamics*, Vol. 11.
- Bruls, O., and Cardona, A., 2010, On the Use of Lie Group Time Integrators in Multibody Dynamics, *Journal of Computational and Nonlinear Dynamics* Vol.5, No.3, pp.1-13.
- Bruls, O., Cardona, A., and Arnold, M., 2012, Lie group generalized- $\alpha$  time integration of constrained flexible multibody systems, *Mechanism and Machine Theory*, Vol. 48, pp.121-137.
- Bryson, A.E., and Ho, Y.C., 1975, *Applied Optimal Control*, Wiley, NY.
- Burdick, J.W., 1989, On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds, *Proceedings, 1989 IEEE Int. Conf. Robot Automation*, Scottsdale AZ, pp264-270.
- Callejo, A., and Garcia de Jalon, J., 2014, A Hybrid Direct-Automatic Differentiation Method for the Computation of Independent Sensitivities in Multibody Systems, *Int. J. Numer. Meth. Engrg.*, vol. 100, pp.933-952.
- Chablat, D., and Wenger, P., 1998, Working modes and aspects in fully parallel manipulators, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 1964-1969.
- Chang, K-H, and Joo, S-H, 2006, Design parameterization and tool integration for CAD-based mechanism optimization, *Advances in Engineering Software*, Vol. 37, pp. 779-796.
- Chiaverine, S., Oriolo, G., and Maciejewski, A.A., 2016, *Redundant Robots*, Springer Handbook on Robotics, pp. 221-242.
- Chung, J., and Hulbert, G.M., 1993, A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized- $\alpha$  Method, *Journal of Applied Mechanics*, Vol. 60, pp. 371-374.
- Coddington, E.A., and Levinson, N., 1955, *Theory of Ordinary Differential Equations*, McGraw-Hill, New York.
- Conconi, M., and Carricato, M., 2009, A New Assessment of Singularities of Parallel Kinematic Chains, *IEEE Transactions on Robotics*, Vol. 25, No. 4, pp. 757-770.

- Corwin, L.J., and Szczerba, R.H., 1982, Multivariable Calculus, Marcel Dekker, New York.
- Coste, M., 2012, A Simple Proof That Generic 3-RPR Manipulators Have Two Aspects, *J. of Mechanisms and Robotics*, Vol. 4, pp. 01108, 1-6.
- Coste, M., Wegner, P., and Chablat, D., 2018, Hidden Cusps, Advances in Robot Kinematic 2016, Lenarcic and Merlet (eds.), pp. 129-138, Springer Proceedings in Advanced Robotics , Springer International Publishing AG, Cham, Switzerland.
- d'Alembert, J. le R., 1743, *Traite de dynamique*, Paris.
- DeMers, D., and Kreutz-Delgado, K., 1996, Canonical Parameterization of Excess Motor Degrees of Freedom with Self-Organizing Maps, *IEEE Transactions on Neural Networks*, Vol. 7, No. 1, pp. 43-55.
- Dontchev, A.L. and Rockafellar, R.T., 2014, Implicit Functions and Solution Mappings, Second Ed., Springer, New York.
- Dopico, D., Gonzalez, F., Luaces, A., Saura, M., and Garcia-Vallejo, D., 2018, Direct sensitivity analysis of multibody systems with holonomic and nonholonomic constraints via an index-3 augmented Lagrangian formulation with projections, *Nonlinear Dynamics*, Vol. 93, pp. 2039-2056.
- Dopico, D., Zhu, Y., Sandu, A., and Sandu, C., 2015, Direct and adjoint sensitivity analysis of Ordinary differential equation multibody formulations, *Journal of Computational and Nonlinear Dynamics*, vol. 10, pp. 011012-1-7.
- Dundas, B.I., 2018, *A Short Course in Differential Topology*, Cambridge University Press, Cambridge, UK.
- Eich-Soellner, E., and Fuhrer, K., 1998, *Numerical Methods in Multibody Dynamics*, B. G. Teubner, Stuttgart.
- Euler, L., 1736, *Mechanica, sive motus sciuntia analytice exposita*, St. Petersburg.
- Fari, C., Ferreira, F., Erlhagen, W., Monteiro, S., and Bicho, E., 2018, Position-based kinematics for 7-DOF serial manipulators with global configuration control, joint limit, and singularity avoidance, *Mechanism and Machine Theory*, Vol. 121, pp. 317-334.
- Featherstone, R., 2008, *Rigid Body Dynamics Algorithms*, Springer, New York.
- Featherstone, R., and Orin, D.E., 2016, Dynamics, *Springer Handbook of Robotics*, Second Ed. Springer-Verlag, Berlin.
- Frank, P.M., 1978, *Introduction to System Sensitivity Theory*, Academic Press, NY.
- Garcia de Jalon, J., Callejo, A., and Hidalgo, A.F., 2012, Efficient Solution of Maggi's Equations, *J. of Computational and Nonlinear Dynamics*, Vol 7, p. 021003.
- Garcia-Naranjo, L.C., and Marrero, J.C., 2013, Non-Existence of an Invariant Measure for a Homogeneous Ellipsoid Rolling on the Plane, *Regular and Chaotic Dynamics*, Vol. 18, No.4, p. 372.
- Gear, C.W., and Petzold, L.R., 1984, ODE Methods for the Solution of Differential/Algebraic Systems, *SIAM Journal of Numerical Analysis*, Vol. 21, No. 4, pp. 716-728.

- German, H.C., 2006, Computer-Aided Design Sensitivity Analysis for Dynamic Multibody Systems, Ph.D. Thesis, Department of Mechanical Engineering, The University of Iowa.
- Ghosal, A., and Ravani, B., 2001, A Differential-Geometric Analysis of Singularities of Point Trajectories of Serial and Parallel Manipulators, *Journal of Mechanical Design*, Vol. 123, pp. 80-89.
- Goldstein, H., 1980, Classical Mechanics, Second Ed., Addison-Wesley, Reading, Mass.
- Gosselin, C., and Angeles, J., 1990, Singularity Analysis of Closed-Loop Kinematic Chains, *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 3, pp. 281-290.
- Gosselin, C., and Schreiber, L.-T., 2018, Redundancy in Parallel Mechanisms: A Review, *Applied Mechanics Reviews*, Vol. 70, p. 010802.
- Guillemin, V., and Pollack, A., 1974, Differential Topology, AMS Chelsea Publishing, Providence, RI.
- Gutierrez-Lopez, M.D., Callejo, A., Garcia de Jalon, J., 2012, Computation of Independent Sensitivities Using Maggi's Formulation, 2nd Joint International Conference on Multibody System Dynamics, Stuttgart.
- Hadamard, J., 1902, Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*. pp. 49–52.
- Hairer, E., and Wanner, G., 1996, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Second Ed., Springer-Verlag, Berlin.
- Hairer, E., Lubich, C., and Wanner, G., 2006, Geometric Numerical Integration, Second Ed., Springer-Verlag, Berlin.
- Hairer, E., Norsett, S.P., and Wanner, G., 1993, Solving Ordinary Differential Equations I: Nonstiff Problems, Second Ed., Springer-Verlag, Berlin.
- Hart, W.L., 1957, Analytical Geometry and Calculus, Heath, Boston.
- Haug, E. J., Choi, K. K., Kuhl, J. K., and Wargo, J. D., 1995, Virtual Prototyping Simulation for Design of Mechanical Systems, *Journal of Mechanical Design*, Vol. 117, pp. 63-70.
- Haug, E. J., Wang, J. T., and Wu, J. K., 1992, Dexterous Workspaces of Manipulators, Part I: Analytical Criteria, *Mechanics of Structures and Machines*, Vol. 20, No. 3, pp. 321-361.
- Haug, E.J., 1984, Computer Aided Analysis and Optimization of Mechanical System Dynamics, Springer-Verlag, Berlin.
- Haug, E.J., 1987, Design Sensitivity Analysis of Dynamic Systems, Mota Soares, C.A. (ed.), Computer Aided Optimal Design: Structural and Mechanical Systems, Springer, Berlin, pp. 705–755.
- Haug, E.J., 1989, Computer-Aided Kinematics and Dynamics of Mechanical Systems, Vol. I: Basic Methods, Allyn and Bacon, Boston.
- Haug, E.J., 1992, Intermediate Dynamics, Prentice-Hall, Englewood Cliffs, NJ.
- Haug, E.J., 2016, An Ordinary Differential Equation Formulation for Multibody Dynamics: Holonomic Constraints, *Journal of Computing and Information Science in Engineering*, Vol. 16, No. 2, p. 021007.

- Haug, E.J., 2017a, An Ordinary Differential Equation Formulation for Multibody Dynamics: Nonholonomic Constraints, *Journal of Computing and Information Science in Engineering*, Vol. 17, No. 1, p 0110009.
- Haug, E.J., 2017b, An Index 0 Differential-Algebraic Equation Formulation for Multibody Dynamics-Holonomic Constraints, *Mechanics Based Design of Structures and Machines*, Vol. 45, No. 4, pp. 479-506.
- Haug, E.J., 2018a, An Index 0 Differential-Algebraic Equation Formulation for Multibody Dynamics-Nonholonomic Constraints, *Mechanics Based Design of Structures and Machines*, Vol. 46, No. 1, pp. 38-65.
- Haug, E.J., 2018b, Simulation of Friction and Stiction in Multibody Dynamics Model Problems, *Mechanics Based Design of Structures and Machines*, Vol. 46, No. 3, pp. 296-317.
- Haug, E.J., 2018c, Simulation of Spatial Multibody Systems with Friction, *Mechanics Based Design of Structures and Machines*, Vol. 46, No.3, pp.347-375.
- Haug, E.J., 2018d, Extension of Maggi and Kane Equations to Holonomic Dynamic Systems, *Journal of Computational and Nonlinear Dynamics*, Vol. 13, pp.121003.
- Haug, E.J., 2020a, Well-posed formulation of holonomic mechanical dynamics and design sensitivity analysis, *Mechanics Based Design of Structures and Machines*, Vol 48, No 1, pp. 111-121.
- Haug, E.J., 2020b, Well-Posed Formulations for Nonholonomic Mechanical System Dynamics, *Journal of Computational and Nonlinear Dynamics*, Vol. 15, p. 101003.
- Haug, E.J., 2021a, Multibody Dynamics on Differentiable Manifolds, *Journal of Computational and Nonlinear Dynamics*, Vol. 16, p. 041003.
- Haug, E.J., 2021b, Parallel Manipulator Domains of Singularity Free Functionality, *Mechanics Based Design of Structures and Machines*, Vol. 49, No. 5, pp 615-639.
- Haug, E.J., 2021c, Parallel Manipulator Dynamics Embedded in Singularity Free Domains of Functionality, *Mechanics Based Design of Structures and Machines*, Vol. 49, No. 6, pp 763-780.
- Haug, E.J., 2022a, Manipulator Kinematics and Dynamics on Differentiable Manifolds: Part I Kinematics, *Journal of Computational and Nonlinear Dynamics*, Vol. 17, p. 021002.
- Haug, E.J., 2022b, Manipulator Kinematics and Dynamics on Differentiable Manifolds: Part II Dynamics, *Journal of Computational and Nonlinear Dynamics*, Vol. 17, p. 021003.
- Haug, E.J., and Arora, J.S., 1979, *Applied Optimal Design*, Wiley, NY.
- Haug, E.J., and Peidro, A., 2022, Redundant Manipulator Kinematics and Dynamics on Differentiable Manifolds, *Journal of Computational and Nonlinear Dynamics*, in review.
- Haug, E.J., and Yen, J., 1992, Implicit Numerical Integration of Constrained Equations of Motion Via Generalized Coordinate Partitioning, *Journal of Mechanical Design*, Vol. 114, pp. 296-304.
- Haug, E.J., Luh, C.M., Adkins, F.A., and Wang, J., 1996, Numerical Algorithms for Mapping Boundaries of Manipulator Workspaces, *Journal of Mechanical Design*, Vol. 118, pp. 228-234.

- Haug, E.J., Mani, N.K., and Krishnaswami, P., 1984, Design Sensitivity Analysis and Optimization of Dynamically Driven Systems, Computer Aided Analysis and Optimization of Mechanical System Dynamics, E.J. Haug (Ed.), Springer-Verlag, Heidelberg, pp. 555-636.
- Haug, E.J., Negrut, D., and Engstler, C., 1999, Implicit Runge-Kutta Integration of the Equations of Multibody Dynamics in Descriptor Form, Mechanics of Structures and Machines, Vol. 27, No. 3, pp. 337-364.
- Haug, E.J., Wu, S.M., and Yang, S.M., 1986, Dynamics of Mechanical Systems with Coulomb Friction, Stiction, Impact, and Constraint Addition and Deletion-I Theory, Mechanism and Machine Theory, Vol. 21, No. 5, pp. 401-406.
- Hilber, H.M., Hughes, T.J.R., and Taylor, R.L., 1977, Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics, Earthquake Engineering and Structural Dynamics, Vol. 5, No. 3, pp. 283-292.
- Hirsch, M.W., 1976, Differential Topology, Springer-Verlag, New York.
- Hopf, H., 1940, Systeme symmetrischer Bilinearformen und euklidsche Modelle der projektiven Raume, Vierteljschr. Naturforsch. Ges. Zurich.
- Husty, M.L., 2009, Non-singular assembly mode change in 3-RPR-parallel manipulators, Computational Kinematics, Springer-Verlag, pp. 51-60.
- Innocenti, C., and Parenti-Castelli, V. 1998, Singularity-Free Evolution from One Configuration to Another in Serial and Fully-Parallel Manipulators, Journal of Mechanical Design, Vol. 120, No. 1, pp. 73-79.
- Kane, T.R., and Levinson, D.A., 1985, Dynamics: Theory and Applications, McGraw-Hill, New York.
- Kato, T., 1980, Perturbation Theory for Linear Operators, Second Ed., Springer Verlag, New York.
- Kelley, C.T., 1995, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia.
- Kreyszig, E., 2011, Advanced Engineering Mathematics, Eighth Ed., Wiley, New York.
- Lagrange, J.L., 1788, Mechanique Analytique, Paris.
- Lanczos, C., 1962, The Variational Principles of Mechanics, Second Ed., University of Toronto Press, Toronto.
- Laulusa, A., and Bauchau, O.A., 2008, Review of Classical Approaches for Constraint Enforcement in Multibody Systems, Journal of Computational and Nonlinear Dynamics, Vol. 3, p. 011004.
- Lee, H. Y., and Liang, C. G., 1988, Displacement analysis of the general spatial 7-link 7R mechanism. Mechanism and machine theory, Vol. 23, No. 3, pp.219-226.
- Lee, J.M., 2010, Introduction to Topological Manifolds, Second Ed., Springer, New York.
- Lee, J.M., 2013, Introduction to Smooth Manifolds, Second Ed., Springer, New York.
- Lee, S., and Bejczy, A.K., 1991, Redundant Arm Kinematic Control Based on Parameterization, Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation, Sacramento, CA, pp. 458-465.

- Li, H., Gosselin, C.M., and Richard, M.J., 2006, Determination of maximal singularity-free zones in the workspace of planar three-degree-of-freedom parallel mechanisms, *Mechanism and Machine Theory*, Vol. 41, pp. 1157-1167.
- Liang, C.G., and Lance, G.M., 1987, A Differentiable Null Space Method for Constrained Dynamic Analysis, *Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 109, pp. 405-411.
- Lovelock, D., and Rund, H., 1975, *Tensors, Differential Forms, and Variational Principles*, Wiley, New York.
- Luces, M., Mills, J.K., and Benhabib, B., 2017, A Review of Redundant Parallel Kinematic Manipulators, *J Intel Robot Sys*, Vol. 86, pp. 175-198.
- Luck, C.L., and Lee, S., 1995, Topology-Based Analysis for Redundant Manipulators under Kinematic Constraints, *Proceedings, 34<sup>th</sup> IEEE Conf. on Design and Control*, New Orleans, pp.1603-1608.
- Luh, C.M., Adkins, F.A., Haug, E.J., and Qiu, C.C., 1996, Working Capability Analysis of Stewart Platforms, *Journal of Mechanical Design*, Vol. 118, pp. 220-227.
- Lynch, K.M., and Park, F.C., 2017, *Modern Robotics*, Cambridge University Press, Cambridge, UK.
- Maggi, G.A., 1896, *Principii della Teoria Matematica del Movimento dei Corpi: Corso de Meccanica Razionale*, Ulrico Hoepli, Milano.
- Maggi, G.A., 1901, *Di Alcune Nuove Forme delle Equazioni della Dinamica Applicabili ai Sistemi Anolonomi*, Rendiconti della Regia Academia dei Lincei, Serie V, Vol. X, pp. 287-291.
- Magnus, K. (ed.), 1978, *Dynamics of Multibody Systems*, Springer-Verlag, Berlin. (Symposium Munich/Germany August 29–September 3, 1977).
- Marques, F, Flores, P., Pimenta Claro, J.C., and Lankarani, H.M., 2016, A survey and comparison of several friction force models for dynamic analysis of multibody mechanical systems, *Nonlinear Dynamics*, Vol. 86, pp.1407–1443.
- Marques, F, Flores, P., Pimenta Claro, J.C., and Lankarani, H.M., 2019, Modeling and analysis of friction including rolling effects in multibody dynamics: a review, *Multibody Sus Dyn*, Vol. 45, pp.223-244.
- Marques, F., Flores, P., and Lankarani, H.M., 2021, An investigation of a novel LuGre-based friction force model, *Mechanism and Machine Theory*, Vol. 166, p. 104493.
- Mendelson, B., 1962, *Introduction to Topology*, Allyn and Bacon, Boston.
- Merlet, J.-P., 2006, *Parallel Robots*, Second Ed., Springer, The Netherlands.
- Merlet, J.-P., Gosselin, C., and Huang, T., 2016, *Parallel Mechanisms*, Springer Handbook of Robotics, Second Ed. Springer-Verlag, Berlin.
- Muller, A, and Hufnagel, T, 2012, Model-Based Control of Redundantly Actuated parallel manipulators in Redundant Coordinates, *Rob. Auton. Sys.*, Vol 64, No. 4, pp 563-571.
- Muller, A. and Terze, Z, 2016, Geometric methods and formulations in computational multibody system dynamics, *Acta Mech*, Vol 227, pp. 3327-3350.

- Muller, A., and Zlatanov, D., eds., 2019, Singular configurations of Mechanisms and Manipulators, Springer, Cham, Switzerland.
- Murray, M., Li, Z., and Sastry, S.S., 1994, A Mathematical Introduction to Robotic Manipulation, CRC Press, Boca Raton, FL.
- Negrut, D., Jay, L.O., and Khude, N., 2009, A Discussion of Low-Order Numerical Integration Formulas for Rigid and Flexible Multibody Dynamics, Journal of Computational and Nonlinear Dynamics, Vol. 4, pp. 121008 1.
- Negrut, D., Haug, E.J., and German, H.C., 2003, An Implicit Runge-Kutta Method for Integration of Differential-Algebraic Equations of Multibody Dynamics, Multibody System Dynamics, Vol 9, pp 121-142.
- Negrut, D., Rampalli, R., Ottersson, G., and Sajdak, A., 2007, On an Implementation of the Hilber-Hughes-Taylor Method in the Context of Index 3 Differential-Algebraic Equations of Multibody Dynamics, Journal of Computational and Nonlinear Dynamics, Vol. 2, pp. 73-85.
- Negrut, D., Sandu, A., Haug, E. J., Potra,F., and Sandu, C., A., 2003, Rosenbrock-Nystrom State Space Implicit Approach for the Dynamic Analysis of Mechanical Systems: II - The Method and Numerical Examples, Multibody System Dynamics, Vol. 217, No. 4, pp. 273-281.
- Neimark, J.I., and Fufaev, N.A., 1972, Dynamics of Nonholonomic Systems, American Mathematical Society, Providence, Rhode Island.
- Newmark, N.M., 1959, A Method of Computation for Structural Dynamics, Journal of the Engineering Mechanics Division, ASCE, Vol. 85, No. EM3, pp. 67-94.
- Newton, I., 1687, Philosophiae naturalis principia mathematica, London.
- Orlandea, N., Chace, M.A., and Calahan, D.A., 1977, A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems, Parts I and II, Journal of Engineering for Industry, Vol. 99, pp. 773-784.
- Palais, B., Palais, R., and Rodi, S., 2009, A Disorienting Look at Euler's Theorem on the Axis of Rotation, American Mathematical Monthly, Vol. 116, No. 10, pp. 892-909.
- Papastavridis, J.G., 1990, The Maggi or Canonical Form of Lagrange's Equations of Motion of Holonomic Mechanical Systems, J. of Applied Mechanics, Vol. 57, pp. 1004-1010.
- Pars, L.A., 1965, A Treatise on Analytical Dynamics, Reprint by Ox Bow Press (1979), Woodbridge, Conn.
- Peidro, A., Marin, J.M., Gil, A., and Reinoso, O., 2015, Performing nonsingular Transitions Between Assembly Modes in Analytical Parallel Manipulators by Enclosing Quadruple Solutions, J. of Mechanical Design, Vol. 137, p.122302.
- Peidro, A., Reinoso, O., Gil, A., Marín, J. M., and Paya, L., 2018. A method based on the vanishing of self-motion manifolds to determine the collision-free workspace of redundant robots, Mechanism and Machine Theory, Vol. 128, pp. 84-109.
- Pennestrì, E., Rossi, V., Salvini, P, Valentini, P.P., 2016, Review and comparison of dry friction force models, Nonlinear Dynamics, Vol 83, pp. 1785-1801.
- Petzold, L.D., 1982, Differential/Algebraic Equations are not ODE's, SIAM Journal of Scientific and Statistical Computing, Vol. 3, No. 3, pp. 367-384.

- Petzold, L.R., 1986, Order Results for Implicit Runge-Kutta Methods Applied to Differential/Algebraic Systems, Vol. 23, No.4, pp. 837-853.
- Potra, F.A., and Rheinboldt, W.C., 1991, On the Numerical Solution of the Euler-Lagrange Equations, Mechanics of Structures and Machines, Vol. 19, No. 1, pp. 1-18.
- Qiu, C. C., Luh, C. M., and Haug, E. J., 1995, Dexterous Workspaces of Manipulators, Part III: Calculation of Continuation Curves at Bifurcation Point, Mechanics of Structures and Machines, Vol. 23, No. 1, pp. 115-130.
- Rabier, P.J., and Rheinboldt, W.C., 2000, Nonholonomic Motion of Rigid Mechanical Systems from a DAE Viewpoint, SIAM, Philadelphia.
- Rabier, P.J., and Rheinboldt, W.C., 2002, Theoretical and Numerical Analysis of Differential-Algebraic Equations, Handbook of Numerical Analysis, Vol VIII (ed. P.G. Ciarlet and J.L. Lions), Elsevier Science B.V., Amsterdam, pp. 183-540.
- Rheinboldt, W.C., 1991, On the Existence and Uniqueness of Solutions of Nonlinear Semi-Implicit Differential-Algebraic Equations, Numerical Analysis, Theory, Methods, & Applications, Vol 16, No. 7/8, pp. 647-661.
- Saha, S.K., and Angeles, J., 1991, Dynamics of Nonholonomic Mechanical Systems Using a Natural Orthogonal Complement, ASME Journal of Applied Mechanics, Vol. 58, pp.238-243.
- Schiehlen, W. (ed.), 1990, Multibody Systems Handbook, Springer-Verlag, Berlin.
- Schiehlen, W. (ed.), 1993, Advanced Multibody System Dynamics: Simulation and Software Tools, Kluwer Academic Publishers, Dordrecht.
- Schiehlen, W., Eismann, W., 1994, Reduction of Nonholonomic Systems, Collection of Papers Dedicated to Professor P. S. Theocaris, Kounadis, A.N. (ed.), National Technical University of Athens, pp. 207 - 220.
- Schlichtkrull, H., 2015, Differentiable Manifolds, Department of Mathematical Sciences, University of Copenhagen, Denmark, ISBN 978-87-7078-568-6, available at <http://web.math.ku.dk/noter/filer/geom2-2015.pdf>.
- Schweizer, B., and Pu, L., 2016, Solving Differential-Algebraic Equation Systems: Alternative Index-2 and Index-1 Approaches for Constrained Mechanical Systems, Journal of Computational and Nonlinear Dynamics, Vol 11, p. 044501.
- Sefrioui, J. and Gosselin, C., 1995, On the Quadratic Nature of the Singularity Curves of Planar Three-Degree-of-Freedom Parallel Manipulators, Mech. Mach Theory, Vol. 30, No.4, pp. 533-551.
- Serban, R., and Haug, E.J., 1998, Kinematic and Kinetic Derivatives in Multibody System Analysis, Mechanics of Structures and Machines, Vol. 26, No. 2, pp.145-173.
- Shimizu, M., Kaleuya, H., Yoon, W-K, and Kitagaki, K., 2008, Analytical Inverse Kinematic Computation of 7-DOF Redundant Manipulators with Joint Limits and its Application to Redundancy Resolution, IEEE Transactions on Robotics, Vol. 24, No. 5, pp. 1131-1142.
- Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., 2010, Robotics; Modelling, Planning and Control, Springer-Verlag, London.
- Sneddon, I.N., 1957, Elements of Partial Differential Equations, McGraw-Hill, New York.

- Spivak, M, 1999, A Comprehensive Introduction to Differential Geometry, Volume One, Third ed., first published 1970, Publish or Perish, Houston.
- Strang, G., 1980, Liner Algebra and its Applications, Second Ed., Academic Press, New York.
- Stuepnagel, J., 1964, On the Parametrization of the Three-Dimensional Rotation Group, SIAM Review, Vol. 6, No. 4, pp. 422-430.
- Terze, Z., Muller, A. and Zlatar, D., 2015, Lie-group integration method for constrained multibody systems in state space, Multibody System Dynamics, Vol.34, No.3, pp.275-305.
- Teschl, G, 2012, Ordinary Differential Equations and Dynamical Systems, American Math Society, Providence, Rhode Island.
- Tomovic, R., and Vukobratonic, M., 1972, General Sensitivity Theory, American Elsevier, NY.
- Tseng, F.-C., Ma, Z.-D., and Hulbert, G.M., 2003, Efficient numerical solution of constrained multibody dynamics systems, Computer methods in applied mechanics and engineering, Vol. 192, pp 439-472.
- Uicker, J.J., Ravani,B., and Sheth, P.N., 2013, Matrix Methods in the Design Analysis of Mechanisms and Multibody Systems, Cambridge University Press, New York.
- Wang, X., 2007, Design Sensitivity Analysis Of Rigid and Flexible Multibody Systems, Ph.D. Thesis, Department of Mechanical Engineering, The University of Iowa.
- Wang, X., Haug, E.J., and Pan, W., 2005, Implicit Numerical Integration for Design Sensitivity Analysis of Rigid Multibody Systems, Mechanics Based Design of Structures and Machines, vol. 33, no. 1, pp. 1–30.
- Wang, Y., Zhao, C., Wang, X., Zhang, P., and Liu, H., 2021, Inverse Kinematics of 7-DOF Spraying Robot with 4R 3-DOF Non-spherical Wrist, J. of Intel. & Rob. Sys., Vol. 101, No. 68.
- Wehage, R. A., and Haug, E. J., 1982, Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems, Journal of Mechanical Design, Vol. 104, No. 1, pp. 247-255.
- Wittenburg, J., 1977, Dynamics of Systems of Rigid Bodies, B.G. Teubner, Stuttgart.
- Wojtyra, M, 2016, Modeling of static friction in closed-loop kinematic chains-Uniqueness and parametric sensitivity problems, Multibody System Dynamics, Vol. 39, No.4, pp. 337-361.

## Index

### A-stable, 309

absolute error tolerance, 308, 457  
abstract definition of manifolds, 168  
acceleration, 20, 34  
acceleration analysis, 200  
acceleration constraint, 11, 120  
acceleration equations, 200  
action and reaction, 254  
actuator force, 283  
actuators, 251, 283  
adaptive explicit RK methods, 308  
Add Function, 206, 228, 414, 443, 595, 604  
admissible displacement, 252-253, 520  
admissible ranges of stiction forces, 626, 630  
aerodynamic drag, 354  
agricultural and construction equipment, 2  
algebraic constraint equations, 118  
algebraic representation of a geometric vector, 27  
algebraic vector, 20, 27-28  
ambient space, 179  
an alternative DAE formulation, 577  
analytical dynamics, 15, 251  
analytical solution of stiction inequalities, 628  
analytical stiction criteria for three mass model  
angular acceleration, 71  
angular acceleration as a function of Euler parameter derivatives, 96  
angular velocity, 4, 70-71, 104, 266  
angular velocity form of equations of motion, 266  
AppData, 597  
AppData Function, 202, 213, 215, 217, 220, 231, 235, 238, 412, 418, 424, 428, 431, 434, 436, 440, 447, 449, 453, 457, 460, 463, 593, 600, 604, 607, 665, 676, 679, 682, 686, 689, 692, 695  
applied force, 254, 335  
applied generalized force, 386  
approaching a singular configuration, 461  
articulated vehicle, 561  
aspects, 735  
assembled configuration, 198-199  
assembly modes, 722  
atlas, 171, 176, 180, 302, 776, 790  
Atol, 464  
ATran, 415, 443, 595, 604  
automobile suspension, 5, 158  
average step-size/equation error, 598  
avoidance of singular behavior, 700  
avoiding obstacles and singularities, 791  
axis of symmetry, 274

### backward differentiation, 305

ball and socket joint, 140  
ball joints, 5  
balls out, 4  
bearing load, 435  
bearings, 4  
BEval, 415  
bifurcation, 15  
bifurcation point, 193  
bilinear, 93  
bodies of revolution, 274  
body fixed reference frame, 51, 59  
body in cylindrical joint with ground, 456, 686  
boundaries of components, 733  
building block constraints, 137-138, 143, 649  
bump stop, 283  
Butcher tableau, 306

**c-space**, 711  
calculation of constraint reaction forces, 364  
Cartesian components of a vector, 24-25  
Cartesian coordinate simulation, 635, 637  
Cartesian generalized coordinates, 14, 126, 137, 271, 381  
Cartesian reference frame, 24  
center of mass, 268-269  
centrifugal force, 4  
centripetal acceleration, 34  
centroid, 262, 268-269, 273  
centroidal Newton-Euler equations of motion, 269  
centroidal planar variational equation of motion, 264  
centroidal reference frame, 264, 275-276, 280  
chain of bodies, 775  
chain rule of differentiation, 43, 119, 369, 399, 528, 531  
characterizing a set-valued inverse mapping, 789  
chart, 171-172, 180, 302, 367, 399, 662, 717, 728, 770, 776, 790  
chart-to-chart continuation process, 779  
closed kinematic loop, 9, 706  
closed loop four-bar mechanism, 235  
closed loop mechanism, 237  
coefficient of dynamic friction, 617  
coefficient of static friction, 614, 617  
coil spring, 2, 428  
collinear, 68, 250, 260  
column matrix, 28, 36  
column vector, 28, 36  
compact, 172

compatible charts, 180  
 complex composite bodies, 282  
 complexity of computation with Euler angles, 105  
 component, 45, 702, 716-718, 720, 726, 732, 773, 776, 788  
 components of a vector, 25, 61  
 components of constraint manifolds, 179  
 components of regular manipulator configuration space, 735  
 composite bodies, 273, 278  
 composite joints, 145  
 compound manipulator regular configuration manifold, 730, 773  
 compound kinematically redundant manipulators, 787  
 compound manipulator, 700, 703, 706, 708, 713, 729  
 compound manipulator configuration space, 772, 787  
 compound manipulator forward kinematics, 751  
 compound manipulator forward velocity mapping, 743-744  
 compound manipulator inverse kinematic configuration mapping, 789  
 compound manipulator inverse kinematics, 753  
 compound manipulator inverse velocity mapping, 744  
 compound manipulator kinematics, 750  
 compound manipulator obstacle avoidance using self-motion manifolds, 793  
 compound manipulator ode of dynamics in input coordinates, 757  
 compound manipulator ode of dynamics in output coordinates, 758  
 compound manipulator self-motion manifold, 791  
 computational algorithms for implicit manipulator kinematics, 745  
 computational components of code, 205, 226, 341, 414, 443, 593, 604, 667, 684  
 computational dynamics, 249, 418  
 computational flow, 226  
 computational kinematics, 202  
 computationally characterize set-valued inverse mappings, 776  
 computer aided kinematic analysis, 211  
 computer-aided design, 282  
 computer-aided engineering, 282  
 computing functions, 228, 414  
 computing inverse matrices, 754  
 condition number, 15, 20, 48, 174, 185, 200, 352, 358, 367, 375, 403, 426, 510, 531, 534, 537, 555, 558, 560, 621, 662, 688  
 configuration coordinates, 727, 729  
 configuration degrees of freedom, 535  
 configuration space, 161, 168  
 configuration space differentiable manifolds, 700  
 configuration space generalized coordinates, 521  
 configuration tangent space, 520  
 configuration tangent space parameterization, 520  
 connecting rods, 3  
 conservation of energy, 353  
 conservative system, 186, 325, 353, 425, 449, 452, 540, 543  
 constitutive relations, 614  
 constrained equations of motion, 336  
 constraint addition-deletion, 632  
 constraint contact forces, 459, 648, 660  
 constraint drift, 360, 523  
 constraint error control, 377, 379, 394, 406, 551  
 constraint generalized forces, 335  
 constraint geometry, 614, 637  
 constraint Jacobian, 11, 120, 172, 252, 291, 356  
 constraint manifold, 176, 181, 252, 662  
 constraint reaction force, 156, 249, 254-255, 335-336, 338, 364, 396, 553, 614, 638, 648  
 constraint reaction forces and friction generalized forces, 642  
 constraint reaction torque, 338, 642, 648  
 constraint surface contacts, 648  
 constraints at the velocity level, 512  
 constraints between pairs of bodies, 119  
 contact forces, 649  
 contact geometries, 637  
 continuation of integration, 14  
 continuous dependence on problem data, 470  
 continuous friction, 651  
 continuous friction approximation, 617  
 continuous friction model, 624, 632, 635, 687  
 continuous reparameterization of friction, 613  
 continuously differentiable basis, 360  
 continuously differentiable function, 302  
 continuously differentiable parameterization, 776  
 control fuel feed, 465  
 control of error in satisfying constraints, 556  
 converges quadratically, 48, 746  
 coordinate vectors, 61  
 Coriolis forces, 57, 72, 335  
 CorrectB, 443  
 correction of constraint errors, 580  
 Coulomb friction, 614

- Coulomb friction force, 616, 634  
 Coulomb friction inequalities, 614, 626  
 Coulomb friction limits on stiction forces, 614  
 counterclockwise as positive, 25  
 counterclockwise rotation, 59, 101  
 coupler, 5  
 crank, 5  
 criteria for reparameterization, 414, 443  
 criteria for singular behavior, 193, 200  
 criteria for stiction, 613  
 cross product, 26  
 cylindrical Joint, 143, 457, 648, 651
- d'Alembert variational equation of motion, 756**
- d'Alembert variational formulation, 470, 737  
 d'Alembert's equation of motion, 254, 553  
 d'Alembert's Principle, 11, 249-250, 253, 334, 471  
 DAE, 12, 249, 272, 336, 554, 573, 575, 669  
 DAE are not ODE, 573-574, 577, 579  
 DAE formulations, 573  
 DAE of multibody dynamics, 573  
 dampers, 283  
 damping coefficient, 283  
 definition of Euler angles, 101  
 deformable components, 2  
 degrees of freedom, 3, 161, 170, 176, 179, 291, 449, 462, 709, 787  
 delay in onset of stiction, 641, 670  
 dependence of initial conditions on design variables, 470  
 dependent generalized coordinates, 12, 300, 506  
 derivative operators, 20  
 derivatives for implicit integration, 367, 399, 528, 556  
 derivatives of friction forces, 654  
 derivatives of friction generalized forces, 645  
 derivatives of generalized forces, 483, 492  
 derivatives of kinematic constraints, 483  
 derivatives of kinetic quantities, 483  
 derivatives of planar constraints, 483  
 derivatives of planar generalized forces, 492  
 derivatives of planar kinetics, 490  
 derivatives of spatial constraints, 486  
 derivatives of spatial generalized forces, 495  
 derivatives of spatial kinetics, 491  
 derivatives with respect to model parameters, 98  
 design dependence in variational equations of motion, 470  
 design dependent initial conditions, 472, 475
- design of bearings, 4, 328  
 design sensitivity analysis, 270  
 design variables, 264, 302, 470-471  
 design variation, 194, 264  
 diagonal matrix, 37  
 diffeomorphism, 170, 175, 357  
 differentiable manifold, 176, 179, 700, 702, 717-718, 724, 728, 731, 775-776, 778, 790  
 differentiable manifold components and mechanism singularities, 195  
 differentiable manifold formulation, 196  
 differentiable manifold structure, 735  
 differentiable manifold theory, 777  
 differentiable manifolds, 180, 788  
 differential, 57, 72, 252  
 differential constraint, 513-516, 519, 526, 535, 539, 547, 562, 565  
 differential equations of manipulator dynamics, 700  
 differential form, 252, 513  
 differential geometry, 14-15, 20, 162, 171, 175, 179, 700  
 differential mass, 260, 266  
 differential topology, 162, 171  
 differential-algebraic equations, 12, 249, 272, 319, 334-336, 573, 575  
 differentials, 58, 351  
 differentiation index, 554, 574-575  
 differentiation rules, 70  
 dimension of the configuration space, 755  
 directed line segment, 22  
 direction cosine matrix, 53, 62  
 direction cosines, 25, 52, 62  
 discontinuous Coulomb friction, 613  
 discrete time grid, 305  
 disjoint components, 721, 773  
 disk rolling on x-y plane, 517, 541, 571  
 distance constraint, 119, 126, 138, 203, 221, 260, 266  
 distance driver, 130, 150  
 domains of functionality, 179  
 dot1 constraint, 138, 141  
 dot2 constraint, 138, 142  
 dot product, 25  
 double pendulum, 131, 418, 597, 609  
 drift, 501, 504  
 drift from constraints, 364  
 driving constraints, 197, 356  
 driving simulator, 704  
 DSA, 470  
 duality of parameterization, 717

- dumbbell, 256  
 dynamic coefficient of friction, 616  
 dynamic performance optimization, 778  
 dynamic system design sensitivity analysis, 470  
 dynamics of a particle on a unit sphere, 181  
 dynamics of manipulators, 700
- earth-fixed reference frame**, 250  
 effective coefficient of friction, 624  
 eigenvalues, 277  
 ellipsoid, 544  
 ellipsoid rolling without slip on a moving surface, 544  
 empty set, 44  
 end effector, 7, 703, 714, 775  
 end-effector pose, 783  
 energy dissipation, 2  
 energy input, 2  
 equal matrices, 37  
 equal vectors, 28  
 equations of motion with friction, 660  
 error control, 316  
 error control parameters, 410, 414, 443  
 439, 464  
 error tolerance, 358, 419  
 estimate of solution error, 308  
 Euclidean space, 20, 44  
 Euclidian space differential geometry, 700  
 Euler angle singularity, 103-104  
 Euler angles, 21, 101, 782  
 Euler parameter derivatives, 90  
 Euler parameter differentials and infinitesimal rotation, 95  
 Euler parameter form of equations of motion, 270  
 Euler parameter generalized coordinates, 20  
 Euler parameter identities, 87  
 Euler Parameter Identities and Derivative Operators, 112  
 Euler parameter normalization condition, 79, 120, 153, 197, 648  
 Euler parameters, 78-79, 348, 518, 542, 545  
 Euler's theorem, 76  
 evaluation of constraint reaction forces, 339  
 evaluation of error in approximate solutions, 584  
 exact differential form, 513  
 exact differential theorem, 513  
 examples of nonholonomic constraints, 514  
 examples of systems with independent generalized coordinates, 294  
 excavator, 706  
 explicit integration of Index 0 DAE, 398, 619  
 explicit integration of Index 1 DAE, 586  
 explicit integration of nonholonomic Index 0 DAE, 555  
 explicit integration of partitioned equations of motion, 509  
 explicit numerical integration, 305, 366, 505, 527, 539, 660  
 explicit numerical integration for Index 1 DAE, 582  
 explicit Nystrom4, 622, 636, 688  
 explicit parallel manipulator, 719  
 explicit parallel manipulator configuration space, 719  
 explicit parallel manipulator forward kinematics, 750  
 explicit parallel manipulator forward velocity mapping, 743  
 explicit parallel manipulator inverse kinematics, 745  
 explicit parallel manipulator inverse velocity mapping, 743  
 explicit parallel manipulator ODE of dynamics in input coordinates, 760  
 explicit parallel manipulator ODE of dynamics in output coordinates, 760  
 explicit parallel manipulator regular configuration manifold, 723  
 explicit parallel manipulator regular configuration space, 723, 743  
 explicit parallel manipulator velocity mappings, 742  
 explicit parallel manipulators, 700, 703, 706, 713  
 explicit RK, 537, 540  
 explicit Runge-Kutta methods for first order ODE, 307  
 explicit Runge-Kutta methods for second order ODE, 314  
 extent of components, 180  
 external force, 260, 263, 266  
 external torque, 263  
 externally applied torque, 351
- feedback control**, 2  
 first form of the fundamental equation of motion, 254  
 first law, 250  
 first order initial-value problem, 302  
 first order kinematic ODE, 526  
 first order ODE, 249, 305  
 first order SDIRK, 317  
 fly-ball governor, 4, 157, 242, 465

force elements, 283  
 force of friction, 254  
 forces of action and reaction, 250  
 forces of constraint, 11, 254, 648  
 forces of contact, 250  
 forward and inverse configuration mappings, 741  
 forward and inverse kinematics, 711  
 forward and inverse kinematics computation, 737  
 forward and inverse kinematics for compound manipulators, 751  
 forward and inverse velocity mappings, 741  
 forward elimination, 505  
 forward kinematic mapping, 700, 702-704, 706, 712, 715, 775, 788-789  
 forward kinematics, 711, 717, 720, 787  
 forward kinematics computation, 737  
 forward kinematics for serial manipulators, 745  
 four-bar linkage, 5  
 four-bar linkages, 5  
 four-bar mechanism, 235  
 four-bar mechanism with internal rotation driver, 235  
 four-body mechanism, 462  
 four-body slider, 691  
 four-mass spatial system, 637  
 frame of reference, 250  
 free length, 283  
 friction, 2, 254  
 friction approximation, 618  
 friction coefficients, 674  
 friction data, 665  
 friction effects, 674, 688  
 friction force, 396, 614, 643-644, 648, 682, 686  
 friction force function, 617  
 friction forces in joints, 651  
 friction generalized forces, 665, 682  
 friction torque, 686  
 front-end loader, 8, 706-707, 729, 731  
 front-end loader compound manipulator, 765  
 Full DAE, 336, 470, 579  
 Full DAE are well posed, 478  
 Full DAE nonholonomic formulation, 568  
 Full DAE of motion, 478, 505, 507  
 full pivoting, 505  
 full rank, 40, 252, 291, 506, 711  
 fundamental equation, 11, 574  
 fundamental theorem of algebra, 41

**GamEval**, 444, 595, 604  
**GamsqqdEval**, 444, 595, 604  
 Gaussian elimination, 505-506

general compound manipulator, 731  
 general explicit parallel manipulator, 723  
 general form of ode of multibody dynamics, 300  
 general implicit manipulator, 727  
 general serial manipulator, 717  
 generalized applied force, 371  
 generalized constraint reaction forces, 334  
 generalized coordinate partitioning, 505, 510  
 generalized coordinate vector, 197  
 generalized coordinates, 8, 20, 57, 251, 538, 561, 574, 714, 719, 729, 741, 787  
 generalized force, 272, 284, 377  
 generalized form of second order ODE, 320  
 generalized friction forces, 660  
 generalized revolute joint friction forces, 643  
 generalized speeds, 502  
 generalized translational joint friction forces, 645  
 generic 7-DOF serial manipulator, 781  
 geometric nonlinearity, 2  
 geometric vector, 20, 22  
 geometric vector analysis, 22  
 geometry of joints, 648  
 global inverse kinematic mapping, 719  
 global inverse kinematics, 723  
 global reference frame, 51, 119  
 global setting, 179  
 globally differentiable forward kinematics, 703  
 globally differentiable inverse kinematics, 703  
 globally one-to-one, 101  
 globally valid forward kinematic mapping, 715  
 globally valid inverse kinematic mapping, 705  
 globally valid inverse kinematics, 703  
 gravitational acceleration, 250  
 gravitational force, 250, 263  
 gravity, 2  
 ground body, 119

**Hausdorff**, 45  
 HHT, 319, 583, 586, 588, 591, 597, 600, 603, 607, 609  
 hidden constraints, 575, 577  
 high frequency response, 421-422  
 highly damped response, 303  
 highly oscillatory, 423  
 holonomic constraints, 118-119, 168, 197, 251, 288, 356, 396, 470, 513, 516, 518-519, 538, 542, 561, 564, 574, 710, 772, 787  
 holonomic stiction constraint, 632  
 holonomic velocity constraint equation, 252  
 hydraulic actuator, 8, 704, 706-707, 729  
 hydraulic dampers, 2

**identity matrix**, 37  
 ill conditioning, 375  
 imbedded error estimation, 310  
 impact, 283, 421, 428, 433  
 implicit function theorem, 20, 36, 45-46, 174, 198, 357, 473, 507, 715, 775  
 implicit Index 0 DAE integration algorithm, 403, 664  
 implicit integration algorithm, 533, 559  
 implicit integration of Index 1 DAE, 586  
 implicit integration of the Index 0 DAE, 620  
 implicit manipulator configuration space, 725, 727  
 implicit manipulator forward kinematics, 747  
 implicit manipulator forward velocity mapping, 743  
 implicit manipulator inverse kinematics, 748  
 implicit manipulator inverse velocity mapping, 743  
 implicit manipulator kinematics, 725, 745  
 implicit manipulator ODE, 760  
 implicit manipulator ODE of dynamics in input coordinates, 760  
 implicit manipulator ODE of dynamics in output coordinates, 760  
 implicit manipulator regular configuration manifold, 727  
 implicit manipulator velocity mappings, 743  
 implicit manipulators, 700, 703, 706, 713, 725  
 implicit numerical integration, 367, 399, 483, 505, 660  
 implicit numerical integration algorithms, 305, 375, 399, 530, 557, 662  
 implicit numerical integration of Index 1 DAE, 586  
 implicit numerical integration of Index 2 DAE, 588  
 implicit numerical integration of Index 3 DAE, 589  
 implicit regular configuration space, 726-727  
 implicit Runge-Kutta, 664  
 Implicit Runge-Kutta Integration of Index 0 DAE, 401  
 implicit Runge-Kutta methods for first order ODE, 309  
 implicit Runge-Kutta methods for second order ODE, 317  
 implicit trapezoidal integration, 309, 317, 530, 632, 636  
 implicit trapezoidal integration of Index 0 DAE, 400  
 inaccuracy of Maggi and Kane equations for holonomic constraints, 504  
 IndODEfunct, 415, 444  
 independent generalized coordinates, 3, 12, 294-296, 300, 348, 506  
 independent holonomic constraints, 300  
 independent velocity coordinates, 523  
 index, 574-575  
 Index 0 DAE, 396-397, 410, 439, 476, 553, 636, 665, 682, 688  
 Index 0 DAE for nonholonomic systems, 553  
 Index 0 DAE formulation, 346, 396, 512, 561, 660  
 Index 0 DAE friction formulation, 638  
 Index 0 DAE initial-value problem, 555  
 Index 0 equations of motion, 660  
 Index 1 DAE formulation, 579, 598, 662  
 Index 2 DAE formulation, 580, 598  
 Index 3 DAE formulation, 580, 599, 623, 688  
 index of a DAE, 573, 575  
 induced topology, 45  
 inertia matrix, 268, 273, 275, 280  
 inertia properties of complex bodies, 278  
 inertia tensor, 268  
 inertial reference frame, 11, 250, 266  
 infinitesimal displacement, 73  
 infinitesimal rotation, 73  
 infinitesimal variation, 72, 261, 267  
 initial angular velocity, 353  
 initial conditions, 251, 291, 379, 403, 460, 509, 575  
 initial conditions as functions of problem data, 471  
 initial conditions for Maggi's equations, 504  
 initial configuration, 292, 353, 451, 471  
 initial estimate, 48, 200  
 initial generalized coordinates, 413, 441  
 initial time, 251, 291  
 initial velocity conditions, 292, 413, 441, 451, 472  
 initial-value problem, 251, 295, 297, 301, 305, 527  
 input coordinates, 700, 702, 705, 710, 714, 719, 727, 741, 775, 787  
 input equations, 710, 730, 772, 787  
 input generalized forces, 756  
 input-output equations, 717, 725, 727, 775  
 input-output model, 704-705  
 integrable differential form, 514  
 integrable differential theorem, 514  
 integrating factor, 514

integration error tolerance, 406, 454, 464  
integration error tolerance, 623  
integration of Index 1 DAE, 586  
integration of Index 2 DAE, 588  
integration of Index 3 DAE, 589  
integration on a constraint manifold, 305  
intermediate variables, 663  
internal force, 260, 266, 283  
internal force element, 285, 287  
internal forces in planar systems, 286  
internal forces in spatial systems, 283  
internal gravitational attraction, 260  
invariants of the orientation transformation, 75  
inverse configuration mapping, 742  
inverse Euler parameter mapping theorem, 80  
inverse image, 45  
inverse kinematic mapping, 700, 702, 712, 715, 778  
inverse kinematics, 711-712, 787  
inverse kinematics for explicit parallel manipulators, 745  
inverse matrix, 40  
isolated singular point, 194  
iterative solution of nonlinear equations, 305

### **Jacobian**, 119, 770, 787

Jacobian matrix, 48  
Jacobian of the residual, 531  
JacobODE, 415, 444  
joint constraint reaction forces and torques, 337  
joint contact forces, 651  
Joint Data Table, 665, 682  
joint definition frame, 336-337  
joint definition point, 338  
joint designs, 651  
joints, 20

### **k-dimensional differentiable manifold**, 724

Kane's equations, 501  
kilograms, 251  
kinematic acceleration equation, 119-120  
kinematic analysis, 202, 220, 777  
kinematic closed loop, 708, 729  
kinematic configuration space, 711  
kinematic constraint, 11, 251, 709, 730  
kinematic degrees of freedom, 539, 542, 563, 565  
kinematic differential equation, 512, 523  
kinematic drivers, 130, 150  
kinematic modeling, 291  
kinematic simulation, 460  
kinematic singularities, 191, 709

kinematic velocity and acceleration equations, 575  
kinematic velocity equation, 119-120, 288  
kinematically admissible, 252  
kinematically admissible virtual displacements, 11, 289-290, 356, 524  
kinematically admissible virtual velocities, 249, 291, 301  
kinematically redundant manipulators, 768  
kinematically redundant serial manipulator, 775  
kinetic degrees of freedom, 539, 542, 563, 565  
kinetic energy, 290-291, 540, 543  
kinetic modeling, 291

### **L-stability**, 309

Lagrange multiplier, 12, 272, 364, 626, 682  
Lagrange multiplier theorem, 12, 41, 272, 334, 477, 553, 575  
Lagrange multiplier theorem of linear algebra, 36  
Lagrange multipliers, 41, 249, 335, 338-339, 399, 435, 574, 637, 648, 660  
Lagrangian generalized coordinates, 12, 14-15, 301  
lane change maneuver, 540  
Larry Bird Paradox, 579, 582, 607  
laws of motion, 250  
leaf springs, 2  
likely not stiff, 691  
linearly independent, 39, 173  
Lipschitz continuity, 302  
local, 742  
local forward kinematic mapping, 723  
local generalized coordinates, 173, 357  
local inverse kinematic mapping, 717  
local manifold parameterizations, 777  
local parameterization, 170, 172, 177, 348, 775  
locally defined independent generalized coordinates, 348  
locally defined one parameter inverse kinematic mapping, 770  
locally independent generalized coordinates, 14  
locally one-to-one, 170  
lock-up, 15, 148  
lock-up configuration, 194  
lubricated bearings, 3  
lumped mass-spring model, 428, 430, 676

### **machine tools**, 2

Maggi's equations, 501, 503  
Maggi/Kane ODE initial value problem, 504  
magnitude of a vector, 22

manifold component, 746  
 manifold parameterizations, 700  
 manifolds, 14  
 manipulator configuration coordinates, 775  
 manipulator configuration manifold, 770  
 manipulator configuration space, 702, 710  
 manipulator control, 702  
 manipulator coordinates, 710, 715, 719, 725, 787  
 manipulator equations of motion are well posed, 700  
 manipulator kinematic generalized coordinates, 707  
 manipulator kinematics, 702  
 manipulator mechanism, 710  
 manipulator platform, 704  
 manipulator regular configuration manifold, 778  
 manipulator working capability, 710  
 manipulator workspaces, 708  
 manipulators, 2, 7, 187, 700  
 mapping from orientation transformation matrix to Euler parameters, 79  
 mapping one-dimensional compound manipulator self-motion manifolds, 791  
 mapping one-dimensional serial manipulator self-motion manifold components, 778  
 mass, 250  
 massless body, 145  
 massless coupler, 126, 138  
 material handling and construction manipulators, 729  
 material handling equipment, 706  
 matrix, 36  
 matrix algebra, 20, 36  
 matrix calculus notation, 42  
 matrix condition number, 48  
 matrix dimension, 36  
 matrix norm, 48  
 matrix product, 38  
 matrix version of Newton-Raphson algorithm, 177, 745, 798  
 maximal singularity free components, 700  
 maximal, singularity free, path connected, components, 179  
 McPherson strut, 5, 149  
 measure of integration error, 308  
 mechanism, 710  
 mechanism assembled configurations, 180  
 mechanism generalized coordinates, 709-710  
 mechanism generalized forces, 756  
 mechanism mass matrix, 756  
 mechanism model, 615  
 mechanism velocity coupling terms, 756  
 mechanisms with closed kinematic loops, 729  
 meters, 251  
 metric space, 44  
 model compound manipulator, 765  
 model explicit parallel manipulator, 763  
 model implicit manipulator, 725, 764  
 model kinematically redundant compound manipulator, 792  
 model manipulators, 700  
 model serial manipulator, 762  
 models with friction are not necessarily stiff, 624  
 modern form of Maggi's equations, 503  
 moment of inertia, 256  
 momentum, 353  
 mosquito tracking prey, 516  
 multibody dynamics, 10  
 multibody systems, 1-2, 118  
 multivariable calculus, 20, 42

**n - m parameter inverse mapping**, 788

n-dimensional Euclidean space, 28, 43  
 n-vector, 28  
 near singular configuration, 156, 691  
 negative of a vector, 22  
 Newmark method, 319  
 Newton's equation of motion, 260  
 Newton's first law, 615  
 Newton's laws of motion, 249-250  
 Newton's third law, 266  
 Newton-Euler equations of motion, 258  
 Newton-Raphson method, 20, 48, 199, 310, 531  
 358, 402  
 newtons, 251  
 noncentroidal equations of motion, 536  
 noncentroidal mass matrix, 290  
 noncentroidal Newton-Euler equations of motion, 269  
 noncentroidal planar variational equation of motion, 263  
 noncentroidal reference frame, 258, 264, 273, 289, 353, 410, 439, 490  
 noncentroidal variational equation of motion, 269, 271  
 nonholonomic constraint, 513-514, 519, 526  
 nonholonomic multibody system dynamics, 512  
 nonholonomic system kinematics, 519  
 nonholonomic system kinetic ODE, 526  
 nonholonomic systems, 187, 501, 512  
 nonholonomic velocity constraints, 517  
 nonintegrability of angular velocity, 95

- nonlinear effects, 4  
 nonlinear equations of constraint, 3  
 nonredundant manipulators, 700, 709  
 nonsingular matrix, 40  
 nonsingular submatrix, 775  
 norm, 44  
 not necessarily stiff, 697  
 not simply connected, 721  
 not stiff, 423, 600, 607, 622-623, 636, 639, 671, 694-695  
 null space, 41, 172, 291, 360  
 null vectors, 520  
 number of stages, 532  
 numerical algorithms, 10  
 numerical damping, 583, 607  
 numerical error control, 20  
 numerical integration, 10  
 numerical integration functions, 415  
 numerical integration methods, 249  
 numerical integration of Index 1 DAE, 586  
 numerical minimization method, 199  
 numerical solution error tolerance, 372  
 numerical solution of Coulomb stiction criteria, 632  
 numerical solution of second order ODE, 305  
 Nystrom 4, 186, 314, 322, 353, 379, 406, 410, 419, 586, 591, 624, 671, 682, 694
- obstacle avoidance**, 773, 778
- ODE, 12, 249  
 ODE existence theorem, 303, 470, 476  
 ODE formulations, 346  
 ODE initial-value problem, 537  
 ODE of dynamics, 301, 737  
 ODE for explicit parallel manipulators, 760  
 ODE for implicit manipulators, 760  
 ODE for serial manipulators, 760  
 ODE of manipulator dynamics in input coordinates, 758-759  
 ODE of motion, 184, 351, 362, 472, 526  
 ODEfunct, 415, 444, 595, 605  
 one parameter inverse kinematic mappings, 773  
 one rotation degree of freedom in the plane, 54  
 one step methods, 305  
 one-dimensional configuration trajectory, 179  
 one-dimensional self-motion manifold component, 784  
 one-to-one, 175  
 onset of the stiction event, 639  
 onstraint reaction force, 338, 642  
 open ball, 44
- open loop mechanisms, 237  
 open set, 44  
 optimum step size, 308  
 order of rotations cannot be interchanged, 60  
 ordinary differential equations, 12, 249, 294  
 ordinary differential equations of motion, 3  
 organization of the text, 17  
 orientation, 59  
 orientation degrees of freedom, 53, 63  
 orientation generalized coordinates, 542  
 orientation identities, 55  
 orientation in space, 59  
 orientation of bodies, 20  
 orientation of reference frames, 76  
 orientation output coordinates, 782  
 orientation transformation matrix, 20, 53, 55, 62, 126  
 orthogonal matrix, 41, 53, 63, 68  
 orthogonal reference frame, 22, 24  
 orthogonal transformation matrix, 137  
 orthogonal triad, 63  
 orthogonal vectors, 25  
 orthogonality, 22  
 orthonormal eigenvectors, 277  
 output coordinates, 700, 702, 705, 710, 714, 719, 725, 727, 741, 787  
 output equation, 710, 730, 772, 787
- P5 Function**, 225
- parallel axis theorem, 276  
 parallel manipulator, 9, 704, 706, 719, 769  
 parallel manipulator singularities, 708  
 parallelogram mechanism, 191  
 parallelogram rule, 23  
 Param, 444  
 parameterization, 171, 176, 393, 403, 408, 528, 534, 556  
 parameterization of a particle on a unit sphere, 168  
 parameterization of the regular compound manipulator configuration space, 790  
 parameterization of the regular constraint manifold, 301  
 parameterization of the self-motion manifold, 778  
 parameterization of velocity space, 512  
 parameterizations of set-valued inverse mappings, 768  
 parameterized system of equations, 661  
 parPart, 415, 444, 595, 604  
 partial velocities, 502  
 particle on a unit sphere, 168, 181, 246

particles as bodies, 119  
 partitioning of vectors, 415  
 partitioning the DAE of dynamics, 507  
 path connected set, 45, 161, 722  
 path connected, singularity free component, 771  
 periods of stiction, 687  
 Petzold Imperative, 573-574, 577, 579, 582, 599-600, 607, 609  
 phase differences, 426  
 piecewise smooth form of analysis, 777  
 PJDT, 412  
 planar articulated vehicle, 538, 561, 571  
 planar body differential equations of motion, 264  
 planar Cartesian generalized coordinates, 57  
 planar distance constraint, 126  
 planar double pendulum, 131, 294, 322, 343, 377, 418, 482  
 planar double pendulum with unilateral spring, 597  
 planar joint constraint reactionforce and torque, 338  
 Planar Joint Data Table, 203, 412  
 planar kinematic analysis, 246  
 Planar Mass Data Table, 412  
 planar parallel manipulator, 725  
 planar platform, 705, 719  
 planar rigid body, 51  
 planar rigid body equations of motion, 257  
 Planar RSDA Data Table, 413  
 planar RSDA generalized forces, 287  
 planar RSDA torque, 287  
 planar serial manipular, 703, 714  
 planar slider-crank, 132, 134, 211, 296, 326, 343  
 planar system dynamics with friction, 665  
 planar system kinematics, 20  
 planar three-wheel transporter, 535  
 Planar TSDA Data Table, 412  
 planar TSDA generalized forces, 286  
 plane of symmetry, 274  
 PMDT, 412  
 point definition of an orthogonal reference frame, 83  
 polar moment of inertia, 262, 273, 277  
 position analysis, 199  
 positive definite, 277, 290-291, 295, 362, 398, 476, 507, 527, 555  
 positive definiteness of system mass matrix, 290  
 positive semidefinite, 276, 290  
 precession, 381  
 principal axes, 277  
 principal moments of inertia, 277  
 principal products of inertia, 277  
 principal subdeterminants, 295  
 prismatic joint, 128  
 problems of multibody dynamics with friction are not well posed, 624  
 product rule of differentiation, 33, 43  
 products of inertia, 276  
 PRSDAT, 413  
 PTSDAT, 412  
**QAEval**, 667, 684  
**QAsLamEva**, 684  
**QAsLamEval**, 668  
**QAsqqdEval**, 667, 684  
**qPart**, 415, 444, 595, 604  
 quadratically convergent, 199  
 quick return mechanism, 215, 424, 600, 674  
**radius of precession**, 353  
 range, 41  
 rank, 40  
 reaction forces, 4  
 real Euclidean space, 42  
 real-time computation of forward and inverse kinematics, 754  
 real-time computation of the ODE of dynamics, 766  
 real-time kinematics algorithms, 739  
 real-time manipulator kinematics and dynamics, 702  
 reciprocating motion, 3, 5  
 recursive dynamics methods, 739  
 reduced mass matrix, 362  
 redundant compound manipulators, 769  
 redundant constraint, 195  
 redundant input manipulators, 768  
 redundant manipulators, 700  
 redundant serial manipulators, 769  
 reference frames for location and orientation in a plane, 51  
 reference frames for location and orientation in space, 59  
 regular compound manipulator configuration space, 788  
 regular configuration manifolds, 768  
 regular configuration space, 118, 161, 168, 172, 176, 179, 349, 356, 367, 520, 771  
 regular constraint manifold, 504  
 regular explicit parallel manipulator configuration spac, 720

regular manipulator configuration space, 700, 711, 731  
 regular mechanism configuration components, 195  
 regular serial manipulator configuration space, 715, 717, 775  
 regular velocity space, 520  
 regular velocity tangent space, 522  
 relative angle driver, 204  
 relative degrees of freedom, 139  
 relative distance driver, 204, 224  
 relative error tolerance, 308  
 relative rotation driver, 131, 151, 224  
 relative topology, 775, 788  
 reordering of variables, 505  
 reparameterization, 185-186, 353-354, 379, 387, 426, 432, 449, 452, 457, 465, 468, 531, 537, 540, 543, 551, 558, 623, 632, 636, 639, 662, 688, 691  
 ResidInd, 415, 444  
 ResidODE, 415, 444  
 residual equation, 399, 556, 586, 588  
 resultant force, 250, 269, 271  
 resultant torque, 269, 271  
 results for highly oscillatory system, 422  
 results for stiff systems, 419  
 revolute joint, 127, 144, 203, 642, 650, 652  
 revolute joint reaction and friction forces, 642  
 revolute-spherical composite joint, 150, 224  
 right-hand orthogonal reference frames, 24  
 rigid body, 59  
 rigid multibody systems, 2  
 rigid-body motion, 262  
 RK, 549, 563  
 RK integrators, 306, 532  
 RKF, 549  
 RKFN, 316, 322, 410, 421, 586, 591, 597, 600, 603, 607, 609, 671, 673, 682  
 robot, 7  
 robotic manipulator, 297  
 rocker arm, 5  
 roll without slip, 515, 535, 538, 541, 544, 561, 563  
 rotating disk with translating body, 436, 678, 695  
 rotation, 3  
 rotation driver, 134  
 rotation is not a vector quantity, 60  
 rotational bearings, 3  
 rotational drivers, 232  
 rotational spring-damper-actuator, 283, 287  
 row matrix, 36  
 row rank (column rank), 40  
 row vector, 36  
 RSDA, 285, 287, 439, 591, 674  
 Runge-Kutta, 305, 307, 322  
 Runge-Kutta discretization, 533  
 Runge-Kutta implicit numerical integration, 373  
 Runge-Kutta integration of second order ODE, 313  
 Runge-Kutta methods, 305  
 Runge-Kutta methods for first order ODE, 305  
 Runge-Kutta methods for second order ODE, 311  
 Runge-Kutta numerical integration methods, 249  
 Runge-Kutta-Fehlberg, 308  
 Runge-Kutta-Fehlberg-Nystrom, 316

**scalar product**, 25, 29, 44  
 SDIRK, 310, 317, 322, 406, 410, 421, 532, 558, 563, 671, 673, 682  
 SDIRK54, 310  
 SDIRK54 formula, 318  
 SDIRK54 integration Jacobian, 402  
 SDIRK formulas, 309  
 SDIRK integration, 532, 558  
 second law, 250  
 second order accurate, 320  
 second order initial-value problem, 303, 311, 322  
 second order integration algorithm, 311  
 second order ODE, 13, 184, 257, 297, 311, 352, 362-363, 508  
 second order ODE of dynamics, 249  
 second order ODE of manipulator dynamics, 700, 756  
 second order partitioned ODE of dynamics, 508  
 second order stage equation, 313  
 second order tableau, 317  
 second order trapezoidal formula, 317, 319, 372  
 second order ODE, 251  
 self-motion manifold, 768, 778, 791  
 self-motion manifold component, 778  
 self-motion manifolds with disjoint components, 780  
 self-motion transformation, 773  
 sensitivity analysis, 264  
 serial chain, 706  
 serial chain mechanism, 714  
 serial manipulator, 8, 713-714, 770, 775  
 serial manipulator configuration space, 717, 770, 775  
 serial manipulator forward kinematics, 745  
 serial manipulator forward velocity mapping, 742  
 serial manipulator inverse kinematic configuration mapping, 776

serial manipulator inverse kinematics, 749  
 serial manipulator inverse velocity mapping, 742  
 serial manipulator kinematic configuration space, 715  
 serial manipulator ODE of dynamics in input coordinates, 760  
 serial manipulator ODE of dynamics in output coordinates, 760  
 serial manipulator regular configuration manifold, 717  
 serial manipulator regular configuration space, 742  
 serial manipulator velocity mappings, 741  
 serial manipulators, 700, 703, 741  
 serial robotic manipulator, 7  
 set-valued, 626, 789  
 set-valued configuration inverse mappings, 768  
 set-valued inverse kinematic mapping, 769, 773, 775  
 sets of singularities, 195  
 shock absorber, 2, 5  
 simulation of three mass planar mechanism, 621  
 simulation results and analysis, 425, 429, 432, 435, 437  
 single degree of freedom joints, 775  
 single valued inverse kinematic mapping, 789  
 singly diagonal implicit Runge-Kutta, 309, 317  
 singular behavior, 9, 15, 193, 661, 727  
 singular configurations, 15, 118, 148, 161, 195, 200, 241, 435, 459, 688, 711, 716, 721, 726, 735, 775, 787-788  
 singular manipulator configuration, 713  
 singular matrix, 40  
 singular points, 194  
 singular value decomposition, 173, 182, 356, 502, 522, 555, 618, 776, 789  
 singularities, 15, 708  
 singularities due to time dependent drivers, 192  
 singularity, 9, 711  
 singularity avoidance, 196, 735, 778  
 singularity free components, 196, 708, 775  
 singularity free differentiable manifold component, 737  
 singularity free domains of functionality, 709  
 singularity free manifold components for front-end loader, 731  
 singularity information, 196, 735  
 six-step algorithm for compound manipulator forward kinematics, 751  
 six-step algorithm for compound manipulator inverse kinematics, 753  
 SJDT, 440  
 skew-symmetric matrix, 29, 39, 73  
 slider-crank, 2-3, 434, 609, 671, 688  
 slider-crank, 2-body model with rotation driver, 134  
 slider-crank, 3-body model, 132  
 slip velocities, 633  
 SMDT, 440  
 solid modeling, 282  
 solution error tolerance, 48, 374  
 space of kinematically admissible virtual velocities, 291  
 sparse matrices, 10, 123, 414, 509  
 spatial disk rolling without slip on x-y plane, 541, 563  
 spatial distance constraint, 138  
 spatial double pendulum, 390, 452-453, 482, 609  
 spatial joint constraint reaction force and torque, 336  
 Spatial Joint Data Table, 221, 225, 440  
 spatial multibody systems, 439  
 spatial rigid body, 257, 266  
 spatial robotic manipulator, 328  
 spatial rotational spring-damper-actuator, 285  
 Spatial RSDA Data Table, 441  
 spatial RSDA torque, 285  
 spatial serial manipulator, 703  
 spatial slider-crank, 156, 238, 459, 609, 688  
 spatial system dynamics, 439  
 spatial TSDA, 283  
 Spatial TSDA Data Table, 440  
 spatial TSDA generalized forces, 285  
 spherical constraint, 138, 222  
 spherical coordinates, 12  
 spherical joint, 140  
 spin stabilized top, 381, 406, 607  
 spinning top with tip constrained to be a unit distance from a fixed point, 449  
 spinning top with tip fixed, 447  
 spring constant, 283  
 spring-damper-actuators, 591  
 springs, 283  
 square matrix, 37  
 SRSDAT, 441  
 stability of the spinning top, 353  
 stage equations, 306-307, 532, 558-559  
 stage variables, 306, 532, 557-558  
 Standard International units (SI), 251  
 state space approach, 579  
 static Coulomb friction in a constrained system, 614

static equilibrium, 626  
 static friction, 250, 614, 626  
 stationary Cartesian reference frame, 32  
 steady state, 4  
 steer angle, 540  
 steering rack, 5  
 step size, 308, 532  
 step size control, 317  
 Stewart platform, 704  
 stiction, 250, 614, 622, 626, 693  
 stiction constraint, 632  
 stiction forces, 614, 626, 630  
 stiff, 303, 597, 623, 675  
 stiff behavior, 15, 303, 305  
 stiff differential equations, 318  
 stiff event, 306, 433  
 stiff system, 303, 309, 419, 421  
 stiff system characteristics, 380  
 stiff transients, 421  
 stiffly accurate, 309-310  
 stiffness, 306, 389, 636, 669  
 strong form of the third law, 250  
 structure of constraint equations, 137  
 strut, 5  
 strut composite joint, 149, 223  
 STSDAT, 440  
 subJacobians, 47, 127-129, 139  
 submanifolds, 775  
 subscript notation, 43  
 sum of matrices, 37  
 surge waves in a coil spring, 428, 676  
 suspension spring, 5  
 symbolic computation, 10  
 symmetric matrix, 39  
 symmetric top, 348, 381  
 system acceleration constraints, 520, 526  
 system acceleration equations, 198  
 system assembly, 199  
 system constraint Jacobian, 123, 198  
 system constraints and virtual displacements, 288  
 system first order ODE of motion, 526  
 system generalized coordinates, 119, 122, 197  
 system holonomic constraints, 288  
 system kinematic acceleration equation, 123  
 system kinematic velocity equation, 123  
 system kinetic energy, 291, 299  
 system ODE, 526  
 system variational equation of motion, 289-290, 296  
 system velocity constraints, 288, 519  
 system velocity equations, 198  
 system virtual displacement constraints, 288  
 systems with friction, 187, 396  
**tangent plane**, 168, 174, 182  
 tangent space, 168, 172, 182, 349  
 tangent space equations of motion, 353  
 tangent space formulation, 360, 410  
 tangent space generalized coordinates, 169, 182, 346, 349, 356  
 tangent space Index 0 DAE, 415  
 tangent space Index 0 DAE with friction, 618  
 tangent space kinematics, 396  
 tangent space kinematics with time dependent constraints, 356  
 tangent space kinematics with time independent constraints, 361  
 tangent space ODE, 410, 415, 439  
 tangent space parameterization, 118, 173, 179, 181, 346, 356-357, 473, 510, 660  
 tangent space parameterization of the regular configuration space, 172  
 transformation of planar polar moment of inertia, 277  
 Taylor series approximation, 305-306  
 the only good DAE is one with an equivalent ODE, 579  
 The Petzold Imperative; DAE are not ODE, 577  
 the regular configuration space is a differentiable manifold, 175  
 the regular serial manipulator configuration differentiable manifold, 776  
 the serial manipulator self-motion manifold, 778  
 the spatial mass data table, 440  
 theorem of the alternative, 193  
 theory of ODE, 302  
 third derivative for variable step size error control, 320, 584  
 third derivatives for trapezoidal error control, 403  
 third law, 250  
 third order stiction redundancy, 637, 691  
 three body double slider-crank, 213  
 three degree of freedom robotic manipulator, 343  
 three mass model with friction and stiction, 615  
 three orientation degrees of freedom in space, 65  
 three-body double slider-crank, 213  
 three-body mechanism, 431  
 three-dimensional space, 20  
 three-particle model with friction, 669  
 three-slider mechanism, 669  
 three-step algorithm for explicit parallel manipulator forward kinematics, 750

- three-step algorithm for implicit manipulator inverse kinematics, 748  
 three-step algorithm for serial manipulator inverse kinematics, 749  
 three-wheel vehicle, 515  
 thrust bearing, 652  
 tie rod, 5  
 tilde, 29  
 time dependent constraint, 252  
 time dependent driver, 205  
 time dependent holonomic constraint, 545  
 time derivative of a vector, 32  
 time derivatives and variations in rotation and displacement, 57  
 time derivatives and variations of Euler parameters, 93  
 time derivatives of vectors, 32  
 time grid, 199  
 time increment, 252  
 time independent constraints, 471  
 time step, 200  
 time step control, 320  
 time-dependent vector, 32  
 tires, 2  
 top tip fixed-spin stabilized, 482  
 top tip fixed-transient, 482  
 top tip on x-y plane, 482  
 top with tip fixed, 381, 406  
 top with tip sliding on x-y plane, 384  
 topological analysis, 10  
 topological space, 44, 179  
 topological subspace, 45, 711  
 topological vector space, 45  
 topology, 44  
 torque control algorithm, 466  
 torque due to friction, 652  
 torsional springs and dampers, 377  
 total energy, 185, 325, 353, 421, 425, 449, 550, 607  
 trace, 79  
 trailing arm, 5  
 trajectories in a constraint manifold, 180  
 trajectory planning, 777  
 transformation of coordinates, 61  
 transformation of inertia matrices, 275  
 transient oscillations, 421  
 transient top, 384, 407  
 translational coefficients of friction, 651  
 translational joint, 128, 145, 203, 643, 653  
 translational joint reaction and friction forces, 643  
 translational spring-damper-actuator, 283, 603  
 transpose of a matrix, 36  
 trapezoidal integrator, 309, 322, 372, 387, 406, 410, 421, 530, 557, 563, 588, 591, 597, 600, 603, 607, 609, 620, 622, 664, 671, 673, 682, 686, 691, 693  
 trapezoidal Jacobian, 401  
 trial design, 200  
 triple slider-crank, 187  
 TSDA, 283, 286, 436, 439, 591, 603, 678  
 two body slider-crank, 211  
 two body spatial pendulum, 153  
 two chassis articulated vehicle, 538  
 two-bar manipulator, 792  
 two-bar mechanism, 231  
 two-bar spatial mechanism with two drivers, 155  
 two-body impact damper, 433  
 two-body slider-crank, 211  
 two-body spatial pendulum, 153
- unilateral spring**, 418, 420
- unilateral TSDA, 283
- union, 627
- unit coordinate vectors, 24
- unit of time, 251
- unit sphere, 11
- unit vector, 11, 22
- units, 251
- universal joint, 145, 147, 223
- user components of code, 202, 220, 340, 410, 439, 591, 603, 665, 682
- usolve, 415, 443
- variable distance constraint, 455
- variable step integrator, 421
- variable step size, 306, 326, 389, 419, 422
- variational equation of motion, 251, 254, 288-289, 301, 335, 351, 502, 574
- variational equation of motion for planar systems, 761
- variational equation of planar motion, 262
- variational equations of motion, 11, 249, 286, 288, 362
- variational equations of motion for systems of particles, 254
- variational methods in mechanics**, 253
- vector, 22
- vector calculus, 251
- vector identities, 30
- vector locating the centroid, 412
- vector of input coordinates, 775

vector of output coordinates, 775  
vector of problem data, 471  
vector product, 26, 30  
vector space, 44  
vector sum, 23  
vehicle mechanics, 3  
vehicles, 2  
velocity, 20, 32-33  
velocity analysis, 200  
velocity and acceleration analysis, 200  
velocity constraint, 11, 119-120, 515-516, 518  
velocity equation, 70, 200  
velocity space generalized coordinates, 522  
velocity tangent space, 512, 520  
velocity tangent space parameterization, 522  
virtual displacement, 20, 193, 251-252, 261, 267, 520, 574  
virtual displacement equations, 334  
virtual work, 249, 251, 283-284, 653  
virtual work of constraint reaction forces is zero, 574

**wave nature of motion**, 429  
wave speed, 429  
weight, 250  
well posed, 303, 363, 397, 470, 527, 624, 758  
well posed Full DAE of motion, 477  
well posed holonomic ODE, 476  
well posed holonomic variational equations, 477  
well posed Index 0 initial-value problem, 477  
well posed nonholonomic equations, 567  
well posed ODE and Index 0 DAE of motion, 476  
well posed problem, 470  
well posed variational equations of motion, 477  
wheels do not slip laterally, 539  
windshield wiper, 5, 135, 217

**xPart**, 415, 444, 595, 604

**you get what you pay for**, 336, 579

**zero matrix**, 37  
zero vector, 22