INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
A general, robust procedure for the kinematic and friction force analysis of single loop, one degree-of-freedom spatial mechanisms

Gutkowski, Lawrence Joseph, Ph.D.
The Ohio State University, 1990
A General, Robust Procedure for the Kinematic and Friction Force Analysis of Single Loop, One Degree-of-Freedom Spatial Mechanisms

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Lawrence Joseph Gutkowski, B.S.M.E., M.S.

* * * *

The Ohio State University

1990

Dissertation Committee

Gary L. Kinzel
Kenneth J. Waldron
Kos Ishii

Approved by

Gary L. Kinzel
Adviser
Mechanical Engineering
This work is dedicated to my wife, Judy.
ACKNOWLEDGEMENTS

I wish to thank the members of my committee, Dr. Gary Kinzel, Dr. Kenneth Waldron, and Dr. Kos Ishii, for their valuable suggestions regarding this dissertation. Their time spent in reviewing this work is appreciated. Special thanks are extended to my adviser, Dr. Gary Kinzel, for the patience and guidance offered over the past years. I have the utmost respect for his commitment and sense of responsibility to the educational process within the engineering discipline. I feel very fortunate to have had him as an adviser.

Unending thanks must go to my wife, Judy. Her belief in me goes beyond the call of duty. I doubt this dissertation would ever have been completed without her patience, understanding, and emotional support. For the years she has worked so I could study, I will always be grateful. She has truly been the sponsor of this research.

I also wish to thank my parents for an upbringing in which education was valued. Any abilities I might have now are necessarily built upon the education (and attitude toward that education) received as a child. My parents were always there to participate in and encourage that early learning process—thank you.
VITA

August 25, 1958 ............... Born—Garfield Heights, Ohio

1980 ......................... B.S.M.E., The Ohio State University, Columbus, Ohio

1984 .......................... M.S., The Ohio State University, Columbus, Ohio


PUBLICATIONS


FIELD OF STUDY

Major Field: Mechanical Engineering
# Table of Contents

<table>
<thead>
<tr>
<th>ACKNOWLEDGEMENTS</th>
<th>iii</th>
</tr>
</thead>
<tbody>
<tr>
<td>VITA</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xiv</td>
</tr>
</tbody>
</table>

## CHAPTER 1 Introduction

1.1 Overview ........................................................................... 1  
1.2 Basic Mechanism Terminology and Concepts .......................... 2  
1.3 Common Kinematic and Force Analysis Assumptions ............... 10  
1.4 General Types of Mechanism Analysis ................................ 13  
1.5 Research Scope and Statement of Objectives ...................... 14  

## CHAPTER II Mathematical Representations of Mechanisms

2.1 Introduction .................................................................... 16  
2.2 Coordinate Systems ..................................................... 17  
2.3 Coordinate Transformation Matrices ................................ 20  
2.3.1 Transformation Matrix Development ............................. 20  
2.3.2 Transformation Matrix Products ................................. 26  
2.3.3 Inverse Transformation Matrices ................................. 27  
2.3.4 Closed Loop Transformation Matrix Product .................. 29  
2.4 Rigid-Body-Affixed Coordinate Systems ............................. 30  
2.4.1 Unconstrained Motion ................................................ 31  
2.4.2 Constrained Motion ................................................... 36  
2.5 Systematic Approaches to Modeling Mechanism Kinematic Structure | 42  


2.5.1 Denavit and Hartenberg Approach ................................... 43
  Basic Principles ......................................................................... 43
  Planar Slider Crank Example ........................................................ 46
  Observations ............................................................................. 49
2.5.2 Sheth and Uicker Approach ................................................ 53
  Basic Principles ......................................................................... 54
  Shape Matrices. .......................................................................... 57
  Pair Matrices. ............................................................................. 60
  Planar Slider Crank Example ....................................................... 73
  Indirect Specification of Shape Matrices ....................................... 79

III Development of Three Dimensional Surface Contact Joints .......... 86
  3.1 Mathematical Development of Pair Matrix ................................. 87
    3.1.1 Development of a Floating Coordinate System at a Point on a Surface ...................................................... 90
    3.1.2 Floating Coordinate System and Its Transformations on a Spherical Surface .................................................. 93
    3.1.3 Development of the Transformation Between Two Floating Coordinate Systems in Contact ............................ 95
    3.1.4 Discussion ......................................................................... 98
  3.2 "Definition to Floating" and "Floating to Definition" System Transformations for Various Surfaces .............................. 99
    3.2.1 Spherical Surface ............................................................... 100
    3.2.2 Cylindrical Surface ........................................................... 101
    3.2.3 Plane Surface .................................................................... 103
    3.2.4 Conical Surface .................................................................. 105
    3.2.5 Paraboloid Surface ............................................................. 107
  3.3 Sphere-Cylinder Surface Contact Example .................................. 109
    3.3.1 Specifying the Pairing Element Coordinate Systems .......... 111
    3.3.2 Determining the Shape Matrices ........................................ 113
    3.3.3 Determining the Pair Matrices .......................................... 114
    3.3.4 Forming the Closed Loop Matrix Product ......................... 117
  3.4 Bicubic Surface Patch .............................................................. 118
    3.4.1 Background ....................................................................... 118
    3.4.2 Establishing a Floating Coordinate System on a Surface Patch ................................................................. 124
| IV Development of Kinematic Analysis Equations | 127 |
| 4.1 Introduction | 127 |
| 4.2 The Forward and Inverse Kinematic Problems | 127 |
| 4.3 Matrix Based Displacement Analysis | 129 |
| 4.3.1 Open Loop Mechanism | 130 |
| 4.3.2 Closed Loop Mechanism | 133 |
| 4.4 Matrix Based Kinematic Velocity Analysis | 134 |
| 4.4.1 Open Loop Mechanism | 136 |
| 4.4.2 Closed Loop Mechanism | 137 |
| 4.5 Matrix Based Kinematic Acceleration Analysis | 141 |
| 4.5.1 Open Loop Mechanism | 141 |
| 4.5.2 Closed Loop Mechanism | 144 |

| V Numerical Solutions for the Kinematic Equations | 148 |
| 5.1 Introduction | 148 |
| 5.2 Optimization Approach to the Displacement Analysis of Mechanisms | 150 |
| 5.2.1 Background to the Optimization Approach | 151 |
| 5.2.2 A Displacement Based Objective Function for a Closed Loop Mechanism | 153 |
| 5.2.3 The Davidon-Fletcher-Powell Minimization Method | 154 |
| 5.2.4 Discussion of Optimization Approach | 160 |
| 5.3 Newton's Method Approach to the Displacement Analysis of Mechanisms | 164 |
| 5.3.1 Background to the Newton's Method Approach | 165 |
| 5.3.2 Nonlinear Displacement Equations for a Newton's Method Solution | 170 |
| 5.4 A Robust Solution Procedure Using Singular Value Decomposition | 174 |
| 5.4.1 Background | 175 |
| 5.4.2 Matrix Based Linear Systems of Equations | 177 |
| 5.4.3 Singular Value Decomposition | 180 |
| 5.4.4 Use of Singular Value Decomposition with Newton's Method | 184 |
| 5.4.5 Numerical Computation of the Jacobian | 190 |
| 5.5 Numerical Formulation and Solution for the Velocity and Acceleration Analyses | 193 |
| 5.5.1 Formulation of the Velocity Analysis System of Equations | 194 |
5.5.2 Formulation of the Acceleration Analysis System of Equations ...................................................... 199
5.6 Calculation of Kinematic Properties From the Joint Variable Solutions ...................................................... 204
5.6.1 Angular Velocities of the Links ...................................................... 205
5.6.2 Angular Accelerations of the Links ...................................................... 209
5.7 Position, Velocity, and Acceleration of Points of Interest ...................................................... 211

VI Dynamic Force Analysis ...................................................... 214
6.1 Dynamic Force Equilibrium ...................................................... 215
6.2 Joint Force Constraint Equations ...................................................... 223
6.2.1 Joint Force Constraint Equations: Frictionless ...................................................... 225
   Revolute Joint ...................................................................................... 226
   Prismatic Joint ...................................................................................... 226
   Helical Joint ...................................................................................... 226
   Cylindrical Joint ...................................................................................... 226
   Planar Joint ...................................................................................... 227
   Spherical Joint ...................................................................................... 227
   Surface Contact Joint ...................................................................................... 227
6.2.2 Joint Force Constraint Equations: Coulomb Friction ...................................................... 230
   Revolute Joint ...................................................................................... 231
   Prismatic Joint ...................................................................................... 234
   Helical Joint ...................................................................................... 234
   Cylindrical Joint ...................................................................................... 235
   Planar Joint ...................................................................................... 235
   Spherical Joint ...................................................................................... 236
   Surface Contact Joint ...................................................................................... 246
6.3 Solutions to the Force System of Equations ...................................................... 250
6.3.1 Solution to Linear Frictionless Force Analysis ...................................................... 251
6.3.2 Solution to Nonlinear Friction Force Analysis ...................................................... 253

VII Example Mechanism Analyses ...................................................... 260
7.1 Kinematic Analysis of a Planar RRRR Mechanism ...................................................... 261
7.2 Kinematic Analysis of a “Planar” RSSR Mechanism ...................................................... 266
7.3 Kinematic Analysis of a Spatial Slider Crank Mechanism ...................................................... 272
7.4 Bent-Shaft Universal Joint—A Surface Contact Mechanism ...................................................... 275
## List of Figures

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The use of more than one coordinate system to define the same point. 19</td>
</tr>
<tr>
<td>2</td>
<td>Two unconstrained bodies with coordinate systems affixed. 32</td>
</tr>
<tr>
<td>3</td>
<td>Two constrained bodies with coordinate systems affixed. 38</td>
</tr>
<tr>
<td>4</td>
<td>Two joined bodies with strategically affixed coordinate systems. 41</td>
</tr>
<tr>
<td>5</td>
<td>Denavit and Hartenberg definition of link local coordinate systems. 45</td>
</tr>
<tr>
<td>6</td>
<td>Slider crank mechanism parameters using Denavit and Hartenberg notation. 47</td>
</tr>
<tr>
<td>7</td>
<td>A “bent-shaft” universal joint. 52</td>
</tr>
<tr>
<td>8</td>
<td>Link coordinate systems for Sheth and Uicker approach. 56</td>
</tr>
<tr>
<td>9</td>
<td>Geometric relationship between the (xyz) and (uvw) coordinate systems involved in the Sheth-Uicker link shape matrix. 58</td>
</tr>
<tr>
<td>10</td>
<td>Description and specifications for Revolute Pair matrix. 62</td>
</tr>
<tr>
<td>11</td>
<td>Description and specifications for Prismatic Pair matrix. 63</td>
</tr>
<tr>
<td>12</td>
<td>Description and specifications for Helical Pair–Type 1 matrix. 64</td>
</tr>
<tr>
<td>13</td>
<td>Description and specifications for Helical Pair–Type 2 matrix. 65</td>
</tr>
<tr>
<td>14</td>
<td>Description and specifications for Cylindric Pair matrix. 66</td>
</tr>
<tr>
<td>15</td>
<td>Description and specifications for Spherical Pair–Type 1 matrix. 67</td>
</tr>
<tr>
<td>16</td>
<td>Description and specifications for Spherical Pair–Type 2 matrix. 68</td>
</tr>
<tr>
<td>Page</td>
<td>Description and specifications for Spherical Pair-Type 3 matrix.</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>18</td>
<td>Description and specifications for Spherical Pair-Type 4 matrix.</td>
</tr>
<tr>
<td>19</td>
<td>Description and specifications for Spherical Pair-Type 5 matrix.</td>
</tr>
<tr>
<td>20</td>
<td>Description and specifications for Planar Pair matrix.</td>
</tr>
<tr>
<td>21</td>
<td>Sheth-Uicker parameters for the planar slider-crank mechanism.</td>
</tr>
<tr>
<td>22</td>
<td>General binary link and its coordinate systems.</td>
</tr>
<tr>
<td>23</td>
<td>A three-dimensional surface contact joint.</td>
</tr>
<tr>
<td>24</td>
<td>Floating systems on two surfaces in contact.</td>
</tr>
<tr>
<td>25</td>
<td>Definition and floating systems for a spherical surface.</td>
</tr>
<tr>
<td>26</td>
<td>Definition and floating systems for a cylindrical surface.</td>
</tr>
<tr>
<td>27</td>
<td>Definition and floating systems for a planar surface.</td>
</tr>
<tr>
<td>28</td>
<td>Definition and floating systems for a conical surface.</td>
</tr>
<tr>
<td>29</td>
<td>Definition and floating systems for a paraboloid surface.</td>
</tr>
<tr>
<td>30</td>
<td>A mechanism possessing a surface contact joint.</td>
</tr>
<tr>
<td>31</td>
<td>Pairing element coordinate systems for mechanism with surface contact joints.</td>
</tr>
<tr>
<td>32</td>
<td>Four views of a Coons surface patch with circular boundary curves.</td>
</tr>
<tr>
<td>33</td>
<td>The boundary curves of a surface patch.</td>
</tr>
<tr>
<td>34</td>
<td>Schematic of an RSSR mechanism with an idle degree-of-freedom.</td>
</tr>
<tr>
<td>35</td>
<td>General joint force configuration between two links.</td>
</tr>
<tr>
<td>36</td>
<td>The revolute joint friction model.</td>
</tr>
<tr>
<td>37</td>
<td>Bearing forces acting on the &quot;ball&quot; of a spherical joint.</td>
</tr>
<tr>
<td>38</td>
<td>A view looking down the instant screw axis of a sphere sectioned at the plane where the bearing loads are applied.</td>
</tr>
</tbody>
</table>
39 A planar four-bar mechanism .................................................. 262
40 Closed form displacement results, \( \theta_4 \), for planar RRRR. .......... 263
41 Closed form velocity results, \( \omega_{4,1} \), for planar RRRR. ............ 263
42 Closed form acceleration results, \( \alpha_{4,1} \), for planar RRRR. ........ 264
43 Matrix based solution displacement results, \( \theta_4 \), for planar RRRR. ... 267
44 Matrix based solution velocity results, \( \omega_{4,1} \), for planar RRRR. .... 267
45 Matrix based solution acceleration results, \( \alpha_{4,1} \), for planar RRRR. ... 268
46 Matrix based solution displacement results, \( \theta_4 \), for quasi planar RSSR. ... 269
47 Matrix based solution velocity results, \( \omega_{4,1} \), for quasi planar RSSR. ... 270
48 Matrix based solution acceleration results, \( \alpha_{4,1} \), for quasi planar RSSR. ... 270
49 Spatial slider crank analyzed by Sandor and Erdman.[22] ............... 273
50 Displacement, velocity, and acceleration of the RSSP slider as obtained by Sandor and Erdman.[22] .................................................. 274
51 Pairing elements coordinate systems for RSSP mechanism. .......... 276
52 Matrix based solution displacement results for the RSSP slider. .... 277
53 Matrix based solution velocity results for the RSSP slider. ............ 277
54 Matrix based solution acceleration results for the RSSP slider. .... 278
55 Bent-shaft universal joint mechanism example. ......................... 279
56 Pairing element coordinate systems for the bent-shaft universal joint links. ................................................................. 281
57 Output shaft displacement results for the bent-shaft universal joint using a surface contact joint. ........................................ 282
58 Output shaft velocity results for the bent-shaft universal joint using a surface contact joint. ................................................. 283

xii
59 Output shaft acceleration results for the bent-shaft universal joint using a surface contact joint. 284
60 Displacement of the contact point on the input and output shafts of the bent-shaft universal joint using a surface contact joint. 285
61 Equivalent lower pair model for the bent-shaft universal joint 286
62 Output shaft displacement results for the bent-shaft universal joint using lower pair joints. 288
63 Output shaft velocity results for the bent-shaft universal joint using lower pair joints. 289
64 Output shaft acceleration results for the bent-shaft universal joint using lower pair joints. 290
65 Required input shaft torque for the bent-shaft universal joint 290
66 A swash plate mechanism with spherical follower. 291
67 Matrix based solution displacement results for the swash plate mechanism. 292
68 Matrix based solution velocity results for the swash plate mechanism. 293
69 Matrix based solution acceleration results for the swash plate mechanism. 293
70 Input torque required for frictionless and friction cases of the swash plate mechanism. 294
71 A slider crank mechanism analyzed with Coulomb friction 295
<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39</td>
</tr>
<tr>
<td>2</td>
<td>85</td>
</tr>
<tr>
<td>3</td>
<td>135</td>
</tr>
<tr>
<td>4</td>
<td>159</td>
</tr>
<tr>
<td>5</td>
<td>169</td>
</tr>
<tr>
<td>6</td>
<td>191</td>
</tr>
<tr>
<td>7</td>
<td>197</td>
</tr>
<tr>
<td>8</td>
<td>202</td>
</tr>
<tr>
<td>9</td>
<td>254</td>
</tr>
<tr>
<td>10</td>
<td>259</td>
</tr>
</tbody>
</table>
CHAPTER I

Introduction

1.1 Overview

The work presented in this dissertation is intended to provide the basis for a
general approach to spatial mechanism kinematic and force analysis. Although
various well established methods exist for mechanism kinematic and force analysis,
many of these employ explicit or implicit simplifying assumptions that limit their
extension to more complicated analyses. Additionally, some of the existing analysis
techniques rely on specific mechanism geometries for successful solutions. These do
not provide a general procedure. The approach described in the following course of
research was developed specifically with generality and future expansion in mind.
Specific extensions might have to do with simply providing "new" joint types or
(on a more complex scale) dealing with dimensional tolerances of mechanism links
and joint clearances.

In this interest of generality and extension, a very modular approach to kine-
matic and force analysis has been taken in setting up the analysis procedures.
This provides a "building block" approach to mechanism modeling. Such an ap-
proach also lends itself well to computer implementation which has become almost
requisite for kinematic analyses. By modularizing the modeling and analysis of a
mechanism, the enhancement of the associated analysis program will itself become
a "building block" exercise. Advancing the capabilities of an existing program is facilitated by the ability to provide "plug-in" modules as new analysis tools are needed and developed.

1.2 Basic Mechanism Terminology and Concepts

A mechanism is typically described as a collection of interconnected members that can move relative to each other for the purpose of transmitting motion or force. One might think of the mechanism as a "black box" into which input motions or forces are applied and output motions or forces are produced. The black box represents highly nonlinear transfer functions relating the mechanism's output(s) to its input(s). It is the goal of a kinematic analysis to determine the form (either algebraically or numerically) of these so called transfer functions.

The members that make up a mechanism are referred to as "links" and the interconnections between the members as "joints." Kinematically speaking, a joint between two links is intended to permit constrained relative motion between those links. The locations on a link at which joint connections are made are often referred to as "pairing elements." A link may have any number of pairing elements which means it may be connected to any number of other links. The most common links have two, three, or four pairing elements and are called binary, ternary, and quaternary links respectively.

Any type of contact between two bodies that allows relative motion can be considered a kinematic joint. Joints are usually divided into two groups called lower pairs and higher pairs. A lower pair joint is defined as one in which links are joined through surface contact. A higher pair joint is characterized by line or point con-
tact. It has been shown that only six lower pair joints exist. They are the revolute, prismatic, cylindrical, spherical, helical, and planar joints. The lower pair joints have received the most attention in kinematic analysis. They are easily described in mathematical terms. Because of their widespread use, the lower pair joints have been given a symbolic notation that can be used to describe a mechanism's kinematic structure. For instance, the closed loop, four-link mechanism using four revolute joints is often referred to as an \(-R-R-R-R-\) mechanism. The planar slider crank is indicated by \(R-R-R-P-\). A six-axis open loop industrial robot mechanism containing six revolute joints would be labeled as \(-R-R-R-R-R-R-\).

Higher pair joints must be dealt with on a more individual basis. It is not possible to define a "minimum" number of higher pair joint types that describe all of the possibilities. As such, the mathematical modeling of higher pair joints is more complex than the modeling of the lower pair joints. It is much more difficult to develop a generalized analysis scheme when higher pair joints are considered. The joint formed by two meshing gears represents a higher pair characterized by line contact. The bent shaft universal joint as analyzed by Beggs [66] uses point contact, and therefore, is also a higher pair joint.

The number of independent movements permitted by a given joint is said to be the "degree-of-freedom" (dof) of that joint. It is well known that six independent quantities (i.e., dofs) are necessary to completely describe the general spatial location and orientation of one body relative and unattached to another. If these two bodies are connected by a kinematic joint, the number of quantities needed to describe their relative location and orientation will be less than six—and greater than zero if relative movement is to be possible. The actual number depends on the
type of joint. A simple hinge joint (revolute joint) for instance would possess one 
degree-of-freedom (i.e., a rotation about the hinge axis). A cylindrical joint would 
possess two degrees-of-freedom (i.e., a rotation about its axis and a translation 
along its axis).

Mechanisms will often be classified as “open loop” or “closed loop” mechanisms. 
This description has to do with kinematic structure and not with any sort of a 
mechanism control system (i.e., open or closed loop control as defined in classical 
control theory). An open loop mechanism is one in which the constituent links are 
connected in a chain, end to end, with the last link unconnected to the first. A 
closed loop mechanism results when the “last” link in a chain is connected to the 
first such that a physical loop results. It is common for complex mechanisms to 
consist of multiple closed loops. Another common classification for a mechanism 
has to do with the overall motion its links exhibit. Mechanisms whose link motions 
are constrained to move in a single plane (or a number of parallel planes) are called 
planar mechanisms. Similar to the planar mechanism is the spherical mechanism 
whose links are constrained to move on the surface of a sphere (or a number of 
concentric spherical surfaces). The most general mechanism allows one or more of 
its links to have motions that are not bounded by planar or spherical motion.

In order for a mechanism to be generally useful, it must have one of its links 
“fixed” or specified as a frame of reference. This is often called the “ground” link 
or “frame”. A mechanism in a factory might have its ground link fastened to the 
factory floor which in turn makes the Earth the frame of reference. A mechanism 
located in outer space might have its ground link attached to the wall of a space 
station wherein the space station becomes the frame of reference. (However, we
could also view the mechanism in an inertial reference frame relative to the fixed stars in order to investigate the mechanism dynamics).

A kinematic analysis usually consists of a position, a velocity, and an acceleration analysis. The position analysis basically involves specifying values for a certain number of joint dofs (i.e., independent dofs) within the mechanism and solving for all of those that remain (i.e., dependent dofs). The number of independent dofs that need to be specified in order to make the mechanism’s configuration determinant is often referred to as the degree-of-freedom of the mechanism. The dof of a given mechanism is a function of the number of links, the number of joints, the types of joints, and the link geometry. The dof of a mechanism is not necessarily equal to the sum of its constituent joint dofs. In fact, for closed loop mechanisms this is certainly not the case. Thus there are two conceptual facets when speaking of degrees-of-freedom in the kinematic sense. One aspect has to do with joint dofs while the other concerns the mechanism dofs. (Note: The dof of a joint is often referred to as its connectivity, and the dof of a mechanisms is sometimes called its mobility).

It is possible to predict a mechanism’s dof using an equation most often credited to Grüber. This equation is based solely on the number of links, the number of joints, and the types of joints (i.e., their dofs) found in a mechanism. Geometry is not considered at all. A general mechanism’s dof is predicted to be:

\[ D_m = 6 (n - j - 1) + \sum_{i=1}^{j} f_i \]  

where  
\[ D_m = \text{predicted dof of mechanism} \]  
\[ n = \text{number of links in mechanism} \]  
\[ j = \text{number of joints in mechanism} \]
\[ f_i = \text{dof of } i\text{-th joint} \]

If \( D_m \) is greater than zero, it would theoretically require \( D_m \) independent inputs to make the mechanism determinant. If \( D_m \) is zero, the "mechanism" should actually be a statically determinant structure. When \( D_m \) is less than zero, a statically indeterminant structure should result. Since this equation does not take link geometry into account, it can often produce erroneous result. A very good example is for the case of the planar RRRR mechanism (all revolute axes are parallel). Grübler's equation would indicate a \( D_m \) of \(-2\) for this mechanism (i.e., a statically indeterminant structure should be realized). However, it is well known that the planar RRRR mechanism actually has one dof and moves quite readily.

The explanation as to why Grübler's equation fails in certain situations has to do with mechanism geometry. In the case of the planar RRRR mechanism, the condition that all of the revolute axes be parallel is what allows that mechanism to move. In other words, this mechanism exploits a very special geometric condition in order to have mobility. If the four revolute axes were oriented such than none were parallel to any other, the RRRR mechanism would indeed be a statically indeterminant structure with a dof of \(-2\). As it turns out, most planar mechanisms will be seen as structures by Grübler's equation, yet they are actually able to move because of special geometry.

It is possible to modify Grübler's equation such that it deals correctly with planar mechanisms. The development of his equation begins with \( n \) unconnected bodies. Since 6 parameters are needed to describe the spatial location and orientation of one of the bodies relative to some reference, \( 6n \) parameters are needed to specify the whole system. As the bodies are joined, the \( 6n \) parameters needed are
reduced as joints (constraints) are added. The fact that 6 parameters per body are needed in general gives rise to the coefficient of 6 in Grübler's equation. If instead, Grübler's equation is developed with planar motion strictly in mind, only 3 parameters are needed to specify the planar location and orientation of a body. Hence, the coefficient to Grübler's equation can be changed to a 3 when it is applied to a planar mechanism. Of course, one must be able to determine that a particular mechanism is in fact planar in order for this method to be reliable (i.e., a certain level of insight is required).

Unfortunately, there are mechanisms that fail both the general and the planar forms of Grübler's equation. These again, are due to special geometry features that are not considered in the equation. In a manner of speaking, such a mechanism is not specialized enough to be considered truly planar, yet there is something about its geometry that makes it less than a general spatial mechanism.

Many practical mechanisms in use possess a single degree-of-freedom. The obvious reason for this is that only one driving force (input quantity) is needed to run the mechanism. Multiple dof mechanisms often require elaborate control schemes to coordinate the multiple input quantities. A very good example of this lies in an industrial robot. These typically have six or more dof and as a result require sophisticated computer control systems to be useful.

Certain instances of multi-dof mechanisms can behave (and be used) as single-dof mechanisms. Such mechanisms contain one or more so called idle degrees-of-freedom. An idle dof exists when a given link in a mechanism is able to move relative to all of the others without requiring that they also move. A good example of this is a four link mechanism made up of two revolute and two spherical joints
(RSSR). If the revolute axes are parallel to each other, the gross motion of the mechanism will behave as though all it were a planar RRRR mechanism (all of its axes parallel). However, due to the two spherical joints, the third link is able to spin about its own axis without restraint and without affecting the positions of the other links. This constitutes an idle degree-of-freedom.

The spherical joints make this a spatial mechanism. Using the general form of Grübler's equation, one correctly gets a $D_m$ of 2 for this mechanism. The typical "crank" input associated with a planar RRRR device represents one of these dof. The second dof is manifested as an angular displacement of link 3 about an axis through the two spherical joints. But because this represents an idle degree-of-freedom, it is possible to let this input "float" at any value desirable since it will not affect the overall motion of the mechanism. An important point to note is that idle-dofs are usually detected by inspection which requires some insight on the part of the analyst.

Whether or not Grübler's equation is valid for a mechanism, a mechanism will possess a certain number of degrees-of-freedom. As mentioned previously, this represents the number of independent quantities that must be specified in order to perform a position analysis. The results of the position analysis will produce values for all of the remaining (dependent) joint dofs in the mechanism based on the specified inputs. The velocity analysis then requires that the first time rates of change of these independent dofs be specified in order to determine the first time rates of change of the dependent joint dofs. Similarly, the second time rates of change for the input dofs must be specified for the acceleration analysis which will yield the second time rates of change for the dependent joint dofs. Once the
positional, velocity, and acceleration values are known for the all of the joint dofs, one may calculate any displacement, velocity, or acceleration information of any point or body within the mechanism.

The position analysis of a mechanism is usually the most difficult when compared to the velocity or acceleration analysis. It is the nature of the position analysis to be highly nonlinear in terms of the unknown (dependent) joint dofs. The velocity and acceleration analyses are more straightforward in that they are linear in the unknown first and second time rates of change of the unknown joint dofs. When attempting to generalize any of the stages of kinematic analysis, susceptibility to various mathematical pathologies such as over and underconstrained systems of equations is also common. For instance, a dependent system of equations is obtained when analyzing a planar mechanism as a spatial mechanism (three of the equations are trivial but are none-the-less linearly dependent).

Once a kinematic analysis has been finished for a given mechanism, it is possible to perform a dynamic force analysis. Obviously, the acceleration analysis results are needed to determine the inertia forces experienced by the links of the mechanism. A force analysis in which the joints are assumed to be frictionless is relatively straightforward. Force balances are written for each link in the mechanism (except the ground link) as are equations representing force constraints for each joint degree-of-freedom. This system of equations will be linear in terms of the unknown joint (bearing) forces and is easily solved.

The case where joint friction is considered, however, is more involved in that it becomes a nonlinear problem. Joint friction adds nonzero bearing forces about the joint degree-of-freedoms to the system of equations. These friction forces are
related to bearing force components that are perpendicular to the joint surfaces. The friction forces will always oppose the direction of relative motion in the joints. Hence, their directions can be determined from the velocity analysis. This opposition to the relative velocity becomes the source of the nonlinearity in the friction force problem.

1.3 Common Kinematic and Force Analysis Assumptions

All engineering or scientific analysis procedures rely on certain sets of assumptions that may limit the analysis scope and validity. These assumptions are often necessary in order to achieve tractable solutions. Other times, assumptions are used to filter out negligible effects or effects which are of no interest to the objectives in mind. It is very important for an analysis procedure to be up front with the assumptions that are employed in the course of solution. There are a number of assumptions that are made often in kinematic and force analyses of mechanisms. The most common assumptions have to do with link rigidity, dimensional tolerances, joint clearances, and joint friction.

The vast majority of kinematic analyses assume that every link in a mechanism is completely rigid. This greatly simplifies the analysis procedure in that the distances between and orientations of pairing elements on a given link will be constant. When links are assumed to be elastic, the configuration of a mechanism is not only a function of the input dofs but also a function of the mechanism loading. Elastic links will deform under load according to their mechanical properties (which in general could be nonlinear behavior). Hence, the kinematic analysis cannot be performed independently of the force analysis unless rigid links are assumed. This
can be a valid assumptions for mechanisms whose links are massive compared
to applied loads (i.e., the strain displacements of the links are sufficiently small).
However, there are mechanisms that by necessity or function are designed to behave
as flexible devices. For these mechanisms, the rigidity assumption would produce
invalid results.

It is also very common for kinematic analyses to assume that all links and joints
are dimensionally and geometrically perfect. This assumption would probably be
quite valid for a one-of-a-kind mechanism whose actual link dimensions could be
measured and used in an analysis. However, if a kinematic analysis is to predict the
behavior of mass produced mechanisms, it is obvious that manufacturing tolerances
will cause the dimensions of a real life link to deviate from its theoretical design
dimensions. This will of course affect the actual kinematic behavior.

Another aspect of the "perfect" mechanism has to do with the treatment of
joint clearances. Most kinematic analyses to date assume that no play exists in the
joints of a mechanism. In reality this would be impossible. A joint (especially the
lower pair joints) must have a degree of clearance between the mating parts in order
for the joint to be movable. This brings up a number of interesting considerations.
First of all, if clearance exists in a lower pair joint, the ideal of surface contact
yields to the reality of point or line contact. Hence, a real life "lower" pair joint
most likely behaves as a higher pair joint in the strict sense of the terminology.

Additionally, the manifestation of "play" in a particular joint is mathematically
dealt with as additional degrees-of-freedom in the joint. Hence, a real life revolute
joint is no longer a 1 dof joint. Additional dofs must be considered in order to deal
with the joint clearances. The existence of clearances also leads to the possibility
of momentary contact loss between the two pairing elements comprising the joint. In this case, the "joint" actually becomes a problem involving the dynamics of two independent bodies. Finally, the concept of planar versus spatial mechanisms becomes muddled. A mechanism (say an RRRR) that is planar in the ideal sense will become a spatial mechanism if joint clearances allow any of the links to move out of the plane. This motion could be induced by out of plane loads from external or inertial forces and torques.

Although the assumptions dealing with link rigidity, dimensional tolerances, and joint clearances have been discussed from their effect on kinematic analysis, they also affect the force analysis. A further assumption directly related to the force analysis has to do with joint friction. Many analyses treat joint friction as negligible. Those that do consider friction must then make an assumption as how to model the friction mathematically. One of the most common models is based on Coulomb friction. This technique states that the friction force is proportional to the component of bearing force normal to the joint surfaces at the point of joint contact. This type of friction is independent of the relative velocity between the mating joint surfaces. The proportionality constant is commonly known as the coefficient of friction. This type of friction model is probably more appropriate for joints that are not lubricated. This might be called "dry friction." In certain circumstances it may be more appropriate to use a viscous friction model where a film of lubrication exists. This type of model tends to be much more complicated than the Coulomb friction model. It is nonlinearly related to the lubricant viscosity, the sliding velocity, and the lubricant pressure.
1.4 General Types of Mechanism Analysis

The literature dealing with motion related studies of mechanisms seems to deal with three major areas. These areas involve the kinematic analysis of mechanisms, the kinematic synthesis of mechanisms, and the dynamic response of mechanisms. Kinematic analysis involves determining the motion characteristics of a fully specified mechanism. Kinematic synthesis on the other hand starts out with desired motion properties and then attempts to design a mechanism that will produce those properties. Kinematic analysis and kinematic synthesis are complements of each other; it is common for a design loop to include aspects of both synthesis and analysis. The third major type of mechanism study is the dynamic response analysis. This involves the determination of mechanism motion due to applied forces.

Kinematic analysis of a mechanism can be approached from two different aspects. In one situation, the input to a given mechanism is specified. It is then desired to solve for the output that results. In the other situation, the output required of a given mechanism may be specified and the corresponding input is to be determined. This second approach is often called the "inverse kinematics problem." Note that the inverse problem is not the same thing as a synthesis problem—the inverse problem is performed on a known mechanism.

It is also common in the literature to see mechanisms being analyzed from the standpoint of dynamic response. In this case, a forcing function is applied as an input to the mechanism and one desires to determine how the mechanism behaves dynamically. The forcing function(s) are usually function(s) of time, and therefore, the response solution is expressed as function of time. The response
would be comprised of the positions, velocities, and accelerations of all of the links in the mechanism. Dynamic response analyses are often used to investigate the effects of contact loss in mechanism joints. When contact loss takes place, there are no kinematic constraints that exist within the joints. A dynamic response analysis is the only means by which to analyze this occurrence. Additionally, a dynamic analysis can be used to assess the impact effects when joint contact is reestablished after a separation occurs.

1.5 Research Scope and Statement of Objectives

It is the goal of the research described in this dissertation to provide a kinematic and friction force analysis procedure for spatial mechanisms. A "building block" (i.e., modular) approach to mechanism modeling will be pursued to facilitate the development of a general and extendable procedure. A method for dealing with higher pair joints will be discussed and modeled in terms of this general approach.

The procedure will be robust in its behavior with "special case" situations such as underconstrained mechanisms (e.g., those with idle dofs) or overconstrained mechanisms (e.g., planar mechanisms analyzed as spatial). The mathematical consequences of these special cases are to be automatically (detected and) handled by the procedure with a minimum of insight required of the analyst. A computer program will be written to verify the principles developed in this course of research. Since a major goal of this research is concerned with a completely general approach to spatial mechanism analysis, the solution procedure will be numerical (as opposed to algebraic) in nature.

As with any program of research, a certain number of assumptions are necessary
in order to provide a reasonable scope of investigation. The scope of analysis characterized by the work in this dissertation is based on the following assumptions and conditions:

1. The kinematic/force analysis will be limited to single closed loop, single dof mechanisms. The procedures developed, however, are applicable and extendable to more complex mechanisms such as multi-dof mechanisms, multi-loop mechanisms, etc. Note that multi-dof mechanisms will be considered by this research where all but one of them are idle-dofs.

2. All mechanisms will be assumed perfect. Links are taken to be rigid and their dimensions as exact. All joints are assumed to have no clearances.

3. The friction force analysis will use a friction model based on the Coulomb (i.e., dry friction) model.
CHAPTER II
Mathematical Representations of Mechanisms

2.1 Introduction

This chapter will present a systematic, mathematical means of modeling mechanisms. The approach described makes use of coordinate transformation matrices. A transformation matrix describes the relative spatial location and orientation between two coordinate systems. By embedding coordinate systems in the links of a mechanism, transformation matrices can also be construed to describe the relative positions and orientations of the links. In the context of mechanism analysis, the transformation matrices will be determined by link dimensions and joint pair types. For a given mechanism, the link dimensions will be constant, and the degrees-of-freedom of the joints will be variable. The transformation matrix description of a mechanism yields kinematic equations directly that relate mechanism motion to joint dof variables. The solution to these equations will be discussed in the next chapter.

Denavit and Hartenberg are credited with bringing this matrix technique into widespread use. Their approach is most easily applied to mechanisms containing joints that permit motion along well defined axes such as the revolute and prismatic joints. In fact, link definition using their method is based on the perpendicular distance and the angle of twist between successive joint axes. Cylindrical,
planar, and spherical joints are considered by using revolute and prismatic joints in kinematically equivalent combinations. An alternative matrix based technique for mechanism modeling has been developed by Sheth and Uicker. Their methods will be shown to be more "building block" oriented than Denavit and Hartenberg's technique and thereby more amenable to computer implementation. Additionally, Sheth and Uicker's approach does not rely on axial motion types of joints. A means of modeling joints formed by three-dimensional surface contact will be presented in Chapter III for use with this Sheth and Uicker method of modeling mechanisms.

2.2 Coordinate Systems

For the purposes of kinematic analysis, coordinate systems are assumed to be right-handed, three-dimensional, rectangular cartesian systems. That is, they consist of three mutually perpendicular unit vectors, \( \hat{i} \), \( \hat{j} \), and \( \hat{k} \), taken in this order, such that

\[
(i \times j) \cdot \hat{k} = 1. \tag{2.1}
\]

These unit vectors all emanate from the same point. This point is called the origin of the coordinate system. The \( \hat{i}, \hat{j}, \hat{k} \) unit vectors define the traditional cartesian \( x, y, z \) axes respectively. The term coordinate system is synonymous with reference frame, frame of reference, frame, and system.

Coordinate systems are useful for the definition of points. A point is a location in space relative to the origin of a specified coordinate system. A three-dimensional point is defined by a triple of numbers, say \((a, b, c)\). This triple corresponds to coefficients, in order, for the \( \hat{i}, \hat{j}, \hat{k} \) unit vectors. A point may then be located within a specified coordinate system by following, from the system's origin, a path
defined by \( a \) units in the \( x \) direction, \( b \) units in the \( y \) direction, and \( c \) units in the \( z \) direction.

Note that when specifying the coordinates of a point, it is important to indicate the coordinate system being used for definition. The same point observed from two different coordinate systems will have different coordinates for each system as demonstrated in Figure 1. The same point \( P \) has been specified in the two different coordinate systems labeled 1 and 2. Point \( P \)'s coordinates in systems 1 and 2 are \((2, 4, 5)\) and \((4, 3, 2)\) respectively.

Figure 1 also reveals that the coordinates of a point in a given system are identical to the components of a position vector anchored at the origin of the system and directed to the point. Once again, the components of such a position vector are only meaningful if the defining coordinate system is specified. Therefore, a position vector quantity is formally symbolized and "pronounced" as:

\[ I \vec{r}_{p/q} = \text{the position of point } p \text{ relative to point } q \text{ defined in system } I. \]

This vector specification has point \( q \) as its "tail" and point \( p \) as its "tip." Note the use of a "pre-superscript" to indicate the coordinate system (or reference frame) being used. The pre-superscript may be omitted, however, if doing so is not ambiguous. This type of vector notation is also useful (necessary!) for velocity and acceleration vectors—here too, the frames of reference used are of extreme significance and must be indicated.

\(^{1}\)To be precise, \( I \vec{r}_{p/q} \) is actually the position of point \( p \) relative to the origin of a system whose origin is coincident with point \( q \). The system is taken to be parallel with system \( I \).
Figure 1: The use of more than one coordinate system to define the same point.
2.3 Coordinate Transformation Matrices

Referring again to Figure 1, the point $P$ is described relative to two different coordinate systems, 1 and 2. As noted, this results in two different sets of coordinates for the point $P$. A given set of coordinates, therefore, has meaning only if a coordinate system is also specified in which the coordinates are defined.

The difference between systems 1 and 2 in Figure 1 is seen easily—system 2 has a different attitude than system 1 (i.e., it is rotated from system 1), and the origin of system 2 has a different location from the origin of system 1 (i.e., it is translated from the origin of system 1). It is possible to describe the rotation and translation of a coordinate system relative to another system in quantifiable manner. Furthermore, this (mathematical) description allows the coordinates of a point given in one system to be transformed into coordinates defined in another system. This process will be shown to be conveniently carried out through the use of transformation matrices. The coordinates of a point defined in one system can be operated upon by a transformation matrix to yield coordinates for that same point but relative to a different system.

2.3.1 Transformation Matrix Development

Refer again to Figure 1—if the coordinates of $P$ in system 2 are given, and the orientation and location of system 1 relative to 2 is known, the ability to determine (mathematically) the coordinates of $P$ in system 1 will be useful. As stated previously, the coordinates of a point defined in a given system are identical to the components of the position vector of that point relative to the origin of the
system. Hence,

the coordinates of point \( P \) in system 1 \( = 1r_{p/o1} \)

the coordinates of point \( P \) in system 2 \( = 2r_{p/o2} \).

Therefore, the task of coordinate transformation can be restated as: "Given \( 2r_{p/o2} \) along with the orientation and location of system 1 relative to system 2, determine \( 1r_{p/o1} \)."

An expression for \( 1r_{p/o1} \) can be written using vector addition:

\[
1r_{p/o1} = 1r_{p/o2} + 1r_{o2/o1}. \tag{2.2}
\]

Note that each term of (2.2) is expressed in terms of system 1. Vector addition is carried out by adding together the respective \( x, y, z \) components of each vector—this can only be valid if all of the components are expressed relative to the same system. Since the position of \( P \) relative \( O_2 \) is given in system 2 coordinates, these known components of \( 2r_{p/o2} \) must somehow be converted to components meaningful in system 1. This would yield \( 1r_{p/o2} \) which is the first term on the right-hand side of (2.2).

The left-hand side of (2.2), \( 1r_{p/o1} \), is the position of \( P \) relative to the origin of system 1 defined in system 1 coordinates. Expressing its components as \( 1x_1, 1y_1, \) and \( 1z_1 \), this vector may be written as:

\[
1r_{p/o1} = 1x_1\hat{i}_1 + 1y_1\hat{j}_1 + 1z_1\hat{k}_1. \tag{2.3}
\]

The second term of (2.2)'s right-hand side, \( 1r_{o2/o1} \), is simply the position of the origin of system 2 relative to the origin of system 1 defined in system 1. Let
its components be represented by \( ^1x_0, ^1y_0, \) and \( ^1z_0 \). Thus,

\[
^1\vec{r}_{O_2/O_1} = ^1x_0 \hat{i}_1 + ^1y_0 \hat{j}_1 + ^1z_0 \hat{k}_1. \tag{2.4}
\]

The \( x, y, z \) components of (2.4) are considered to be known if the relative orientation and location of the systems are known.

As stated, the first term on the right-hand side of (2.2), \( ^1\vec{r}_{P/O_2} \), is obtainable by manipulation of \( ^2\vec{r}_{P/O_2} \). Let the components of \( ^2\vec{r}_{P/O_2} \) be expressed as \( ^2x_2, ^2y_2, \) and \( ^2z_2 \). These are directed along \( \hat{i}_2, \hat{j}_2, \) and \( \hat{k}_2 \) respectively. Thus,

\[
^2\vec{r}_{P/O_2} = ^2x_2 \hat{i}_2 + ^2y_2 \hat{j}_2 + ^2z_2 \hat{k}_2. \tag{2.5}
\]

The projection of this vector onto the \( x \)-axis of system 1 may be obtained by dotting the vector with \( \hat{i}_1 \). This projection is simply the \( x \)-component in system 1 of \( ^1\vec{r}_{P/O_2} \) or \( ^1x_2 \). Hence,

\[
^1x_2 = (^2x_2 \hat{i}_2 + ^2y_2 \hat{j}_2 + ^2z_2 \hat{k}_2) \cdot \hat{i}_1 \tag{2.6}
\]

\[
= ^2x_2 (\hat{i}_2 \cdot \hat{i}_1) + ^2y_2 (\hat{j}_2 \cdot \hat{i}_1) + ^2z_2 (\hat{k}_2 \cdot \hat{i}_1). \tag{2.7}
\]

The \( y \) and \( z \) components in system 1 may similarly be obtained by dotting the components of \( ^2\vec{r}_{P/O_2} \) with \( \hat{j}_1 \) and \( \hat{k}_1 \) respectively:

\[
^1y_2 = ^2x_2 (\hat{i}_2 \cdot \hat{j}_1) + ^2y_2 (\hat{j}_2 \cdot \hat{j}_1) + ^2z_2 (\hat{k}_2 \cdot \hat{j}_1), \tag{2.8}
\]

\[
^1z_2 = ^2x_2 (\hat{i}_2 \cdot \hat{k}_1) + ^2y_2 (\hat{j}_2 \cdot \hat{k}_1) + ^2z_2 (\hat{k}_2 \cdot \hat{k}_1). \tag{2.9}
\]

At this point, the component forms defined in system 1 are available for each term of (2.2). Thus, the following component (scalar) equations may be written according to the vector equation of (2.2):

\[
^1x_1 = ^2x_2 (\hat{i}_2 \cdot \hat{i}_1) + ^2y_2 (\hat{j}_2 \cdot \hat{i}_1) + ^2z_2 (\hat{k}_2 \cdot \hat{i}_1) + ^1x_0 \tag{2.10}
\]
\[ \begin{align*}
1y_1 &= 2x_2(j_2 \cdot j_1) + 2y_2(j_2 \cdot j_1) + 2z_2(\hat{k}_2 \cdot \hat{j}_1) + 1y_0 \\
1z_1 &= 2x_2(\hat{i}_2 \cdot \hat{k}_1) + 2y_2(\hat{j}_2 \cdot \hat{k}_1) + 2z_2(\hat{k}_2 \cdot \hat{k}_1) + 1z_0
\end{align*} \] (2.11) (2.12)

These may be written in matrix form as:

\[
\begin{pmatrix}
1x_1 \\
1y_1 \\
1z_1
\end{pmatrix} =
\begin{bmatrix}
\hat{i}_2 \cdot \hat{i}_1 & \hat{j}_2 \cdot \hat{i}_1 & \hat{k}_2 \cdot \hat{i}_1 \\
\hat{i}_2 \cdot \hat{j}_1 & \hat{j}_2 \cdot \hat{j}_1 & \hat{k}_2 \cdot \hat{j}_1 \\
\hat{i}_2 \cdot \hat{k}_1 & \hat{j}_2 \cdot \hat{k}_1 & \hat{k}_2 \cdot \hat{k}_1
\end{bmatrix}
\begin{pmatrix}
2x_2 \\
2y_2 \\
2z_2
\end{pmatrix} +
\begin{pmatrix}
1x_0 \\
1y_0 \\
1z_0
\end{pmatrix}
\] (2.13)

The dot product between any two unit vectors simply evaluates to the cosine of the angle between those two unit vectors. Hence, the 3 x 3 matrix in (2.13) may be written as:

\[
\begin{bmatrix}
\cos(\hat{i}_2, \hat{i}_1) & \cos(\hat{j}_2, \hat{i}_1) & \cos(\hat{k}_2, \hat{i}_1) \\
\cos(\hat{i}_2, \hat{j}_1) & \cos(\hat{j}_2, \hat{j}_1) & \cos(\hat{k}_2, \hat{j}_1) \\
\cos(\hat{i}_2, \hat{k}_1) & \cos(\hat{j}_2, \hat{k}_1) & \cos(\hat{k}_2, \hat{k}_1)
\end{bmatrix}
\] (2.14)

where \(\cos(\hat{a}, \hat{b}) = \) the cosine of the angle between \(\hat{a}\) and \(\hat{b}\). Obviously, this matrix is solely a function of the relative angular orientation (or rotation) between the two systems. Note also that the 3 x 1 column vector on the right-hand side of (2.13) is equal to the displacement of the origin of system 2 relative to the origin of system 1 in system 1 coordinates. As such, Equation (2.13) may be symbolically written as:

\[ 1r_{P/O_1} = R_{12} 2r_{P/O_2} + D_{12} \] (2.15)

where \(R_{12}\) converts the system 2 components of a vector into components aligned with the axes of system 1, and \(D_{12}\) accounts for the offset between the origins of system 2 and 1. This equation clearly transforms the coordinates of a point \(P\) defined in system 2 into coordinates defined in system 1.

It is easily verified that (2.13) may be written as:

\[
\begin{pmatrix}
1x_1 \\
1y_1 \\
1z_1
\end{pmatrix} =
\begin{bmatrix}
\hat{i}_2 \cdot \hat{i}_1 & \hat{j}_2 \cdot \hat{i}_1 & \hat{k}_2 \cdot \hat{i}_1 & 1x_0 \\
\hat{i}_2 \cdot \hat{j}_1 & \hat{j}_2 \cdot \hat{j}_1 & \hat{k}_2 \cdot \hat{j}_1 & 1y_0 \\
\hat{i}_2 \cdot \hat{k}_1 & \hat{j}_2 \cdot \hat{k}_1 & \hat{k}_2 \cdot \hat{k}_1 & 1z_0
\end{bmatrix}
\begin{pmatrix}
2x_2 \\
2y_2 \\
2z_2
\end{pmatrix},
\] (2.16)
By forming a partitioned matrix (based on the rotation and displacement matrices), the following can be written:

\[
\begin{bmatrix}
{^1x_P/O_1} \\
1
\end{bmatrix} = \begin{bmatrix} R_{12} & D_{12} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix}
{^2x_P/O_2} \\
1
\end{bmatrix}
\]

\[= T_{12} \begin{bmatrix}
{^2x_P/O_2} \\
1
\end{bmatrix}\]

(2.17)

where \( T_{12} = \) a 4 x 4 transformation matrix that converts system 2 coordinates into system 1

\( R_{12} = \) a 3 x 3 rotation matrix

\( D_{12} = \) a 3 x 1 displacement matrix (vector)

\( 0^T = \) a transposed 3 x 1 column vector of zeros

\( 1 = 1. \)

Note that the bottom row of \( T_{12} \) will always be \( \{0001\} \).

Notice that the \( x, y, z \) components in (2.16) have been augmented with a 1 to form a 4 x 1 column vector. This type of vector description is referred to as the homogeneous representation. The augmented 1 is actually a scale factor that is applied simultaneously to the \( x, y, z \) components and does not have to be 1. For the purposes herein, however, the scale factor is not useful and the 1 merely serves as a way of combining the effects of coordinate system rotation and translation into a single 4 x 4 transformation matrix.

A general 4 x 4 transformation matrix, \( T_{IJ} \), that transforms point coordinates from system \( J \) into system \( I \) can now be defined by determining \( R_{IJ} \) and \( D_{IJ} \). The matrix \( R_{IJ} \) is defined by (2.14). Physically speaking, the columns of \( R_{IJ} \) represent the components of the \( i, j, k \) of system \( J \) as seen in system \( I \). Hence, column 1 contains the components of \( i \) of system \( J \), column 2 contains \( j \), and column 3
contains \( \hat{k} \). The \( 3 \times 1 \) matrix \( D_{IJ} \) is determined by the \( x, y, z \) position of the origin of system \( J \) relative to the origin of system \( I \) defined in system \( I \) coordinates.

With this physical significance in mind, it is possible to define the transformation from system 2 to 1, \( T_{12} \), for the two coordinate systems in Figure 1—this can be done by observation.

First, the components of system 2's \( i, j, k \) unit vectors as seen from system 1 must be established in order to define \( R_{12} \). The \( i \) axis of system 2 has the components \((0,0,-1)\) when observed from system 1. Similarly the \( j \) and \( k \) axes of system 2 have the system 1 components of \((0,1,0)\) and \((1,0,0)\) respectively. Hence, using these, in order, as the columns of \( R_{12} \) yields:

\[
R_{12} = \begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
-1 & 0 & 0
\end{bmatrix}.
\]  

Next, the location of the origin of system 2 relative to system 1 in system 1 coordinates (i.e., \( ^1\vec{r}_{O_2/O_1} \)) must be determined. Using View B of Figure 1, the \( x \)-component of this location is 0. View A shows the \( y \)- and \( z \)-components to be 1 and 9 respectively. Therefore,

\[
D_{12} = \begin{bmatrix}
0 \\
1 \\
9
\end{bmatrix},
\]

and, according to (2.17),

\[
T_{12} = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
-1 & 0 & 0 & 9 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

This relation for \( T_{12} \) may now be used to convert any point defined in system 2 into system 1 coordinates (for the configuration shown in Figure 1). By using this
transformation on the homogeneous representation for \( P \) in system 2, it can be seen that the expected coordinates for \( P \) in system 1 are obtained:

\[
\begin{bmatrix}
2 \\
4 \\
5 \\
1
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
-1 & 0 & 0 & 9 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
4 \\
3 \\
2 \\
1
\end{bmatrix}.
\] (2.22)

### 2.3.2 Transformation Matrix Products

Assuming from this point on that all coordinates and vector components will be expressed in homogeneous form, the process of coordinate transformation may be written as:

\[
I_{p/q} = T_{IJ} J_{p/q}
\] (2.23)

where  \( I_{p/q} \) = the position, defined in system \( I \), of point \( p \) relative to point \( q \) (expressed in homogeneous form)

\( J_{p/q} \) = the position, defined in system \( J \), of point \( p \) relative to point \( q \) (expressed in homogeneous form)

\( T_{IJ} \) = the matrix transforming system \( J \) coordinates to system \( I \) coordinates.

Equation (2.23) is useful in demonstrating how an “overall” transformation matrix can be developed from the matrix product of a number of “intermediate” transformation matrices. Consider the following development based on a position vector \( \mathbf{r} \) observed from three different coordinate systems \( I, J, \) and \( K \). Using (2.23):

\[
J_{\mathbf{r}} = T_{JK} K_{\mathbf{r}}
\] (2.24)

\[
I_{\mathbf{r}} = T_{IJ} J_{\mathbf{r}}
\] (2.25)
\[ I_\gamma = T_{IK} K_\gamma. \]  
(2.26)

The \( J_\gamma \) term in Equation (2.25) can be replaced by the right hand side of Equation (2.24) to yield:

\[ I_\gamma = T_{IJ} T_{JK} K_\gamma. \]  
(2.27)

Comparing Equations (2.27) and (2.26) shows:

\[ T_{IK} = T_{IJ} T_{JK}. \]  
(2.28)

Equation (2.28) may be used recursively to yield the following general relationship:

\[ T_{1N} = T_{12} T_{23} \cdots T_{I-1,I} T_{IJ} T_{J,J+1} \cdots T_{N-1,N}. \]  
(2.29)

Equation (2.29) might be called the "chain rule" for coordinate transformation matrix products.

### 2.3.3 Inverse Transformation Matrices

Equation (2.23) can be used to represent the transformation of a vector defined in system \( I \) into a vector defined in system \( J \) by simply exchanging the \( I \) and \( J \) indices wherever they appear:

\[ J_\gamma P/Q = T_{JI} I_\gamma P/Q. \]  
(2.30)

However, a similar result can be achieved by multiplying both sides of (2.23) by the matrix inverse of \( T_{IJ} \):

\[ T_{IJ}^{-1} I_\gamma P/Q = T_{IJ}^{-1} T_{IJ} J_\gamma P/Q. \]  
(2.31)

\[ = I_\gamma P/Q \]  
(2.32)

\[ = J_\gamma P/Q \]  
(2.33)
where \( I_4 \) = the 4 x 4 identity matrix.

By comparing (2.30) and (2.33), it is evident that

\[
T_{ij}^{-1} = T_{jj}.
\]

Hence,

\[
T_{jj} T_{jj} = I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\]

The actual process of inverting a transformation matrix turns out to be easier than inverting a general 4 x 4 matrix. To show this, it is useful to begin with

\[
T_{jj} T_{jj}^{-1} = I_4.
\]

Using Equation (2.34),

\[
T_{jj} T_{jj} = I_4
\]

or, in partitioned form

\[
\begin{bmatrix} R_{jj} & D_{jj} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} R_{jj} & D_{jj} \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} I_3 & 0 \\ 0^T & 1 \end{bmatrix}
\]

where \( I_3 = \) the 3 x 3 identity matrix

\( 0 = \) a column of 3 zeros

\( 0^T = \) a row of 3 zeros.

This partitioned matrix equation yields the two nontrivial matrix equations

\[
R_{jj} R_{jj} = I_3
\]

and

\[
R_{ij} D_{jj} + D_{ij} = 0.
\]
Solving Equations (2.39) and (2.40) for $R_{JJ}$ and $D_{JJ}$ will provide the information to construct $T_{ij}^{-1}$. From (2.39)

$$R_{JJ} = R_{jj}^{-1},$$

and from (2.40)

$$D_{JJ} = -R_{jj}^{-1}D_{jj}.$$  

The $3 \times 3$ "rotation" partition of the transformation matrices is always orthogonal because of the choice of mutually perpendicular coordinate system axes. Therefore, the inverse of the rotation partition is obtained simply by taking the transpose of the partition. Hence,

$$R_{JJ} = R_{jj}^T,$$

and

$$D_{JJ} = -R_{jj}^TD_{jj}.$$  

Therefore, the inverse to any transformation can be calculated according to

$$T_{ij}^{-1} = T_{jj}$$

$$= \begin{bmatrix} R_{JJ} & D_{JJ} \\ 0^T & 1 \end{bmatrix}$$

$$= \begin{bmatrix} R_{jj}^T & -R_{jj}^TD_{jj} \\ 0^T & 1 \end{bmatrix}.$$  

### 2.3.4 Closed Loop Transformation Matrix Product

An interesting consequence of (2.35) is seen when both sides of (2.29) are post-multiplied by $T_{N1}$:

$$T_{1N}T_{N1} = T_{12}T_{23} \cdots T_{I-1,J}T_{IJ}T_{J,J+1} \cdots T_{N-1,N}T_{N1}$$  

(2.48)
or

\[ I = T_{1,1} = T_{12} T_{23} \cdots T_{I-1,I} T_{I,J} T_{J,J+1} \cdots T_{N-1,N} T_{N1}. \]  

Equation (2.49) shows that when the matrix product chain rule is employed on a closed loop of coordinate systems (i.e., the final transformation returns to the initial coordinate system), the resulting matrix is the constant identity matrix. This property is fundamental to the matrix method of kinematic analysis.

2.4 Rigid-Body-Affixed Coordinate Systems

As has been demonstrated, a coordinate transformation matrix contains relative rotation and translation information between two coordinate systems. Hence, given two completely defined coordinate systems, their relative orientation and location can be quantified with a transformation matrix. Alternatively, one might specify a desired orientation and location for a “moveable” coordinate system relative to a “fixed” one. This can essentially be done by specifying the elements of the corresponding transformation matrix that would be due to such a relative orientation and location.

In the study of kinematics, the goal of analysis is to determine motion properties (which includes position) of a collection of bodies. If a coordinate system is rigidly embedded in each body, then describing the relative orientations and locations between the coordinate systems is synonymous with describing the relative orientations and locations between the bodies. Hence, coordinate transformation matrices can be used to describe kinematic (motion) properties of general bodies. In other words, as a body is rotated and translated so too is its locally attached coordinate system rotated and translated by the same amount. Thus position
information concerning the coordinate system directly correlates to position information concerning the corresponding body.

2.4.1 Unconstrained Motion

Consider the two independent bodies, $B_I$ and $B_J$ as shown in Figure 2. Coordinate systems $I$ and $J$ are embedded in $B_I$ and $B_J$ respectively. Without loss of generality, body $B_I$ will be considered the frame of reference—it contains the coordinate system in which it is desired to have all coordinates resolved. For a given orientation and location of body $B_J$ relative to body $B_I$, it is possible to determine the matrix that transforms system $J$ coordinates to system $I$ coordinates (viz., $T_{IJ}$). This allows all points defined on body $B_J$ to be converted to coordinates defined in system $I$ using:

$$J_{P^I} = T_{IJ} I_{P^J}. \quad (2.50)$$

As previously shown, matrix $T_{IJ}$ consists of 12 nontrivial elements; 9 elements are found in the rotation partition, and 3 elements are found in the displacement partition. However, these 12 elements are not all independent of each other. This means fewer than 12 parameters are needed to describe the orientation and location of one body (coordinate system) relative to another body (coordinate system). The following discussion will show that three elements are sufficient to describe the rotation partition of $T_{IJ}$ and three more are required for the displacement partition.

Using Equations (2.39) and (2.43) yields:

$$R_{IJ} R_{IJ}^T = I_3. \quad (2.51)$$
Figure 2: Two unconstrained bodies with coordinate systems affixed.
If element \( kl \) of \( R_{ij} \) is denoted by \( r^{kl} \) then
\[
\begin{bmatrix}
  r_{11} & r_{12} & r_{13} \\
  r_{21} & r_{22} & r_{23} \\
  r_{31} & r_{32} & r_{33}
\end{bmatrix}
\begin{bmatrix}
  r_{11} & r_{21} & r_{31} \\
  r_{12} & r_{22} & r_{32} \\
  r_{13} & r_{23} & r_{33}
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1
\end{bmatrix},
\] (2.52)
or
\[
(r^{11})^2 + (r^{12})^2 + (r^{13})^2 = 1 \tag{2.53}
\]
\[
r^{21} r^{11} + r^{22} r^{12} + r^{23} r^{13} = 0 \tag{2.54}
\]
\[
r^{31} r^{11} + r^{32} r^{12} + r^{33} r^{13} = 0 \tag{2.55}
\]
\[
r^{11} r^{21} + r^{12} r^{22} + r^{13} r^{23} = 0 \tag{2.56}
\]
\[
(r^{21})^2 + (r^{22})^2 + (r^{23})^2 = 1 \tag{2.57}
\]
\[
r^{31} r^{21} + r^{32} r^{22} + r^{33} r^{23} = 0 \tag{2.58}
\]
\[
r^{11} r^{31} + r^{12} r^{32} + r^{13} r^{33} = 0 \tag{2.59}
\]
\[
r^{21} r^{31} + r^{22} r^{32} + r^{23} r^{33} = 0 \tag{2.60}
\]
\[
(r^{31})^2 + (r^{32})^2 + (r^{33})^2 = 1. \tag{2.61}
\]

Only 6 out of these 9 equations involving the elements of \( R_{ij} \) are unique—Equations (2.54) and (2.56) are identical, (2.55) and (2.59) are identical, and (2.58) and (2.60) are identical. Hence, Equations (2.53)–(2.61) can be used to determine the elements of \( R_{ij} \) if only 6 of the 9 \( r^{kl} \) are unknown. This means that 3 of the \( r^{kl} \) must be specified to completely define \( R_{ij} \). However, not any three may be specified; they must not all be in the same row or column of \( R_{ij} \). This can be reasoned physically by recalling that the columns of \( R_{ij} \) represent the components of the \( i, j, k \) axes of system \( J \) as seen in system \( I \). If the three elements specified are all in the same column, then information about only one of either \( i, j, \) or \( k \) is specified. Absolutely no information is provided about the remaining two axes.
This discussion also holds for all three elements occurring in the same row since the rows of $R_J$ represent the components of the $\hat{i}, \hat{j}, \hat{k}$ axes of system $I$ as seen from system $J$.²

The $3 \times 1$ displacement partition, $D_{ij}$, of $T_{ij}$ obviously requires the $x, y, z$ coordinates of the origin of system $J$ relative to the origin of system $I$. Hence, the need for these three parameters along with the three necessary to specify the rotation partition, shows that a total of six variables are needed to completely define $T_{ij}$. Since the relative location and orientation bodies $B_I$ and $B_J$ are described by $T_{ij}$, it is clear that six parameters are needed to describe the relative location and orientation between two independent, unattached bodies. In other words, general unconstrained spatial motion between two bodies must be described (or specified) using six degrees-of-freedom. These degrees-of-freedom are such that three describe the relative orientation and three describe the relative position between the two bodies.

The manifestation and visualization of the three positional degrees-of-freedom (i.e., the $D_{ij}$ partition) is intuitive and unambiguous. Knowing (or specifying) the $x, y, z$ components of the position of the origin of system $J$ relative to the origin of system $I$ in system $I$ coordinates is all that is necessary. The rotational degrees-of-freedom, however, may be expressed according to many different schemes.

Although three appropriate elements of $R_{ij}$ could be given to specify the orientation of a body, these elements are in fact direction cosines and are difficult to visualize in terms of a physical environment (i.e., they have no immediately intuitive meaning without a fair amount of analytical interpretation). Therefore,

²This is because $R_{ij}^T = R_{jj}$. 
a number of schemes have been developed that specify the rotation dofs as angular displacements about physically significant axes. These methods analytically construct the elements of $R_{ij}$ in terms of these angular displacements.

Listed below are three common methods used to define the rotation partition of spatial coordinate transformation matrices [77]. The matrices are developed by constructing intermediate coordinate systems between the $I$ and $J$ systems. These intermediate systems are strategically situated such that a rotation from one to another is about an easily defined axis (like an $x$, $y$, or $z$ axis). The rotation matrix, $R_{ij}$, is then constructed by forming the matrix product of these simpler intermediate rotation matrices. Note that no matter what scheme is employed a general spatial rotation matrix will be a function of only three parameters.

$\mathbf{R}_{(\alpha \beta \gamma)}$ : Defined by a rotation, $\alpha$, about $z_I$; a rotation, $\beta$, about $y_I$; and a rotation, $\gamma$, about $x_I$ (these rotations must be performed in the specified order).

$$
\mathbf{R}_{(\alpha \beta \gamma)} = 
\begin{bmatrix}
    C\alpha C\beta & -S\alpha C\beta & S\beta \\
    S\alpha C\gamma + C\alpha S\beta S\gamma & C\alpha C\gamma - S\alpha S\beta S\gamma & -C\beta S\gamma \\
    S\alpha S\gamma - C\alpha S\beta C\gamma & C\alpha S\gamma + S\alpha S\beta C\gamma & C\beta C\gamma 
\end{bmatrix}
$$

(2.62)

$C\alpha = \cos \alpha; \quad S\alpha = \sin \alpha; \quad$ etc.

$\mathbf{R}_{(\phi, \theta)}$ : Defined by a rotation of $\phi$ about the unit vector $\hat{u}$ defined in system $I$ (only two components of $\hat{u}$ are needed—the third can be calculated because $u_x^2 + u_y^2 + u_z^2 = 1$).

$$
\mathbf{R}_{(\phi, \theta)} = 
\begin{bmatrix}
    u_x^2 V\phi + C\phi & u_x u_y V\phi - u_z S\phi & u_x u_z V\phi + u_y S\phi \\
    u_x u_y V\phi + u_z S\phi & u_y^2 V\phi + C\phi & u_y u_z V\phi - u_x S\phi \\
    u_x u_z V\phi - u_y S\phi & u_y u_z V\phi + u_x S\phi & u_z^2 V\phi + C\phi 
\end{bmatrix}
$$

(2.63)

$C\phi = \cos \phi; \quad S\phi = \sin \phi; \quad V\phi = \text{versine} \phi = (1 - \cos \phi)$
$R_{(\psi \theta \phi)}$ : Defined by a rotation, $\psi$, of system I about $z_I$ to get system $I'$; a rotation, $\theta$, of system $I'$ about $x'_I$ to get system $I''$; and a rotation, $\phi$, of system $I''$ about $z''_I$ to get system $J$ (this is the Euler rotation matrix).

$$R_{(\psi \theta \phi)} = \begin{bmatrix} 
C\psi C\phi - S\psi C\theta S\phi & -C\psi S\phi - S\psi C\theta C\phi & S\psi S\theta \\
S\psi C\phi + C\psi C\theta S\phi & -S\psi S\phi + C\psi C\theta C\phi & -C\psi S\theta \\
S\theta S\phi & S\theta C\phi & C\theta 
\end{bmatrix}$$

(2.64)

$c\psi = \cos \psi; \quad s\psi = \sin \psi; \quad \text{etc.}$

### 2.4.2 Constrained Motion

Useful mechanisms are those that convert motion and/or transmit force in a known and predictable manner. Obviously, a system of two or more bodies wherein the relative motions of the bodies are completely independent of each other is of little mechanical use. The motion of one body will impart no motion nor transmit any force to the other bodies. In fact, if there are $n$ bodies, there must be $6(n - 1)$ degrees-of-freedom specified (if the bodies are all unconstrained and one of them is used as the reference frame) to completely determine their relative configuration.

A mechanism is made up of a collection of bodies called links that are attached to each other by joints. These joints are formed by physical contact between the links and constrain the relative motion between them. The constraints of motion are a function of the body features that comprise a given joint. In order for a joint to be useful, it must constrain motion in a predictable manner.

Take for instance the two bodies $B_I$ and $B_J$ shown in Figure 3. They are connected by a simple revolute (pin) joint. The ideal revolute joint will allow a relative motion between two bodies only in the form of a rotation about the revolute's axis. Assuming that the physical dimensions of both bodies are known,
the position of a point \( p \) in body \( B_j \) as seen in system \( I \) can only be a function the revolute rotation angle. In other words, knowing the rotation angle will locate all points of interest in body \( B_j \) relative to system \( I \). Suddenly, the 6 dof situation of unconstrained motion has been reduced to a 1 dof situation when two bodies are constrained by a revolute joint.

As already mentioned, a kinematic joint is classified by the number of degrees-of-freedom it permits between two joined bodies. Obviously, a joint must have between one and five degrees of freedom. If it had six dofs, it would allow general unconstrained motion, and hence, it would not be a joint at all. Zero dofs would imply that the "joint" restrained all motion between two bodies (i.e., as if they were welded together); kinematically speaking the two bodies would behave (and thus analyzed) as one body.

Kinematic joints are typically formed by physical contact between adjacent surfaces, edges (lines), or points on two bodies. Joints formed by surface contact are called lower pair joints. The six possible lower pair joints are shown in Table 1. The revolute, helical, and prismatic joints each allow one dof (i.e., they constrain five of the six dofs of general spatial motion). The cylindrical joint allows two dofs of motion; the spherical and planar joints allow three dofs of motion.

The geometries of the lower pair joints are analytically well defined. As such, their mathematical representations are straightforward. If coordinate systems are strategically embedded in bodies attached by lower pair joints, the transformation matrices that relate the motions of the bodies can be developed very easily and are quite intuitive.

Consider the two bodies in Figure 4. These bodies are identical to those shown
Figure 3: Two constrained bodies with coordinate systems affixed.
Table 1: Representations and degrees-of-freedom of the lower pair joints.

**Lower Pair Joints**

<table>
<thead>
<tr>
<th>Joint</th>
<th>Pictorial</th>
<th>Schematic</th>
<th>Abbreviation</th>
<th>DOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revolute</td>
<td><img src="image" alt="Revolute Pictorial" /></td>
<td><img src="image" alt="Revolute Schematic" /></td>
<td>R</td>
<td>1: $\theta$</td>
</tr>
<tr>
<td>Prismatic</td>
<td><img src="image" alt="Prismatic Pictorial" /></td>
<td><img src="image" alt="Prismatic Schematic" /></td>
<td>P</td>
<td>1: $s$</td>
</tr>
<tr>
<td>Helical</td>
<td><img src="image" alt="Helical Pictorial" /></td>
<td><img src="image" alt="Helical Schematic" /></td>
<td>H</td>
<td>1: $\theta$ or $s$</td>
</tr>
<tr>
<td>Cylindrical</td>
<td><img src="image" alt="Cylindrical Pictorial" /></td>
<td><img src="image" alt="Cylindrical Schematic" /></td>
<td>C</td>
<td>2: $\theta$, $s$</td>
</tr>
<tr>
<td>Spherical</td>
<td><img src="image" alt="Spherical Pictorial" /></td>
<td><img src="image" alt="Spherical Schematic" /></td>
<td>S</td>
<td>3: $\theta$, $\phi$, $\psi$</td>
</tr>
<tr>
<td>Planar</td>
<td><img src="image" alt="Planar Pictorial" /></td>
<td><img src="image" alt="Planar Schematic" /></td>
<td>F</td>
<td>3: $x$, $y$, $\theta$</td>
</tr>
</tbody>
</table>
in Figure 3, however, their embedded coordinate systems have been relocated and reoriented—this is a valid action since the theory presented thus far imposes no conditions on the placement and orientation of the body-affixed coordinate systems. With the coordinate systems as shown, it is very easy to write the elements of $T_{IJ}$ in terms of the joint's degree-of-freedom, $\theta$. This is done using the fact that the columns of $R_{IJ}$ are the components of $i_J$, $j_J$, and $k_J$ as seen in system $I$, and that $D_{IJ}$ is the position of $O_J$ relative to $O_I$ as seen in system $I$. Therefore,

$$T_{IJ} = T(\theta)_{IJ} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & \overline{O_J O_I} \cos \theta \\ \sin \theta & \cos \theta & 0 & \overline{O_J O_I} \sin \theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.65)$$

where $\overline{O_J O_I}$ is the distance between the two origins. The location of $O_I$ on body $B_I$ was obviously selected because of some important feature on $B_I$ and $B_J$—namely, this is the location of the revolute joint. Similarly, the location of $O_J$ on body $B_J$ would be selected based on some important feature on $B_J$—perhaps the location of a joint between $B_J$ and another body. At any rate, it can be reasoned that $\overline{O_J O_I}$ is related to a physical dimension (length) between two important features on body $B_J$. Thus, the transformation matrix is dependent upon (constant) body dimensions and (variable) joint degrees-of-freedom.

In general, let $\hat{f}_{IJ}$ represent the dofs permitted by a joint between two bodies $B_I$ and $B_J$. Then $T_{IJ}$ becomes a function of $\hat{f}_{IJ}$:

$$T_{IJ} = T(\hat{f}_{IJ})_{IJ} = \begin{bmatrix} r(\hat{f}_{IJ})^{11} & r(\hat{f}_{IJ})^{12} & r(\hat{f}_{IJ})^{13} & d(\hat{f}_{IJ})^{11} \\ r(\hat{f}_{IJ})^{21} & r(\hat{f}_{IJ})^{22} & r(\hat{f}_{IJ})^{23} & d(\hat{f}_{IJ})^{21} \\ r(\hat{f}_{IJ})^{31} & r(\hat{f}_{IJ})^{32} & r(\hat{f}_{IJ})^{33} & d(\hat{f}_{IJ})^{31} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.66)$$
Figure 4: Two joined bodies with strategically affixed coordinate systems.
The form of $T_{jj}$'s elements will be related to body dimensions and the joint dofs, $f_{ij}$. Specifying the elements of $f_{ij}$ will define $T_{jj}$ and put body $B_j$ at a known location and orientation relative to body $B_i$. The ease with which the elements of $T_{jj}$ are expressed in terms of $f_{ij}$ is extremely dependent upon how the coordinate systems are embedded in the bodies relative to the motions permitted by the joints.

2.5 Systematic Approaches to Modeling Mechanism Kinematic Structure

The relative orientation and location between two kinematically joined bodies can be expressed by embedding a coordinate system in each of the bodies and then developing the transformation matrix between those two systems. The transformation matrix will be a function of the degrees-of-freedom of motion permitted by the joint and the dimensions of the bodies. As such, transformation matrices become an ideal means of quantifying the configuration of a collection of bodies.

A number of researchers have exploited the properties of transformation matrices to develop kinematic equations of motion for mechanisms. The basic process followed is to embed a local coordinate system in each link of a mechanism, develop the transformation matrix between adjacent links, and then use the transformation matrix product chain rule to form an "overall" transformation matrix for the mechanism. This overall matrix will be a function of all of the link dimensions and joint degrees-of-freedom in the mechanism. As will be shown, this technique can be used to generate position, velocity, and acceleration information about a mechanism.

As discussed previously, the ease with which a transformation matrix may be
written between the coordinate systems of two adjoining bodies is dependent upon the placement of the coordinate systems. The following techniques demonstrate systematic methods of "anchoring" the local coordinate systems in the bodies of a mechanism. The first method is based on the classical work of Denavit and Hartenberg [18]. Their technique is perhaps the most widely known and used approach especially for the "axial" types of joints (i.e., those which exhibit motion along or about a well defined axis). The second method is due to a paper by Sheth and Uicker. Their method will be shown to be more amenable to "non-axial" types of joints and is more "building block" oriented than Denavit and Hartenberg's technique.

2.5.1 Denavit and Hartenberg Approach

Denavit and Hartenberg developed a procedure for systematically defining and affixing local coordinate systems to links of a mechanism. The result of their technique was a generalized transformation matrix that could be used easily to describe binary links connected by axial joints. These types of joints include directly the revolute, prismatic, cylindrical, and helical joints. Spherical and planar joints are indirectly considered by constructing kinematic equivalents made up of revolute and prismatic joint combinations.

Basic Principles

Their approach begins by looking at a link, say link \( i \), connected between two other links, say \( i - 1 \) and \( i + 1 \), as shown in Figure 5. These links are assumed to be members in a chain of links all of which are connected in series by axial joints.
The goal is to establish useful local coordinate systems affixed to each link. The axis of the joint connecting links \( i \) and \( i + 1 \) will define the \( z \)-axis of system \( i \) (i.e., \( z_i \)). Its positive direction is chosen according to convenience. Similarly, the joint between links \( i - 1 \) and \( i \) defines the \( z_{i-1} \) axis. The common normal between \( z_{i-1} \) and \( z_i \) will define the \( x_i \) axis. The intersection of the \( x_i \) and \( z_i \) axes defines the origin of system \( i \). The \( y_i \) axis is obtained by crossing the \( z_i \) axis into the \( x_i \) axis using the right-hand rule. This technique will always (regardless of the mechanism configuration) define a locally fixed coordinate system on each link.

Once the local coordinate systems have been established in the above manner, it is possible to describe the relative location and orientation of adjacent systems with relatively few parameters. The distance along \( x_i \) between \( z_{i-1} \) and \( z_i \) (i.e., along their common normal) is \( a_i \). The distance along the \( z_{i-1} \) axis between the origin of system \( i - 1 \) and the intersection of the common normal with \( z_{i-1} \) is \( s_i \). The angle between the \( x_{i-1} \) and \( x_i \) axes (looking down the \( z_{i-1} \) axis) is \( \theta_i \). Finally, the angle between \( z_{i-1} \) and \( z_i \) (looking down the \( x_i \) axis) is \( \alpha_i \).

With these four parameters, \( \alpha_i, s_i, \theta_i, \) and \( \alpha_i, \) the transformation matrix \( T_{i-1,i} \) can be written as:

\[
T_{i-1,i} = \begin{bmatrix}
\cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\
\sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\
0 & \sin \alpha_i & \cos \alpha_i & s_i \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (2.67)

\[
T = T(\theta_i, \alpha_i, s_i)_{i-1,i}
\] (2.68)

Depending on the type of joint connecting links \( i - 1 \) and \( i \), one or more of the parameters found in \( T_{i-1,i} \) will be variable. For instance, a revolute joint will cause \( \theta_i \) to be variable; a prismatic joint will cause \( s_i \) to be variable; and a cylindrical joint will cause both \( \theta_i \) and \( s_i \) to be variable. These variables obviously
Figure 5: Denavit and Hartenberg definition of link local coordinate systems.
represent directly the degree(s)-of-freedom permitted by the joint considered. The parameters in $T_{i-1,i}$ found to be constants are related to the physical dimensions of links $i - 1$ and $i$. Although they may not be given explicitly on the mechanical drawings of the links, they would be derivable from the dimensions supplied.

Planar Slider Crank Example

The usefulness of this technique can be demonstrated with the planar slider crank mechanism shown in Figure 6. The three revolute joints define axes $z_1$, $z_2$, and $z_3$ (conveniently chosen positive “out of the plane of the paper”). The prismatic joint defines the $z_4$ axis. With the information for $\theta_i$, $\alpha_i$, $\alpha_i$, and $s_i$ ($i = 1, 2, 3, 4$) as shown in Figure 6, $T_{12}$, $T_{23}$, $T_{34}$, and $T_{41}$ can be written using Equation (2.67). Note when $i = 1$ for a closed loop mechanism with $n$ links, the matrix $T_{i-1,i}$ is actually written as $T_{n1}$ and not $T_{01}$.

For the slider crank being used, $s_1$, $\theta_2$, $\theta_3$, and $\theta_4$ are the only variables encountered. All other parameters that make up the $T_{i-1,i}$’s are constant. Hence, inserting the constant values yields:

\[
T_{12} = T(\theta_2)_{12} = \begin{bmatrix}
\cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 \\
\sin \theta_2 & \cos \theta_2 & 0 & l_2 \sin \theta_2 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \quad (2.69)
\]

\[
T_{23} = T(\theta_3)_{23} = \begin{bmatrix}
\cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cos \theta_3 \\
\sin \theta_3 & \cos \theta_3 & 0 & l_3 \sin \theta_3 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \quad (2.70)
\]

\[
T_{34} = T(\theta_4)_{34} = \begin{bmatrix}
\cos \theta_4 & 0 & \sin \theta_4 & l_4 \cos \theta_4 \\
\sin \theta_4 & 0 & -\cos \theta_4 & l_4 \sin \theta_4 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \quad (2.71)
\]
Figure 6: Slider crank mechanism parameters using Denavit and Hartenberg notation

\[
\begin{align*}
\theta_1 &= 180^\circ \\
a_1 &= 0 \\
\alpha_1 &= 90^\circ \\
S_1 &= \text{variable}
\end{align*}
\begin{align*}
\theta_2 &= \text{variable} \\
a_2 &= l_2 \\
\alpha_2 &= 0^\circ \\
S_2 &= 0
\end{align*}
\begin{align*}
\theta_3 &= \text{variable} \\
a_3 &= l_3 \\
\alpha_3 &= 0^\circ \\
S_3 &= 0
\end{align*}
\begin{align*}
\theta_4 &= \text{variable} \\
a_4 &= l_4 \\
\alpha_4 &= 90^\circ \\
S_4 &= 0
\end{align*}
\]
If the matrix product chain rule is used on a closed loop of coordinates systems (see Equation (2.49)), the following can be written:

\[ T_{41} = T(s_1)_{41} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & s_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \] (2.72)

Expanding Equation (2.73) using (2.69)-(2.72) and using

\[ \mu = \theta_2, \ 
\nu = \theta_2 + \theta_3, \ 
\xi = \theta_2 + \theta_3 + \theta_4 \]

yields

\[ I_4 = \begin{bmatrix} -\cos \xi & \sin \xi & 0 & l_2 \cos \mu + l_3 \cos \nu + l_4 \cos \xi + s_1 \sin \xi \\ -\sin \xi & -\cos \xi & 0 & l_2 \sin \mu + l_3 \sin \nu + l_4 \sin \xi - s_1 \cos \xi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \] (2.75)

As discussed previously, at most six independent elements exist in a transformation matrix. Three can be picked from the rotation partition (not all three from the same row or column) along with the three from the displacement partition. This being the case, six independent (in general) elements of \( T_{1,1} \) may be equated with the corresponding elements of \( I_4 \). This results in six equations in terms of \( s_1, \theta_2, \theta_3, \) and \( \theta_4 \). Without further analysis it appears as though the slider crank example yields an overconstrained problem—six equations in four unknowns. However, due to the fact that the slider crank has special geometry (it is a planar mechanism), it will be found that only three of these equations will actually be independent. Trivial equations \((0 = 0 \text{ and } 1 = 1)\) are of no use.
Three useful equations from (2.75) for the slider crank at hand are:

\[ 1 = -\cos(\theta_2 + \theta_3 + \theta_4) \]  
\[ 0 = l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3) + l_4 \cos(\theta_2 + \theta_3 + \theta_4) + s_1 \sin(\theta_2 + \theta_3 + \theta_4) \]  
\[ 0 = l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3) + l_4 \sin(\theta_2 + \theta_3 + \theta_4) - s_1 \cos(\theta_2 + \theta_3 + \theta_4) \]  

Thus with three equations and four unknowns, one of the unknowns must be specified to make the slider crank's configuration determinant. This is consistent with the familiar one degree-of-freedom behavior exhibited by a slider crank mechanism.

Assuming \( \theta_2 \) is the input to the mechanism, the following relations result:

\[ s_1 = -l_2 \sin \theta_2 \pm \sqrt{l_3^2 - (l_2 \cos \theta_2 - l_4)^2} \]  
\[ \theta_3 = \tan^{-1}\left(\frac{-s_1 - l_2 \sin \theta_2}{l_4 - l_2 \cos \theta_2}\right) - \theta_2 \]  
\[ \theta_4 = \pi - \theta_2 - \theta_3. \]  

Due to the square root appearing in the relation for \( s_1 \), Equation (2.79), two solutions are possible. This is also consistent with the known behavior of a slider crank. For a given crank angle, \( \theta_2 \), the mechanism can be assembled in two different ways. The equation for \( s_1 \) also contains information regarding kinematic limit positions \(^3\) imposed by link dimensions. If the argument of the square root equals zero, a kinematic limit position for \( \theta_2 \) has been reached. Arguments less than zero represent impossible values for \( \theta_2 \).

**Observations**

The Denavit and Hartenberg method of mechanism modeling is systematic and relatively straightforward. However, it does require a bit of "practice" on the part

\(^3\)These are not limit positions caused by mechanical or geometric interference between the links.
of the analyst to determine the correct placement of link parameters \( (\theta_i, \alpha_i, \alpha_i, \text{ and } s_i) \) as well as their correct signs. Nevertheless, the parameters have direct qualitative meaning regarding the kinematic shape of the links. For instance, \( \alpha \) is often called the "twist" angle. In the case of a binary link with revolutes on both ends, \( \alpha \) is a measure of how the two revolute axes are twisted relative to each other about their common normal.

Modeling a mechanism using Denavit and Hartenberg's approach begins with locating the locally attached "z-axes" for each link. Establishing these axes is obvious for links with joints that allow motion along or about a single axis (i.e., revolute, prismatic, cylindrical, and helical joints). The generalized Denavit and Hartenberg matrix deals very well (and intuitively) with these one and two degree-of-freedom connections. However, the motion permitted by joints such as the spherical and planar joints cannot be quantified in terms of a single well-defined axis. As such, the use of the Denavit and Hartenberg matrix to incorporate these joints cannot be done directly. Instead, these "non-axial" joints are modeled by representing them with a kinematically equivalent chain of axial-joint-interconnected links.

For instance, two links connected by a planar joint will be allowed the same freedom of motion if instead they are connected by two prismatic joints and a revolute. The prismatic joints must lie in the plane of the actual planar joint and must not be collinear (they are usually taken to be perpendicular to each other). The revolute's axis must be perpendicular to the plane of the planar joint. In symbolic notation, the chain "-F-" \(^4\) is replaced by "-P-P-R-". Similarly, a spherical joint may be modeled using a chain of three revolute joints whose axes

\(^4\)The use of "F" for a planar joint is logical in that this joint is also called a flat pair.
all intersect at the same point. In general, none of the axes are to be collinear if the spherical joint's three degrees-of-freedom are to be maintained. The chain "–S–" is, therefore, replaced by "–R–R–R–".

A goal of this research was to analyze mechanisms having joints formed by contact between general three dimensional surfaces. Such a joint might be referred to as a three dimensional cam and follower. A practical example of a three dimensional cam and follower is the "bent-shaft" universal joint as analyzed by Beggs [12] and shown in Figure 7. The axes of the input and output shafts are skew in general. Rotary motion from the input shaft is transmitted to the output shaft through surface contact at the bent ends of the shafts. For the general case, the point of contact on each shaft will change throughout the mechanism's motion. The relative motion permitted by the surface contact joint is a five degree-of-freedom motion and clearly is not resolvable in terms of a single axis. Furthermore, a kinematically equivalent replacement chain made up of the common axial joints is by no means apparent for the surface joint as it was for the planar joint or spherical joint. Hence, Denavit and Hartenberg's approach to mechanism modeling does not appear to be well suited for mechanisms with other than axial joints.

A final observation on the Denavit and Hartenberg technique has to do its amenability to computerized implementation. Since another objective of this research was to develop a computer programmable technique for spatial mechanism analysis, this is a very important consideration. Methods of analysis well suited to computerization are generally those that can be approached in a "building-block" approach.

---

5The only motion restrained by surface contact is relative displacement between the two bodies along the surface normal at the point of contact. If unconstrained motion is a six dof motion, a joint enforcing just one constraint must have five dofs.
Figure 7: A "bent-shaft" universal joint.
fashion. In the case of mechanism analysis, a desirable procedure is to first pro-
vide the computer program with the relevant dimensions of each individual link
and then, after all links have been defined, specify the manner in which they are
joined together. The Denavit and Hartenberg method does not lend itself well to
this type of process.

As is evident, the general D-H matrix contains both link "shape" information
as well as joint dof information. Separating this information is necessary if a true
building-block environment is to be achieved—the link shape definition must exist
independently of the joints used to connect the links. This is not possible with the
D-H approach. In fact the parameters, \( \theta, a, \alpha, \) and \( s \), for a given link, \( i \), cannot
all be defined until link \( i - 1 \) is attached to link \( i \) by a specified joint. Sheth and
Uicker [69] simply state this dependence as:

The four parameters of the D-H notation depend not only on the shape
of the link under consideration, but also on the shape of the previous
link in the kinematic loop being considered.

This does not bode well for a building-block approach in which it is desired to
define each link independently from the others.

2.5.2 Sheth and Uicker Approach

Sheth and Uicker [69] have presented a generalized, systematic approach to mod-
eling mechanisms that goes beyond some of the limitations previously mentioned
with the Denavit and Hartenberg method. The overall concept is similar to the one
employed by the D-H technique. Local coordinate systems are attached to each
link, and the transformation matrices between these systems are functions of the
joint degrees-of-freedom. Forming the transformation product around a closed loop of links must equal the identity matrix. This yields up to six independent equations in terms of constant link dimensions and variable joint degrees-of-freedom. The Sheth and Uicker method, however, clearly "decouples" the transforming effects of the constant link dimensions from the effects of the joint variables. This results in a procedure more amenable to a building block implementation—a given link can exist as a completely quantifiable entity without knowing what joint or link properties precede it. Additionally, the process with which local link coordinate systems are assigned to the individual links is not dependent on associated joints exhibiting axial behavior. This facilitates the analysis of mechanisms with higher pair joints.

Kinzel and Hall [39] described a similar concept wherein the joint characteristics were mathematically separate from the link characteristics. Their purpose was to provide a convenient means for describing the clearance effects of joints and the dimensional tolerance effects of links. Kinzel and Hall's method, however, still relied on the traditional (and somewhat limiting) D-H parameters.

Basic Principles

Consider two links, \( i - 1 \) and \( i \), as shown in Figure 8. For simplicity, the two links are joined by a revolute joint shown in an exploded view. Each link has two attached coordinate systems rather than just one—they are labeled as systems \((uvw)\) and \((xyz)\). These systems are rigidly affixed to the links at their pairing element locations. The systems' orientations are not theoretically important, however, aligning them with the motion characteristics and/or geometric features of
their associated joints will make the mathematical description of the joints more straightforward. The rules for this alignment will be discussed later.

As with the Denavit and Hartenberg approach, the goal is to systematically develop transformation matrices between adjacent links which, when multiplied consecutively around a closed loop, yield the identity matrix. For the links in Figure 8, the transformation matrix of interest would be $T_{i-1,i}$ (i.e., the transformation matrix converting system $(xyz)_i$ coordinates to system $(xyz)_{i-1}$ coordinates). The Sheth-Uicker (i.e., S-U) method develops this in two stages.

The first step is to determine the transformation from system $(xyz)_i$ to system $(uvw)_i$. This will be called $S_i$ (the shape matrix). $S_i$ is solely a function of the shape of link $i$ since both systems $(xyz)_i$ and $(uvw)_i$ are embedded in link $i$. Next the transformation from system $(uvw)_i$ to system $(xyz)_{i-1}$ must be formed. This transformation will labeled as $P_{i-1,i}$ (the pair matrix). It is entirely a function of the degrees-of-freedom permitted by the joint pair. The transformation $T_{i-1,i}$ may then be calculated from

$$T_{i-1,i} = S_{i-1} P_{i-1,i} \quad (2.82)$$

where

$T_{i-1,i} = $ the transformation matrix from system $(xyz)_i$ to system $(xyz)_{i-1}$

$S_{i-1} = $ the transformation matrix from system $(xyz)_{i-1}$ to system $(uvw)_{i-1}$ (the constant shape matrix)

$P_{i-1,i} = $ the transformation matrix from system $(uvw)_i$ to system $(xyz)_{i-1}$ (the variable pair matrix).

A single closed loop mechanism possessing $n$ links could then be described with transformation matrices as follows:

$$I_4 = S_1 P_{12} S_2 P_{23} S_3 P_{34} \cdots S_{n-1} P_{n-1,n} S_n P_{n,1} \quad (2.83)$$
Figure 8: Link coordinate systems for Sheth and Uicker approach.
The S matrices are constant for a given mechanism, and the P matrices are functions of the degrees-of-freedom of the joints. The form of (2.83) is identical to the form used for Denavit and Hartenberg approaches. Hence, any solution technique that works with the D-H method will also work (without modification!) with the S-U formulation.

Shape Matrices. As discussed previously, the specification of a transformation matrix between two coordinate systems is a function of six independent parameters. Three of these concern the relative orientation of the systems, and three concern their relative locations. Therefore, the link shape transformation matrix, \( S_i \), is made to be a function of six parameters in the form of three rotation angles and three displacements. For a given link, these parameters will be constant.

Consider, the two coordinate systems \((uvw)_i\) and \((xyz)_i\), attached to link \(i\) in Figure 9. The transformation from system \((xyz)_i\) to \((uvw)_i\) can be formulated by considering the matrix product of six simple one dof transformations. First, the common normal to \(w_i\) and \(z_i\) is constructed. This normal intersects \(w_i\) at \(p\) and \(z_i\) at \(q\). A vector, \(\vec{\gamma}_i\), is directed from \(p\) to \(q\) along this common normal. Starting at system \((uvw)_i\), system \((xyz)_i\) is obtained by a translation of \(c_i\) along \(w_i\), a rotation of \(\gamma_i\) about \(w_i\), a translation of \(a_i\) along \(\vec{\gamma}_i\), a rotation of \(\alpha_i\) about \(t_i\), a translation of \(b_i\) along \(z_i\), and a rotation of \(\beta_i\) about \(z_i\). The dimensions are defined as follows:

- \(c_i\) is the distance from the origin of \((uvw)_i\), \(O_{uvw}\), to the point of intersection, \(p\), of \(w_i\) with \(\vec{\gamma}_i\). It is defined positive in the direction of \(w_i\).

- \(\gamma_i\) is the angle required to rotate \(u_i\) about \(w_i\) into \(\vec{\gamma}_i\). It is defined positive
Figure 9: Geometric relationship between the \((xyz)\) and \((uvw)\) coordinate systems involved in the Sheth-Uicker link shape matrix
counterclockwise looking down \( w_i \).

- \( \alpha_i \) is the length of the common normal (i.e., the distance between \( p \) and \( q \)). It is defined positive in the direction of \( i \).

- \( \alpha_i \) is the angle required to rotate \( w_i \) about \( i \) into \( z_i \). It is defined positive counterclockwise looking down \( i \).

- \( b_i \) is the distance from the point of intersection, \( q \), of \( i \) with \( z_i \) to the origin of \( (xyz)_i \), \( O_{xyz} \). It is defined positive in the direction of \( z_i \).

- \( \beta_i \) is the angle required to rotate \( i \) about \( z_i \) into \( x_i \). It is defined positive counterclockwise looking down \( z_i \).

Symbolically, these displacements and rotations can be mathematically described using rigid body translation and rotation matrices. In other words, system \((uvw)_i\) can be rotated and translated such that it is coincident and parallel to system \((xyz)_i \). The concatenation of these matrices will yield the transformation matrix from system \((xyz)_i \) to \((uvw)_i \), namely the shape matrix \( S_i \). Thus,

\[
S = D_w(c) \, R_\gamma(\gamma) \, D_x(\delta) \, R_x(\alpha) \, D_z(b) \, R_z(\beta) \quad (2.84)
\]

where

- \( D_w(r) = \) a translation of \( r \) in the \( z \)-direction
- \( D_x(r) = \) a translation of \( r \) in the \( x \)-direction
- \( R_z(\phi) = \) a rotation of \( \phi \) about the \( z \)-axis
- \( R_x(\phi) = \) a rotation of \( \phi \) about the \( x \)-axis.

General forms for translation matrices along \( z \)- and \( x \)-axes are

\[
D_z(r) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & r \\
0 & 0 & 0 & 1
\end{bmatrix} \quad (2.85)
\]
\[ D_x(r) = \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.86) \]

and general forms for rotation matrices about \( z \)- and \( x \)-axes are

\[ R_z(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.87) \]

\[ R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.88) \]

Using these general definitions for the translation and rotation matrices, Equation (2.84) can be used to express the shape matrix for a given link as a function of the six parameters \( \alpha, \alpha, \beta, b, c, \) and \( \gamma \):

\[ S(\alpha, \alpha, b, \beta, c, \gamma) = \begin{bmatrix} C\beta C\gamma - C\alpha S\beta S\gamma & -S\beta C\gamma - C\alpha C\beta S\gamma & S\alpha S\gamma & bS\alpha S\gamma + aC\gamma \\ C\beta S\gamma + C\alpha S\beta C\gamma & -S\beta S\gamma + C\alpha C\beta C\gamma & -S\alpha C\gamma & -bS\alpha C\gamma + aS\gamma \\ S\alpha S\beta & S\alpha C\beta & C\alpha & bC\alpha + c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.89) \]

where \( S\alpha = \sin \alpha, C\alpha = \cos \alpha, \) etc. Note the six parameters \( \alpha, \alpha, b, \beta, c, \) and \( \gamma \) will be constant for a given link.

**Pair Matrices.** Once the pairing element coordinate systems \((uvw)\) and \((xyz)\) have been established for each link in a mechanism, the links may be "mathematically" joined to each other with pair matrices, \( P \). These pair matrices are transformations between the \((xyz)\) and \((uvw)\) systems of adjacent links. A given pair matrix will be a function of the degrees-of-freedom permitted by the joint being modeled. The pair matrices are completely unrelated to the shape of the
links they connect. As such, a standard library of pair matrices suitable for use with any link shape can be developed. Additionally, since no special "motion" characteristics have been imposed on the pair matrices (i.e., like axial motion for instance), any valid transformation may be used. This permits the straightforward development of pair matrices for higher pair types of joints.

The pair matrices for the common lower pair joints are presented in Figures 10–17. Note that joining two links with a particular joint may force the alignment of certain pairing element axes on the links being joined. For instance, joining two links, say i−1 and i, with a revolute pair will force the two links into a relative orientation where the origins of (xyz)i−1 and (uvw)i are coincident and the z_{i-1} and w_i-axes are aligned in the same direction. The pair variable, \theta_i, will be measured about the w_i-axis from the x_{i-1} to the u_i-axes positive counterclockwise looking down w_i. Obviously, this will affect the analyst's choice for the location and orientation of the pairing element coordinate systems.

Observe some pairing element types have more than one way to mathematically describe their motion characteristics. The spherical joint, for instance, can be modeled with any three rotations about intersecting axes. A planar joint could be described using cartesian type degrees-of-freedom, or it might be described using polar degrees-of-freedom. For another example, the helical joint may be described using its rotation angle, or it may be described using its axial translation. Its rotation and translation are related to each other by the pitch of the screw thread. The form of the pairing element used is based on the manner in which the analyst desires the degrees-of-freedom to be expressed by the solution.
Revolute Pair

Pair Variables \( \theta \)

Specifications
- The origins of \((x_yz)_{i-1}\) and \((uvw)_{i}\) are coincident.
- \(z_{i-1}\) and \(w_i\) are collinear and in the same direction.
- \(\theta\) is measured about \(z_{i-1}\) from \(x_{i-1}\) to \(u_i\) positive counterclockwise looking down \(z_{i-1}\).

Pair Matrix

\[
P(\theta) = \begin{bmatrix}
\cos \theta & -\sin \theta & 0 & 0 \\
\sin \theta & \cos \theta & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (2.90)

Coordinate Systems

Figure 10: Description and specifications for Revolute Pair matrix.
Prismatic Pair

Pair Variables $s$

Specifications

- $z_{i-1}$ and $w_i$ are collinear and in the same direction.
- $x_{i-1}$ and $u_i$ are parallel and in the same direction.
- $s$ is measured along $z_{i-1}$ from $x_{i-1}$ to $u_i$ positive in the direction of positive $z_{i-1}$.

Pair Matrix

$$P(s) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ \hspace{1cm} (2.91)

Coordinate Systems

![Diagram of coordinate systems](Figure 11: Description and specifications for Prismatic Pair matrix.)
Helical Pair–Type 1

Pair Variables $\theta$

Specifications

- Behavior is described in terms of joint rotation.
- $z_{i-1}$ and $w_i$ are collinear and in the same direction.
- $\theta$ is measured about $z_{i-1}$ from $x_{i-1}$ to $u_i$ positive counterclockwise looking down $z_{i-1}$.
- $s$ is the distance along $z_{i-1}$ from $x_{i-1}$ to $u_i$ positive in the direction of positive $z_{i-1}$. It is a function of $\theta$.
- The lead of the helix is $L$.
- Translation of joint, $s$, along $z_{i-1}$ is $\frac{L\theta}{2\pi}$ ($\theta$ measured in radians).

Pair Matrix

$$P(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & \frac{L\theta}{2\pi} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.92)$$

Coordinate Systems

Figure 12: Description and specifications for Helical Pair–Type 1 matrix.
Helical Pair–Type 2

Pair Variables \( s \)

Specifications

- Behavior is described in terms of joint translation.
- \( z_{i-1} \) and \( w_i \) are collinear and in the same direction.
- \( s \) is measured along \( z_{i-1} \) from \( x_{i-1} \) to \( u_i \); positive in the direction of positive \( z_{i-1} \).
- \( \theta \) is the angle about \( z_{i-1} \) from \( z_{i-1} \) to \( u_i \) positive counterclockwise looking down \( z_{i-1} \). It is a function of \( s \).
- The lead of the helix is \( L \).
- The rotation of the joint, \( \theta \), about \( z_{i-1} \) is \( \frac{2\pi s}{L} \) (\( \theta \) measured in radians).

Pair Matrix

\[
P(s) = \begin{bmatrix}
\cos \frac{2\pi s}{L} & -\sin \frac{2\pi s}{L} & 0 & 0 \\
\sin \frac{2\pi s}{L} & \cos \frac{2\pi s}{L} & 0 & 0 \\
0 & 0 & 1 & s \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (2.93)

Coordinate Systems

Figure 13: Description and specifications for Helical Pair–Type 2 matrix.
Cylindric Pair

Pair Variables \( \theta, s \)

Specifications
- \( z_{i-1} \) and \( w_i \) are collinear and in the same direction.
- \( \theta \) is measured about \( z_{i-1} \) from \( x_{i-1} \) to \( u_i \) positive counterclockwise looking down \( z_{i-1} \).
- \( s \) is measured along \( z_{i-1} \) from \( x_{i-1} \) to \( u_i \) positive in the direction of positive \( z_{i-1} \).

Pair Matrix

\[
P(\theta, s) = \begin{bmatrix}
\cos \theta & -\sin \theta & 0 & 0 \\
\sin \theta & \cos \theta & 0 & 0 \\
0 & 0 & 1 & s \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(2.94)

Coordinate Systems

Figure 14: Description and specifications for Cylindric Pair matrix.
Spherical Pair–Type 1

Pair Variables $\theta, \phi, \psi$

Specifications

- This is an Euler type of transformation matrix.
- The transformation is constructed as a product of three simple rotation matrices: $R_\phi(\theta) R_\phi(\phi) R_\phi(\psi)$.
- The origins of $(xyz)_{i-1}$ and $(uvw)_i$ are coincident.
- $R_\phi(\theta)$ rotates $x_{i-1}$ about $z_{i-1}$ to describe a “floating” axis, $u'_i$. $R_\phi(\phi)$ rotates $z_{i-1}$ about $u'_i$ into $w_i$. $R_\phi(\psi)$ rotates $u'_i$ about $w_i$ into $u_i$.
- All angles are measured positive counterclockwise looking down the axes about which they describe a rotation.
- $C\theta = \cos \theta$, $S\theta = \sin \theta$, etc. below.

Pair Matrix

$$P(\theta, \phi, \psi) = \begin{bmatrix}
C\theta C\psi - S\theta C\phi S\psi & -C\theta S\psi - S\theta C\phi C\psi & S\theta S\phi & 0 \\
S\theta C\psi + C\theta C\phi S\psi & -S\theta S\psi + C\theta C\phi C\psi & -C\theta S\phi & 0 \\
S\phi S\psi & S\phi C\psi & C\phi & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

(2.95)

Coordinate Systems

![Coordinate Systems Diagram]

Note: The $y$-axes have been omitted for clarity.

Figure 15: Description and specifications for Spherical Pair–Type 1 matrix.
Spherical Pair–Type 2

Pair Variables \( \theta, \phi, \psi \)

Specifications

- This is an Euler type of transformation matrix.
- The transformation is constructed as a product of three simple rotation matrices: \( R_x(\theta) R_\phi(\phi) R_x(\psi) \).
- The origins of \((xyz)_{i-1}\) and \((uvw)_i\) are coincident.
- \( R_x(\theta) \) rotates \( z_{i-1} \) about \( x_{i-1} \) to describe a "floating" axis, \( w'_i \). \( R_\phi(\phi) \) rotates \( x_{i-1} \) about \( w'_i \) into \( u_i \). \( R_x(\psi) \) rotates \( w'_i \) about \( u_i \) into \( w_i \).
- All angles are measured positive counterclockwise looking down the axes about which they describe a rotation.
- \( C\theta = \cos \theta, \ S\theta = \sin \theta, \) etc. below.

Pair Matrix

\[
P(\theta, \phi, \psi) = \begin{bmatrix}
C\phi & -S\phi C\psi & S\phi S\psi \\
C\theta S\phi & -S\theta S\psi + C\theta C\phi C\psi & -S\theta C\psi - C\theta C\phi S\psi \\
S\theta S\phi & C\theta S\psi + S\theta C\phi C\psi & C\theta C\psi - S\theta C\phi S\psi
\end{bmatrix}
\]

Coordinate Systems

Figure 16: Description and specifications for Spherical Pair–Type 2 matrix.
Spherical Pair–Type 3

Pair Variables $\theta, \phi, \psi$

Specifications

- This is a screw axis type of transformation matrix based on a rotation, $\psi$, about an arbitrarily oriented axis, $A_i$, through the origin.
- $\theta$ and $\phi$ give the orientation of the screw axis, $A_i$, relative to system $(xyz)_{i-1}$. With $z_{i-1}$ as the "North Pole," $\theta$ and $\phi$ give the longitude and latitude, respectively, of a point through which $A_i$ is directed from the origin.
- The origins of $(xyz)_{i-1}$ and $(uvw)_{i}$ are coincident.
- $\theta$ is the angle measured about $z_{i-1}$ through which $A_i$ is rotated from the $x_{i-1}z_{i-1}$ plane.
- $\phi$ is the angle between $z_{i-1}$ and $A_i$.
- All angles are measured positive counterclockwise looking down the axes about which they describe a rotation.
- It is useful to write the pair matrix in terms of the direction cosines of $A_i$, $(a_x, a_y, a_z)$. Thus, $a_x = \cos \theta \sin \phi; \ a_y = \sin \theta \sin \phi; \ a_z = \cos \phi$.
- $C\psi = \cos \psi, S\psi = \sin \psi, V\psi = \text{versin} \psi = 1 - \cos \psi$ below.

Pair Matrix

$$P(\theta, \phi, \psi) = \begin{bmatrix}
    a_x^2 V\psi + C\psi & a_x a_y V\psi - a_z S\psi & a_x a_z V\psi + a_y S\psi & 0 \\
    a_x a_y V\psi + a_z S\psi & a_y^2 V\psi + C\psi & a_y a_z V\psi - a_x S\psi & 0 \\
    a_x a_z V\psi - a_y S\psi & a_y a_z V\psi + a_x S\psi & a_z^2 V\psi + C\psi & 0 \\
    0 & 0 & 0 & 1
  \end{bmatrix}$$ (2.97)

Coordinate Systems

Note: The $(uvw)_i$ system has been omitted for clarity.

Figure 17: Description and specifications for Spherical Pair–Type 3 matrix.
Spherical Pair—Type 4

<table>
<thead>
<tr>
<th>Pair Variables</th>
<th>$\theta, \phi, \psi$</th>
</tr>
</thead>
</table>

Specifications

- This is a transformation matrix formed by rotations about three mutually perpendicular $z$-axes, and is the spherical joint representation put forth by Sheth and Uicker [69].
- The origins of $(xyz)_{i-1}$ and $(uvw)_{i}$ are coincident.
- The matrix is constructed using $R_x(\theta) R_x(\pi/2) R_x(\phi) R_x(\pi/2) R_x(\psi)$. The constant $R_x(\pi/2)$ matrices ensure the mutual perpendicularity of the $z$-axes about which the joint's degrees-of-freedom are are defined.
- All angles are measured positive counterclockwise looking down the axes about which they describe a rotation.
- $C\theta = \cos \theta, S\theta = \sin \theta$, etc. below.

Pair Matrix

$$P(\theta, \phi, \psi) = \begin{bmatrix} C\theta C\phi C\psi + S\theta S\psi & -C\theta C\phi S\psi + S\theta C\psi & C\theta S\phi & 0 \\ S\theta C\phi C\psi - C\theta S\psi & -S\theta C\phi S\psi - C\theta C\psi & S\theta S\phi & 0 \\ S\phi C\psi & -S\phi S\psi & -C\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.98)$$

Figure 18: Description and specifications for Spherical Pair—Type 4 matrix.
### Spherical Pair–Type 5

<table>
<thead>
<tr>
<th>Pair Variables</th>
<th>$\theta, \phi, \psi$</th>
</tr>
</thead>
</table>

#### Specifications

- This is a transformation matrix formed by a $z$-axis, a $y$-axis, and an $x$-axis rotation (in that order). Paul [59] presents this matrix as the "Roll, Pitch, and Yaw" transformation.
- The origins of $(xyz)_i$ and $(uvw)_i$ are coincident.
- The matrix is constructed using $R_x(\theta) R_y(\phi) R_z(\psi)$. $\theta$, $\phi$, and $\psi$ correspond to the "roll," "pitch," and "yaw" coordinates, respectively.
- All angles are measured positive counterclockwise looking down the axes about which they describe a rotation.

#### Pair Matrix

$$ P(\theta, \phi, \psi) = \begin{bmatrix} C\theta C\phi & C\theta S\phi S\psi - S\theta C\psi & C\theta S\phi C\psi + S\theta S\psi & 0 \\ S\theta C\phi & S\theta S\phi S\psi + C\theta C\psi & S\theta S\phi C\psi - C\theta S\psi & 0 \\ -S\phi & C\phi S\psi & C\phi C\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.99) $$

---

Figure 19: Description and specifications for Spherical Pair–Type 5 matrix.
Planar Pair

Pair Variables \( r, s, \theta \)

Specifications

- The \((xy)_{i-1}\) and \((uv)_{i}\) planes are coplanar.
- The \(x_{i-1}\) and \(w_{i}\) axes are parallel and point in the same direction.
- The origin of \((uvw)_{i}\) is displaced from the origin of \((xyz)_{i-1}\) by \(r\) units along \(x_{i-1}\) and \(s\) units along \(y_{i-1}\).
- The angle from \(x_{i-1}\) to \(u_{i}\) is \(\theta\) and is measured about \(w_{i}\). \(\theta\) is measured positive counterclockwise looking down \(w_{i}\).

Pair Matrix

\[
P(r, s, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & r \\ \sin \theta & \cos \theta & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.100}
\]

Coordinate Systems

Figure 20: Description and specifications for Planar Pair matrix.
Planar Slider Crank Example

For sake of comparison, the Sheth-Uicker mechanism description method will be used on the same planar slider crank mechanism previously modeled with the Denavit-Hartenberg method (see pages 46–49). The same slider-crank mechanism depicted in Figure 6 is shown in Figure 21 but in an exploded rendering.

The first step in the S-U method is to affix the \((uvw)\) and \((xyz)\) coordinate systems to the pairing elements of each link. Recall that the pairing element matrices imply the "direction" of transformation to be from the \((xyz)\) system of link \(i - 1\) to the \((uvw)\) system of link \(i\). Therefore, if one wishes to proceed clockwise (or counterclockwise) around a loop of links, the \((uvw)\) systems will be at the "beginning" of the links, and the \((xyz)\) systems will be at the "end" of the links.

By definition of the revolute pair matrix (page 62), a revolute joint forces the alignment of axes \(z_{i-1}\) and \(w_i\) and the coincidence of their origins. Rotation between the \((xyz)_{i-1}\) and \((uvw)_i\) systems is permitted about the \(z_{i-1}/w_i\) axis. Therefore, the placement of the \(z\)-and \(w\)-axes for all of the revolute pairing elements is obvious. In this particular example, all of the revolute axes have been chosen as positive out of the paper, however, this is not necessary. The \(x\)- and \(u\)-axes are placed in planes perpendicular to the \(z\)- and \(w\)-axes, respectively. Their orientations in these planes are arbitrary. However, since the revolute pair variable, \(\theta_i\), measures the angle from \(x_{i-1}\) to \(u_i\), it is useful to align the \(x\) and \(u\) axes with significant features of the links (i.e., along the links’ centerlines for instance). Once the \(x\)- and \(z\)- or \(u\)- and \(w\)-axes have been established, the \(y\) and \(v\) axes are defined using the right-hand rule.
Figure 21: Sheth-Uicker parameters for the planar slider-crank mechanism.

<table>
<thead>
<tr>
<th>Link</th>
<th>Shape Matrix Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1 = 0$</td>
<td>$c_2 = 0$</td>
</tr>
<tr>
<td>$\gamma_1 = 0^\circ$</td>
<td>$\gamma_2 = 180^\circ$</td>
</tr>
<tr>
<td>$a_1 = 0$</td>
<td>$a_2 = l_2$</td>
</tr>
<tr>
<td>$\alpha_1 = 90^\circ$</td>
<td>$\alpha_2 = 0^\circ$</td>
</tr>
<tr>
<td>$b_1 = 0$</td>
<td>$b_2 = 0$</td>
</tr>
<tr>
<td>$\beta_1 = 90^\circ$</td>
<td>$\beta_2 = 0^\circ$</td>
</tr>
</tbody>
</table>
By definition of the prismatic pair matrix (page 63), a prismatic joint enforces the alignment of axes \( z^i \) and \( w^i \). Translation between the \((xyz)_{i-1}\) and \((uvw)_i\) permitted along the \(z_{i-1}/w_i\) axis. The \(x^i\) and \(u^i\) axes are placed in planes perpendicular to the \(z^i\) and \(w^i\), respectively. Their orientations in these planes are arbitrary. The \(x^i\) and \(u^i\) axes are constrained to be parallel and in the same direction. However, since the translation, \(s^i\), between the two systems is measured as the distance from \(x^i\) to \(u^i\), it is again useful to align the \(x^i\) and \(u^i\) axes with significant features of the links.

With the links' pairing element coordinate systems specified as shown in Figure 21, the six shape parameters \(c, \gamma, a, \alpha, b,\) and \(\beta\) may be determined for each link. These will all be constant as the \((xyz)\) and \((uvw)\) systems are all rigidly fixed to the links. The rules for determining the shape parameters were listed on page 57. Equation (2.89) may then be used to determine the shape matrices for each link. The values for the shape constants are shown in Figure 21. The resulting link shape matrices are:

\[
S_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2.101)}
\]

\[
S_2 = \begin{bmatrix} -1 & 0 & 0 & -l_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2.102)}
\]

\[
S_3 = \begin{bmatrix} -1 & 0 & 0 & -l_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(2.103)}
\]
Note these shape matrices can be determined by inspection (very easily in this particular example). In general, the first three elements of the first column in matrix \( S_i \) represent the direction cosines of \( x_i \) as seen in system \((uvw)_i\). The first three elements of the second and third columns are the direction cosines for the \( y_i \) and \( z_i \) axes, respectively, also as seen in system \((uvw)_i\). The first three element of the fourth column represents the coordinates of system \((xyz)_i\)'s origin as seen in system \((uvw)_i\). The fourth row of \( S_i \) is always \( (0 0 0 1) \).

The matrix loop for this slider crank now becomes

\[
I_4 = T_{12} T_{23} T_{34} T_{41} \tag{2.105}
\]

\[
= [S_1 P_{12}] [S_2 P_{23}] [S_3 P_{34}] [S_4 P_{41}] \tag{2.106}
\]

where \( P_{41} \) is a prismatic pair matrix and \( P_{12}, P_{23}, \) and \( P_{34} \) are revolute pair matrices. Thus,

\[
P(\theta_2)_{12} = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.107}
\]

\[
P(\theta_3)_{23} = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.108}
\]

\[
P(\theta_4)_{34} = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.109}
\]
Equations (2.101)–(2.104) and (2.107)–(2.110) may now be used in Equation (2.106) to yield the closed loop matrix product. Using the following substitutions

\[ \begin{align*}
\mu & = \theta_2 \\
\nu & = \theta_2 + \theta_3 \\
\xi & = \theta_2 + \theta_3 + \theta_4,
\end{align*} \]

the matrix product evaluates to

\[ I_4 = \begin{bmatrix}
-\sin \xi & 0 & -\cos \xi & l_2 \sin \mu - l_3 \sin \nu + l_4 \sin \xi - s_1 \cos \xi \\
0 & 1 & 0 & 0 \\
\cos \xi & 0 & -\sin \xi & -l_2 \cos \mu + l_3 \cos \nu - l_4 \cos \xi - s_1 \sin \xi \\
0 & 0 & 0 & 1
\end{bmatrix}. \tag{2.111} \]

As with the slider-crank analysis performed with the Denavit and Hartenberg method, only three independent (nontrivial) equations exist because this is a planar mechanism (due to special geometry). Three useful equations from (2.111) for the slider crank being analyzed are:

\[ \begin{align*}
1 & = -\sin(\theta_2 + \theta_3 + \theta_4) \tag{2.112} \\
0 & = l_2 \sin(\theta_2) - l_3 \sin(\theta_2 + \theta_3) + \\
&\quad l_4 \sin(\theta_2 + \theta_3 + \theta_4) - s_1 \cos(\theta_2 + \theta_3 + \theta_4) \tag{2.113} \\
0 & = -l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3) + \\
&\quad l_4 \cos(\theta_2 + \theta_3 + \theta_4) - s_1 \sin(\theta_2 + \theta_3 + \theta_4). \tag{2.114}
\end{align*} \]

Assuming \( \theta_2 \) is the input to the mechanism, expressions for the three remaining unknowns, \( s_1, \theta_3, \) and \( \theta_4 \), may be obtained using the above equations. Thus, after manipulation,

\[ \begin{align*}
s_1 & = l_2 \cos \theta_2 \pm \sqrt{l_3^2 - (l_2 \sin \theta_2 - l_4)^2} \tag{2.115} \\
\theta_3 & = \tan^{-1} \left( \frac{l_2 \sin \theta_2 - l_4}{l_2 \cos \theta_2 - s_1} \right) - \theta_2 \tag{2.116} \\
\theta_4 & = \frac{3\pi}{2} - \theta_2 - \theta_3. \tag{2.117}
\end{align*} \]
Note the similarity between Equations (2.115)-(2.117) and the corresponding equations from the Denavit and Hartenberg approach, Equations (2.79)-(2.81). Using the Sheth and Uicker approach, two solutions still exist due to the square root in Equation (2.115). The resulting equations from both schemes are easily verified as correct using simple geometrical techniques. The reason different equations are obtained from one method to another has to do with how the local coordinate systems are embedded in the links.

The Denavit and Hartenberg method leaves no options as to the orientations and locations of the local coordinate systems—they are completely defined by set rules. The Sheth and Uicker method, however, allows the analyst some freedom in choosing the orientation of the local coordinate systems. For example, the pairing element coordinate system for a “Sheth and Uicker” revolute joint requires only that the $z$-axis (or $w$-axis) be aligned with the revolute axis. The origin of the $(xyz)$ system (or $(uvw)$ system) may be located on the $z$ (or $w$) axis anywhere convenient. The $x$-axis (or $u$-axis) may then be placed through this origin at any convenient orientation as long as it is perpendicular to the $z$ (or $w$) axis.

For a real mechanism, the relative position and orientation between two joined links is usually measured relative to physical features of the links. The Sheth and Uicker method allows the analyst to place the pairing element coordinate systems in a manner consistent with position/orientation measurement features of the real mechanism. This flexibility, however, can result in apparently different analysis results for the same mechanism due to differing placements and orientations of the pairing element coordinate systems. This is the reason the slider-crank results from the Denavit/Hartenberg and the Sheth/Uicker techniques are not identical—the
links’ locally affixed reference frames are not consistent. The predicted behavior of the slider-crank as observed in the reference frame, however, will be the same regardless of what local coordinate systems are used.

**Indirect Specification of Shape Matrices**

One of the difficulties in applying the Denavit and Hartenberg method to mechanism analysis has to do with determining the parameters $a$, $\alpha$, $\theta$, and $s$. It takes a fair amount of practice to visualize a link in terms of these quantities. The parameters $c$, $\gamma$, $a$, $b$, and $\beta$ needed for the Sheth and Uicker shape matrices are certainly not any better in terms of ease of visualization. In fact, with the Sheth and Uicker method, rather than just four parameters per link, now six are needed. Fortunately it is possible to indirectly determine the link shape matrices without explicitly determining the six parameters.

Consider the general binary link shown in Figure 22. The pairing element systems $(uvw)$ and $(xyz)$ would be located on the link according to the type of joints used on the link. The origins of these two systems are labeled $O_u$ and $O_x$ respectively. A third coordinate system, $(XYZ)$, has been affixed to the link and is called the link’s definition system. Its origin is labeled $O_i$. System $(XYZ)$ can be thought of as the reference from which all link dimensions and features are measured. It would be possible to dimension all features of a part from such a system on an actual mechanical drawing of that part. Since systems $(uvw)$ and $(xyz)$ are rigidly affixed to the link, it is then also possible to physically define their axes relative to $(XYZ)$—doing so will allow the transformation matrix between systems $(XYZ)$ and $(uvw)$ and between systems $(XYZ)$ and $(xyz)$ to be
developed. These can in turn be used to form the transformation between systems (uvw) and (xyz) which is precisely the definition of the shape matrix. For ease of description, the three coordinate systems will be referred to as:

\[
\begin{align*}
\text{system (uvw)} & \equiv \text{system } u \\
\text{system (xyz)} & \equiv \text{system } x \\
\text{system (XYZ)} & \equiv \text{system } l.
\end{align*}
\]

Note the three points, \(p_1, p_2, \text{ and } p_3\), shown on system \(u\) in Figure 22. These are points defined in system \(l\)—their coordinates in system \(l\) would be identical to the components of \(\vec{r}_1, \vec{r}_2, \text{ and } \vec{r}_3\) also shown in the figure. Let

\[
\begin{align*}
\vec{r}_1 &= \{x_1, y_1, z_1\}^T \\
\vec{r}_2 &= \{x_2, y_2, z_2\}^T \\
\vec{r}_3 &= \{x_3, y_3, z_3\}^T.
\end{align*}
\]

Points \(p_1\) and \(p_2\) lie on the \(w\)-axis such that the positive direction of \(w\) is obtained by proceeding from point \(p_1\) to point \(p_2\). Point \(p_3\) is simply placed on the positive \(u\)-axis. These three points uniquely define system \(u\) relative to system \(l\).

The coordinates of \(O_u\) can be determined from two observations. First, a line through \(p_1\) and \(p_2\) is perpendicular to a line through \(p_3\) and \(O_u\). Second, \(O_u\) must lie somewhere on an infinite line passing through points \(p_1\) and \(p_2\). These two observations may be stated mathematically as

\[
\begin{align*}
0 &= (\vec{r}_2 - \vec{r}_1) \cdot (\vec{r}_3 - \vec{O}_u) \quad (2.121) \\
\vec{O}_u &= \vec{r}_1 + \mu(\vec{r}_2 - \vec{r}_1) \quad (2.122)
\end{align*}
\]

where \(\vec{O}_u\) is the vector from \(O_l\) to \(O_u\) as defined in system \(l\). In component form this may be written as

\[
\vec{O}_u = \{x_0, y_0, z_0\}^T. \quad (2.123)
\]
System \((XYZ)\) - Link definition system, origin \(O_l\)
System \((uvw)\) - Pairing element system, origin \(O_u\)
System \((xyz)\) - Pairing element system, origin \(O_x\)

Figure 22: General binary link and its coordinate systems.
Substituting Equation (2.122) into (2.121) yields

\[ 0 = (\vec{r}_2 - \vec{r}_1) \cdot (\vec{r}_3 - \vec{r}_1 + \mu(\vec{r}_2 - \vec{r}_1)). \]  

(2.124)

This equation may be solved for \( \mu \):

\[ \mu = \frac{(\vec{r}_2 - \vec{r}_1) \cdot (\vec{r}_3 - \vec{r}_1)}{(\vec{r}_2 - \vec{r}_1) \cdot (\vec{r}_2 - \vec{r}_1)}. \]  

(2.125)

Using the components of \( \vec{r}_1, \vec{r}_2, \) and \( \vec{r}_3 \) (which are identical to the coordinates of \( p_1, p_2, \) and \( p_3, \) respectively), \( \mu \) may be expressed as

\[ \mu = \frac{(x_2 - x_1)(x_3 - x_1) + (y_2 - y_1)(y_3 - y_1) + (z_2 - z_1)(z_3 - z_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \]  

(2.126)

The coordinates of \( O_u \) in system \( I \) may be calculated by using this value for \( \mu \) in Equation (2.122). In other words, this value for \( \mu \) allows the calculation of \( x_o, y_o, \) and \( z_o \) in terms of the coordinates of \( p_1, p_2, \) and \( p_3. \)

Now that the coordinates of \( O_u \) are known, a vector, \( \vec{u}' \), in the same direction as the \( u \)-axis may be written as:

\[ \vec{u}' = \{(x_3 - x_o), (y_3 - y_o), (z_3 - z_o)\}^T. \]  

(2.127)

Similarly, a vector, \( \vec{w}' \), having the same direction as the \( w \)-axis can be expressed as

\[ \vec{w}' = \{(x_2 - x_1), (y_2 - y_1), (z_2 - z_1)\}^T. \]  

(2.128)

The cross product of \( \vec{w}' \) and \( \vec{u}' \) can be used to obtain a vector, \( \vec{v}' \), having the same direction as the \( v \)-axis as follows:

\[ \vec{v}' = \vec{w}' \times \vec{u}' \]  

(2.129)

\[ \text{Sheth and Uicker used } \vec{w}' = \{(x_2 - x_o), (y_2 - y_o), (z_2 - z_o)\}^T \text{ which is valid only if } p_2 \text{ is on the positive } w \text{-axis—a condition never explicitly stated. The only requirement of Equation (2.128) is } p_2 > p_1 \text{ along the } w \text{-axis.} \]
If \( \vec{u}', \vec{v}', \) and \( \vec{w}' \) are divided by their respective magnitudes, the result will be the direction cosines for the \( u-, v-, \) and \( w- \) axes relative to system \( I. \) From previous discussions, it is known that the first three columns of a transformation matrix, say, \( T_{IJ} \), consist of the direction cosines of system \( J \)'s axes relative to system \( I. \) It is also known that the fourth column represents the coordinates of system \( J \)'s origin relative to system \( I. \) Therefore, the transformation from the pairing element system, \( u, \) to the link definition system, \( l, \) may be formed using \( \vec{u}', \vec{v}', \vec{w}', x_o, y_o, \) and \( z_o. \) Thus,

\[
T_{lu} = \begin{bmatrix}
\frac{(x_3 - x_o)}{|\vec{u}'|} & \frac{(y_3 - y_o)(x_3 - z_o) - (y_o - y_o)(x_2 - z_1)}{|\vec{v}'|} & \frac{(x_2 - x_1)}{|\vec{w}'|} \\
\frac{(y_3 - y_o)}{|\vec{u}'|} & \frac{(x_3 - x_o)(x_2 - z_1) - (x_o - x_1)(x_3 - z_o)}{|\vec{v}'|} & \frac{(y_2 - y_1)}{|\vec{w}'|} \\
\frac{(z_3 - z_o)}{|\vec{u}'|} & \frac{(x_3 - x_o)(y_3 - y_o) - (x_o - x_1)(y_2 - y_1)}{|\vec{v}'|} & \frac{(z_2 - z_1)}{|\vec{w}'|} \\
0 & 0 & 1
\end{bmatrix}
\]

(2.131)

where

\[
|\vec{u}'| = (x_3 - x_o)^2 + (y_o - y_o)^2 + (z_3 - z_o)^2
\]
\[
|\vec{w}'| = (x_2 - x_1)^2 + (y_o - y_o)^2 + (z_2 - z_1)^2
\]
\[
|\vec{v}'| = [(y_3 - y_1)(x_3 - z_o) - (y_o - y_o)(x_2 - z_1)]^2 +
[(x_3 - x_o)(x_2 - z_1) - (x_2 - x_1)(x_3 - z_o)]^2 +
[(x_3 - x_o)(y_3 - y_o) - (x_o - x_1)(y_2 - y_1)]^2
\]

\[
x_o = x_1 + \mu(x_2 - x_1)
\]
\[
y_o = y_1 + \mu(y_2 - y_1)
\]
\[
z_o = z_1 + \mu(z_2 - z_1)
\]
\[
\mu = \frac{(x_2 - x_1)(x_3 - x_1) + (y_2 - y_1)(y_3 - y_1) + (z_2 - z_1)(x_3 - z_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}
\]

The transformation from system \( s \) to system \( l, T_{ls}, \) may be determined in exactly the same manner by choosing three points on its axes (i.e., two points on
its $z$-axis and one point on its $x$-axis according to the rules set forth. Having calculated $T_{lu}$ and $T_{lx}$, the transformation from system $x$ to system $u$, $T_{ux}$, may be determined using

$$T_{ux} = T_{lu}^{-1} T_{lx} \quad (2.132)$$

$$= T_{ul} T_{lx} \quad (2.133)$$

By definition, $T_{ux}$ transforms system $(xyz)$ coordinates to system $(uvw)$ coordinates and, therefore, is identical to the previously described link shape matrix. $T_{ul}$ is inverted easily as outlined in Section 2.3.3. Forming the shape matrices is now seen to be as simple as locating points on the pairing element coordinate system axes. A reasonable scheme for generating a link's shape matrix would then be as shown in Table 2.
Indirect Formation of Sheth-Uicker Shape Matrix

1. Make a sketch (free-hand or to scale) of the link for which a shape matrix is to be formed.

2. Place the link definition coordinate system, \( I \), in a strategic location on the sketch. The strategy is based on knowing that the coordinates of points on the pairing element system axes need to be described relative to this definition system.

3. Place the pairing element coordinate systems, \( u \) and \( x \), in strategic locations and orientations on the sketch at the link's pairing element sites. The strategy is based on the type of joints used as well between what link features the analyst wishes the joint degrees-of-freedom to be measured. The convention employed is that system \( u \) on link \( i \) will be joined with system \( x \) on link \( i - 1 \).

4. Form the transformation from system \( u \) to system \( l \), \( T_{lu} \).
   (a) Specify the coordinates, in system \( l \), of two points on the \( w \)-axis such that the direction from point 1 to point 2 is in the positive direction of the \( w \)-axis. This gives \( x_1, y_1, z_1 \) and \( x_2, y_2, z_2 \).
   (b) Specify the coordinates, in system \( l \), of a point on the positive \( u \)-axis. This gives \( x_3, y_3, z_3 \).
   (c) Use these coordinates in Equation (2.131) to get \( T_{lu} \).

5. Form the transformation from system \( x \) to system \( l \), \( T_{lx} \).
   (a) Specify the coordinates, in system \( l \), of two points on the \( z \)-axis such that the direction from point 1 to point 2 is in the positive direction of the \( z \)-axis. This gives \( x_1, y_1, z_1 \) and \( x_2, y_2, z_2 \).
   (b) Specify the coordinates, in system \( l \), of a point on the positive \( x \)-axis. This gives \( x_3, y_3, z_3 \).
   (c) Use these coordinates in Equation (2.131) to get \( T_{lx} \).

6. Invert \( T_{lu} \) to get \( T_{ul} \).

7. Form \( T_{ux} \) using Equation (2.133). \( T_{ux} \) is the shape matrix for this link.
CHAPTER III

Development of Three Dimensional Surface Contact Joints

The mechanism modeling method described by Sheth and Uicker clearly separates the mathematical effects of link shape from the joint degrees-of-freedom. This is done by writing the transformation between adjacent links as a product of two intermediate transformations—one contains just link shape information (i.e., the shape matrix), and the other contains just joint information (i.e., the pairing element matrix). By separating the link and joint characteristics, it is now possible to "design" pairing element matrices based purely on the motion constraints desired (or known to exist) between two coordinate systems. These two coordinate systems are the \((xyz)\) and \((uvw)\) systems of adjacent links.

For instance, the revolute pair matrix was designed to align the \(z\)-axis of link, say, \(i - 1\) with the \(w\)-axis of link \(i\), force their origins to be coincident, and allow relative rotation about the \(z/w\)-axis. Similarly, the prismatic pair matrix was designed to align the \(z\)-axis of link \(i - 1\) with the \(w\)-axis of link \(i\), force the \(x\)-axis of link \(i - 1\) to be parallel with the \(u\)-axis of link \(i\), and allow relative translation of their origins along the \(z/w\)-axis. The implementation of aligning certain axes in these two joints was a matter of design—other choices could have been made. The characteristic relative motions permitted were matters of simulating known joint
behaviors.

Using the concepts presented, it is possible to develop pair matrices that simulate the kinematic behavior of joints consisting of two contacting three-dimensional surfaces. This might also be described as a three-dimensional cam and follower. It is assumed that the surfaces treated are analytically defined using parametric forms (in terms of two parameters) and they are convex. This means they can have only one contact point.

3.1 Mathematical Development of Pair Matrix

Consider the surface contact joint shown in Figure 23. The "tail end" (system $(xyz)_{i-1}$) of link $i-1$ is connected to the "front end" (system $(uvw)_i$) of link $i$ via two surfaces in contact. Surface $\Sigma_{i-1}$ is rigidly attached to link $i-1$ and is defined in terms of system $(xyz)_{i-1}$. Likewise, surface $\Sigma_i$ is rigidly attached to link $i$ and is defined in terms of system $(uvw)_i$. The pair matrix for this type of joint must mathematically describe the relative motion permitted between systems $(xyz)_{i-1}$ and $(uvw)_i$.

The exploded view in Figure 23 shows the points of contact on the two surfaces; point $A$ is on $\Sigma_{i-1}$, and point $B$ is on $\Sigma_i$. Since the transformation (i.e., the pair matrix) from system $(uvw)_i$ to system $(xyz)_{i-1}$ is not obvious, it will be useful to form the transformation from a product of a number of more straightforward transformations. Specifically, "floating" coordinate systems will be placed at points $A$ and $B$. They are considered floating since they will always be situated at the point of contact on each surface. The pair matrix may then be formed using

$$P_{i-1,i} = T_{i-1,A} T_{AB} T_{Bi} \tag{3.1}$$
Figure 23: A three-dimensional surface contact joint.
where

\[ \begin{align*}
    P_{i-1,i} & = \text{the pair matrix for the surface contact joint} \\
    T_{B_i} & = \text{transformation from system } (uvw)_i \text{ to the intermediate floating system } B \\
    T_{AB} & = \text{transformation from the floating intermediate system } B \text{ to the floating intermediate system } A \\
    T_{i-1,A} & = \text{transformation from the floating intermediate system } A \text{ to system } (xyz)_{i-1}.
\end{align*} \]

Considering surface \( \Sigma_{i-1} \), the floating system at point \( A \) will be based on two perpendicular vectors at \( A \). One vector is normal to the surface at \( A \) (viz., \( \hat{n}_{i-1} \)), and the other vector is tangent to the surface also at \( A \) (viz., \( \hat{e}_{i-1} \)). It is assumed the normal vector is an outward normal to the surface. The cross product of these two vectors yields a third vector producing a set of three mutually perpendicular vectors. This set of mutually perpendicular vectors can be used to establish the floating coordinate system with its origin at point \( A \). A similar procedure is used to construct floating system \( B \) on \( \Sigma_i \).

The benefit of defining systems \( A \) and \( B \) as stated is apparent from two key observations related to simple contacting surfaces. First, the surfaces must contact at a point, and second, their surface normals at the point of contact must be collinear. This means the origins of systems \( A \) and \( B \) must be coincident, and the axis of system \( A \) selected to be along the surface normal of \( \Sigma_{i-1} \) must be collinear with the axis of system \( B \) selected to be along the surface normal of \( \Sigma_i \). The normals will actually be of opposite direction since both of them are assumed to be outward normals.\(^1\) Thus, based on the manner in which systems \( A \) and \( B \) have been defined, the transformation from system \( B \) to system \( A \) (or visa versa) can

\(^1\)This assumption can be used to prevent the mathematical possibility of the “outside” of one surface contacting the “inside” of another surface.
only be a single degree-of-freedom rotation about the surfaces' common normals.

The development of a surface's normal and tangent vectors at a specified point is relatively straightforward (assuming the surface is definable analytically). By making use of parametric representations for points on the surface, these vectors can be described, relative to the surface definition system, as a function of the two parameters that locate the specified point. If these vectors are used as a basis for the surface's floating coordinate system, constructing the transformation from the floating system to the surface definition system (or visa versa) is performed easily using previously presented techniques.

3.1.1 Development of a Floating Coordinate System at a Point on a Surface

A surface described in cartesian coordinates is often presented in the form

\[ F(x, y, z) = \text{constant}. \]  \hspace{1cm} (3.2)

Points satisfying Equation (3.2) will lie on the defined surface. When the surface is described parametrically, points on its surface may be represented using

\[ x = f(\theta, \phi) \]  \hspace{1cm} (3.3)

\[ y = g(\theta, \phi) \]  \hspace{1cm} (3.4)

\[ z = h(\theta, \phi) \]  \hspace{1cm} (3.5)

or

\[ \vec{P} = (x, y, z) = ( f(\theta, \phi), g(\theta, \phi), h(\theta, \phi) ) \]  \hspace{1cm} (3.6)

2The parametric representation of points on a surface requires two parameters. Given the analytical definition of a surface, the location of a point on that surface is a two degree-of-freedom operation. For example, locating a point on the Earth's surface requires the two "parameters" called longitude and latitude.
where $\theta$ and $\phi$ are the surface parameters, and $\vec{P}$ represents a point on the surface relative to the surface definition system. Note $f(\theta, \phi)$, $g(\theta, \phi)$, and $h(\theta, \phi)$ must satisfy Equation (3.2) when substituted for $x$, $y$, and $z$, respectively.

It is possible to develop an analytical form for the normal to a surface by using the gradient of the function describing the surface. This may be written as

$$\vec{n}(x, y, z) = \nabla F = \nabla F$$

$$= \left( \frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right)$$

where $\vec{n}$ is a vector normal to the surface at a point with coordinates $x, y, z$ satisfying Equation (3.2). The expression for $\vec{n}$ may be written in terms of a parametric representation by substituting the parametric expressions $f(\theta, \phi)$, $g(\theta, \phi)$, and $h(\theta, \phi)$ for $x$, $y$, and $z$ in the components of $\nabla F$ (note this is done after forming the gradient in cartesian coordinates). The unit normal may be obtained by dividing the components of (3.8) by the magnitude of $\vec{n}$, i.e.,

$$\hat{n} = \frac{\vec{n}}{|\vec{n}|}$$

A vector tangent to a surface may be obtained easily using Equation (3.6). Allowing one of the parameters to vary while holding the other constant will describe a curve embedded in the surface. A tangent to such a curve will necessarily be tangent to the surface. Hence, the components of Equation (3.6) may be partially differentiated with respect to one of the parameters to yield components for a tangent vector. Thus,

$$\vec{t}_\theta = \left( \frac{\partial f(\theta, \phi)}{\partial \theta}, \frac{\partial g(\theta, \phi)}{\partial \theta}, \frac{\partial h(\theta, \phi)}{\partial \theta} \right)$$

$$\vec{t}_\phi = \left( \frac{\partial f(\theta, \phi)}{\partial \phi}, \frac{\partial g(\theta, \phi)}{\partial \phi}, \frac{\partial h(\theta, \phi)}{\partial \phi} \right)$$
where
\[ \hat{t}_\theta = \text{a vector tangent to the surface along curves of constant } \phi \]
\[ \hat{t}_\phi = \text{a vector tangent to the surface along curves of constant } \theta. \]

The choice of which tangent vector to use (there will always be two) is up to the discretion of the analyst. A unit tangent vector may be obtained by dividing the components of the tangent vector by the vector's length.

A rectangular cartesian coordinate system may now be formed very easily at a point on a surface. For the purposes herein, the unit normal vector will be chosen (arbitrarily) to be the $z$-axis of the system. One of the unit tangent vectors is then chosen to be the $x$-axis. The $y$-axis is obtained by forming the cross product of $\hat{n} \times \hat{t}$. Both the normal and tangent vectors are functions of only the two parameters that locate the point on the surface. Hence, if a point is defined on a surface in terms of its two parameters, a well defined, meaningful coordinate system is automatically established in terms of those two parameters.

Using $\hat{t}$, $\hat{n} \times \hat{t}$, and $\hat{n}$ for the $x, y, z$ axes of a floating system actually defines the direction cosines for the axes relative to the surface's definition system. Now recall that the rotation partition of a transformation matrix, say $T_{IJ}$, is simply a columnwise presentation of the direction cosines of system $J$'s $x, y, z$ axes relative to system $I$'s axes. Therefore, information for the rotation partition of the transformation from the surface floating system to the surface definition system is completely specified using the components of $\hat{t}$, $\hat{n} \times \hat{t}$, and $\hat{n}$. Additionally, the displacement partition of a transformation matrix, say $T_{IJ}$, represents the position of system $J$'s origin relative to system $I$. This information for the floating system is given in Equation (3.6). Hence, all of the pieces are now available to construct the transformation from a floating system at a point on a surface to the
system in which the surface is defined. All of these may be reduced to expressions in terms of two parameters. Therefore, this transformation is a function of two "degrees-of-freedom."

3.1.2 Floating Coordinate System and Its Transformations on a Spherical Surface

As an example, consider the surface of a sphere with radius \( r \). It may be described in cartesian form as

\[
F(x, y, z) = x^2 + y^2 + z^2 = r^2.
\]  

(3.12)

Its definition coordinate system is located at the center of the sphere. Parametrically, the points on its surface may be defined as

\[
x_s = r \cos \theta \sin \phi
\]

(3.13)

\[
y_s = r \sin \theta \sin \phi
\]

(3.14)

\[
z_s = r \cos \phi
\]

(3.15)

or

\[
\vec{P} = r(\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi)
\]

(3.16)

where the principal values for \( \theta \) and \( \phi \) are \( 0 \leq \theta \leq 2\pi \) and \( 0 \leq \phi \leq \pi \). Note that placing (3.13)-(3.15) into Equation (3.12) does, in fact, always satisfy Equation (3.12) for all values of \( \theta \) and \( \phi \).

The normal for a point on the surface is obtained by taking the gradient of Equation (3.12):

\[
\vec{n} = \nabla \vec{F} = (2x, 2y, 2z).
\]

(3.17)
Using Equations (3.13)–(3.15) for \(x\), \(y\), and \(z\) in Equation (3.17) yields the normal vector as a function of the surface parameters \(\theta\) and \(\phi\):

\[
\mathbf{n} = 2r(\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi).
\] (3.18)

The unit normal would then be equal to

\[
\mathbf{n} = \frac{2r(\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi)}{2r \sqrt{(\cos \theta \sin \phi)^2 + (\sin \theta \sin \phi)^2 + (\cos \phi)^2}} \quad \text{(3.19)}
\]

\[
= (\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi). \quad \text{(3.20)}
\]

A tangent vector may be obtained by differentiating the components of Equation (3.16) with respect to \(\phi\):

\[
\mathbf{t} = r(\cos \theta \cos \phi, \sin \theta \cos \phi, -\sin \phi). \quad \text{(3.21)}
\]

The magnitude of this vector can be shown to be \(r\), hence, the unit tangent is

\[
\mathbf{\hat{t}} = (\cos \theta \cos \phi, \sin \theta \cos \phi, -\sin \phi). \quad \text{(3.22)}
\]

By letting the components (i.e., direction cosines) of the \(z\)-and \(x\)-axes of the floating system be given by \(\mathbf{n}\) and \(\mathbf{\hat{t}}\), respectively, the \(y\)-axis components (i.e., direction cosines) are obtained by crossing \(\mathbf{n}\) into \(\mathbf{\hat{t}}\):

\[
\begin{align*}
x\text{-axis direction cosines} &= (\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi) \quad \text{(3.23)} \\
y\text{-axis direction cosines} &= (\cos \theta \cos \phi, \sin \theta \cos \phi, -\sin \phi) \quad \text{(3.24)} \\
\text{\(y\)-axis direction cosines} &= \mathbf{n} \times \mathbf{\hat{t}} \\
&= \begin{vmatrix} i & j & k \\ \cos \theta \sin \phi & \sin \theta \sin \phi & \cos \phi \\ \cos \theta \cos \phi & \sin \theta \cos \phi & -\sin \phi \end{vmatrix} \quad \text{(3.25)} \\
&= (-\sin \theta, \cos \theta, 0) \quad \text{(3.26) and (3.27)}
\end{align*}
\]
The components of Equations (3.24), (3.27), and (3.23) may be used directly to establish the rotation partition of the transformation from the floating system to the surface definition system. The \( x, y, \) and \( z \) direction cosines are used to fill columns one, two, and three of the transformation matrix, respectively. Additionally, column four of the matrix represents the position of the floating system’s origin relative to the surface definition system—this is given by the components of Equation (3.16).

The transformation from the floating system at a point on a sphere’s surface to the sphere’s definition system may now be written as

\[
T_{\text{definition},\text{floating}} = \begin{bmatrix}
\cos \theta \cos \phi & -\sin \theta & \cos \theta \sin \phi & r \cos \theta \sin \phi \\
\sin \theta \cos \phi & \cos \theta & \sin \theta \sin \phi & r \sin \theta \sin \phi \\
-\sin \phi & 0 & \cos \phi & r \cos \phi \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (3.28)

If the transformation from the definition system to the floating system is needed, the transformation in (3.28) may be inverted (see Section 2.3.3):

\[
T_{\text{floating},\text{definition}} = \begin{bmatrix}
\cos \theta \cos \phi & \sin \theta \cos \phi & -\sin \phi & 0 \\
-\sin \theta & \cos \theta & 0 & 0 \\
\cos \theta \sin \phi & \sin \theta \sin \phi & \cos \phi & -r \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (3.29)

### 3.1.3 Development of the Transformation Between Two Floating Coordinate Systems in Contact

Referring back to Equation (3.1) for the general surface contact pair matrix, the general process for obtaining \( T_{i-1,A} \) and \( T_{Bi} \) is as just outlined. \( T_{i-1,A} \) represents a transformation from a floating system to a surface definition system. \( T_{Bi} \) represents a transformation from a surface definition system to a floating system. The remaining transformation needed is \( T_{AB} \). This is the transformation between the two floating systems at the point of contact.
As previously stated, the transformation between the floating systems is necessarily a one degree-of-freedom rotation about an axis collinear with the surfaces' normals at the point of contact. The outward normals to the surfaces were chosen as the z-axes of the floating systems. As will be seen, the transformation from floating system B to floating system A is most easily developed in two stages.

Figure 24 shows the floating systems on two surfaces $\Sigma_{i-1}$ and $\Sigma_i$. The origins of systems $A$ and $B$ are assumed to be coincident, but the surfaces have been separated for clarity. First, a transformation from system $B$ to an intermediate system $B'$ is made. It is labeled $T_{B'B}$ and is formed by a rotation of $\pi$ about the $x$-axis of system $B$. This aligns the $x'$-axis of system $B'$ in the same direction as the $z$-axis of system $A$. Next, a transformation about the $x'$-axis of $B'$ may be written. It is label $T_{A'B'}$ and is intended to rotate the $x'$-axis of $B'$ into the $x$-axis of system $A$. The angle, $\psi$, required to do this may be thought of simply as the angle between the $x$-axes of system $B$ and $A$ measured about the $x$-axis of system $A$.

The general transformation from system $B$ to $A$, or the "floating to floating" transformation, may now be expressed as

$$T_{\text{floating floating}} = T_{A'B'} T_{B'B}$$

or

$$T_{ff} = \begin{bmatrix}
\cos \psi & -\sin \psi & 0 & 0 \\
\sin \psi & \cos \psi & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \pi & -\sin \pi & 0 \\
0 & \sin \pi & \cos \pi & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}$$

$$= \begin{bmatrix}
\cos \psi & \sin \psi & 0 & 0 \\
\sin \psi & -\cos \psi & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}$$
Note: Points $A, B'$, and $B$ are coincident; they have been shown as separated for clarity.

Figure 24: Floating systems on two surfaces in contact.
This transformation is not a function of the surfaces in contact. The only requirements for validity of this transformation are that the surfaces contact at a point, and that the floating coordinate systems are defined as presented above.

3.1.4 Discussion

It is now evident that a surface contact joint as discussed must be a five degree-of-freedom entity. The pair matrix is made up of three transformations:

\( T_{Di} \) is the transformation from \( \Sigma_i \)'s definition system to \( \Sigma_i \)'s floating system,

\( T_{AB} \) is the transformation from \( \Sigma_i \)'s floating system at point of contact to \( \Sigma_{i-1} \)'s floating system at point of contact,

\( T_{i-1,A} \) is the transformation from \( \Sigma_{i-1} \)'s floating system to \( \Sigma_{i-1} \)'s definition system.

As reasoned previously from a geometrical standpoint, a surface contact joint possesses five degrees-of-freedom. The only motion it constrains is displacement along the surface normals at the point of contact. The mathematical development of the pair matrix for this type of joint agrees with this observation.

The transformation from a surface definition system to its floating system (or visa versa) is a function of the surface's two parameters. There are two of these types of transformations involved here (one for each surface), thus, four degrees-of-freedom are used. The transformation between floating systems adds an additional degree-of-freedom to make a total of five. This is consistent with the behavior of actual mechanisms.
The bent shaft universal joint described by Beggs [12] is obviously a one degree-of-freedom mechanism. It consists of three links joined by two revolute joints and one surface contact joint. Using Grubler's equation, Equation (1.1), the predicted mechanism degree-of-freedom is one:

\[ D_m = 6(3 - 3 - 1) + (1 + 1 + 5) \]

\[ = 1. \]

This supports the conclusion that a surface contact joint is a five degree-of-freedom joint.

3.2 "Definition to Floating" and "Floating to Definition" System Transformations for Various Surfaces

The usefulness of a surface contact pair matrix is now contingent on developing "definition to floating" and "floating to definition" coordinate system transformations for surfaces likely to be in contact. Some obvious surface types are spherical, cylindrical, and planar surfaces. However, any surface representable by parametric means will work. Ellipsoids, paraboloids, and cones may also be used. Note it is not required that similar surfaces be in contact. A sphere may contact a cylinder or a plane without special procedures.

Due to the coordinate system labeling conventions employed in this work, the floating to definition transformations represent \( T_{i-1,A} \), and the definition to floating transformations represent \( T_{Bi} \). When two links, say \( i-1 \) and \( i \), are involved in a surface contact joint, surface \( \Sigma_{i-1} \) is attached to the "end" of link \( i-1 \) and is, therefore, defined in system \((xyz)_{i-1}\). The contact point on \( \Sigma_{i-1} \) will be labeled \( A \). Likewise, surface \( \Sigma_i \) is attached to the "beginning" of link \( i \) and is defined in system
The contact point on $\Sigma_i$ will be labeled $B$. It is obvious that the placement of systems $(xyz)_{i-1}$ and $(uvw)_i$ will affect how the surfaces are positioned and oriented on links $i - 1$ and $i$ respectively.

### 3.2.1 Spherical Surface

The definition and floating systems are for the typical spherical surface is shown in Figure 25. The relationships between the parameters $\theta$ and $\phi$ relative to the definition coordinate system are also shown. The transformations for a spherical surface have already been developed resulting in Equations (3.28) and (3.29). For
completeness, they will be restated here as

\[
T_{df} = \begin{bmatrix}
\cos \theta \cos \phi & -\sin \theta & \cos \theta \sin \phi & r \cos \theta \sin \phi \\
\sin \theta \cos \phi & \cos \theta & \sin \theta \sin \phi & r \sin \theta \sin \phi \\
-\sin \phi & 0 & \cos \phi & r \cos \phi \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  
(3.33)

and

\[
T_{fd} = \begin{bmatrix}
\cos \theta \cos \phi & \sin \theta \cos \phi & -\sin \phi & 0 \\
-\sin \theta & \cos \theta & 0 & 0 \\
\cos \theta \sin \phi & \sin \theta \sin \phi & \cos \phi & -r \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  
(3.34)

where \( T_{df} \) is the floating to definition transformation, and \( T_{fd} \) is the definition to floating transformation.

3.2.2 Cylindrical Surface

The function describing a cylindrical surface relative to the coordinate systems shown in Figure 26 can be stated as

\[ F(x, y, z) = x^2 + y^2 = r^2 \]  
(3.35)

where \( r \) is the radius of the cylinder, and \(-\infty < z < \infty\). Parametrically, a point on the cylinder’s surface may be written as

\[ \vec{P} = (r \cos \theta, r \sin \theta, z) \]  
(3.36)

where \( \theta \) and \( z \) are the parameters. The normal at any point on the surface is given as

\[ \vec{n} = \nabla F = (2x, 2y, 0) \]  
(3.37)

or, using the \( x, y, z \) components of (3.36),

\[ \vec{n} = 2(r \cos \theta, r \sin \theta, 0). \]  
(3.38)
The unit normal is then
\[ \hat{n} = (\cos \theta, \sin \theta, 0). \] (3.39)

The tangent at any point on the surface is given by
\[ \hat{t} = \frac{\partial \mathbf{P}}{\partial \theta} = (-r \sin \theta, r \cos \theta, 0) \] (3.40)
for which the unit tangent vector becomes
\[ \hat{t} = (-\sin \theta, \cos \theta, 0) \] (3.41)

Letting \( \hat{n} \) and \( \hat{t} \) represent the \( z \)- and \( x \)-axes of the floating system, respectively, implies the \( y \)-axis is given by
\[ y \text{-axis} = \hat{n} \times \hat{t} \] (3.42)
\[ = (0, 0, 1). \] (3.43)
The transformation from the floating system to the definition system may now be formed using Equations (3.41), (3.43), (3.39), and (3.36) as columns one through four, respectively, of the matrix:

\[
T_{df} = \begin{bmatrix}
-\sin \theta & 0 & \cos \theta & r \cos \theta \\
\cos \theta & 0 & \sin \theta & r \sin \theta \\
0 & 1 & 0 & z \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (3.44)

The transformation from the definition to the floating system is obtained by inverting Equation (3.44):

\[
T_{fd} = \begin{bmatrix}
-\sin \theta & \cos \theta & 0 & 0 \\
0 & 0 & 1 & -z \\
\cos \theta & \sin \theta & 0 & -r \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (3.45)

3.2.3 Plane Surface

The function describing a plane surface relative to coordinate systems shown in Figure 27 can be stated as

\[
F(x, y, z) = z = 0
\] (3.46)

where \(-\infty < x < \infty\) and \(-\infty < y < \infty\). A point on the plane’s surface may be written as

\[
\vec{P} = (x, y, 0).
\] (3.47)

Technically speaking, \(\vec{P}\) can be considered the parametric representation of points on the plane’s surface. The unit normal at any point on the surface is given (by inspection) as

\[
\hat{n} = (0, 0, 1).
\] (3.48)

The unit tangent at any point on the surface is given (by inspection) as

\[
\hat{t} = (1, 0, 0).
\] (3.49)
Letting \( \hat{n} \) and \( \hat{i} \) represent the \( z \)- and \( x \)-axes of the floating system, respectively, implies the \( y \)-axis is given by

\[
y\text{-axis} = \hat{n} \times \hat{i}
\]

\[
y\text{-axis} = (0, 1, 0). \tag{3.51}
\]

The transformation from the floating system to the definition system may now be formed using Equations (3.49), (3.51), (3.48), and (3.47) as columns one through four, respectively, of the matrix:

\[
T_{df} = \begin{bmatrix}
1 & 0 & 0 & x \\
0 & 1 & 0 & y \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}. \tag{3.52}
\]
The transformation from the definition to the floating system is obtained by inverting Equation (3.52):

\[
T_{fd} = \begin{bmatrix}
1 & 0 & 0 & -x \\
0 & 1 & 0 & -y \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]  

(3.53)

### 3.2.4 Conical Surface

The function describing a conical surface relative to the coordinate systems shown in Figure 28 can be stated as

\[
F(x, y, z) = \frac{x^2 + y^2}{r^2} - \frac{z^2}{c^2} = 0
\]

(3.54)
where $r$ is the radius of the cone when $z = c$, and $0 < z < \infty$. Parametrically, a point on the cone's surface may be written as

$$\vec{P} = \left(\frac{z}{c} r \cos \theta, \frac{z}{c} r \sin \theta, z\right)$$

(3.55)

where $\theta$ and $z$ are the parameters. The normal at any point on the surface is given as

$$\vec{n} = \nabla \vec{P} = \left(\frac{2x}{r^2}, \frac{2y}{r^2}, -\frac{2z}{c^2}\right).$$

(3.56)

or, using the $x, y, z$ components of (3.55),

$$\vec{n} = \frac{2z}{c} \left(\frac{\cos \theta}{r}, \frac{\sin \theta}{r}, -\frac{1}{c}\right).$$

(3.57)

The unit normal is then

$$\hat{n} = \frac{\vec{n}}{\sqrt{\left(\frac{c \cos \theta}{\sqrt{c^2 + r^2}}\right)^2 + \left(\frac{c \sin \theta}{\sqrt{c^2 + r^2}}\right)^2 + \left(-\frac{r}{\sqrt{c^2 + r^2}}\right)^2}}.$$  

(3.58)

The tangent at any point on the surface is given by

$$\vec{t} = \frac{\partial \vec{P}}{\partial \theta} = \left(-\frac{z}{c} r \sin \theta, \frac{z}{c} r \cos \theta, 0\right)$$

(3.59)

for which the unit tangent vector becomes

$$\hat{t} = (-\sin \theta, \cos \theta, 0)$$

(3.60)

Letting $\hat{n}$ and $\hat{t}$ represent the $z$- and $x$-axes of the floating system, respectively, implies the $y$-axis is given by

$$y\text{-axis } = \hat{n} \times \hat{t} = \left(\frac{r \cos \theta}{\sqrt{c^2 + r^2}}, \frac{r \sin \theta}{\sqrt{c^2 + r^2}}, \frac{c}{\sqrt{c^2 + r^2}}\right).$$

(3.61)
The transformation from the floating system to the definition system may now be formed using Equations (3.60), (3.62), (3.58), and (3.55) as columns one through four, respectively, of the matrix:

\[
T_{df} = \begin{bmatrix}
-\sin\theta & \frac{r \cos\theta}{\sqrt{c^2 + r^2}} & \frac{c \cos\theta}{\sqrt{c^2 + r^2}} & \frac{rz \cos\theta}{c} \\
\cos\theta & \frac{r \sin\theta}{\sqrt{c^2 + r^2}} & \frac{c \sin\theta}{\sqrt{c^2 + r^2}} & \frac{rz \sin\theta}{c} \\
0 & \frac{c}{\sqrt{c^2 + r^2}} & -\frac{r}{\sqrt{c^2 + r^2}} & z \\
0 & 0 & \frac{c}{\sqrt{c^2 + r^2}} & 1
\end{bmatrix}
\] (3.63)

The transformation from the definition to the floating system is obtained by inverting Equation (3.63):

\[
T_{fd} = \begin{bmatrix}
-\sin\theta & \cos\theta & 0 & 0 \\
\frac{r \cos\theta}{\sqrt{c^2 + r^2}} & \frac{r \sin\theta}{\sqrt{c^2 + r^2}} & \frac{c}{\sqrt{c^2 + r^2}} & -\frac{z \sqrt{c^2 + r^2}}{c} \\
\frac{c \cos\theta}{\sqrt{c^2 + r^2}} & \frac{c \sin\theta}{\sqrt{c^2 + r^2}} & -\frac{r}{\sqrt{c^2 + r^2}} & 0 \\
0 & 0 & \frac{c}{\sqrt{c^2 + r^2}} & 1
\end{bmatrix}
\] (3.64)

3.2.5 Paraboloid Surface

The function describing a paraboloid surface relative to the coordinate systems shown in Figure 29 can be stated as

\[
F(x, y, z) = \frac{x^2 + y^2}{r^2} - \frac{z}{c} = 0
\] (3.65)

where \( r \) is the "radius" of the paraboloid when \( z = c \), and \( 0 < z < \infty \). Parametrically, a point on the paraboloid's surface may be written as

\[
\vec{P} = \left( \sqrt{\frac{x}{c}} r \cos\theta, \sqrt{\frac{y}{c}} r \sin\theta, z \right)
\] (3.66)

where \( \theta \) and \( z \) are the parameters. The normal at any point on the surface is given as

\[
\vec{n} = \nabla F = \left( \frac{2x}{r^2}, \frac{2y}{r^2}, -\frac{1}{c} \right).
\] (3.67)
or, using the $x, y, z$ components of (3.66),

$$
\vec{n} = \left( \sqrt{\frac{2}{c}} \left( \frac{2 \cos \theta}{r} \right), \sqrt{\frac{2}{c}} \left( \frac{2 \sin \theta}{r} \right), -\frac{1}{c} \right).
$$

(3.68)

The unit normal is then

$$
\hat{n} = \left( \frac{2\sqrt{zc} \cos \theta}{\sqrt{4zc + r^2}}, \frac{2\sqrt{zc} \sin \theta}{\sqrt{4zc + r^2}}, -\frac{r}{\sqrt{4zc + r^2}} \right)
$$

(3.69)

The tangent at any point on the surface is given by

$$
\vec{t} = \frac{\partial \vec{\rho}}{\partial \theta} = \left( -\sqrt{\frac{2}{c}}(r \sin \theta), \sqrt{\frac{2}{c}}(r \cos \theta), 0 \right)
$$

(3.70)

for which the unit tangent vector becomes

$$
\hat{t} = (-\sin \theta, \cos \theta, 0)
$$

(3.71)
Letting \( \hat{n} \) and \( \hat{i} \) represent the \( z- \) and \( x- \)axes of the floating system, respectively, implies the \( y- \)axis is given by

\[
\text{y-axis} = \hat{n} \times \hat{i} = \begin{pmatrix}
\frac{r\cos\theta}{\sqrt{4zc + r^2}} & \frac{r\sin\theta}{\sqrt{4zc + r^2}} & \frac{2\sqrt{zc}}{\sqrt{4zc + r^2}} \n\end{pmatrix}
\] (3.72)

(3.73)

The transformation from the floating system to the definition system may now be formed using Equations (3.71), (3.73), (3.69), and (3.66) as columns one through four, respectively, of the matrix:

\[
T_{df} = \begin{bmatrix}
-sin\theta & \frac{r\cos\theta}{\sqrt{4zc + r^2}} & \frac{2\sqrt{zc}\cos\theta}{\sqrt{4zc + r^2}} & \frac{r\cos\theta}{\sqrt{4zc + r^2}} \\
\frac{r\sin\theta}{\sqrt{4zc + r^2}} & \frac{r\sin\theta}{\sqrt{4zc + r^2}} & \frac{2\sqrt{zc}\sin\theta}{\sqrt{4zc + r^2}} & \frac{r\sin\theta}{\sqrt{4zc + r^2}} \\
0 & \frac{2\sqrt{zc}}{\sqrt{4zc + r^2}} & \frac{r}{\sqrt{4zc + r^2}} & z \\
0 & \frac{2\sqrt{zc}}{\sqrt{4zc + r^2}} & -\frac{r}{\sqrt{4zc + r^2}} & 1
\end{bmatrix}
\] (3.74)

The transformation from the definition to the floating system is obtained by inverting Equation (3.74):

\[
T_{df} = \begin{bmatrix}
-sin\theta & \frac{r\cos\theta}{\sqrt{4zc + r^2}} & \frac{2\sqrt{zc}\cos\theta}{\sqrt{4zc + r^2}} & 0 \\
\frac{r\sin\theta}{\sqrt{4zc + r^2}} & \frac{r\sin\theta}{\sqrt{4zc + r^2}} & \frac{2\sqrt{zc}\sin\theta}{\sqrt{4zc + r^2}} & 0 \\
0 & \frac{2\sqrt{zc}}{\sqrt{4zc + r^2}} & -\frac{r}{\sqrt{4zc + r^2}} & 0 \\
0 & \frac{2\sqrt{zc}}{\sqrt{4zc + r^2}} & -\frac{r}{\sqrt{4zc + r^2}} & 1
\end{bmatrix}
\] (3.75)

### 3.3 Sphere-Cylinder Surface Contact Example

An example of a mechanism containing a three-dimensional surface contact joint is shown in Figure 30. The surfaces involved are a sphere and a cylinder. The sphere and cylinder have radii of \( a \) and \( b \), respectively. It will be informative to go through the procedure required to model this mechanism.
Figure 30: A mechanism possessing a surface contact joint.
3.3.1 Specifying the Pairing Element Coordinate Systems

The first step in modeling a mechanism is to establish the pairing element coordinate systems on the links as shown in Figure 31. These systems are chosen in a way that is consistent with the joints actually used to connect the links. For instance, both pairing elements on link 1 are revolute joints. Therefore, the $z_1$ and $w_1$ axes must be aligned with the revolute axes. For consistency, the $z_1$ and $w_1$ have been chosen as positive "out of the page." This automatically defines $w_2$ and $z_3$ since the revolute pair matrix will align them (in the same direction) with $z_1$ and $w_1$ respectively. The $x_1$ and $u_1$ may then be placed at any orientation in a plane perpendicular to $z_1$ and $w_1$, respectively. In this particular example, the $x_1$ and $u_1$ axes have been aligned with the centerline of link 1. The $y_1$ and $v_1$ axes are now automatically established using the right-hand rule.

Moving to link 2, $w_2$ is already defined (from the preceding paragraph). It makes sense to align $u_2$ with a significant feature of link 2, which in this case is link 2's centerline. Note the revolute pair matrix measures the revolute angle from $x_{i-1}$ to $u_i$ about the $z_{i-1}/w_i$ axis. This is the motivation for aligning $x_1$ and $u_2$ with body features of the links — the revolute angle will then be more meaningful and easily visualized. With $w_2$ and $u_2$ defined, $v_2$ is located using the right-hand rule.

Next, system $(xyz)_2$ must be located realizing that it is the definition system for the spherical surface. Referring back to Figure 25, the definition system for a sphere is located at the center of the sphere with $\phi$ measured from the $z$ axis and $\theta$ measured from the $x$-axis (where $\theta$ and $\phi$ are used to locate a point on the surface). With the choice for system $(xyz)_2$ shown in Figure 31, the sphere will be
Figure 31: Pairing element coordinate systems for mechanism with surface contact joints.

\[ \mathbf{I} = \mathbf{T}_{12} \mathbf{T}_{23} \mathbf{T}_{31} = [\mathbf{S}_1 \mathbf{P}_{12}][\mathbf{S}_2 \mathbf{P}_{23}][\mathbf{S}_3 \mathbf{P}_{31}] \]
located properly on the link. Since a sphere is symmetric about all three axes, the orientation of system \((xyz)_2\) is actually not important. However, the orientation shown will help visualize a point on the surface given \(\theta\) and \(\phi\).

Finally, the pairing element systems on link 3 are addressed. System \((uvw)_3\) is the definition system for the cylindrical surface. Figure 26 shows the general cylindrical surface in terms of its definition system. For the example at hand, the axis of the cylinder is perpendicular to the page of Figure 31. Hence, \(w_3\) is aligned with the desired axis of the cylinder. Axis \(u_3\) is then placed in a plane perpendicular to \(w_3\). The parameter \(\theta\) (not to be confused with \(\theta\) for the sphere) will be measured about \(w_3\) from \(u_3\)—this is by definition from Figure 26. Axis \(u_3\) is obtained using the right-hand rule. System \((xyz)_3\) is now constructed using \(z_3\) (previously specified) and defining \(x_3\) to be along link 3's centerline. Axis \(y_3\) is determined using the right-hand rule.

3.3.2 Determining the Shape Matrices

When all of the pairing element matrices have been specified, the shape matrices may be formed. For complex link shapes, it may be easiest to calculate them using the indirect method of Table 2. This would require establishing a link definition coordinate system on each link. Such a system is usually placed and oriented relative to significant physical features of the link. This system is then used to define points on the pairing element system axes in a manner that allows the shape matrix to be calculated. The indirect method of determining shape matrices, however, is more suited for a computer based approach as it is numerically intensive (albeit straightforward).
In this particular case, the shape matrices can be formed by inspection based on the definition of a transformation matrix. The shape matrix for a given link represents the transformation from its \((xyz)\) system to its \((uvw)\) system. Hence, the first three columns of the shape matrix are the direction cosines of the \(x, y, z\) axes relative to system \((uvw)\). The fourth column contains the coordinates of the system \((xyz)\)'s origin relative to system \((uvw)\). Thus, observing the coordinate systems in Figure 31, the shape matrices are seen to be

\[
S_1 = \begin{bmatrix}
-1 & 0 & 0 & -l_1 \\
0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \tag{3.76}
\]

\[
S_2 = \begin{bmatrix}
0 & 0 & -1 & -l_2 \\
1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \tag{3.77}
\]

\[
S_3 = \begin{bmatrix}
0 & -1 & 0 & 0 \\
1 & 0 & 0 & l_3 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \tag{3.78}
\]

### 3.3.3 Determining the Pair Matrices

The links in this mechanism are connected by two revolute joints and a surface contact joint. Using the pair matrix from Figure 10, the pair matrices, \(P_{12}\) and \(P_{31}\), are

\[
P_{12} = \begin{bmatrix}
\cos \theta_{12} & -\sin \theta_{12} & 0 & 0 \\
\sin \theta_{12} & \cos \theta_{12} & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \tag{3.79}
\]

\[
P_{31} = \begin{bmatrix}
\cos \theta_{31} & -\sin \theta_{31} & 0 & 0 \\
\sin \theta_{31} & \cos \theta_{31} & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \tag{3.80}
\]
The pair matrix for the surface contact joint, $P_{23}$, is developed from three intermediate transformation matrices as per Equation (3.1). The transformation from system $(uvw)_{3}$ to system $(xyz)_{2}$ (i.e., the pair matrix) is made by

1. transforming from $(uvw)_{3}$ to a floating system on link 3's cylindrical surface (this is the "definition to floating" transformation for a cylinder, $T_{fd}$, from Equation (3.45) )

2. transforming from the floating system on link 3's surface to a floating system on link 2's surface (this is the "floating to floating" transformation, $T_{ff}$ from Equation (3.32) )

3. transforming from the floating system on link 2's spherical surface to $(xyz)_{2}$ (this is the "floating to definition" transformation for a sphere, $T_{df}$, from Equation (3.33) ).

Using items 1–3, the contact pair matrix will be

$$P_{23} = T_{df} T_{ff} T_{fd}$$  (3.81)

where

$$T_{fd} = \begin{bmatrix}
-sin\theta_3 & cos\theta_3 & 0 & 0 \\
0 & 0 & 1 & -z_3 \\
cos\theta_3 & sin\theta_3 & 0 & -b \\
0 & 0 & 0 & 1
\end{bmatrix}$$  (3.82)

$$T_{ff} = \begin{bmatrix}
cos\psi_{23} & sin\psi_{23} & 0 & 0 \\
sin\psi_{23} & -cos\psi_{23} & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$  (3.83)

$$T_{df} = \begin{bmatrix}
cos\theta_2 cos\phi_2 & -sin\theta_2 & cos\theta_2 sin\phi_2 & \alpha cos\theta_2 sin\phi_2 \\
sin\theta_2 cos\phi_2 & cos\theta_2 & sin\theta_2 sin\phi_2 & \alpha sin\theta_2 sin\phi_2 \\
-sin\phi_2 & 0 & cos\phi_2 & \alpha cos\phi_2 \\
0 & 0 & 0 & 1
\end{bmatrix}$$  (3.84)
Note the constants $a$ and $b$ in matrices (3.82) and (3.84) are the radii for the sphere and cylinder respectively.

Multiplying (3.82)–(3.84) according to Equation (3.81) yields $P_{23}$ (written column by column due to its complexity) as

$$P_{23} = \begin{bmatrix}
-(C\theta_2 C\phi_2 C\psi_{23} - S\theta_2 S\psi_{23})S\theta_3 - C\theta_2 S\phi_2 C\theta_3 \\
-(S\theta_2 C\phi_2 C\psi_{23} + C\theta_2 S\psi_{23})S\theta_3 - S\theta_2 S\phi_2 C\theta_3 \\
S\phi_2 C\psi_{23} S\theta_3 - C\phi_2 C\theta_3 \\
0 \\
(C\theta_2 C\phi_2 C\psi_{23} - S\theta_2 S\psi_{23})C\theta_3 - C\theta_2 S\phi_2 S\theta_3 \\
(S\theta_2 C\phi_2 C\psi_{23} + C\theta_2 S\psi_{23})C\theta_3 - S\theta_2 S\phi_2 S\theta_3 \\
-S\phi_2 C\psi_{23} C\theta_3 - C\phi_2 S\theta_3 \\
0 \\
-C\theta_2 S\phi_2 \\
-S\theta_2 S\phi_2 \\
-C\phi_2 \\
0 \\
-(C\theta_2 C\phi_2 S\psi_{23} + S\theta_2 C\psi_{23})z_3 + (a + b)C\theta_2 S\phi_2 \\
-(S\theta_2 C\phi_2 S\psi_{23} - C\theta_2 C\psi_{23})z_3 + (a + b)S\theta_2 S\phi_2 \\
S\phi_2 S\psi_{23} z_3 + (a + b)C\phi_2 \\
1
\end{bmatrix}$$

(3.85)

where $C\theta_2 = \cos \theta_2$, $S\theta_2 = \sin \theta_2$, etc. The pair variables in $P_{23}$ are

$\theta_2$ = parameter associated with spherical surface on link 2

$\phi_2$ = parameter associated with spherical surface on link 2

$\psi_{23}$ = angle between $x$-axis of floating system on link 2 and $x$-axis of floating system on link 3; measured about common normal of surface at point of contact (the floating systems' $x$-axes)

$\theta_3$ = parameter associated with cylindrical surface on link 3

$z_3$ = parameter associated with cylindrical surface on link 3.
3.3.4 Forming the Closed Loop Matrix Product

Having calculated all of the shape and pair matrices, the closed loop matrix product can be written as

\[
I = T_{12} T_{23} T_{31}
\]  
(3.86)

\[
= [S_1 P(\theta_{12})_{12}] [S_2 P(\phi, \psi_{23}, \theta_3, z_3)_{23}] [S_3 P(\theta_{31})_{31}].
\]  
(3.87)

This matrix equation will produce 12 nontrivial equations in the 7 variables \(\theta_3\), \(\theta_{12}\), \(\phi\), \(\psi_{23}\), \(\theta_3\), and \(z_3\). As will be discussed in the next chapter, no more than six of these equations can be independent. For this particular example, six of the equations will be independent. Thus with six equations and seven variables, if one of the variables is specified, the remaining six can be calculated. This is consistent with the predicted one degree-of-freedom behavior for this mechanism (using Grüber’s equation).

The actual process of calculating the elements of Equation (3.87) will not be shown here. All of the elements of its constituent matrices have been presented in this section, however. It is obvious merely by the complexity of \(P_{23}\) that the right-hand side of (3.87) is unwieldy. This type of analysis is certainly more suited to numerical solution than it is to an algebraic solution. The important concept here, however, is the ability to form the constituents of equations like (3.87) for any mechanism. The process by which the resulting equations are solved (algebraic or numerical) is of secondary importance.
3.4 Bicubic Surface Patch

General surfaces are often not describable in terms of the usual analytical forms such as spheres, cones, ellipsoids, planes, etc. A common means for mathematically describing a "non-analytical" surface is through the use of surface patches.\[67, 63\]

A very complex surface may be constructed using a collection of these relatively easy to define patches. A single "linear Coons surface patch" (with lines of constant "u" and "w") is shown in Figure 32. Although the use of surface patches for surface contact joints has not been implemented in this dissertation, the potential to do so exists without undue effort. As with the analytical surfaces, the only requirement is the ability to define a well behaved floating coordinate system at a point on the surface.

3.4.1 Background

A patch is basically bounded by four space curves each of which are defined by parameters ranging from 0 to 1. The surface described by the patch is a function of these boundary curves. Their effects are blended together to form a smooth surface. Consider the patch shown in Figure 33. The patch is defined by families of curves in two "directions" on the surface of the patch. These directions will be called the \(u\) and \(w\) directions. If the patch is mapped to a unit square, these \(u\) and \(w\) curves are orthogonal.

The four boundary curves are labeled as \(P(0, w), P(1, w), P(u, 0), \) and \(P(u, 1)\). These must be specified since they are used to define the surface through a "blending" process. A particular implementation of a surface patch is the bicubic surface patch. This technique uses parametric cubic splines as the boundary curves.
<table>
<thead>
<tr>
<th>Top View</th>
<th>Isometric View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front View</td>
<td>Side View</td>
</tr>
</tbody>
</table>

Figure 32: Four views of a Coons surface patch with circular boundary curves.
Splines are piecewise continuous cubics connected together to form a curve. Cubics allow zero and first order continuity to be satisfied between adjoining pieces. Given a set of discrete points that lie on a curve, cubic splines are easily generated to pass through the points and approximate the curve. If that curve happens to be embedded in a surface, the corresponding cubic spline will also be (approximately) embedded in the surface. If a surface is "meshed" with cubic splines, these can act as boundary curves for individual surfaces patches approximating the surface.

A parametric cubic is given as [63]

\[ \vec{P}(t) = \vec{B}_1 + \vec{B}_2 t + \vec{B}_3 t^2 + \vec{B}_4 t^3 \]  

(3.88)

where \( \vec{P}(t) \) contains the \( x, y, z \) coordinates of the point on the curve specified by the parameter \( t \). A bicubic surface patch requires knowledge of the corner point

Figure 33: The boundary curves of a surface patch.
coordinates and the slope in each of the \( u \) and \( w \) directions at the corner points. For a given boundary curve, it meets the corners when \( t = 0 \) or \( t = 1 \). Thus, for a given boundary curve, \( P(0), P(1), P'(0), \) and \( P'(1) \) are known. Therefore (dropping the vector symbol),

\[
P(0) = B_1 \tag{3.89}
\]
\[
P(1) = B_1 + B_2 + B_3 + B_4 \tag{3.90}
\]
\[
P'(0) = B_2 \tag{3.91}
\]
\[
P'(1) = B_2 + 2B_3 + 3B_4. \tag{3.92}
\]

In matrix form, this becomes

\[
\begin{bmatrix}
P(0) \\
P(1) \\
P'(0) \\
P'(1)
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 \\
3 & 2 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
B_3 \\
B_4
\end{bmatrix} \tag{3.93}
\]

or

\[
P = M B. \tag{3.94}
\]

The coefficients, \( B \), may be solved using

\[
B = M^{-1} P \tag{3.95}
\]

where

\[
M^{-1} =
\begin{bmatrix}
2 & -2 & 1 & 1 \\
-3 & 3 & -2 & -1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix} \tag{3.96}
\]

Therefore, Equation (3.88) may be written as

\[
P(t) = F_1(t)P(0) + F_2(t)P(1) + F_3(t)P'(0) + F_4(t)P'(1) \tag{3.97}
\]
and, the $F_i(t)$ are given by

$$F_1(t) \quad F_2(t) \quad F_3(t) \quad F_4(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \\ 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$ (3.98)

These $F_i$ are used to blend adjacent pieces of the cubic spline together to ensure zero and first order continuity.

Thus far, the development has been concerned with a single curve. If two curves on opposite sides of a patch are known, a lofted surface between them may be established by linearly interpolating between them. Suppose the two boundary curves in the u direction are considered. These occur at $w = 0$ and $w = 1$. Hence, call these u direction curves $P(u, 0)$ and $P(u, 1)$. Then

$$Q(u, w) = P(u, 0)(1 - w) + P(u, 1)w$$ (3.99)

where $Q(u, w)$ is a vector containing the $x, y, z$ components of a point on the lofted surface specified by $(u, w)$. Using Equations (3.97) and (3.98), the lofted surface between the two curves is given by

$$Q(u, w) = P(0, w)(1 - 3u^2 + 2u^3) + P(1, w)(3u^2 - 2u^3) + P_u(0, w)(u - 2u^2 + u^3) + P_u(1, w)(-u^2 + u^3)$$ (3.100)

where

$$P_u(u, w) = \frac{\partial P(u, w)}{\partial u} \bigg|_{u=u_i}.$$ (3.101)

A similar expression can be determined by lofting a surface between the other two boundary curves. These two lofted surfaces are then added together\(^3\) to form the

\(^3\)The corner point values, tangents, and cross-derivatives are subtracted in a weighted amount. Otherwise, their effects would be added twice since both sets of opposite boundary curves contain the full effect of the corner points.
bicubic surface. The final form of the approximated surface is

\[ Q(u, w) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} N P N^T \begin{bmatrix} w^3 \\ w^2 \\ w \\ 1 \end{bmatrix} \]  

(3.102)

where \( N = M^{-1} \) given in Equation (3.96) and

\[
P = \begin{bmatrix}
P(0,0) & P(0,1) & P_w(0,0) & P_w(0,1) \\
P(1,0) & P(1,1) & P_w(1,0) & P_w(1,1) \\
P_u(0,0) & P_u(0,1) & P_{u,w}(0,0) & P_{u,w}(0,1) \\
P_u(1,0) & P_u(1,1) & P_{u,w}(1,0) & P_{u,w}(1,1)
\end{bmatrix}. \]  

(3.103)

Also

\[ P_u(u_i, w_j) = \left. \frac{\partial P(u, w)}{\partial u} \right|_{u=u_i, w=w_j} \]
\[ P_w(u_i, w_j) = \left. \frac{\partial P(u, w)}{\partial w} \right|_{u=u_i, w=w_j} \]
\[ P_{u,w}(u_i, w_j) = \left. \frac{\partial^2 P(u, w)}{\partial u^2} \right|_{u=u_i, w=w_j} \]

The matrix \( P \) can be broken down into the following categories:

\[
P = \begin{bmatrix}
\text{corner coordinates} & \text{\( u \)-tangent vectors} \\
\text{\( u \)-tangent vectors} & \text{twist vectors}
\end{bmatrix}
\]  

(3.104)

All of the elements of \( P \) must be specified to define one surface patch. The twist vectors are difficult to ascertain, and therefore, are sometimes assumed as zero. This implementation is called an \( F \)-patch due to Ferguson. This assumption may causes some flattening in the surface, however, smaller patches will counteract this effect.

Scherrer [67] modeled the articular surfaces of a canine scapula and proximal humerus using bicubic surface patches. A grid was laid out on the surfaces to
provide the locations of the patch corner points. A coordinate measuring device was used to determine the coordinates of the corner points. Cubic splines were passed through the corner points, along the mesh lines defining the patches, to calculate tangents at the corner points. "Measuring" a tangent would be a difficult if not impossible task.

3.4.2 Establishing a Floating Coordinate System on a Surface Patch

Given the definition of a surface patch as in Equation (3.102), it is fairly simple to construct a floating coordinate system at any point on the surface. If the parameters \( u \) and \( w \) are specified on the surface, the floating coordinate system must have the associated point as its origin. Furthermore, the \( z \)-axis of the system must be the outward surface normal at this point. The surface normals for the analytical surfaces were obtained directly using the mathematical definitions of the surfaces. The surface patches will require an indirect approach.

A surface normal at a specified point is easily obtained if two noncollinear surface tangents can be determined at that same point. The cross product of the two tangent vectors will produce a mutually perpendicular vector normal to the surface. The tangent vectors can be obtained at any point, \((u, w)\), by taking the partial derivative of the patch's defining equation. Therefore, tangents may be obtained by partially differentiating Equation (3.102) once with respect to \( u \), and once with respect to \( w \). Hence,

\[
\vec{t}_u = \frac{\partial Q(u, w)}{\partial u} \tag{3.105}
\]
\[
\mathbf{NPN}^T \begin{bmatrix}
3u^2 & 2u & 1 & 0
\end{bmatrix} \begin{bmatrix}
w^3 \\
w^2 \\
w \\
1
\end{bmatrix} \tag{3.106}
\]

\[
\mathbf{\hat{t}_w} = \frac{\partial Q(u, w)}{\partial w} \tag{3.107}
\]

\[
= \begin{bmatrix}
u^3 & u^2 & u & 1\end{bmatrix} \mathbf{NPN}^T \begin{bmatrix}
w^2 \\
2w \\
1 \\
0
\end{bmatrix} \tag{3.108}
\]

Now consider a "rule" stating the \(u\)-direction crossed into the \(w\)-direction must produce an outward normal. If surface patches adhere to this rule when defined, the cross product \(\mathbf{\hat{t}_u} \times \mathbf{\hat{t}_w}\) will always produce an outward normal at the point \((u, w)\). Therefore, the \(z\) axis for the floating system will be given by

\[
\hat{z} = \frac{\mathbf{\hat{t}_u} \times \mathbf{\hat{t}_w}}{|\mathbf{\hat{t}_u} \times \mathbf{\hat{t}_w}|} \tag{3.109}
\]

Since \(\hat{z}\) must be perpendicular to \(\mathbf{\hat{t}_u}\), let

\[
\hat{x} = \frac{\mathbf{\hat{t}_u}}{|\mathbf{\hat{t}_u}|} \tag{3.110}
\]

The \(y\) axis is then given by

\[
\hat{y} = \hat{z} \times \hat{x}. \tag{3.111}
\]

The components of the \((\hat{x}, \hat{y}, \hat{z})\) unit vectors represent the direction cosines of these axes as seen in the definition system of the surface. The definition system is the system in which the corner point coordinates and tangents are defined. Hence, if the components of \(\hat{x}, \hat{y}, \) and \(\hat{z}\) are used to fill the columns of a \(3 \times 3\) matrix, the rotation partition of the "floating-to-definition" transformation matrix will be known. Additionally, the displacement partition is determined from \(Q(u, w)\) as this will give the coordinates of the floating system's origin relative to the definition
Therefore, the floating-to-definition system may be stated as

\[
T_{floating, definition} = \begin{bmatrix}
\hat{x}_x & \hat{y}_x & \hat{z}_x & Q(u, w)_x \\
\hat{x}_y & \hat{y}_y & \hat{z}_y & Q(u, w)_y \\
\hat{x}_z & \hat{y}_z & \hat{z}_z & Q(u, w)_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  
(3.112)

The definition-to-floating transformation may be obtained by inverting the above matrix according to

\[
T_{definition, floating} = \begin{bmatrix}
T_{fd}^T & -T_{fd}^T D_{fd} \\
0^T & 1
\end{bmatrix}
\]  
(3.113)

where \( T_{fd} \) is the floating-to-definition transformation.
CHAPTER IV

Development of Kinematic Analysis Equations

4.1 Introduction

This chapter will present the general theory used in developing useful equations for the displacement, velocity, and acceleration analysis of mechanisms. The approach is based on the use of transformation matrices to describe the relative positions of the links that make up a mechanism. These matrices are functions of link geometry and joint pair variables. The joint pair variables are the entities for which displacement, velocity, and acceleration information is being sought. The zero-th, first, and second time derivatives of the joint pair variables are the dependent variables of the displacement, velocity, and acceleration analyses, respectively. With values for these basic entities, all other quantities related to the displacement, velocity, and acceleration characteristics of the links are calculable. Examples of such quantities are the position, velocity, and acceleration of a link's center of gravity (useful for a force analysis). Also of interest, might be the angular velocity and angular acceleration of a link.

4.2 The Forward and Inverse Kinematic Problems

Consider an open loop chain of binary links with the initial link specified as the ground link. The links are assumed to be connected by mathematically well de-
fined joints. Assume also, that the kinematic equations are available to relate the mechanism's link positions with its joint variables. If the joint variables are given, the determination of each link's whereabouts relative to the ground link is straightforward. The given values for the joint variables are simply placed in the kinematic equations (actually matrix equations consisting of coordinate system transformations). The results of these equations can be interpreted to directly yield the locations and orientations of coordinate systems embedded in each link. This establishes the location and orientation of the links. This process might be referred to as the “forward kinematics problem.”

On the other hand, consider the same open loop mechanism, but in this case, the desired positions and orientations of some (or all of the links) are to be specified. Given the same kinematic equations as discussed in the preceding paragraph, it is now the joint variables that must be determined given some (or all) of the link locations and orientations. This process is called the “inverse kinematics problem.”

The kinematic equations used for either the forward or inverse kinematic analysis tend to be highly nonlinear in terms of the joint variables. This is not a concern in the forward kinematic solution since the arguments (i.e., the joint variables) of the nonlinear terms are known. Their given values are simply inserted into the kinematic equations to yield the link positions. The inverse kinematics problem is much less straightforward in that the arguments of the nonlinear terms are the unknowns of the analysis. Not only will the methods of solution be more complex in this case, but additionally, the solutions will tend to be multivalued. It will be common for more than one solution to exist.

The nonlinear and multivalued characteristics of the inverse kinematics problem
are evident in the solutions for the planar slider crank mechanism presented in Sections 2.5.1 and 2.5.2. Although this is a closed loop mechanism, its analysis involved determining values for joint variables caused by the specification of its input link position (its position is specified by the crank angle, $\theta_2$). The equations developed were nonlinear in terms of the joint variables, and the solutions to the equations were multivalued due to the occurrence of a square root function in the expression for the slider pair variable.

The kinematic solution process used throughout this research addresses the inverse kinematics problem. The goal of solution is to determine the dependent joint variables of a mechanism based on the specified motion of the mechanism's input link(s). The motion of an input link is often specified by controlling one or more of its pairing elements' joint variables. Since the class of mechanisms treated by this research is that of one degree-of-freedom behavior, the specification of only one joint variable will be necessary in order to solve for the remaining joint variables. However, all of the techniques presented are directly applicable to mechanisms possessing more than one degree-of-freedom.

4.3 Matrix Based Displacement Analysis

The first step in approaching either the forward or inverse kinematics problem is the development of kinematic equations in terms of a mechanism's joint variables. Well defined procedures for doing this have already been presented through the use of Denavit and Hartenberg's or Sheth and Uicker's approaches. Their mechanism modeling techniques have been shown to produce kinematic equations in terms of a mechanism's link dimensions and joint variables. The solution to these equa-
tions for the unknown joint variables in terms of the input joint variable(s) is the goal of the displacement analysis. Hence, an example of a displacement analysis has already been performed on the slider crank mechanism of Sections 2.5.1 and 2.5.2. The following paragraphs provide a general formulation for the development of the displacement analysis kinematic equations. This formulation is valid for mechanisms modeled using a transformation matrix based approach.

4.3.1 Open Loop Mechanism

Consider an open chain of connected binary links numbered 1 to \( N \). If a coordinate system is attached to each body, it is possible to write a transformation matrix "across" each joint (i.e., between coordinate systems of successive bodies). These transformations will describe the relative motion between adjacent bodies in the chain and will be functions of the joint dofs. If \( \hat{f}_{ij} \) represents the joint degrees-of-freedom permitted by the joint between links \( i \) and \( j \), the transformation from system \( j \) to system \( i \) will be a function of \( \hat{f}_{ij} \), i.e.,

\[
T_{ij} = T(\hat{f}_{ij})_{ij}. \tag{4.1}
\]

Using the matrix product chain rule from (2.29), the following can be written:

\[
T(\hat{f}_{12}, \hat{f}_{23}, \ldots, \hat{f}_{N-1,N})_{1N} = T(\hat{f}_{12})_{12} T(\hat{f}_{32})_{23} \cdots T(\hat{f}_{N-1,N})_{N-1,N}. \tag{4.2}
\]

Hence, the transformation from link \( N \) to link 1, \( T_{1N} \), must be a function of all of the joint dofs possessed by the joints connecting the links together. For ease of notation, all of the joint degrees of freedom will be expressed as generalized
components of a vector $\vec{Q}$:

\[
\vec{Q} = \begin{bmatrix}
\vec{f}_{12} \\
\vec{f}_{23} \\
\vdots \\
\vec{f}_{N-1,N}
\end{bmatrix} = \begin{bmatrix}
q_1 \\
q_2 \\
q_3 \\
q_F
\end{bmatrix}
\]  

(4.3)

where $F$ = the total number of joint dofs that constrain the links.

Now assume that appropriate values for the $F$ joint dofs, $\vec{Q}$, must be found to place system $N$ instantaneously at a precise orientation and location relative to system $1$.\(^1\) This would then be classified as an inverse kinematics problem since the joint variables are the unknowns.

The desired orientation and location of system $N$ relative to system $1$ can be described instantaneously by a transformation matrix, say $C_{1N}$. The first three columns of $C_{1N}$ will contain the direction cosines of system $N$'s $x, y, z$ axes relative to system $1$, and the fourth column will contain the location of system $N$'s origin relative to system $1$'s origin. Since the matrix product from Equation (4.2) also represents the transformation from system $N$ to system $1$ (but as a function of the joint degrees-of-freedom), then

\[
T(\vec{Q})_{1N} = C_{1N}.
\]  

(4.4)

On an element level, Equation (4.4) may be written as

\[
\begin{bmatrix}
\dot{t}(\vec{Q})^{11} & \dot{t}(\vec{Q})^{12} & \dot{t}(\vec{Q})^{13} & \dot{t}(\vec{Q})^{14} \\
\dot{t}(\vec{Q})^{21} & \dot{t}(\vec{Q})^{22} & \dot{t}(\vec{Q})^{23} & \dot{t}(\vec{Q})^{24} \\
\dot{t}(\vec{Q})^{31} & \dot{t}(\vec{Q})^{32} & \dot{t}(\vec{Q})^{33} & \dot{t}(\vec{Q})^{34} \\
0 & 0 & 0 & 1
\end{bmatrix}
= \begin{bmatrix}
c^{11} & c^{12} & c^{13} & c^{14} \\
c^{21} & c^{22} & c^{23} & c^{24} \\
c^{31} & c^{32} & c^{33} & c^{34} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(4.5)

\(^1\)Since an open chain is being considered, this situation is identical to that found in the typical robotic manipulator.
The matrix equation (4.5) implies sixteen scalar equations which are functions of $\vec{Q}$. Four of the equations are trivial (i.e., $0 = 0$ or $1 = 1$). Of the remaining twelve equations, only 6 are independent. If $T$ is again presented in terms of a rotation partition and a displacement partition (see Equation (2.17)), three equations will come from the rotation partition and three will come from the displacement partition.

The equations generated from the displacement partition (i.e., column 4) are simply:

\begin{align*}
    t(\vec{Q})^{14} &= c^{14} \\
    t(\vec{Q})^{24} &= c^{24} \\
    t(\vec{Q})^{34} &= c^{34}
\end{align*}

(4.6) \hspace{1cm} (4.7) \hspace{1cm} (4.8)

However, there are nine equations represented by the rotation partition—only three will be useful. Generally speaking, any three may be picked such that they are not all in the same column or same row. For example, three independent equations in general might be

\begin{align*}
    t(\vec{Q})^{11} &= c^{11} \\
    t(\vec{Q})^{21} &= c^{21} \\
    t(\vec{Q})^{32} &= c^{32}
\end{align*}

(4.9) \hspace{1cm} (4.10) \hspace{1cm} (4.11)

(although these would not be the only three possible).

Considering Equations (4.6)-(4.11), it is obvious that $\vec{Q}$ cannot consist of more than six unknowns (for the inverse kinematics problem) since only six equations are available for solution. If the constituent matrices, $T_{ij}$, that make up $T_{IN}$
are expressed and manipulated algebraically, Equations (4.6)-(4.11) will also be algebraic. Theoretically, they may then be solved to yield a general algebraic kinematic solution. However, these equations are often extremely nonlinear (as evidenced by the form of (2.62)-(2.64)) and not easily solved. Hence, it is common practice to employ numerical solution techniques.

4.3.2 Closed Loop Mechanism

Suppose now that links 1 to $N$ are connected by joints to form a closed chain mechanism. Using the matrix product chain rule on a closed loop (see Equation (2.49)), the following may be written:

$$T(\vec{Q})_{1,1} = T(f_{12})_{12} T(f_{23})_{23} \cdots T(f_{N-1,N})_{N-1,N} T(f_{N1})_{N1}$$

$$= T(f_{12}, f_{23}, \ldots, f_{N-1,N}, f_{N1})_{1,1}$$

$$= I_4$$

where

$$\vec{Q} = \begin{bmatrix} f_{12} \\ f_{23} \\ \vdots \\ f_{N-1,N} \\ f_{N1} \end{bmatrix}$$

On an element level, (4.14) can be written as:

$$T(\vec{Q})_{1,1} = \begin{bmatrix} t(\vec{Q})^{11} & t(\vec{Q})^{12} & t(\vec{Q})^{13} & t(\vec{Q})^{14} \\ t(\vec{Q})^{21} & t(\vec{Q})^{22} & t(\vec{Q})^{23} & t(\vec{Q})^{24} \\ t(\vec{Q})^{31} & t(\vec{Q})^{32} & t(\vec{Q})^{33} & t(\vec{Q})^{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, a transformation product around closed loop of connected bodies will always be equal to the constant identity matrix. This may be used, as with the
open loop situation, to produce six equations that are functions of the joint dofs. If no more than six joint dofs are unknown, the system of equations may be solved for those unknown dofs. In the case of a general, closed loop, one degree-of-freedom spatial mechanism, \( \tilde{Q} \) would consist of (at most) seven parameters. One of them would be designated as the input (or independent) dof, and the remaining six could be determined from the system of six equations. The process of determining values for the joint dofs is often called the position or displacement analysis.

In summary, a general procedure for a matrix based approach to the displacement analysis of a single closed loop mechanism would be as listed in Table 3. This process is the basis of most transformation matrix approaches to kinematic displacement analysis. The literature pertaining to matrix based kinematic analysis usually deals with differences and advances in implementation of the items listed in Table 3 — the underlying basis to the problem formulation is essentially identical to the theory presented above.

4.4 Matrix Based Kinematic Velocity Analysis

The general theory for a velocity analysis follows directly from the displacement analysis theory. Completing the displacement analysis for a given mechanism will produce values (algebraically or numerically) for the elements of \( \tilde{Q} \)—a vector of the joint dof variables for the joints in the mechanism. The i-th component of \( \tilde{Q} \) is symbolized by \( \tilde{q}_i \). The total number of joint dofs in the mechanism will be \( F \). Results from the velocity analysis will produce the time rates of change for these \( F \) elements of \( \tilde{Q} \) and are symbolized collectively as \( \dot{\tilde{Q}} \).
Table 3: General procedure for matrix based displacement analysis

Closed Loop Mechanism Displacement Analysis Steps:

1. Attach a coordinate system to each link in the mechanism.

2. Develop transformation matrices between the coordinate systems of adjacent links as functions of the degrees-of-freedom possessed by the joints connecting the links.

3. Form the closed loop matrix product and set it equal to the identity matrix. This will yield sixteen equations.

4. Pick six nontrivial and independent equations from the sixteen available (three each from the rotation and displacement partitions).

5. Specify enough of the joint dofs (i.e., the mechanism input(s)) such that no more than six unknown dofs remain.

6. Solve the system of six (probably highly nonlinear) equations for the unknown joint dofs.
4.4.1 Open Loop Mechanism

The basis for the velocity analysis of an open loop mechanism is obtained by differentiating Equation (4.4) with respect to time to get:

\[ \frac{d}{dt} T(\vec{q})_{1N} = \frac{d}{dt} C_{1N}. \]  

(4.16)

Since \( T_{1N} \) is a function of \( \vec{q} \) (whose components \( q_i \) are functions of time), the left-hand side of (4.16) must be differentiated using the chain rule:

\[ \frac{\partial T(\vec{q})_{1N}}{\partial q_1} \frac{dq_1}{dt} + \frac{\partial T(\vec{q})_{1N}}{\partial q_2} \frac{dq_2}{dt} + \ldots + \frac{\partial T(\vec{q})_{1N}}{\partial q_N} \frac{dq_N}{dt} = \frac{d}{dt} C_{1N} \]  

(4.17)

or

\[ \sum_{i=1}^{F} \frac{\partial T(\vec{q})_{1N}}{\partial q_i} \dot{q}_i = \dot{C}_{1N} \]  

(4.18)

Recall that \( C_{1N} \) is a mathematical means of describing the location and orientation of system \( N \) relative to system 1. Therefore, \( \dot{C}_{1N} \) is a description of how this location and orientation changes as a function of time. More specifically, the "rotation" partition of \( \dot{C}_{1N} \) is related to the angular velocity of system \( N \) relative to system 1, and the "displacement" partition corresponds to the velocity of the origin of system \( N \) relative to the origin of system 1.

If values for the \( \dot{q}_i \) are specified, they may be inserted into Equation (4.18) to determine \( \dot{C}_{1N} \). This assumes that the partial derivatives of \( T_{1N} \) with respect to the joint dofs are calculable (algebraically or numerically). For the velocity portion of an inverse kinematics problem, \( \dot{C}_{1N} \) will be specified and the corresponding joint rates \( (\dot{q}_i, i = 1,2,\ldots,F) \) necessary to produce \( \dot{C}_{1N} \) are the unknowns to be found. Similar to the displacement analysis, Equation (4.18) is capable of producing six independent equations in terms of the unknown \( \dot{q}_i \). However, unlike the unknowns
in the displacement analysis, the \( \dot{q}_i \) appear linearly in (4.18). Thus the inverse kinematics velocity analysis is more easily solved than is the nonlinear displacement analysis. The six independent equations may be chosen in the same manner as with the displacement analysis: three equations come from the “rotation” partition (any three but not all from the same column or row), and three more equations come from the “displacement” partition.

### 4.4.2 Closed Loop Mechanism

The velocity problem formulation for a closed loop mechanism is obtained by differentiating Equation (4.14) with respect to time:

\[
\frac{d}{dt} T(\dot{Q})_{1,1} = \frac{d}{dt} I_4
\]  
\[
(4.19)
\]

or

\[
\sum_{i=1}^{P} \frac{\partial T(\dot{Q})_{1,1}}{\partial q_i} \dot{q}_i = 0_4
\]  
\[
(4.20)
\]

by using the derivative chain rule and because \( I_4 \) is a constant matrix (\( 0_4 \) is a \( 4 \times 4 \) matrix of zeros). Equation (4.20) is also capable of providing six independent, linear equations in terms of the unknown \( \dot{q}_i \). \( \dot{Q} \) will in general consist of both independent and dependent elements. The independent \( \dot{q}_i \)'s are the input joint dof velocities, and the dependent \( \dot{q}_i \)'s are the resulting velocities of the remaining joint dofs. In order to have a unique solution, there may be no more than six unknown (dependent) elements in \( \dot{Q} \).

For ease of notation let

\[
\frac{\partial T(\dot{Q})_{1,1}}{\partial q_i} = \frac{\partial}{\partial q_i} T_{1,1}
\]  
\[
(4.21)
\]
where
\[ t_{qi}^{kl} = \frac{\partial T_{11}^{kl}}{\partial q_i} \]

Since the last row of \( \frac{\partial T_{11}^{kl}}{\partial q_i} \) is always zero, twelve nontrivial equations are implied by (4.20) and may be written as:

\[
\begin{align*}
t_{q_1}^{11} \dot{q}_1 + t_{q_2}^{11} \dot{q}_2 + \ldots + t_{q_p}^{11} \dot{q}_p &= 0 \\
t_{q_1}^{21} \dot{q}_1 + t_{q_2}^{21} \dot{q}_2 + \ldots + t_{q_p}^{21} \dot{q}_p &= 0 \\
t_{q_1}^{31} \dot{q}_1 + t_{q_2}^{31} \dot{q}_2 + \ldots + t_{q_p}^{31} \dot{q}_p &= 0 \\
t_{q_1}^{12} \dot{q}_1 + t_{q_2}^{12} \dot{q}_2 + \ldots + t_{q_p}^{12} \dot{q}_p &= 0 \\
t_{q_1}^{22} \dot{q}_1 + t_{q_2}^{22} \dot{q}_2 + \ldots + t_{q_p}^{22} \dot{q}_p &= 0 \\
t_{q_1}^{32} \dot{q}_1 + t_{q_2}^{32} \dot{q}_2 + \ldots + t_{q_p}^{32} \dot{q}_p &= 0 \\
t_{q_1}^{13} \dot{q}_1 + t_{q_2}^{13} \dot{q}_2 + \ldots + t_{q_p}^{13} \dot{q}_p &= 0 \\
t_{q_1}^{23} \dot{q}_1 + t_{q_2}^{23} \dot{q}_2 + \ldots + t_{q_p}^{23} \dot{q}_p &= 0 \\
t_{q_1}^{33} \dot{q}_1 + t_{q_2}^{33} \dot{q}_2 + \ldots + t_{q_p}^{33} \dot{q}_p &= 0 \\
t_{q_1}^{14} \dot{q}_1 + t_{q_2}^{14} \dot{q}_2 + \ldots + t_{q_p}^{14} \dot{q}_p &= 0 \\
t_{q_1}^{24} \dot{q}_1 + t_{q_2}^{24} \dot{q}_2 + \ldots + t_{q_p}^{24} \dot{q}_p &= 0 \\
t_{q_1}^{34} \dot{q}_1 + t_{q_2}^{34} \dot{q}_2 + \ldots + t_{q_p}^{34} \dot{q}_p &= 0
\end{align*}
\]
Equations (4.24)-(4.26) come from the elements of column one, (4.27)-(4.29) come from column two, and so on. Equations (4.24)-(4.32) are from the 3 x 3 "rotation" partition of which only three equations are independent. When solving this system of equations for the \( \dot{q}_i \)'s, three independent equations may be chosen from these first nine as before: the three may not all be from the same row or column. Equations (4.33)-(4.35) are three more independent equations coming from the 3 x 1 "displacement" partition.

Equations (4.24)-(4.35) may be written in matrix form as follows:

\[
\begin{bmatrix}
\begin{array}{ccccc}
  t_{11} & t_{12} & \ldots & t_{1q} \\
  t_{21} & t_{22} & \ldots & t_{2q} \\
  t_{31} & t_{32} & \ldots & t_{3q} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{q_1} & t_{q_2} & \ldots & t_{qq} \\
\end{array}
\end{bmatrix}
\begin{bmatrix}
  \dot{q}_1 \\
  \dot{q}_2 \\
  \vdots \\
  \dot{q}_{q_2} \\
\end{bmatrix}
\begin{bmatrix}
  0 \\
  0 \\
  0 \\
  \vdots \\
  0 \\
\end{bmatrix}
\]

(4.36)

As written above, Equation (4.36) includes all twelve nontrivial equations implied by (4.20). The matrix of partial derivatives will be a 12 x \( F \) matrix. Since there will be at most six independent equations among these twelve, (4.36) is normally pared down to contain just the independent equations. In the most general sense, the partial derivative matrix will then become a 6 x \( F \) matrix. This matrix is often referred to as the Jacobian. Equation (4.36) may then be written as:

\[
J \ddot{\mathbf{q}} = \mathbf{0}
\]

(4.37)

where \( J \) = the Jacobian matrix.

Assuming that the Jacobian is calculable, (4.37) is a straightforward linear system of equations. In order to be solvable for the general case, the "coefficient
matrix" (i.e., \( J \)) must be square. In other words, \( \dot{\mathbf{Q}} \) must not contain more than six unknowns. If a given closed loop mechanism has \( F_I \) degrees-of-freedom, then \( F_I \) components of \( \dot{\mathbf{Q}} \) must be specified as velocity inputs to the mechanism. The remaining components of \( \dot{\mathbf{Q}} \) will be the unknowns of the problem; these will be \( F_D \) in number where the total number of joint degrees of freedom can now be broken down as:

\[
F = F_I + F_D. \tag{4.38}
\]

It now makes sense to partition (4.37) as follows:

\[
\begin{bmatrix}
J_I & J_D
\end{bmatrix}
\begin{bmatrix}
\dot{Q}_I \\
\dot{Q}_D
\end{bmatrix} = \mathbf{0}
\tag{4.39}
\]

where

\[
\dot{Q}_I = \text{the independent (specified) joint dof velocities}
\]

\[
\dot{Q}_D = \text{the dependent (unknown) joint dof velocities}
\]

\[
J_I = \text{columns of Jacobian associated with independent joint dof velocities}
\]

\[
J_D = \text{columns of Jacobian associated with dependent joint dof velocities}
\]

This can be further written as:

\[
J_I \dot{Q}_I + J_D \dot{Q}_D = \mathbf{0} \tag{4.40}
\]

or

\[
J_D \dot{Q}_D = -J_I \dot{Q}_I. \tag{4.41}
\]

The right-hand side of (4.41) will be completely specified by inputs to the mechanism. Hence, \( J_D \) on the left-hand side of (4.41) must be invertible in order to
solve for the dependent joint dofs, \( \dot{Q}_D \). A necessary condition is that \( \mathbf{J}_D \) be square. However, even if \( \mathbf{J}_D \) is square, it still may be singular due to special mechanism geometry or configuration.

If a numerical solution technique is being employed, the components of \( \dot{Q}_D \) are easily determined using Gaussian elimination techniques. Although the theory presented can also be used for algebraic solutions, it is usually more straightforward to directly differentiate the displacement equations obtained from an algebraic position analysis. This eliminates the need to symbolically invert the (in general) \( 6 \times 6 \) dependent portion of the Jacobian.

### 4.5 Matrix Based Kinematic Acceleration Analysis

The general approach to a matrix based acceleration analysis follows directly from the velocity analysis formulation. As would be expected, the velocity equations can be differentiated with respect to time to yield a system of equations that describe the acceleration characteristics of a mechanism. Although the exercise of performing the differentiation becomes increasingly involved when compared with the development of the velocity equations, the result is again a linear system of equations that, in general, is easily solved.

#### 4.5.1 Open Loop Mechanism

The compact form of the velocity system of equations for an open loop mechanism is given by Equation (4.18). This may be differentiated with respect to time to yield the acceleration system of equations:

\[
\frac{d}{dt} \sum_{i=1}^{P} \frac{\partial T(\dot{q})_{1N}}{\partial q_i} \ddot{q}_i = \frac{d}{dt} \dot{C}_{1N} \tag{4.42}
\]
If just the \(i\)-th term of (4.42) is differentiated, the following will be obtained:

\[
\frac{d}{dt} \left( \frac{\partial T(\vec{q})_{1N}}{\partial q_i} \dot{q}_i \right) = \frac{d}{dt} \left( \frac{\partial T(\vec{q})_{1N}}{\partial q_i} \right) \dot{q}_i + \frac{\partial T(\vec{q})_{1N}}{\partial q_i} \frac{d}{dt} \ddot{q}_i \tag{4.43}
\]

\[
= \left( \sum_{j=1}^{P} \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_i \partial q_j} \dot{q}_j \right) \dot{q}_i + \frac{\partial T(\vec{q})_{1N}}{\partial q_i} \ddot{q}_i \tag{4.44}
\]

Hence, (4.42) may be written as:

\[
\sum_{i=1}^{P} \left[ \left( \sum_{j=1}^{P} \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_i \partial q_j} \dot{q}_j \right) \dot{q}_i + \frac{\partial T(\vec{q})_{1N}}{\partial q_i} \ddot{q}_i \right] = \ddot{\mathbf{C}}_{1N} \tag{4.45}
\]

or, after distributing the outermost summation and rearranging:

\[
\sum_{i=1}^{P} \frac{\partial T(\vec{q})_{1N}}{\partial q_i} \ddot{q}_i + \sum_{i=1}^{P} \sum_{j=1}^{P} \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_i \partial q_j} \dot{q}_i \dot{q}_j = \ddot{\mathbf{C}}_{1N} \tag{4.46}
\]

The matrix \(\ddot{\mathbf{C}}_{1N}\) represents system \(N\)'s second time rate of change of location and orientation with respect to system 1. The \(3 \times 3\) "rotation" portion of \(\ddot{\mathbf{C}}_{1N}\) is related to the angular acceleration of system \(N\) relative to system 1; and the \(3 \times 1\) "displacement" partition represents the acceleration of the origin of system \(N\) relative to the origin of system 1.

The second term on the left-hand side of (4.46) can be manipulated to a form that economizes calculation and also provides some physical significance to the equation. Since the order of differentiation in this term is not significant, it is obvious that:

\[
\frac{\partial^2 T(\vec{q})_{1N}}{\partial q_k \partial q_i} \ddot{q}_k \dot{q}_i + \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_i \partial q_k} \dot{q}_i \ddot{q}_k = 2 \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_k \partial q_i} \dot{q}_k \ddot{q}_i \tag{4.47}
\]

Therefore, the second term on the left-hand side of (4.46) may be written as:

\[
\sum_{i=1}^{P} \sum_{j=1}^{P} \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_i \partial q_j} \dot{q}_i \dot{q}_j = 2 \sum_{i=1}^{P} \left( \sum_{j=i+1}^{P} \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_i \partial q_j} \dot{q}_i \ddot{q}_j + \sum_{i=1}^{P} \frac{\partial^2 T(\vec{q})_{1N}}{\partial q_i^2} \ddot{q}_i^2 \right) \tag{4.48}
\]
Note that the summation definition for the first term of (4.48)'s right-hand side is a means of referencing only elements where \( j > i \). The coefficient of 2 incorporates the effect of elements where \( j < i \). The second term of (4.48)'s right-hand side represents the elements where \( j = i \).

By putting (4.48) into (4.46), the complete system of acceleration equations may now be written as:

\[
\sum_{i=1}^{P} \frac{\partial T(\ddot{q})_{iN}}{\partial q_i} \ddot{q}_i + \sum_{i=1}^{P} \frac{\partial^2 T(\ddot{q})_{iN}}{\partial q_i^2} \dddot{q}_i^2 + 2 \sum_{i=1}^{P} \left( \sum_{j=i+1}^{P} \frac{\partial^2 T(\ddot{q})_{iN}}{\partial q_i \partial q_j} \dot{q}_i \dot{q}_j \right) = \dddot{C}_{1N}.
\]

(4.49)

In the customary jargon of kinematics, the three terms on the left-hand side of (4.49) can be labeled, in order, as the tangential, normal, and Coriolis accelerations. This observation is consistent with other techniques that develop kinematic acceleration equations. The tangential accelerations are related to the second time derivative of the joint dofs; the normal accelerations are related to the square of the first time derivative of the joint dofs; and the Coriolis accelerations are related to the “cross” products of the first time derivatives of the joint dofs.

Assuming that completed displacement and velocity analyses precede an acceleration analysis (and that the partial derivatives of \( T(\ddot{q})_{iN} \) are calculable), everything is known in Equation (4.49) except for the second time derivatives of the joint dofs (i.e., the \( \dddot{q}_i \)'s in the first term of the left-hand side). Hence, the inverse kinematics problem seeks to find the \( \dddot{q}_i \)'s necessary for a specified \( \dddot{C}_{1N} \). To facilitate the solution of (4.49), it is useful to place the known quantities on the right-hand side:

\[
\sum_{i=1}^{P} \frac{\partial T(\ddot{q})_{iN}}{\partial q_i} \ddot{q}_i = \dddot{C}_{1N} - \sum_{i=1}^{P} \frac{\partial^2 T(\ddot{q})_{iN}}{\partial q_i^2} \dddot{q}_i^2 - 2 \sum_{i=1}^{P} \left( \sum_{j=i+1}^{P} \frac{\partial^2 T(\ddot{q})_{iN}}{\partial q_i \partial q_j} \dot{q}_i \dot{q}_j \right),
\]

(4.50)
or

\[
\sum_{i=1}^{F} \frac{\partial T(\ddot{\theta})_{1i}}{\partial q_i} \ddot{q}_i = \ddot{C}_{1N} - A_n - A_c
\]

(4.51)

\[
= B_{open}
\]

(4.52)

where 

\( C_{1N} \) = desired acceleration state of link \( N \)

\( A_n \) = normal acceleration related terms

\( A_c \) = Coriolis acceleration related terms

\( B_{open} \) = right-hand side for open loop mechanism

Since the matrices that make up \( B_{open} \) are known, (they are either given explicitly or calculated from the displacement and velocity analyses), the matrix Equation (4.52) represents a system of equations that are linear in the \( \ddot{q}_i \). The form of (4.52) is identical to the matrix equation derived for the velocity analysis. Hence, the acceleration analysis solution proceeds from this point in exactly the same manner as the process for the velocity analysis. The matrices involved are 4 x 4 in dimension, however the fourth row of all of them are identically zero. Hence, (4.52) represents 12 nontrivial equations. Just as with the velocity analysis development, only 6 (at most) of these equations can be independent: three may be chosen from the 3 x 3 “rotation” partition—not all from the same row or column, and the three from the 3 x 1 “displacement” partition are used as well.

### 4.5.2 Closed Loop Mechanism

If this theory is applied to a single closed loop mechanism, Equation (4.20) from the velocity theory is differentiated with respect to time which yields:

\[
\sum_{i=1}^{F} \frac{\partial T(\ddot{\theta})_{1i}}{\partial q_i} \ddot{q}_i + \sum_{i=1}^{F} \frac{\partial^2 T(\ddot{\theta})_{1i}}{\partial q_i^2} \dddot{q}_i + 2 \sum_{i=1}^{F} \left( \sum_{j=i+1}^{F} \frac{\partial^2 T(\ddot{\theta})_{1j}}{\partial q_i \partial q_j} \ddot{q}_i \dddot{q}_j \right) = 0, \quad (4.53)
\]
or

$$\sum_{i=1}^{p} \frac{\partial T(\ddot{q})_{h,i}}{\partial \dot{q}_i} \ddot{q}_i = -(A_n + A_c) \tag{4.54}$$

$$= B_{\text{closed}} \tag{4.55}$$

where \(B_{\text{closed}}\) = right-hand side for closed loop mechanism.

In a similar fashion to the velocity analysis development, (4.55) will yield the following 12 nontrivial equations:

\[
t_{g1}^{11} \ddot{q}_1 + t_{g2}^{11} \ddot{q}_2 + \ldots + t_{gP}^{11} \ddot{q}_P = b^{11} \tag{4.56}
\]

\[
t_{g1}^{21} \ddot{q}_1 + t_{g2}^{21} \ddot{q}_2 + \ldots + t_{gP}^{21} \ddot{q}_P = b^{21} \tag{4.57}
\]

\[
t_{g1}^{31} \ddot{q}_1 + t_{g2}^{31} \ddot{q}_2 + \ldots + t_{gP}^{31} \ddot{q}_P = b^{31} \tag{4.58}
\]

\[
t_{g1}^{12} \ddot{q}_1 + t_{g2}^{12} \ddot{q}_2 + \ldots + t_{gP}^{12} \ddot{q}_P = b^{12} \tag{4.59}
\]

\[
t_{g1}^{22} \ddot{q}_1 + t_{g2}^{22} \ddot{q}_2 + \ldots + t_{gP}^{22} \ddot{q}_P = b^{22} \tag{4.60}
\]

\[
t_{g1}^{32} \ddot{q}_1 + t_{g2}^{32} \ddot{q}_2 + \ldots + t_{gP}^{32} \ddot{q}_P = b^{32} \tag{4.61}
\]

\[
t_{g1}^{13} \ddot{q}_1 + t_{g2}^{13} \ddot{q}_2 + \ldots + t_{gP}^{13} \ddot{q}_P = b^{13} \tag{4.62}
\]

\[
t_{g1}^{23} \ddot{q}_1 + t_{g2}^{23} \ddot{q}_2 + \ldots + t_{gP}^{23} \ddot{q}_P = b^{23} \tag{4.63}
\]

\[
t_{g1}^{33} \ddot{q}_1 + t_{g2}^{33} \ddot{q}_2 + \ldots + t_{gP}^{33} \ddot{q}_P = b^{33} \tag{4.64}
\]

\[
t_{g1}^{14} \ddot{q}_1 + t_{g2}^{14} \ddot{q}_2 + \ldots + t_{gP}^{14} \ddot{q}_P = b^{14} \tag{4.65}
\]

\[
t_{g1}^{24} \ddot{q}_1 + t_{g2}^{24} \ddot{q}_2 + \ldots + t_{gP}^{24} \ddot{q}_P = b^{24} \tag{4.66}
\]

\[
t_{g1}^{34} \ddot{q}_1 + t_{g2}^{34} \ddot{q}_2 + \ldots + t_{gP}^{34} \ddot{q}_P = b^{34} \tag{4.67}
\]
which can be written as:

\[
\begin{bmatrix}
  \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_p} \\
  \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \cdots & \frac{\partial f_2}{\partial q_p} \\
  \frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \cdots & \frac{\partial f_3}{\partial q_p} \\
  \frac{\partial f_4}{\partial q_1} & \frac{\partial f_4}{\partial q_2} & \cdots & \frac{\partial f_4}{\partial q_p} \\
  \vdots & \vdots & \ddots & \vdots \\
  \frac{\partial f_p}{\partial q_1} & \frac{\partial f_p}{\partial q_2} & \cdots & \frac{\partial f_p}{\partial q_p}
\end{bmatrix}
\begin{bmatrix}
  \ddot{q}_1 \\
  \ddot{q}_2 \\
  \ddot{q}_3 \\
  \ddot{q}_4 \\
  \vdots \\
  \ddot{q}_p
\end{bmatrix}
= \begin{bmatrix}
  b_{11} \\
  b_{21} \\
  b_{31} \\
  b_{41} \\
  \vdots \\
  b_{p1}
\end{bmatrix}. \quad (4.68)
\]

As was done with the velocity analysis, 6 independent equations (in general) may be chosen from (4.68) and expressed in the form:

\[
J \ddot{\vec{Q}} = \vec{b}
\]

where

- \( J \) = the Jacobian matrix (i.e., the 6 rows of the partial derivative matrix in (4.68) chosen as the independent ones)
- \( \vec{b} \) = the 6 elements of the right-hand side of (4.68) that correspond to the six rows of \( J \).

The joint dof acceleration vector, \( \ddot{\vec{Q}} \), may then be partitioned according to the independent (specified) and dependent (unknown) dofs such that the following can be written:

\[
\begin{bmatrix}
  J_I & J_D
\end{bmatrix}
\begin{bmatrix}
  \ddot{Q}_I \\
  \ddot{Q}_D
\end{bmatrix}
= \begin{bmatrix}
  \vec{b}
\end{bmatrix} \quad (4.70)
\]

where

- \( \ddot{Q}_I \) = the independent (specified) joint dof accelerations
- \( \ddot{Q}_D \) = the dependent (unknown) joint dof accelerations
- \( J_I \) = columns of Jacobian associated with independent joint dof accelerations
\( J_D = \) columns of Jacobian associated with dependent joint dof accelerations.

This can be further written as:

\[
J_D \ddot{Q}_D = \ddot{b} - J_I \ddot{Q}_I. \tag{4.71}
\]

The right-hand side of (4.71) is completely specified by the inputs to the mechanism and by results from the displacement and velocity analyses. Therefore, as with the velocity analysis, if \( J_D \) is invertible, the solution for \( \dot{Q}_D \) is straightforward.
CHAPTER V

Numerical Solutions for the Kinematic Equations

5.1 Introduction

Chapter IV presented the general theory for generating systems of equations useful for the displacement, velocity, and acceleration analysis of mechanisms. These equations were based on transformation matrix methods of mechanism modeling. The displacement analysis equations are known to be highly nonlinear in terms of a mechanism's joint variables and, therefore, require nonlinear solution techniques for the inverse kinematics problem. The velocity and acceleration analyses are somewhat more complicated to set up than the displacement analysis, however, they are linear in the unknown time rates of change of the joint variables. The main sources of complexity in the velocity and acceleration analyses are related to forming the time derivatives of the transformation matrix product used to model a given mechanism.

This chapter will present useful techniques for solutions to the displacement, velocity, and acceleration systems of equations. The procedures will be developed for closed loop mechanisms, however, with minor modifications, the concepts are applicable to open loop mechanisms as well. The approach taken will be numerical in nature. As is evident from the complexity of the sphere-cylinder surface contact
mechanism, there is little hope for developing a closed form, algebraic procedure
general enough to encompass all of the mechanism possibilities (see Section 3.3,
especially Equation (3.85) where the pair matrix is shown for the sphere-cylinder
contact joint).

The numerical techniques for the displacement analysis are iterative in nature
and essentially seek values for the unknown joint degrees-of-freedom that cause
the resulting closed loop transformation matrix product to equal the identity ma-
trix. Numerical solution techniques usually require a reasonable estimate to the
solution before the solving process is started. Determining such a starting point is
often difficult for complicated mechanisms. Therefore, two solution methods are
presented for the displacement analysis. One is based on optimization techniques
and is relatively insensitive to the quality of the initial solution estimate. The sec­
ond displacement solution technique is a Newton’s Method based approach that is
more sensitive to the initial guess, however, it proceeds to a solution more quickly
than with the optimization technique. Hence, the more robust optimization tech­
nique is used to generate the solution for the first position desired, and the quicker
Newton’s Method approach is used to solve the remaining positions.1

The Newton’s Method procedure is combined with the use of a technique
known as “singular value decomposition” (SVD) which is very useful in detect­
ing and dealing with pathological characteristics of systems of equations. Simply
put, pathologies arise in situations where the number of independent equations is
greater than or less than the number of unknowns. In the kinematics parlance,

---

1The first position's solution can be used as a reasonable initial guess to the second position's
solution, the second's for the third's, the third's for the fourth's, etc. if the mechanism input
is not changing rapidly.
these situations manifest themselves as overconstrained or underconstrained mechanisms. The SVD approach is able to form solutions for these types of mechanisms without intervention on the part of the analyst.

Both the optimization and Newton's Method approaches to the displacement analysis require the calculation of derivatives of certain functions. A general numerical procedure for these operations will be presented. Derivatives are also required for the velocity and acceleration analyses in order to determine the system of equations used for their respective solutions. These will also be performed numerically.

5.2 Optimization Approach to the Displacement Analysis of Mechanisms

The proper joint variable values for a closed loop mechanism will cause its closed loop transformation product to equal the identity matrix. A matrix other than the identity matrix will be obtained if the wrong values are chosen. This "non-identity" matrix indicates mathematically that the links cannot form a closed loop for the chosen values of joint variables. A measure of how well closure is approached can be ascertained by the degree to which the matrix product approaches the identity matrix. The more closely the matrix product resembles the identity matrix, the more closely the chosen joint variables represent a correct solution. Hall, Root, and Sandgren [32] exploited this observation in the development of an optimization based displacement analysis.
5.2.1 Background to the Optimization Approach

Optimization principles are usually associated with finding the components of a vector, say \( \mathbf{Q} \), such that the function \( F(\mathbf{Q}) \), when evaluated using those components, attains its minimum value. The function, \( F(\mathbf{Q}) \), is called the objective function and the components of \( \mathbf{Q} \) are called the design variables. Note that \( F(\mathbf{Q}) \) is a scalar valued function of a multidimensional vector. The allowable values for the components of \( \mathbf{Q} \) are often subject to certain constraints. These constraints may be in the form of simple inequalities to indicate lower and upper bounds for certain components such as

\[
q_l \leq q_i \leq q_u. \quad (5.1)
\]

The constraints may also be of the forms

\[
h_i(q_1, q_2, \ldots, q_n) = 0 \quad (5.2)
\]

\[
g_i(q_1, q_2, \ldots, q_n) \leq 0 \quad (5.3)
\]

where \( h_i \) and \( g_i \) represent constraints that are functions of one or more design variables.

Optimization procedures are iterative in nature and use various schemes to converge efficiently upon a minimum solution. Some techniques require derivatives of the objective function with respect to the design variables to provide information on where to search for a minimum solution. Although schemes exist that do not require derivative information, the ability to use inexpensively calculated derivatives often leads to convergence more quickly.

Many functions contain a number of local minima. A local minimum exists where the objective function attains a minimum value in the "neighborhood" of
a given value for \( \hat{Q} \). A one-dimensional example of such a function is \( \cos(q) \). For \(-\infty < q < \infty\), \( \cos(q) \) will have an infinite number of local minima\(^2\) expressible by \( q = n \pi \) for \( n = \ldots, -5, -3, -1, 1, 3, 5, \ldots \). In this case, all of the local minima have the same value of \(-1\). However, many objective functions have a number local minima for the objective function that are not the same. The value of \( \hat{Q} \) yielding the smallest minimum is the global minimum.

Optimization procedures cannot directly recognize the difference between a local minimum and a global minimum. Hence, an optimization program will converge to a local minimum just as readily as it will to a global minimum. Awareness of this "feature" is obviously important when interpreting the results of an optimization analysis especially when a reasonable value for the global minimum is not known—there will be no way to positively know if the result obtained is absolutely the global minimum or if it is just a local minimum.

As with most iterative solution schemes, optimization procedures require an estimate for \( \hat{Q} \) from which to begin the minimization search. If an objective function has a number of local minima, the initial estimate for \( \hat{Q} \) will affect the minimum to which the procedure converges. If a reasonable value for the global minimum is not known a priori, performing a number of optimization restarts with different initial estimates for \( \hat{Q} \) is often recommended. Enough of these restarts can provide some confidence in obtaining a "fairly" global minimum (note this still does not guarantee the absolute global minimum will be found).

---

\(^2\)For a one-dimensional problem (and \( q \) unconstrained), a minimum of \( F(q) \) is defined by \( \frac{dF(q)}{dq} = 0 \) and \( \frac{d^2F(q)}{dq^2} \geq 0 \). Analogues to this may be developed for multidimensional problems using the gradient of \( F(\hat{Q}) \), \( \nabla F(\hat{Q}) \), and the Hessian matrix, \( \nabla^2 F(\hat{Q}) \).
5.2.2 A Displacement Based Objective Function for a Closed Loop Mechanism

Referring back to Equation (4.14), a closed loop transformation matrix product is a function of the joint variables (degrees-of-freedom) in a given mechanism. By collecting the joint variables into a vector $\tilde{Q}$, the closed loop matrix product is symbolically expressed as $T(\tilde{Q})_{1,1}$ for an $N$ link mechanism. If values for the joint variables are chosen such that a closed loop is not formed, Equation (4.14) will be of the form

$$T(\tilde{Q})_{1,1} = P$$ (5.4)

where $P$ is a transformation matrix equal to something other than the identity matrix. If the joint variables values are modified so as to eventually close the loop, $P$ will approach, and finally equal, the identity matrix. In element form, Equation (5.4) may be written

$$\begin{bmatrix}
    t(\tilde{Q})_{11} & t(\tilde{Q})_{12} & t(\tilde{Q})_{13} & t(\tilde{Q})_{14} \\
    t(\tilde{Q})_{21} & t(\tilde{Q})_{22} & t(\tilde{Q})_{23} & t(\tilde{Q})_{24} \\
    t(\tilde{Q})_{31} & t(\tilde{Q})_{32} & t(\tilde{Q})_{33} & t(\tilde{Q})_{34} \\
    0 & 0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
    p_{11} & p_{12} & p_{13} & p_{14} \\
    p_{21} & p_{22} & p_{23} & p_{24} \\
    p_{31} & p_{32} & p_{33} & p_{34} \\
    0 & 0 & 0 & 1
\end{bmatrix}.$$ (5.5)

Using Equation (5.4), Hall, et. al. suggested a scalar valued function that can only be zero when closure is obtained:

$$F(\tilde{Q}) = \sum_{i=1}^{4} \sum_{j=1}^{4} [p_{ij} - I_{ij}]^2$$ (5.6)

$$= \sum_{i=1}^{4} \sum_{j=1}^{4} [t(\tilde{Q})_{ij} - I_{ij}]^2$$ (5.7)

where

$$I_{ij} = \begin{cases} 
1 & \text{for } i = j \\
0 & \text{for } i \neq j
\end{cases}$$
Even if incorrect values for the joint variables are chosen, $T(\vec{Q})_{1,1}$ is still a product of valid transformation matrices whose fourth rows are always $(0, 0, 0, 1)$. Hence, $T(\vec{Q})_{1,1}$ must always have $(0, 0, 0, 1)$ as its fourth row. Therefore, the summation over $i$ in Equation (5.7) need only be considered from 1 to 3:

$$F(\vec{Q}) = \sum_{i=1}^{3} \sum_{j=1}^{4} [t(\vec{Q})^{ij} - I^{ij}]^2$$

Equation (5.9) will obviously approach zero as the components of $T(\vec{Q})_{1,1}$ approach those of the identity matrix. Furthermore, because each term is squared, Equation (5.9) cannot be less than zero. Hence, the globally minimum value to which $F(\vec{Q})$ can evaluate is zero. If $F(\vec{Q})$ is not zero, the joint variable values used for $\vec{Q}$ do not close the mechanism, and therefore, do not represent a solution.

5.2.3 The Davidon-Fletcher-Powell Minimization Method

The actual optimization algorithm used to minimize Equation 5.9 is not of great importance. Hall, et. al. used the Davidon-Fletcher-Powell (DFP) method with good success. The DFP approach is a *Quasi-Newton* method in that it is based on the properties of quadratic functions [62]. The multidimensional optimization procedure is iterative in nature and based on performing many one-dimensional line searches in strategically chosen directions to eventually converge upon an objective
function minimum. The manner in which search directions are determined is a distinguishing characteristic of optimization algorithms in general.

In searching for a minimum, the DFP procedure approximates the objective function as a quadratic in the neighborhood of the current estimate for the minimum. Using the first three terms of the Taylor's series, a quadratic approximation for a function \( f(\bar{x}) \) in the neighborhood of \( \bar{x} = \bar{x}_o \) can be written as

\[
f(\bar{x})|_{\bar{x} = \bar{x}_o} \approx f(\bar{x}_o) + \nabla f(\bar{x}_o)^T \Delta \bar{x} + \frac{1}{2} \Delta \bar{x}^T \nabla^2 f(\bar{x}_o) \Delta \bar{x}
\]

or

\[
a + b^T \Delta \bar{x} + \frac{1}{2} \Delta \bar{x}^T C \Delta \bar{x}
\]

where

\[
a = \text{the function evaluated at } \bar{x}_o \\
b = \text{the gradient of the function evaluated at } \bar{x}_o \\
c = \text{the Hessian matrix for the function evaluated at } \bar{x}_o.
\]

The minimum of the function approximation will occur at a point where its gradient vanishes. Therefore, taking the gradient of Equation (5.11) and setting it to zero yields

\[
0 = b + C \Delta \bar{x}
\]

or

\[
\Delta \bar{x} = -C^{-1} b
\]

\[
= -[\nabla^2 f(\bar{x}_o)]^{-1} \nabla f(\bar{x}_o).
\]

If the function happens to be quadratic at \( \bar{x}_o \), Equation (5.14) will increment \( \bar{x}_o \) to the exact point at which the gradient is zero. If the function is not quadratic, it seems logical that \( \Delta \bar{x} \) would provide a reasonable direction in which to search for a vanishing gradient.
Based on the preceding discussion, the DFP method uses the following search direction for the $k$-th iteration:

$$
\mathbf{s}(\mathbf{Q}^{(k)}) = -A^{(k)} \nabla F(\mathbf{Q}^{(k)})
$$

(5.15)

where $A^{(k)}$ is an approximation to the inverse Hessian of $F(\mathbf{Q})$ at $\mathbf{Q}^{(k)}$. The matrix $A$, called the metric, is updated with each iteration and, based on the DFP formulation, will converge to the Hessian of $F(\mathbf{Q})$ as $\mathbf{Q}$ approaches a minimum point. This will be discussed shortly. Given the search direction $\mathbf{s}$, the iterative procedure for improving the components of $\mathbf{Q}$ such that they continue to minimize $F(\mathbf{Q})$ is given by

$$
\mathbf{Q}^{(k+1)} = \mathbf{Q}^{(k)} + \alpha^{(k)} \mathbf{s}(\mathbf{Q}^{(k)})
$$

(5.16)

where

- $\mathbf{Q}^{(k)}$ = the $k$-th estimate for $\mathbf{Q}$
- $\mathbf{s}(\mathbf{Q}^{(k)})$ = the $k$-th direction in which to look for a minimum of $F(\mathbf{Q})$
- $\alpha^{(k)}$ = the distance (i.e., step length) along $\mathbf{s}$, from $\mathbf{Q}^{(k)}$, that the next estimate for $\mathbf{Q}$ is found.

A one-dimensional line search to improve the estimate for $\mathbf{Q}$ is then involved with finding a value for $\alpha$ to minimize $F(\mathbf{Q}^{(k)} + \alpha^{(k)} \mathbf{s}(\mathbf{Q}^{(k)}) )$.

Since, only first derivatives are used for the DFP method, no direct knowledge will be available regarding the Hessian of $F(\mathbf{Q})$. This is why an approximation (i.e., $A$) to the inverse Hessian is used. The DFP method sets $A^{(0)}$ to the identity matrix for the first iteration. However, $A$ can be improved iteratively toward the inverse Hessian with an updating formula to be presented. Note incidently, referring to Equation (5.15), by setting $A^{(0)}$ equal to the identity matrix, the initial search direction is simply one of steepest descent (i.e., along the negative gradient).
If the objective function is approximated by a quadratic, then it is of the form

\[ F(\tilde{Q}) = a + b^T \tilde{Q} + \frac{1}{2} \tilde{Q}^T C \tilde{Q}. \]  

(5.17)

The gradient of \( F(\tilde{Q}) \) is then

\[ \tilde{G}(\tilde{Q}) = \nabla F(\tilde{Q}) = b + C \tilde{Q}. \]  

(5.18)

For two different points, \( \tilde{Q}^{(0)} \) and \( \tilde{Q}^{(1)} \) on this same quadratic hyper-surface, the gradients may be written as

\[ \tilde{G}(\tilde{Q}^{(0)}) = b + C \tilde{Q}^{(0)} \]  

(5.19)

\[ \tilde{G}(\tilde{Q}^{(1)}) = b + C \tilde{Q}^{(1)}. \]  

(5.20)

Subtracting Equation (5.19) from (5.20) gives

\[ \Delta \tilde{G} = \tilde{G}(\tilde{Q}^{(1)}) - \tilde{G}(\tilde{Q}^{(0)}) \]  

(5.21)

\[ = C (\tilde{Q}^{(1)} - \tilde{Q}^{(0)}) \]  

(5.22)

\[ = C \Delta \tilde{Q}. \]  

(5.23)

This yields the important quadratic relation

\[ \Delta \tilde{Q} = C^{-1} \Delta \tilde{G} \]  

(5.24)

where, again, \( C^{-1} \) is equal to the inverse Hessian of \( F(\tilde{Q}) \).

Forcing successive \( A^{(k)} \)'s (in place of \( C^{-1} \)) to satisfy Equation (5.24) will cause \( A^{(k)} \) to converge to \( C^{-1} \) as the minimum is approached. In other words, the iterates must satisfy

\[ \Delta \tilde{Q}^{(k-1)} = A^{(k)} \Delta \tilde{G}^{(k-1)}. \]  

(5.25)
Equation (5.25) can be used to develop the previously mentioned updating relation for $A^{(k)}$ as

$$A^{(k)} = A^{(k-1)} + \frac{\Delta \bar{Q}^{(k-1)} \Delta \bar{G}^{(k-1)T}}{\Delta \bar{Q}^{(k-1)T} \Delta \bar{G}^{(k-1)T}} - \frac{A^{(k-1)} \Delta \bar{G}^{(k-1)} \Delta \bar{G}^{(k-1)T} A^{(k-1)}}{\Delta \bar{G}^{(k-1)T} A^{(k-1)} \Delta \bar{G}^{(k-1)}}$$

(5.26)

where

$$\Delta \bar{Q}^{(k-1)} = \bar{Q}^{(k)} - \bar{Q}^{(k-1)}$$

(5.27)

$$\Delta \bar{G}^{(k-1)} = \nabla F(\bar{Q}^{(k)}) - \nabla F(\bar{Q}^{(k-1)})$$

(5.28)

The updating formula for $A^{(k)}$ can be shown to preserve symmetry and positive definiteness. Using $A^{(0)}$ equal to the identity matrix will assure all successive $A^{(k)}$'s are symmetric and positive definite. This observation can be used to show that the DFP method has a descent property where $F(\bar{Q}^{(k+1)}) < F(\bar{Q}^{(k)})$ for all $\alpha^{(k)} > 0$ [62]. The general DFP method may now be summarized as shown in Table 4.

This technique requires calculation of the objective function's gradient. Hall, et. al. found the use of numerically calculated derivatives to be of sufficient accuracy in determining the gradient for the DFP method. They used the forward difference calculation to calculate a partial derivative at a given point $\bar{Q}_0$ as follows:

$$\left. \frac{\partial F(\bar{Q})}{\partial q_i} \right|_{\bar{Q}=\bar{Q}_0} \approx \frac{F(\bar{Q}_0 : q_i = q_{i0} + \epsilon) - F(\bar{Q}_0)}{\epsilon}$$

(5.29)

where $q_i$ is the $i$-th component of $\bar{Q}$, and $q_{i0}$ is the value of $q_i$ at the point $\bar{Q}_0$. This same approach to numerical derivatives has been applied in this research with very good success. At a particular point $\bar{Q}_0$, the partial derivative of the objective function with respect to say $q_i$ is determined by calculating $T(\bar{Q})_{1,1}$ twice — once at the point $\bar{Q} = \bar{Q}_0$ and then again at a new point obtained by perturbing the $i$-th
Table 4: General procedure for Davidon-Fletcher-Powell minimization algorithm.

**Basic Davidon-Fletcher-Powell Procedure:**

1. Set \( k = 0 \).
2. Set \( \mathbf{Q}^{(0)} \) to the initial estimate for the solution.
3. Calculate \( \nabla F(\mathbf{Q}^{(k)}) \).
4. If \( \nabla F(\mathbf{Q}^{(k)}) \) vanishes within acceptable tolerance, then an optimum has been found. Terminate the process with \( \mathbf{Q}^{(k)} \) as the minimum.
5. If \( k = 0 \), set \( \mathbf{A}^{(0)} \) to the identity matrix. Otherwise, calculate \( \mathbf{A}^{(k)} \) using Equation (5.26).
6. Calculate the search direction, \( \mathbf{s}(\mathbf{Q}^{(k)}) \), using Equation (5.15).
7. Using \( \mathbf{Q}^{(k+1)} = \mathbf{Q}^{(k)} + \alpha^{(k)} \mathbf{s}(\mathbf{Q}^{(k)}) \), perform a one-dimensional line search in terms of \( \alpha^k \) such that \( F(\mathbf{Q}^{(k+1)}) \) is minimized. The resulting value for \( \mathbf{Q}^{(k+1)} \) is the new estimate for the minimum.
8. Set \( k = k + 1 \). Go back to item 3 and continue process until a minimum has been found or allowable number of iterations has been exceeded.
component of $\vec{Q}_0$ by a small amount $\epsilon$. Using the two resulting matrices, Equation (5.9) is used to calculate the objective functions $F(\vec{Q}_0)$ and $F(\vec{Q}_0 : q_i = q_{i_0} + \epsilon)$. Subtracting the objective function at $\vec{Q}_0$ from the perturbed objective function and then dividing by the perturbation amount yields the partial derivative as per Equation (5.29).

The formation of the closed loop transformation matrix with $q_i$ perturbed simply takes the form:

$$T(\vec{Q}_0 : q_i = q_{i_0} + \epsilon)_{1,1} = T_{12} T_{23} \cdots T(q_{i_0} + \epsilon)_{IJ} \cdots T_{N-1,N} T_{N1}$$  \hspace{1cm} (5.30)

where all of the matrices are evaluated at $\vec{Q} = \vec{Q}_0$, and the matrix $T_{IJ}$ is (at least partly) a function of $q_i$ for which $q_{i_0}$ has been perturbed by $\epsilon$. Due to the formulation of the closed loop matrix product, any given joint degree-of-freedom will appear in only transformation matrix.

5.2.4 Discussion of Optimization Approach

The most impressive feature of the optimization approach to displacement analysis is its robustness. Although the quality of the initial estimate for the solution affects the rate of convergence to a minimum, a poor estimate seldom confuses the process. Wildly incorrect initial guesses have been used for joint variables, and yet the technique reliably finds the correct values for the variables to mathematically close the closed loop matrix product.

The normal procedure for the optimization approach is to specify values for the independent degrees-of-freedom (for this research, only one degree-of-freedom mechanisms are considered). Treating these independent degrees-of-freedom as instantaneously constant, the values for the remaining dependent degrees-of-freedom
are sought such that the matrix product converges to the identity matrix (which is necessary if the objective function vanishes).

If the process finds a local minimum not equal to zero, the matrix product cannot be closed and a valid displacement solution has not been found. This situation will occur if, based on the input value(s), the mechanism simply cannot be assembled in this position. In other words, a closed matrix product may not exist for the specified independent degree(s)-of-freedom. The key item of importance here, is knowing that an objective function minimum of zero is the only condition for which a valid solution is produced. The global minimum is known to be zero, and it must be the minimum converged upon. Thus, there is no uncertainty in assessing the validity of the results.

Another interesting capability of optimization procedures is their ability to solve underconstrained systems of equations. These are sets of equations for which there are more unknowns than independent equations. No unique solution exists for such a system of equations. However, optimization procedures will still converge, without special treatment, to one of the solutions. This can be useful in mechanisms possessing a passive (or idle) degree-of-freedom.

For instance, an RSSR mechanism will allow unlimited rotation of the link possessing both of the spherical pairing elements. This rotation is about an axis through the spherical joint centers, and it can take place without affecting the relative positions of any of the other links. In essence, the RSSR mechanism is a two degree-of-freedom mechanism (by Grubler’s equation as well as by observation) for which the idle degree-of-freedom can be functionally ignored. In use, the RSSR can then be considered a one degree-of-freedom mechanism if the “axial rotation”
of the "S-S-" link is not important. Most analyses would require this axial rotation to be suppressed (by setting it to zero for instance) in order to make the system of equations determinant (i.e., the number of equations equals the number of unknowns). However, the optimization approach is simply intent on minimizing the scalar objective function via an iteration strategy that has nothing to do with the number of equations in relation to the number of unknowns. Therefore, the optimization approach will find a solution for an underconstrained mechanism without any insight on the part of the analyst.

Once an underconstrained solution has been found, its usefulness is subject to whether idle degrees-of-freedom exist in the mechanism. If none exist, then the mechanism is truly an "n" degree-of-freedom mechanism \( n > 1 \) for which less than \( n \) inputs have been specified. Thus, even though an optimization solution would provide a closed loop, additional unconstrained degrees-of-freedom would exist in the mechanism rendering an infinite number of possible joint variable values that could close the mechanism. The optimization based solution would simply be one of these and therefore is not really a solution at all in the kinematic sense. However, if only one of the \( n \) degrees-of-freedom is not idle, and that is the one being specified, a useful solution (again, in the kinematic sense) will result.

The ability to find a solution for an underconstrained system of equations is also useful in determining whether a given mechanism is able to assume any closed configuration. If none of the joint variables are specified, and the mechanism is closable, the number of closed configurations will be infinite. The optimization procedure will find just one of them. However, the value so obtained for the intended independent variable is a value for which the mechanism is now known to
be closable. This value for the independent value along with the corresponding dependent variables provide a good starting point to investigate the motion characteristics of the mechanism. Once one closed position is found, the independent variable can be incremented and decremented from this position until kinematic limit positions are reached (i.e., when the optimization procedure can no longer find a minimum of zero for the objective function).

The DFP method as presented is an unconstrained minimization procedure. The design variables are not restricted to any lower or upper bounds. There are also no constraint equation relations involving the design variables. For many practical mechanisms, a constrained procedure is not necessary for a purely kinematic analysis.

The idealized revolute joint for instance, has no restriction on the number of complete relative rotations allowed between the links it connects. Although the revolute angle is usually expressed as a measure between 0 and $2\pi$, it is mathematically acceptable to express its value anywhere outside of this range. Likewise, the parameters $\theta$ and $\phi$ used to locate the contact point on the surface of a sphere are principally expressed as $0 \leq \theta \leq 2\pi$ and $0 \leq \phi \leq \pi$. However, the associated mathematics in the transformation matrix development will hold up even if $\theta$ and $\phi$ are outside of these ranges. Similarly, an ideal prismatic joint will impose no restriction on the relative sliding motion it permits between two links. Kinematically speaking, the sliding degree-of-freedom may extend from $-\infty < s < \infty$.

If a circumstance presents itself that requires lower and/or upper bound specification for any of a mechanism's joint variables, a constrained minimization tech-

\footnote{Note an angular measure can always be converted to a "principal" value by dividing by $2\pi$ and using the remainder.}
nique will have to be employed. Constraints might be applied in situations where physical "stops" or motion limits are known to exist in a joint. Also, note the definitions of the conical and paraboloid surfaces (see Figures 28 and 29) used for contact joints. These are restricted to \( z \geq 0 \). A constrained analysis would be necessary in these cases to reliably maintain the correct sign on \( z \). Another instance for the need of a constrained analysis would be for a contact surface described by a parametric surface patch. The boundary curves of such a patch are usually parameterized curves where the parameters range from 0 to 1. The equations describing the surface patch are valid only within the defined ranges of the parameters. As with other contact surfaces, the parameters would be used to locate the point of contact on the surface. Hence, they must be constrained to the valid ranges.

5.3 Newton's Method Approach to the Displacement Analysis of Mechanisms

The closed loop matrix product for a mechanism provides six (at most) independent nonlinear equations in terms of the mechanism's joint variables. The solution to these equations may be performed using a multi-dimensional Newton's method approach. Newton's method is iterative in nature and converges to a solution by solving a system of equations that linearly approximates the nonlinear functions at a specified point. It is the specified point that is iteratively improved until it converges to a solution for the nonlinear equations. Newton's method is extremely dependent upon the degree to which an initial estimate for the solution agrees with the actual solution.

Although the solution to the linearized set of equations is straightforward, there
are still various numerical deficiencies that may present themselves. Mathematically, these deficiencies arise from underconstrained and overconstrained systems of equations. Existence of these conditions will wreak havoc with the common linear equation solvers which all expect a system of \( n \) independent linear equations in \( n \) unknowns. A mechanism that is over or underconstrained will not meet this criterion, and therefore, must be handled as a special case. Situations also arise where a mechanism becomes instantaneously underconstrained such as in a dead-center position. This situation is doubly troublesome since it will not be known a priori if or where such a configuration occurs—this makes preemptive action difficult to perform.

A technique will be presented that overcomes many of the numerical difficulties involved with over and underconstrained mechanisms. The basic Newton method approach is still taken. However, the solution to the linearized system of equations developed by Newton's method is performed using a technique called Singular Value Decomposition.

5.3.1 Background to the Newton's Method Approach

The traditional approach to solving the nonlinear displacement equations has been through the use of Newton's Method for multi-dimensional problems. This method approximates the nonlinear equations as linear equations about a point chosen as an estimate of the solution. The system of nonlinear equations is in a sense converted to a system of linear equations. The solution to the linearized equations provides corrections that are applied to the estimated solution. If the initial estimate is accurate enough, successive corrections to the estimate will eventually
converge to a solution that satisfies the nonlinear system of equations. In fact, the convergence is fairly rapid for good solution estimates. However, if the initial estimate is not very close to an actual solution, Newton's method is likely to diverge and not produce a valid result.

The nonlinear equations used for a closed mechanism displacement analysis are obtained from Equation (4.15). The equations are formed by equating the elements of $T(\vec{Q})_{1,1}$ with the corresponding elements of the identity matrix. The trivial equations involving the elements of the fourth rows are useless which leaves twelve remaining equations. As developed previously, only three (at most) of the equations coming from the rotation partition can be independent. Any three equations can be used as long as they are not all from the same row or column. Three more (at most) independent equations come from the elements of the displacement partition.

Algebraic examples of developing the nonlinear displacement equations for a mechanism were shown for the planar slider crank in Sections 2.5.1 and 2.5.2. Since the slider crank is a planar mechanism, only three independent equations are found as opposed to the maximum of six independent equations. The resulting nonlinear displacement equations are Equations (2.76)-(2.78) and Equations (2.112)-(2.114) for the two different examples respectively. These two systems of equations were solvable in closed form for the dependent joint variables.

For the more general case where algebraic solutions to the displacement equations are not practical, Newton's method can be used to achieve a numerical solu-
Consider a system of \( m \) equations nonlinear in the components of \( \mathbf{\hat{Q}} \):

\[
\begin{align*}
    f_1(\mathbf{\hat{Q}}) &= 0 \\
    f_2(\mathbf{\hat{Q}}) &= 0 \\
    \vdots \\
    f_m(\mathbf{\hat{Q}}) &= 0
\end{align*}
\]  

(5.31)

where \( \mathbf{\hat{Q}} \) is of dimension \( n \). Neglecting the second and higher order terms of a Taylor series, the \( i \)-th function may be linearized about a point \( \mathbf{\hat{Q}}_0 \) as

\[
    f_i(\mathbf{\hat{Q}})|_{\mathbf{\hat{Q}}=\mathbf{\hat{Q}}_0} \approx f_i(\mathbf{\hat{Q}}_0) + \frac{\partial f_i(\mathbf{\hat{Q}}_0)}{\partial q_1} \Delta q_1 + \frac{\partial f_i(\mathbf{\hat{Q}}_0)}{\partial q_2} \Delta q_2 + \cdots + \frac{\partial f_i(\mathbf{\hat{Q}}_0)}{\partial q_n} \Delta q_n
\]

(5.32)

\[
    \approx f_i(\mathbf{\hat{Q}}_0) + \sum_{j=1}^{n} \frac{\partial f_i(\mathbf{\hat{Q}}_0)}{\partial q_j} \Delta q_j
\]

(5.33)

where \( \Delta q_j \) is a small displacement in the \( j \)-th component of the point \( \mathbf{\hat{Q}}_0 \). The entire system of \( m \) linearized equations can then be expressed as:

\[
\begin{align*}
    f_1(\mathbf{\hat{Q}}) &\approx f_1(\mathbf{\hat{Q}}_0) + \sum_{j=1}^{n} \frac{\partial f_1(\mathbf{\hat{Q}}_0)}{\partial q_j} \Delta q_j \\
    f_2(\mathbf{\hat{Q}}) &\approx f_2(\mathbf{\hat{Q}}_0) + \sum_{j=1}^{n} \frac{\partial f_2(\mathbf{\hat{Q}}_0)}{\partial q_j} \Delta q_j \\
    \vdots \\
    f_m(\mathbf{\hat{Q}}) &\approx f_m(\mathbf{\hat{Q}}_0) + \sum_{j=1}^{n} \frac{\partial f_m(\mathbf{\hat{Q}}_0)}{\partial q_j} \Delta q_j
\end{align*}
\]

or in matrix form

\[
\begin{bmatrix}
    f_1(\mathbf{\hat{Q}}) \\
    f_2(\mathbf{\hat{Q}}) \\
    \vdots \\
    f_m(\mathbf{\hat{Q}})
\end{bmatrix} = \begin{bmatrix}
    f_1(\mathbf{\hat{Q}}_0) \\
    f_2(\mathbf{\hat{Q}}_0) \\
    \vdots \\
    f_m(\mathbf{\hat{Q}}_0)
\end{bmatrix} + \begin{bmatrix}
    \frac{\partial f_1(\mathbf{\hat{Q}}_0)}{\partial q_1} & \frac{\partial f_1(\mathbf{\hat{Q}}_0)}{\partial q_2} & \cdots & \frac{\partial f_1(\mathbf{\hat{Q}}_0)}{\partial q_n} \\
    \frac{\partial f_2(\mathbf{\hat{Q}}_0)}{\partial q_1} & \frac{\partial f_2(\mathbf{\hat{Q}}_0)}{\partial q_2} & \cdots & \frac{\partial f_2(\mathbf{\hat{Q}}_0)}{\partial q_n} \\
    \vdots & \vdots & \ddots & \vdots \\
    \frac{\partial f_m(\mathbf{\hat{Q}}_0)}{\partial q_1} & \frac{\partial f_m(\mathbf{\hat{Q}}_0)}{\partial q_2} & \cdots & \frac{\partial f_m(\mathbf{\hat{Q}}_0)}{\partial q_n}
\end{bmatrix} \begin{bmatrix}
    \Delta q_1 \\
    \Delta q_2 \\
    \vdots \\
    \Delta q_n
\end{bmatrix}
\]

(5.34)
Symbolically, this might be written as

\[ \vec{F}(\vec{Q}) \approx \vec{F}_0 + \mathbf{J} \Delta \vec{Q} \]  

(5.35)

where

\[ \begin{align*}
\vec{F}_0 &= \text{an } m \times 1 \text{ vector containing the values of } f_i(\vec{Q}_0) \\
\mathbf{J} &= \text{an } m \times n \text{ matrix containing the partial derivatives of the } f_i(\vec{Q}); \text{ this matrix is known as the Jacobian} \\
\Delta \vec{Q} &= \text{an } n \times 1 \text{ vector of small displacements from the point } \vec{Q}_0.
\end{align*} \]

According to Equation (5.31), the left-hand side of (5.34) will equal the null vector if \( \vec{Q} \) represents a solution to the nonlinear system of equations. Therefore, if \( \vec{Q}_0 \) is close to a solution, Equation (5.35) may be written as

\[ \mathbf{J} \Delta \vec{Q} = -\vec{F}_0. \]  

(5.36)

If the number of equations equals the number of unknowns (i.e., \( m = n \)) and if \( \mathbf{J} \) is (numerically) invertible, Equation (5.36) can be used to solve for \( \Delta \vec{Q} \). If \( \vec{Q}_0 \) was not an exact estimate to the system solution, the right-hand side of (5.36) will not be null which also means \( \Delta \vec{Q} \) will not be null. The components obtained for \( \Delta \vec{Q} \) will then represent a correction to \( \vec{Q}_0 \) that brings \( \vec{Q}_0 \) closer to a solution for the nonlinear system. \( \vec{Q}_0 \) may then be iteratively improved until it adequately solves the nonlinear system of equations.

Let \( \vec{Q}_0^{(k)} \) represent the solution estimate for the \( k \)-th iteration of a Newton's method analysis. Assume that the number of independent equations in the system is equal to the number of unknowns. The basic Newton method procedure is then as outlined in Table 5.
Table 5: General procedure for multi-dimensional Newton's Method algorithm.

Basic Multi-Dimensional Newton's Method Procedure:

1. Set $k=0$.

2. Set $\tilde{\xi}^{(0)}$ by guessing a solution to the system of equations.

3. Evaluate $f_i(\tilde{\xi}^{(k)})$ for $i = 1, 2, \ldots, n$. This represents $F_0^{(k)}$. If $F_0^{(k)}$ is null within some predetermined tolerance, $\tilde{\xi}^{(k)}$ represents the solution.

4. Evaluate the Jacobian at $\tilde{\xi}^{(k)}$. This represents $J^{(k)}$ which may be obtained numerically or analytically.

5. Use a numerical procedure to solve for $\Delta \tilde{\xi}^{(k)}$ according to

$$\Delta \tilde{\xi}^{(k)} = -J^{(k)^{-1}}F_0^{(k)}.$$  

This assumes the Jacobian is invertible—it may not be!

6. If the correction $\Delta \tilde{\xi}^{(k)}$ is null within some predetermined tolerance, $\tilde{\xi}^{(k)}$ represents the solution.

7. Calculate the new solution estimate, $\tilde{\xi}^{(k+1)}$, using

$$\tilde{\xi}^{(k+1)} = \tilde{\xi}^{(k)} + \Delta \tilde{\xi}^{(k)}$$  

8. Increment $k$ and repeat steps 3–8 until convergence to a solution is attained, the maximum allowable iterations have been exceeded, or the solution begins to diverge.
5.3.2 Nonlinear Displacement Equations for a Newton's Method Solution

As discussed earlier, the closed loop matrix product provides six (at most) independent nonlinear equations useful for displacement analysis. Three are found in the rotation partition, and three are found in the displacement partition. The three equations from the rotation matrix are selected from a total of nine (i.e., there are nine elements in the $3 \times 3$ rotation matrix). The three equations used must not all be in the same row or column.

As before, let the elements of the closed loop matrix product be represented by $t(\tilde{Q})_{i,j}$. Each of these elements will be a function of the joint degrees-of-freedom represented by $\tilde{Q}$. Sandor and Erdman [66] allude to using the following six equations obtained from the elements of the closed loop matrix product in order to establish the nonlinear system of displacement equations:

\[
\begin{align*}
    t(\tilde{Q})^{1,3} + t(\tilde{Q})^{1,1} + t(\tilde{Q})^{3,3} - 2 &= 0 \quad (5.39) \\
    t(\tilde{Q})^{2,1} + t(\tilde{Q})^{1,1} + t(\tilde{Q})^{2,3} - 2 &= 0 \quad (5.40) \\
    t(\tilde{Q})^{3,3} + t(\tilde{Q})^{2,2} + t(\tilde{Q})^{3,3} - 2 &= 0 \quad (5.41) \\
    t(\tilde{Q})^{1,4} &= 0 \quad (5.42) \\
    t(\tilde{Q})^{2,4} &= 0 \quad (5.43) \\
    t(\tilde{Q})^{3,4} &= 0. \quad (5.44)
\end{align*}
\]

Although it is easily proved that the above equations are satisfied by mechanism closure, the motivation for the form of Equations (5.39)–(5.41) is not obvious.

Sandor and Erdman present a nonlinear displacement analysis based on Newton method principles. They make use of “derivative operator” matrices in order
to calculate the Jacobian matrix. The derivative operators are antisymmetric matrices developed to aid in the differentiation of transformation matrices describing revolute and prismatic joints. Using these derivative operators, Sandor and Erdman manipulate the closed loop matrix product (linearized about $\mathbf{Q}_0$) to the following form:

$$\sum_{i=1}^{n} \mathbf{D}_i \Delta \mathbf{Q} = \mathbf{T}(\mathbf{Q}_0)_{1,1} - \mathbf{I}$$

where $\mathbf{D}_i$ is the derivative operator matrix with respect to the $i$-th variable in the loop.

The right-hand side of Equation (5.45) may be written as

$$\mathbf{T}(\mathbf{Q}_0)_{1,1} - \mathbf{I} = \begin{bmatrix} E_1^{1,1} - 1 & E_1^{1,2} & E_1^{1,3} & E_1^{1,4} \\ E_2^{2,1} & E_2^{2,2} - 1 & E_2^{2,3} & E_2^{2,4} \\ E_3^{3,1} & E_3^{3,2} & E_3^{3,3} - 1 & E_3^{3,4} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where $E^{i,j}_i$ represents the error or deviation of the elements of $\mathbf{T}(\mathbf{Q}_0)_{1,1}$ from the identity matrix. The right-hand side of (5.46) clearly must converge to the null matrix upon mechanism closure.

Equation (5.45) can now be shown to represent equations of the form

$$d_1^{j,k} \Delta q_1 + d_2^{j,k} \Delta q_2 + \cdots + d_n^{j,k} \Delta q_n = E^{j,k}_j - I^{j,k}$$

where

$$I^{j,k} = \begin{cases} 1 & \text{for } j = k \\ 0 & \text{for } j \neq k. \end{cases}$$

The diagonals of the derivative operator matrices are zero because of antisymmetry ($d_i^{j,k} = 0$ for $j = k$). Hence, when $j = k$, the equations represented by (5.47) are not functions of the joint variables, $q_i$. In fact, for $j = k$, these equations reduce to
0 = E^{i,j} - 1. Although this is true at mechanism closure, the equation is obviously not a function of the \( q_i \). Hence, Sandor and Erdman use off-diagonal elements for the nonlinear displacement equations generated by the rotation partition because these are functions of the joint variables. Thus the purpose of the first terms in Equations (5.39)–(5.41) is understood—they are off-diagonal elements which do not all appear in the same row or column (an earlier requirement).

In terms of the elements \( E_{j,k} \) in (5.46), mechanism closure is achieved when

\[
E_{j,k} = \begin{cases} 
0 & \text{for } j \neq k \\
1 & \text{for } j = k.
\end{cases}
\]

However, it can be shown (see Appendix A) that merely finding joint variables such that \( E^{2,1} \), \( E^{3,2} \), and \( E^{1,3} \) are zero does not guarantee that elements \( E^{3,1} \), \( E^{1,2} \), and \( E^{2,3} \) are zero nor that elements \( E^{1,1} \), \( E^{2,2} \), and \( E^{3,3} \) are one. In fact, the following can occur even though the off-diagonal elements \( E^{i,j} \), \( E^{j,i} \), and \( E^{j,j} \) are zero:

- It is possible for the diagonal elements of (5.46) to converge to \( \pm 1 \). All of them can simultaneously be positive (the desired result), or any two of them can be negative. The remaining off-diagonal elements will be zero.

- It is possible for the remaining off-diagonal elements of (5.46) to converge to \( \pm 1 \) and the diagonal elements to zero. The off-diagonal elements will either be all positive or any two of them negative.

Therefore, to assure that the diagonal elements do indeed converge to \( +1 \)'s, Sandor and Erdman include diagonal terms in the three nonlinear displacement equations from the rotation partition. The manner in which they accomplish this
is identical to the way in which the diagonal terms are used as the last three terms on the left-hand sides of Equations (5.39)-(5.41). Assuming a good initial estimate to the solution is available, these terms should prevent a solution from converging upon one of the unwanted forms for $T(\mathbf{Q})_{1,1}$ (i.e., forms in which the rotation diagonal elements are either all zero or one of the elements is $+1$ and the other two are $-1$).

It should be noted that Uicker, Denvavit, and Hartenberg [85] addressed this same topic in their paper on the displacement analysis of spatial mechanisms. In fact, Sandor and Erdman cited Uicker's et. al. work in developing the techniques presented above. However, rather than incorporate the diagonal terms into the six nonlinear displacement equations, Uicker et. al. chose to add three additional equations to the system that set each diagonal element to $+1$. This results in a system of nine equations in (usually) six or less unknown joint variables. They used a least squares approach to solve the system of equations since theoretically no exact solution exists for more equations than unknowns. However, since all of the equations are consistent and this is an iterative process, the least squares solution is found to "exactly"\textsuperscript{4} satisfy all nine equations.

Certain conditions required Sandor and Erdman to choose three off-diagonal elements for the displacement equations from the rotation partition. Their displacement formulation manipulated the closed loop matrix product into an equation involving antisymmetric matrices. These provide no functional information from their diagonal elements. The displacement formulation presented in this dissertation does not have this characteristic. The Jacobian needed for the displacement

\textsuperscript{4}"Exactly" is used in the numerical sense assuming the mechanism is being analyzed in a configuration that can satisfy closure.
analysis is calculated using finite difference techniques since joints more general than revolute and prismatic are considered. It is does not appear possible to develop derivative operator matrices for complex joints.

The form of the equations used by Sandor and Erdman, however, has worked very well in a computer program written to verify the principles developed in this work. Since the diagonal elements of the Jacobian are functions of the joint variables, a case might be made for using just the diagonal elements for the three equations from the rotation partition. These equations by themselves could explicitly require the diagonal elements to equal +1 as well as satisfy the requirement that all three equations cannot be from elements all in the same row or column. Although no formal proof can be made, displacement equations of the form suggested by Sandor and Erdman (Equations (5.39)–(5.44)) seem to generate Jacobian matrices less susceptible to numerical pathologies compared to other choices used for the displacement equations.

5.4 A Robust Solution Procedure Using Singular Value Decomposition

Once an acceptable system of nonlinear displacement equations is obtained, they are solved using Newton’s method. As just presented, six nonlinear displacement equations are available through the closed loop transformation matrix formulation. If a particular mechanism had $n$ dependent joint variables, the six equations would be functions of those $n$ variables. Linearizing the system of equations about some point $\mathbf{q}_0$ would yield a linear set of equations of the form shown in Equations (5.34) and (5.35). Solving these linearized set of equations produces corrections
to $\mathbf{Q}_0$ that (hopefully) cause $\mathbf{Q}_0$ to approach a solution to the nonlinear system of equations.

5.4.1 Background

The linearized system of equations can be solved using a variety of techniques such as Gaussian Elimination or LU Decomposition. Although the Jacobian is rarely inverted in actuality, it must possess the ability to be inverted for typical solution methods to work. This implies that the Jacobian must be square and it must be nonsingular. Unfortunately, this cannot be guaranteed.

For a general, well behaved mechanism with $n$ unknown joint variables, six (at most) independent equations in $n$ unknowns will be generated from the closed loop transformation matrix. In order for this system to be solvable, it is reasonable that the number of unknowns must also equal six. If less than six unknowns exist in the mechanism, then hopefully the six displacement equations actually represent less than six independent ones.

This is exactly the case when dealing with planar mechanisms like a planar slider crank. Given an input degree-of-freedom, only three unknown degrees-of-freedom remain. Examining the six displacement equations reveals that only three of them are independent. Hence, the mechanism potentially has a solution since the number of unknowns equals the number of independent equations. Mechanisms that have fewer unknowns than the number of equations available to solve for them are called overconstrained mechanisms. All planar and spherical mechanisms are overconstrained in the spatial sense. This is why Grübler's equation often predicts negative mobilities for planar mechanisms.
Another class of useful mechanisms is referred to as underconstrained. These mechanisms have more unknowns than the number of independent displacement equations. In these cases, there cannot be a unique solution to the displacement analysis. An RSSR mechanism is often used as an underconstrained mechanism. As a mechanism, it possesses two degrees-of-freedom. One independent degree-of-freedom is specified as one of the revolute angles. The second independent joint variable is the idle degree-of-freedom of the "-SS-" link about an axis through the two spherical joints. Although the input revolute angle controls the gross configuration of the mechanism, the idle degree-of-freedom can take on any angular value without constraint in the kinematic sense. Hence, there is no unique solution.

An otherwise numerically well behaved mechanism can also become instantaneously underconstrained as it moves through a singular position. In fact it is called a singular position because the Jacobian matrix temporarily becomes singular in this position. A dead-center position of an planar RRRR mechanism exhibits this characteristic. A (planar) analysis of such a mechanism will proceed nicely until it reaches the dead-center position at which point the Jacobian becomes singular and a unique solution cannot be found.

It is clear that underconstrained and overconstrained mechanisms are useful and therefore need to be dealt with. Unfortunately, it is difficult (and arguably undesirable) to design a computer program that must detect and then properly deal with special case situations. Fortunately, there exists a powerful linear equation solver that not only deals with numerically problematic systems of equations, but it also is able to produce (usually meaningful) solutions in these cases. The technique is known as Singular Value Decomposition (SVD). Its most valuable asset in terms
of kinematic analysis is that it allows all mechanisms to be treated as generally spatial in nature without the usual numerical consequences.

SVD is used in place of Gaussian Elimination or LU Decomposition to solve the linearized system of displacement equations. Recall this solution provides a correction to the current estimate for the nonlinear equation system solution. Hence, SVD becomes part of the iteration loop in converging upon a solution to the nonlinear displacement equations. If the Jacobian is not square or becomes singular (or near singular), SVD will automatically diagnose the problem and usually provide a meaningful answer. This procedure is extremely robust. The literature, as well as experience with its use in this course of research, has not seen it fail!

5.4.2 Matrix Based Linear Systems of Equations

Consider the following matrix equation:

\[ A \vec{x} = \vec{b}. \]  \hspace{1cm} (5.48)

Matrix \( A \) is considered a linear mapping from the vector space \( \vec{x} \) to the vector space \( \vec{b} \). Given the matrix \( A \) and two vectors, \( \vec{x}_1 \) and \( \vec{x}_2 \), the following may be written:

\[ A \vec{x}_1 = \vec{b}_1 \] \hspace{1cm} (5.49)
\[ A \vec{x}_2 = \vec{b}_2. \] \hspace{1cm} (5.50)

Also, due to the linearity of \( A \),

\[ A (\vec{x}_1 + \vec{x}_2) = (\vec{b}_1 + \vec{b}_2). \] \hspace{1cm} (5.51)

Depending on the characteristics of \( A \), there may be one or more vectors, \( \vec{x}_i^* \).
for which
\[ A \vec{x}_i^* = \vec{0}. \] (5.52)

The vectors \( \vec{x}_i^* \) are considered to be in the \textit{nullspace} of \( A \). The collection of all \textit{linearly independent} vectors that, under \( A \), map to \( \vec{0} \) is called the nullspace. The number of these linearly independent vectors is the \textit{nullity} of \( A \). Using Equations (5.51) and (5.52), it can be seen that
\[ A (\vec{x}_1 + \vec{x}_i^*) = (\vec{b}_1 + \vec{0}) = \vec{b}_1. \] (5.53)

More generally, given a vector \( \vec{x}_1 \) mapped to \( \vec{b}_1 \) and given a nullspace of dimension \( k_n \),
\[ A (\vec{x}_1 + c_1 \vec{x}_1^* + c_2 \vec{x}_2^* + \cdots + c_{k_n} \vec{x}_{k_n}^*) = \vec{b}_1 \] (5.55)
where the \( c_i \) are arbitrary, constant coefficients. This says given \( \vec{x}_1 \), mapped by \( A \) into \( \vec{b}_1 \), the vector \( \vec{x}_1 \) may be combined with any linear combination of the nullspace and the result will still be mapped to \( \vec{b}_1 \).

Given \( A \) and the vector space \( \vec{x} \), there is a certain subspace of \( \vec{b} \) "reachable" by \( A \). This subspace of \( \vec{b} \) is called the \textit{range} of \( A \). Its dimension is found to equal the number of linearly independent columns in \( A \). This number of linearly independent columns is known as the \textit{rank} of \( A \). Letting the dimension of \( \vec{x} \) equal \( N \), the rank of \( A \) equal \( k_r \), and the nullity of \( A \) equal \( k_n \), the following may be stated:
\[ N = k_r + k_n. \] (5.56)

Hence, given the dimension of \( \vec{x} \) and the rank of \( A \), the size of the nullspace can be determined. If the nullspace is of dimension zero (i.e, the matrix \( A \) is of full
rank), there will be a unique \( \bar{x} \) that maps to a given \( \bar{b} \). However, if the dimension of the nullspace is not zero, there will be an infinite number of \( \bar{x} \)'s that map to a given \( \bar{b} \). This is due to Equation (5.55) whose ramifications are described in its following sentence.

Perhaps the most common form of Equation (5.48) when \( A \), \( \bar{x} \), and \( \bar{b} \) are of dimension \( N \times N \), \( N \times 1 \), and \( N \times 1 \) respectively. A typical problem is one in which matrix \( A \) and \( \bar{b} \) are given and the goal is to find the corresponding \( \bar{x} \). In order for a unique solution to be possible, \( \bar{b} \) must be in the range of \( A \), and \( A \) must consist of \( N \) linearly independent columns. A square matrix with all linearly independent columns is often called nonsingular. More generally, a nonsingular matrix is simply a matrix of full rank. Conversely, a singular matrix is a rank deficient square matrix which possesses a nullspace with dimension greater than zero.

The solution of Equation (5.48) for \( \bar{x} \) rests on being able to invert \( A \). If \( A \) is singular (i.e., rank deficient), \( A^{-1} \) does not exist, and no unique solution can be found for \( \bar{x} \). A singular square matrix has two or more columns (or rows for that matter) linearly dependent upon each other. This means there are fewer linearly independent equations than there are unknowns. As a result, an infinite number of solutions exist for \( \bar{x} \) that transform by \( A \) into \( \bar{b} \).

When solving a linear system of equations whose coefficient matrix is singular, usually the only information obtained is the fact that no unique solution exists; somewhere during the solution process, a "division by zero" will need to be performed and the procedure can go no further. This occurs whether Gaussian elimination, LU decomposition, or actual matrix inversion (rarely done) is used. Based
on the discussion above, it would be useful to have more information about a rank
deficient system than simply "an infinite number of solutions exist." Realizing that
a singular matrix corresponds to some sort of nullspace, the basis vectors making
up that nullspace might provide some additional useful information. Singular value
decomposition can provide this type of information.

The singular value decomposition technique will readily produce orthonormal
bases for both the range and nullspace of a general matrix. In the case of a rank
deficient situation, SVD will determine the solution of minimum length from the
infinite solutions possible. Hence, rather than just saying a matrix is singular, SVD
produces the solution of minimum length as well as the range and nullspace basis
vectors. Combining the minimum length solution with any linear combination of
the nullspace basis vectors will also be a solution. SVD will of course also deal
with matrices of full rank whose nullspaces are empty.

5.4.3 Singular Value Decomposition

The singular value decomposition of any \( M \times N \) matrix \( A \) is defined as follows [24, 60]:

\[
A = U W V^T
\]  

(5.57)

where

\[
W = \text{an } N \times N \text{ diagonal matrix whose diagonal elements are positive or zero}
\]

\[
U = \text{an } M \times N \text{ column-orthogonal matrix}
\]

\[
V^T = \text{the transpose of an } N \times N \text{ orthogonal matrix.}
\]

The diagonal elements, \( w_i \), of \( W \) are called the singular values of \( A \). The columns
of \( U \) and \( V \) are called the left and right singular vectors respectively.
If $A$ is in fact square, its inverse may be easily expressed in terms of its SVD as

$$A^{-1} = V W^{-1} U^T$$  \hspace{1cm} (5.58)

$$= V \left[ \text{diag} \left( \frac{1}{w_j} \right) \right] U^T$$  \hspace{1cm} (5.59)

where $w_j$ represents the diagonal element in the $j$-th column of $W$. It is clear if one or more $w_j$'s equal zero, $W^{-1}$ cannot exist, and therefore, neither can $A^{-1}$ exist. Hence, $A$ is singular. The number of zero (or near zero) diagonal elements found in $W$ provides information as to "how singular" $A$ really is. A very informative condition number for $A$ can be formed by the ratio of the maximum $w_j$ to the minimum $w_j$. The existence of one or more zero $w_j$'s gives an infinite condition number and indicates a singular matrix. A very large condition number indicates an ill-conditioned matrix (i.e., some of the columns of $A$ are very nearly linearly dependent). A condition number close to 1 indicates the columns of $A$ are very independent (e.g. an orthogonal matrix).

The number of zero diagonal elements in $W$ corresponds to the dimension of $A$'s nullspace. Consequently, the number of nonzero diagonal elements is equal to the rank of $A$ which is precisely the dimension of the range. If no diagonal elements are zero, the matrix is of full rank, and therefore, all columns in $A$ are independent.

The incidence of a given $w_j$ being equal or not equal to zero has special significance. Orthonormal basis vectors for the range of $A$ are given by the columns of $U$ corresponding to the columns of $W$ where $w_j$ is not equal to zero. Similarly, orthonormal basis vectors for the nullspace are supplied by the columns of $V$ that correspond to columns of $W$ where $w_j$ is equal to zero.
Again, assuming $A$ is square, the solution for $\bar{x}$ may now be written as

$$\bar{x} = A^{-1} \bar{b}$$

(5.60)

$$= V \left[ \text{diag} \left( \frac{1}{w_j} \right) \right] U^T \bar{b}. \quad (5.61)$$

If $A$ is singular, there will be some $w_j = 0$ for which $1/w_j = \infty$. Although not immediately evident, these infinite diagonal terms in Equation (5.61) terms may be set to zero in order to produce a valid solution for $\bar{x}$. The process of zeroing these terms is akin to discarding the linearly dependent portions of the system [60].

If $\bar{b}$ is in the range of $A$, an infinite number of solutions exist of which $\bar{x}$ is the smallest one [60]. The complete family of solutions is then obtained by combining this minimum solution with a linear combination of all the nullspace basis vectors found in $V$ (i.e., those columns of $V$ for which $w_j$ is zero in $W$).

If $\bar{b}$ is not in the range of $A$, there is no solution. The infinite diagonal terms of $1/w_j$ are still set to zero. However, the resulting $\bar{x}$ now represents an approximate solution to the system in a least squares sense [60]. In other words, $\bar{x}$ minimizes the residual, $r$, given by

$$r = |A \bar{x} - \bar{b}| \quad (5.62)$$

Consider now the situation where $A$ is rectangular. Two cases are possible. Matrix $A$ may either have more columns than rows, or it may have more rows than columns. From the standpoint of systems of linear equations, these two cases correspond to having fewer equations than unknowns or more equations than unknowns, respectively. The case of fewer equations than unknowns represents an underconstrained system of equations—a unique solution is not expected. Having more equations than unknowns is considered an overconstrained system of equa-
tions for which an exact solution is not expected in general.

A system of fewer equations than unknowns is handled very easily. The rectangular matrix $A$ is simply filled with enough rows of zeros along its bottom so as to make it square. The vector $\vec{b}$ must then be augmented with zeros equal in number to the rows added to $A$. This clearly makes the new $A$ singular. As discussed above, a singular matrix results in an infinite number of solutions. However, this is precisely what should happen with an underconstrained system. Hence, when the SVD is performed on the new $A$, there will be singular values ($w_j = 0$) equal in number to at least the number of zero rows added to $A$. There may be more singular values if the original $A$ had rank deficiencies itself. Equation (5.61) may then be used with the appropriate $1/w_j$'s set to zero to achieve the minimum length solution to the underconstrained system. The corresponding columns of $V$, as before, will represent the nullspace basis. Thus, the entire family of solutions is described.

An overconstrained system of equations has more equations than unknowns. In general there will be no exact solution. However, it may be desirable to find the best approximate solution. Once again, SVD steps in and is able to perform this maneuver. Using Equation (5.61) here as well, the vector $\vec{z}$, obtained when the appropriate $1/w_j$'s are zeroed, will yield a solution that minimizes the residual described in Equation 5.62 in the least squares sense.

The actual process of SVD is a two stage procedure. The first step reduces $A$ to a bidiagonal form using Householder reduction. The second step uses a QR procedure to iteratively reduce the remaining off-diagonal elements to negligible quantities. Algorithms for the SVD are presented in [60] and [24].
5.4.4 Use of Singular Value Decomposition with Newton's Method

When performing the kinematic analysis of mechanisms there are a number of special case mathematical situations that may be encountered. A planar mechanism analyzed as a spatial mechanism yields an overconstrained problem. For example, a planar slider crank mechanism consists of three unknown joint variables given one joint variable as an input. The six equations obtained from the closed loop matrix product should overconstrain the mechanism which means no exact solution should exist to close the matrix loop. However, due to special geometry, the six equations are consistent in the three unknowns which causes the "approximate" overconstrained solution to be exact.

Another special case is one in which idle degrees-of-freedom exist in a mechanism. These represent underconstrained mechanisms in that the idle degree-of-freedom is usually left to "do what it wants" during the mechanisms' motion. Although its time response might be predictable from a dynamic viewpoint, it has no effect from a kinematic stance. Since typical solution methods cannot solve a system of equations (linear or nonlinear) having more unknowns than equations, idle degrees-of-freedom must be suppressed (i.e., constrained to an arbitrary, fixed value) to reduce the number of unknowns. This suppression must be done by the analyst prior to analysis. Note it must be the idle degrees-of-freedom suppressed and not any of the others—this requires some intuition on the part of the analyst to recognize the idle degree-of-freedom.

A third special case (and perhaps the most frustrating) is when a mechanism passes through a singular position. From a mathematical standpoint, this happens
when two or more equations from a system of independent equations become dependent. The result is a system of fewer independent equations than unknowns. As with the explicitly underconstrained problems (i.e., those with idle degrees-of-freedom), conventional means of solution will fail since the inverse to the coefficient matrix becomes undefined. The annoying thing about singular positions is that until such a position is reached, a mechanism's displacement equations are solved without incident, and then as the singular position is approached, the whole system begins to break down.

The three special cases just discussed present difficulties when trying to develop a general kinematics analysis approach. The obvious generalization is to consider all mechanisms as spatial. However, the large class of planar (and spherical) mechanisms are overconstrained from this standpoint. As such, there are numerical difficulties involved if planar mechanisms are analyzed as being generally spatial. The existence of idle degrees-of-freedom also presents problems in that they usually must be artificially constrained in an analysis to reduce the number of unknowns in an otherwise underconstrained system. This requires insight on the part of the analyst not only to recognize idle degrees-of-freedom but also to correctly suppress them. The possibility of passing through a singular position is even more devastating in that their particular configurations can be difficult to detect a priori in an analysis. Even if they were generally and reliably detectable, doing something about them in a general sense to make a solution possible does not appear straightforward.

The use of SVD can eliminate the consequences of the aforementioned special case situations. As presented in the previous section, SVD will produce quanti-
tative results for both over and underconstrained problems. Naturally it can also solve completely specified systems of equations where the number of independent equations equals the number of unknowns.

Consider the displacement analysis of a mechanism using Newton's method. Recall at most six independent displacement equations are available from the closed loop transformation matrix product. Although these are nonlinear equations, the number of unknowns must still equal the number independent equations in order to apply the typical nonlinear solution techniques. If Newton's method is used, the nonlinear equations are linearized about some estimate for the solution. The number of independent linearized equations is directly related to the number of independent nonlinear equations. Hence, if fewer independent nonlinear equations exist than unknowns, the linearized system will also have fewer independent equations than unknowns. A solution will not be possible as the Jacobian (which may not even be square in this case) cannot be inverted. A similar problem exists with overconstrained nonlinear systems of equations. The linearized system will still be overconstrained thereby making inversion of the Jacobian impossible.

The simple solution to this problem is to use SVD when solving the linearized system of displacement equations to yield corrections for the current solution estimate. If the nonlinear system is underconstrained, an infinite number of solutions exist. The solution to the linearized system of equations will also consist of infinite possibilities. Nevertheless, SVD will produce a quantitative solution of the linearized system that is smallest in size. This solution can be used for the Newton's method correction to the current estimate of the nonlinear system solution.

Additionally, SVD will determine the nullspace dimension and its basis vectors.
This may provide diagnostic information as well as have physical significance. From a diagnostic standpoint, the dimension, \( k_n \), of the nullspace (i.e., the number of zero \( w_i \)'s obtained from the SVD) will indicate the number of singularities in the solution. Hence, by supplying \( k_n \) additional independent equations, or by specifying \( k_n \) of the unknowns, the system could be solvable. Traditional methods of solution simply indicate a singularity problem and "pack up their bags" without a word as to what might have caused the problem.

Physically speaking, the form of the nullspace vectors may provide information regarding the existence of idle degrees-of-freedom. If the only nonzero components of a given nullspace basis vector are associated with joint variables attached to the same link, it is likely an idle degree-of-freedom exists for that link. An idle degree-of-freedom is one in which a link is able to move relative to the other links in a mechanism without affecting the relative position of those other links. Kinematically, this passive motion has no constraint. The mathematical concept of a nullspace vector compares well with the physical manifestation of an idle degree-of-freedom. Given a solution to a singular system of equations, any multiple of a corresponding nullspace vector may be added to it without changing the "right-hand side" of the equation. This is exactly what an idle degree-of-freedom does—it can be assigned any value (multiple) without affecting the closure of a mechanism.

Consider the RSSR mechanism shown schematically in Figure 34. It is a four link, four joint mechanism with a total of 8 joint degrees-of-freedom—the revolute have one degree-of-freedom apiece, and the spherical joints have 3 apiece. Grüber's equation predicts the mechanism will have 2 degrees-of-freedom. Experience indicates one of those degrees-of-freedom will be in the form of an idle freedom.
Figure 34: Schematic of an RSSR mechanism with an idle degree-of-freedom about an axis through the spherical joints. Assuming joint variable $q_1$ is the mechanism input, a kinematic analysis will seek values for $\bar{Q} = (q_2, q_3, q_4, q_5, q_6, q_7, q_8)$. Realizing six (at most) independent displacement equations are available, an underconstrained system is inevitable in that seven unknowns exist. If the joint models for the spherical joints place the rotations associated with $q_4$ and $q_5$ along the axis of link 3, using SVD with Newton's method will produce a solution of the form:

$$\bar{Q} = \begin{pmatrix} q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{pmatrix} + c \begin{pmatrix} 0 \\ 0 \\ q_4^2 \\ q_5^2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(5.63)
\[ = Q_{\text{small}} + c Q^* \quad (5.64) \]

Equation (5.64) represents an infinite family of solutions that will satisfy closure for the RSSR mechanism. \( Q_{\text{small}} \) represents the solution of smallest size. \( Q^* \) represents the nullspace basis vector for this underconstrained system. Note, in this case, the nullspace is of dimension one. Any multiple of the nullspace vector may be added to the first term to yield another valid solution. *It must be emphasized that this is true only because no more than one link is affected by the nullspace vector.*

A nullspace vector with nonzero components corresponding to joint variables from two or more *different* links cannot be an idle degree-of-freedom. This is obvious since adding a multiple of such a vector to a valid solution will affect the position of more than one link. The nullspace vector(s) being discussed result from the SVD solution of the *linearized* form of the displacement equations. They cannot be added to a valid solution for the nonlinear equations because the unknowns appear *nonlinearly* in these equations. The idle degrees-of-freedom are an exception to this because these tend to occur when joint rotational or translational axes line up. The nullspace vectors in these situations consist of a "give and take" and take nature. If two rotational degrees-of-freedom line up to form an idle one, the associated nullspace vector will consist of equal but opposite rotations for these degrees-of-freedom. It is because of this lining up that the nullspace vectors from the linearized equations can be applied to the nonlinear solution.

Nullspace information can still be of some use even when it does not represent an idle degree-of-freedom. If \( N \) unknowns are being sought, and a nullspace of dimension \( k_n \) is found, then \( k_n \) of the unknowns must be specified in order to have a solution. The number of independent equations in the system for such a case is
$N - k_n$. This type of information could be useful.

With all of the SVD discussion in mind, the Newton's method displacement analysis described in Table 5 is now enhanced. Specifically, step 5 is now further described in terms of an SVD solution to the linearized system of equations. For completeness, the entire procedure is reiterated in Table 6 with the SVD related information added to step 5.

### 5.4.5 Numerical Computation of the Jacobian

The linearization of the displacement equations requires computation of the Jacobian matrix at a given point $Q_0$. This was presented in Equations (5.34) and (5.35). Since six displacement functions are available, the Jacobian will be $6 \times N$ in size where $N$ is the number of unknown joint variables. Hence,

$$
J = \begin{bmatrix}
\frac{\partial f_1(Q_0)}{\partial q_1} & \frac{\partial f_2(Q_0)}{\partial q_1} & \ldots & \frac{\partial f_1(Q_0)}{\partial q_N} \\
\frac{\partial f_2(Q_0)}{\partial q_1} & \frac{\partial f_2(Q_0)}{\partial q_2} & \ldots & \frac{\partial f_2(Q_0)}{\partial q_N} \\
\frac{\partial f_3(Q_0)}{\partial q_1} & \frac{\partial f_3(Q_0)}{\partial q_2} & \ldots & \frac{\partial f_3(Q_0)}{\partial q_N} \\
\frac{\partial f_4(Q_0)}{\partial q_1} & \frac{\partial f_4(Q_0)}{\partial q_2} & \ldots & \frac{\partial f_4(Q_0)}{\partial q_N} \\
\frac{\partial f_5(Q_0)}{\partial q_1} & \frac{\partial f_5(Q_0)}{\partial q_2} & \ldots & \frac{\partial f_5(Q_0)}{\partial q_N} \\
\frac{\partial f_6(Q_0)}{\partial q_1} & \frac{\partial f_6(Q_0)}{\partial q_2} & \ldots & \frac{\partial f_6(Q_0)}{\partial q_N}
\end{bmatrix}
$$

(5.68)

where $q_1, q_2, \ldots, q_N$ are the unknown joint variables. The functions, $f_1, f_2, \ldots, f_6$ are the six displacement functions shown in Equations (5.39)-(5.44) and restated here as

$$
f_1 = t(Q)^{1,3} + t(Q)^{1,1} + t(Q)^{3,3} - 2 \quad (5.69)$$
$$
f_2 = t(Q)^{2,1} + t(Q)^{1,1} + t(Q)^{2,3} - 2 \quad (5.70)
$$
Basic Multi-Dimensional Newton's Method Procedure Using Singular Value Decomposition:

1. Set $k=0$.

2. Set $\vec{Q}^{(0)}$ by guessing a solution to the system of equations.

3. Evaluate $f_i(\vec{Q}^{(k)})$ for $i = 1, 2, \ldots, n$. This represents $\vec{F}_0^{(k)}$. If $\vec{F}_0^{(k)}$ is null within some predetermined tolerance, $\vec{Q}^{(k)}$ represents the solution.

4. Evaluate the Jacobian at $\vec{Q}_0^{(k)}$. This represents $J^{(k)}$ which may be obtained numerically or analytically.

5. (a) If there are more than six unknowns, $J^{(k)}$ will be $6 \times N$ where $N$ is the number of unknowns. Make $J^{(k)}$ square by adding $N - 6$ rows of zero to it. The same number of zeros must also be augmented to $\vec{F}_0^{(k)}$. If $N \leq 6$, use $J^{(k)}$ as is. Determine the SVD of $J^{(k)}$:

$$J^{(k)} = U \ W \ V^T.$$  (5.65)

(b) If any $w_j$'s (the diagonal elements of $W$) are equal to zero, the corresponding columns, $j$, of $V$ represent the nullspace basis vectors. Save these at each iteration. After setting the $1/w_j$'s to zero for all $w_j = 0$, determine $\Delta \vec{Q}^{(k)}$ according to:

$$\Delta \vec{Q}^{(k)} = -V \ [\text{diag} \ (1/w_j)] \ U^T \vec{F}_0^{(k)}.$$  (5.66)

6. If the correction $\Delta \vec{Q}^{(k)}$ is null within some predetermined tolerance, $\vec{Q}_0^{(k)}$ represents the solution.

7. Calculate the new solution estimate, $\vec{Q}_0^{(k+1)}$, using

$$\vec{Q}_0^{(k+1)} = \vec{Q}_0^{(k)} + \Delta \vec{Q}^{(k)}.$$  (5.67)

8. Increment $k$ and repeat steps 3–8 until convergence to a solution is attained, the maximum allowable iterations have been exceeded, or the solution begins to diverge. If convergence has been achieved, report the solution vector as well the corresponding null space if one exists.
\[ f_3 = t(\ddot{Q})^{3,2} + t(\ddot{Q})^{2,2} + t(\ddot{Q})^{3,3} - 2 \]  
(5.71)

\[ f_4 = t(\ddot{Q})^{1,4} \]  
(5.72)

\[ f_5 = t(\ddot{Q})^{2,4} \]  
(5.73)

\[ f_6 = t(\ddot{Q})^{3,4}. \]  
(5.74)

Obviously, a partial derivative of any of these functions is obtained by taking the same partial derivative of the closed loop matrix product and then using the appropriate elements. For instance,

\[ \frac{\partial f_1}{\partial q_i} = \frac{\partial t(\ddot{Q})}{\partial q_i}^{1,3} + \frac{\partial t(\ddot{Q})}{\partial q_i}^{1,1} + \frac{\partial t(\ddot{Q})}{\partial q_i}^{3,3}. \]  
(5.75)

The partial derivatives with respect to \( q_i \) on the right-hand side are obtained from the partial derivative with respect to \( q_i \) of the closed loop matrix product:

\[ \frac{\partial T(\ddot{Q})_{1,1}}{\partial q_i} = \begin{bmatrix} \frac{\partial t(\ddot{Q})}{\partial q_i}^{1,1} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{1,2} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{1,3} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{1,4} \\ \frac{\partial t(\ddot{Q})}{\partial q_i}^{2,1} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{2,2} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{2,3} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{2,4} \\ \frac{\partial t(\ddot{Q})}{\partial q_i}^{3,1} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{3,2} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{3,3} & \frac{\partial t(\ddot{Q})}{\partial q_i}^{3,4} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \]  
(5.76)

In fact by taking the partial derivative with respect to \( q_i \) of the closed loop matrix product, all of the terms needed for the \( i \)-th column of the Jacobian are available. Therefore, the entire Jacobian can be constructed by forming the partial derivatives of the closed loop matrix product with respect to each of the \( N \) unknown joint variables.

The partial derivative of the closed loop matrix product with respect to \( q_i \) at a given point \( \ddot{Q}_0 \) can be formed using

\[ \frac{\partial T(\ddot{Q})_{1,1}}{\partial q_i} \bigg|_{q_i = \ddot{q}_0} = \frac{T(\ddot{Q}_0 : q_i = q_0 + \epsilon)_{1,1} - T(\ddot{Q}_0)_{1,1}}{\epsilon} \]  
(5.77)
where

\[ T(\tilde{\mathbf{Q}}_0 : q_i = q_{0i} + \epsilon)_{1,j} = T_{12} T_{23} \cdots T(q_{0i} + \epsilon)_{I,J} \cdots T_{N-1,N} T_{N1}. \]

Note all of the T's on the right-hand side are evaluated at \( \tilde{\mathbf{Q}}_0 \) except for the one that is (at least in part) a function of \( q_i \). In this matrix, \( q_{0i} \) (the \( i \)-th component of \( \tilde{\mathbf{Q}}_0 \)) is perturbed by the small amount \( \epsilon \). As stated previously, a given joint variable can occur in only one of the transformation matrices.

5.5 Numerical Formulation and Solution for the Velocity and Acceleration Analyses

As developed in Chapter IV, the velocity and acceleration analyses are linear in the unknown joint velocities and accelerations respectively. The system of equations for the velocity analysis is developed by differentiating the closed loop matrix product with respect to time. Similarly, differentiating the matrix product a second time yields a system of equations useful for the acceleration analysis. Since both systems of equations are linear in their unknowns, their solutions are relatively straightforward. It is common to solve them using Gaussian elimination or LU decomposition.

The inherent linearity of the velocity and acceleration analyses, however, does not preclude the problems associated with underconstrained and overconstrained mechanisms. If a mechanism is mathematically underconstrained (or overconstrained) at the displacement analysis stage, it will also be underconstrained (or overconstrained) at the velocity and acceleration stages. Hence, utilizing the SVD solution approach will be of value here as well. However, since the velocity and acceleration analyses are linear, SVD is applied directly to the systems of equations—
an iterative Newton's method approach is not necessary.

Although the mathematics of solution are straightforward for velocity and acceleration analyses given the SVD approach, the actual process of forming the systems of equations can appear formidable. This is especially true of the acceleration analysis. Presented below are the numerical techniques used within this dissertation for developing the first and second derivatives of the closed loop matrix product necessary for the velocity and acceleration analyses.

5.5.1 Formulation of the Velocity Analysis System of Equations

As discussed in Section 4.4.2, equations for the velocity analysis can be obtained by equating elements of the first time derivative of the closed loop matrix product with elements of the null matrix. Since six (at most) independent equations are available in the displacement analysis, there will also be six (at most) independent equations available to the velocity analysis. Of the equations used, three must come from the rotation partition (not all from the same row or column), and three must come from the displacement partition. With this in mind, the elements used to construct the system of equations will be \((1,3), (2,1),\) and \((3,2)\) from the rotation partition and \((1,4), (2,4),\) and \((3,4)\) from the displacement partition. The six velocity equations will then be

\[
\begin{align*}
\dot{t}_{13} &= 0, \quad \dot{t}_{21} = 0, \quad \dot{t}_{32} = 0, \quad \dot{t}_{14} = 0, \quad \dot{t}_{24} = 0, \quad \dot{t}_{34} = 0
\end{align*}
\]

(5.78)

where the elements \(\dot{t}^{ij}\) represent time derivatives of the elements of the closed loop matrix product. Note the diagonal terms are not used in the rotation partition equations as was done with the displacement analysis. This is not necessary
here due to the linearity of the equations. The equations are solved directly, and therefore, it is not possible for the solution to "converge" to an invalid solution.

The elements of the closed loop matrix product are functions of the joint variables which are functions of time themselves. Hence the elements must be differentiated using the chain rule to get

\[
\begin{align*}
\frac{\partial t^{13}}{\partial q_1} t_1 + \frac{\partial t^{13}}{\partial q_2} t_2 + \cdots + \frac{\partial t^{13}}{\partial q_N} t_N &= 0 \\
\frac{\partial t^{21}}{\partial q_1} t_1 + \frac{\partial t^{21}}{\partial q_2} t_2 + \cdots + \frac{\partial t^{21}}{\partial q_N} t_N &= 0 \\
\frac{\partial t^{32}}{\partial q_1} t_1 + \frac{\partial t^{32}}{\partial q_2} t_2 + \cdots + \frac{\partial t^{32}}{\partial q_N} t_N &= 0 \\
\frac{\partial t^{14}}{\partial q_1} t_1 + \frac{\partial t^{14}}{\partial q_2} t_2 + \cdots + \frac{\partial t^{14}}{\partial q_N} t_N &= 0 \\
\frac{\partial t^{24}}{\partial q_1} t_1 + \frac{\partial t^{24}}{\partial q_2} t_2 + \cdots + \frac{\partial t^{24}}{\partial q_N} t_N &= 0 \\
\frac{\partial t^{34}}{\partial q_1} t_1 + \frac{\partial t^{34}}{\partial q_2} t_2 + \cdots + \frac{\partial t^{34}}{\partial q_N} t_N &= 0
\end{align*}
\]

As presented in Section 4.4.2, this may be written in terms of the Jacobian as

\[ J \ddot{q} = 0. \tag{5.85} \]

This matrix equation is then partitioned according to the dependent and independent joint rates as

\[ J_D \ddot{q}_D = -J_I \dot{q}_I \tag{5.86} \]

\[ = \ddot{\theta}_w \tag{5.87} \]
where

\[ \dot{Q}_I = \text{the } N_I \text{ independent joint dof velocities; } N_I \times 1 \]
\[ \dot{Q}_D = \text{the } N_D \text{ dependent joint dof velocities; } N_D \times 1 \]
\[ J_I = \text{columns of Jacobian associated with independent joint } \]
\[ \text{dof velocities; } 6 \times N_I \]
\[ J_D = \text{columns of Jacobian associated with dependent joint dof } \]
\[ \text{velocities; } 6 \times N_D \]
\[ \dot{R}_v = \text{the known right-hand side of the velocity equation; } 6 \times 1. \]

The procedure used to solve the velocity system of equations at a given position
is shown in Table 7. This procedure assumes the displacement analysis has been
completed since all of the joint variables must be known to evaluate the partial
derivatives.

The partial derivatives of the closed loop matrix product are done numerically
as has been done in previous sections. The numerical derivatives demonstrated up
to this point have all been done with the forward difference method. However, for
the velocity analysis, the central difference method yields better accuracy. Recall
that numerical derivatives were used for both the optimization and the Newton's
method displacement approaches. Since both of these displacement analysis tech­
niques are iterative in nature, errors in the derivative calculation are eventually
iterated to a negligible size. This is not so with the velocity analysis.

The velocity system of equations are linear in the unknowns, and therefore, are
solvable directly. Hence, the accuracy of the Jacobian is very important since it is
calculated only once at a given position. In order to make the partial derivatives
as accurate as reasonably possible, the central difference theorem is used where

\[ \frac{\partial T(\tilde{Q})_{1,1}}{\partial q_i} \bigg|_{\tilde{q}=\tilde{q}_0} = \frac{T(\tilde{Q}_0 : q_i = q_{i0} + \epsilon)_{1,1} - T(\tilde{Q}_0 : q_i = q_{i0} - \epsilon)_{1,1}}{2\epsilon} \quad (5.89) \]
Table 7: General procedure for formulating and solving the velocity system of equations.

<table>
<thead>
<tr>
<th>Formulation and Solution of Velocity Analysis Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Evaluate the partial derivative of the closed loop matrix product with respect to the i-th dependent joint variable.</td>
</tr>
<tr>
<td>2. Place elements (1, 3), (2, 1), (3, 2), (1, 4), (2, 4), and (3, 4) of the partially differentiated matrix in column i of $J_D$.</td>
</tr>
<tr>
<td>3. Repeat steps 1–2 for all of the dependent joint variables. This will completely define $J_D$.</td>
</tr>
<tr>
<td>4. Evaluate the partial derivative of the closed loop matrix product with respect to the i-th independent joint variable.</td>
</tr>
<tr>
<td>5. Place elements (1, 3), (2, 1), (3, 2), (1, 4), (2, 4), and (3, 4) of the partially differentiated matrix in column i of $J_I$.</td>
</tr>
<tr>
<td>6. Repeat steps 4–5 for all of the independent joint variables. This will completely define $J_I$.</td>
</tr>
<tr>
<td>7. Evaluate the right-hand side vector, say $\vec{R}_v$, using the known independent joint rates, i.e., $\vec{R}_v = -J_J \hat{q}_I$.</td>
</tr>
<tr>
<td>8. (a) If there are more than six unknown joint rates, $J_D$ will be $6 \times N_D$. Make $J_D$ square by adding $N_D - 6$ rows of zero to it. The same number of zeros must also be augmented to the right-hand side vector $\vec{R}_v$. If $N_D \leq 6$, use $J_D$ as is.</td>
</tr>
<tr>
<td>(b) Determine the SVD of $J_D = U W V^T$.</td>
</tr>
<tr>
<td>(c) If any $w_j$'s (the diagonal elements of $W$) are equal to zero, the corresponding columns, $j$, of $V$ represent the nullspace basis vectors for the velocity solution.</td>
</tr>
<tr>
<td>9. After setting the $1/w_j$'s to zero for all $w_j = 0$, calculate $\hat{\dot{q}}_D$:</td>
</tr>
<tr>
<td>$$\hat{\dot{q}}_D = -V \left[ \text{diag} \left( \frac{1}{w_j} \right) \right] U^T \vec{R}_v.$$ (5.88)</td>
</tr>
</tbody>
</table>
The error for the forward difference approach is $O(\epsilon)$, while the error for the central difference approach is $O(\epsilon^2)$ [28]. Assuming the closed loop matrix product calculated at $\ddot{\mathbf{Q}}_0$ is already available, only one additional calculation of the matrix product is need for the forward difference approximation. However, the central difference approximation requires two additional calculations—one at "+$\epsilon$" and one at "$-\epsilon$." The additional accuracy obtained with the central difference approach justifies the extra calculation required.

The Jacobian formed for the velocity analysis is different than that formed for the linearized portion of the displacement analysis. The most obvious reason has to do with the functions for which the Jacobians are formed. The functions used for the displacement analysis have the extra diagonal terms added to the rotation partition related equations. These are not necessary in the velocity analysis. Furthermore, the Jacobian used in the displacement analysis was developed with respect to only the dependent joint variables. Recall that the linearized solution is performed to generate corrections to improve a current estimate for the nonlinear displacement solution. The input joint variables are specified (i.e., they are already known exactly), and as such, they do not (should not) become part of the correction process.

The velocity analysis, however, is obviously driven by the velocity of the input joint variables. It makes sense for the time derivatives of the independent joint variables to part of the solution. The mathematical derivation for the differentiation of the closed loop matrix product was performed without any regard as to what joint rates were known or unknown. However, once the system of equations is presented in terms of the Jacobian matrix and the joint rate vector, it makes
sense to partition the joint rates (and therefore, the Jacobian) into dependent and independent parts. The independent quantities can then be brought over to the "right-hand side" of the equation as a known (constant) quantity.

5.5.2 Formulation of the Acceleration Analysis System of Equations

The development and solution to the acceleration equations follows directly from the displacement and velocity analyses. As expected there will be six (at most) linearly independent equations available. These are produced by differentiating the closed loop matrix product twice with respect to time. As with the velocity analysis, elements (1, 3), (2, 1), and (3, 2) from the rotation partition and (1, 4), (2, 4), and (3, 4) from the displacement partition are equated with corresponding elements of the null matrix (i.e., the second time derivative of the identity matrix).

The six acceleration equations will then be

\[ \ddot{\mathbf{r}}^3 = 0, \quad \ddot{\mathbf{r}}^{21} = 0, \quad \ddot{\mathbf{r}}^{32} = 0, \quad \ddot{\mathbf{r}}^{14} = 0, \quad \ddot{\mathbf{r}}^{24} = 0, \quad \ddot{\mathbf{r}}^{34} = 0 \]  

(5.90)

where the elements \( \ddot{\mathbf{r}}^{ij} \) represent second time derivatives of the elements of the closed loop matrix product.

The second derivative of the closed loop matrix product with respect to time is developed in Section 4.5.2 and shown in Equation 4.53. It is restated here for convenience:

\[
\sum_{i=1}^{P} \frac{\partial \mathbf{T}(\mathbf{Q})_{1,1}}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_i + \sum_{i=1}^{P} \frac{\partial^2 \mathbf{T}(\mathbf{Q})_{1,1}}{\partial \mathbf{q}_i^2} \dot{\mathbf{q}}_i^2 + 2 \sum_{i=1}^{P} \left( \sum_{j=i+1}^{P} \frac{\partial^2 \mathbf{T}(\mathbf{Q})_{1,1}}{\partial \mathbf{q}_i \partial \mathbf{q}_j} \dot{\mathbf{q}}_i \dot{\mathbf{q}}_j \right) = O_4. \tag{5.91}
\]

or

\[
\sum_{i=1}^{P} \frac{\partial \mathbf{T}(\mathbf{Q})_{1,1}}{\partial \mathbf{q}_i} \dot{\mathbf{q}}_i = -(\mathbf{A}_n + \mathbf{A}_c) \tag{5.92}
\]

\[ = \mathbf{B}_{\text{closed}} \tag{5.93} \]
where $\mathbf{B}_{\text{closed}}$ consists of the normal and Coriolis accelerations; these are known after the displacement and velocity analyses have been completed.

Extracting the $(1, 3)$, $(2, 1)$, $(3, 2)$, $(1, 4)$, $(2, 4)$, and $(3, 4)$ elements from Equation (5.93) yields a system of equations very similar in form to those obtained for the velocity analysis:

\begin{align*}
\frac{\partial t^{13}}{\partial q_1} \ddot{q}_1 + \frac{\partial t^{13}}{\partial q_2} \ddot{q}_2 + \cdots + \frac{\partial t^{13}}{\partial q_N} \ddot{q}_N &= b^{13} \\
\frac{\partial t^{21}}{\partial q_1} \ddot{q}_1 + \frac{\partial t^{21}}{\partial q_2} \ddot{q}_2 + \cdots + \frac{\partial t^{21}}{\partial q_N} \ddot{q}_N &= b^{21} \\
\frac{\partial t^{32}}{\partial q_1} \ddot{q}_1 + \frac{\partial t^{32}}{\partial q_2} \ddot{q}_2 + \cdots + \frac{\partial t^{32}}{\partial q_N} \ddot{q}_N &= b^{32} \\
\frac{\partial t^{14}}{\partial q_1} \ddot{q}_1 + \frac{\partial t^{14}}{\partial q_2} \ddot{q}_2 + \cdots + \frac{\partial t^{14}}{\partial q_N} \ddot{q}_N &= b^{14} \\
\frac{\partial t^{24}}{\partial q_1} \ddot{q}_1 + \frac{\partial t^{24}}{\partial q_2} \ddot{q}_2 + \cdots + \frac{\partial t^{24}}{\partial q_N} \ddot{q}_N &= b^{24} \\
\frac{\partial t^{34}}{\partial q_1} \ddot{q}_1 + \frac{\partial t^{34}}{\partial q_2} \ddot{q}_2 + \cdots + \frac{\partial t^{34}}{\partial q_N} \ddot{q}_N &= b^{34}.
\end{align*}

Similar to the velocity analysis equations, this may be written as

$$\mathbf{J} \dddot{\mathbf{Q}} = \dddot{\mathbf{b}}.$$  \hspace{1cm} (5.100)

After partitioning $\dddot{\mathbf{Q}}$ and the Jacobian into dependent and independent parts, a system is obtained that is easily solved for the second time derivatives of the dependent joint variables in terms of the independent accelerations:

$$\mathbf{J}_D \dddot{\mathbf{Q}}_D = \dddot{\mathbf{b}} - \mathbf{J}_I \dddot{\mathbf{Q}}_I = \dddot{\mathbf{R}}_a.$$  \hspace{1cm} (5.101)
where

$$\ddot{Q}_I = \text{the } N_I \text{ independent joint dof accelerations; } N_I \times 1$$

$$\ddot{Q}_D = \text{the } N_D \text{ dependent joint dof accelerations; } N_D \times 1$$

$$J_I = \text{columns of Jacobian associated with independent joint dof accelerations; } 6 \times N_I$$

$$J_D = \text{columns of Jacobian associated with dependent joint dof accelerations; } 6 \times N_D$$

$$\ddot{b} = \text{combined normal and Coriolis acceleration terms; } 6 \times 1$$

$$\ddot{R}_a = \text{the known right-hand side of the acceleration equation; } 6 \times 1.$$

The acceleration system of equations at a given position can be solved according to the procedure shown in Table 8. This procedure assumes the displacement and velocity analyses have been completed. All of the joint variables must be known to evaluate the partial derivatives, and the velocities are needed to evaluate the normal and Coriolis acceleration terms.

The normal and Coriolis acceleration matrices are represented by the second and third terms respectively on the left-hand side of Equation (5.91). The only involved aspect of evaluating these terms has to do with determining the second partial derivatives of the closed loop matrix product. The normal acceleration term uses second partial derivatives with respect to the same joint variable. The Coriolis acceleration requires the "mixed" second partial derivatives.

Calculating the partial derivative of the closed loop matrix product for the normal acceleration is relatively straightforward. A given joint variable can occur in only one constituent matrix of the closed loop matrix product:

$$T(\ddot{Q})_{1,1} = T_{12} T_{23} \cdots T(q_i)_{IJ} \cdots T_{N-1,N} T_{N1}. \quad (5.104)$$
Table 8: General procedure for formulating and solving the acceleration system of equations.

Formulation and Solution of Acceleration Analysis Equations

1. Evaluate the partial derivative of the closed loop matrix product with respect to the \( i \)-th dependent joint variable.

2. Place elements \((1, 3), (2, 1), (3, 2), (1, 4), (2, 4), \) and \((3, 4)\) of the partially differentiated matrix in column \( i \) of \( J_D \).

3. Repeat steps 1–2 for all of the dependent joint variables. This will completely define \( J_D \).

4. Evaluate the partial derivative of the closed loop matrix product with respect to the \( i \)-th independent joint variable.

5. Place elements \((1, 3), (2, 1), (3, 2), (1, 4), (2, 4), \) and \((3, 4)\) of the partially differentiated matrix in column \( i \) of \( J_I \).

6. Repeat steps 4–5 for all of the independent joint variables. This will completely define \( J_I \).

7. Evaluate the normal and Coriolis acceleration matrices, \( A_n \) and \( A_c \) respectively, and calculate \( B_{\text{closed}} = -(A_n + A_c) \).

8. Evaluate the right-hand side vector, say \( \mathbf{R}_a \), using the known independent joint rates and the known components of \( B_{\text{closed}} \) i.e., \( \mathbf{R}_a = \ddot{\mathbf{b}} - J_I \ddot{Q}_I \).

9. (a) If there are more than six unknown joint rates, \( J_D \) will be \( 6 \times N_D \). Make \( J_D \) square by adding \( N_D - 6 \) rows of zero to it. The same number of zeros must also be augmented to the right-hand side vector \( \mathbf{R}_a \). If \( N_D \leq 6 \), use \( J_D \) as is.

(b) Determine the SVD of \( J_D = U W V^T \).

(c) If any \( w_j \)'s (the diagonal elements of \( W \)) are equal to zero, the corresponding columns, \( j \), of \( V \) represent the nullspace basis vectors for the acceleration solution

10. After setting the \( 1/w_j \)'s to zero for all \( w_j = 0 \), calculate \( \ddot{Q}_D \):

\[
\ddot{Q}_D = -V \left[ \text{diag} \left( 1/w_j \right) \right] U^T \ddot{R}_a. \tag{5.103}
\]
The second partial derivative with respect to \( q_i \) must then be

\[
\frac{\partial^2 T(\vec{q})_{1,1}}{\partial q_i^2} = T_{12} T_{23} \cdots \frac{\partial^2 T(q_i)_{IJ}}{\partial q_i^2} \cdots T_{N-1,N} T_{N1}. \tag{5.105}
\]

The central difference formula for the second partial derivative with respect to the same variable is given by

\[
\frac{\partial^2 T(q_i)_{IJ}}{\partial q_i^2} \approx \frac{T(q_i + 2\varepsilon)_{IJ} - 2T(q_i)_{IJ} + T(q_i - 2\varepsilon)_{IJ}}{(2\varepsilon)^2}. \tag{5.106}
\]

Hence, this approximation for the second partial with respect to \( q_i \) of the constituent matrix can be used in Equation (5.105) to approximate the second partial of the closed loop matrix product.

Calculation of the mixed second derivative for the Coriolis acceleration is a bit more involved as two cases can arise. The mixed second derivative of the closed loop matrix product is written as

\[
\frac{\partial^2 T(\vec{q})_{1,1}}{\partial q_i \partial q_j}
\]

where \( q_i \) and \( q_j \) are two different joint variables in a mechanism. The approximation to this derivative is performed differently depending on whether or not \( q_i \) and \( q_j \) appear in different constituent transformations

\[
T(\vec{q})_{1,1} = T_{12} T_{23} \cdots T(q_i)_{IJ} \cdots T(q_j)_{KL} \cdots T_{N-1,N} T_{N1} \tag{5.107}
\]

or whether they appear in the same constituent transformation

\[
T(\vec{q})_{1,1} = T_{12} T_{23} \cdots T(q_i, q_j)_{IJ} \cdots T_{N-1,N} T_{N1}. \tag{5.108}
\]

For the case where \( q_i \) and \( q_j \) appear in different matrices, the differentiation of the closed loop matrix product results in

\[
\frac{\partial^2 T(\vec{q})_{1,1}}{\partial q_i \partial q_j} = T_{12} T_{23} \cdots \frac{\partial T(q_i)_{IJ}}{\partial q_i} \cdots \frac{\partial T(q_j)_{KL}}{\partial q_j} \cdots T_{N-1,N} T_{N1}. \tag{5.109}
\]
Hence, only the first partial derivatives of the two matrices containing \( q_i \) and \( q_j \) need to be approximated. The resulting matrices can then be used in place of the original undifferentiated matrices as they are in Equation (5.109). The "matrix product" yielded by this is the second mixed partial derivative being sought.

For the second case where \( q_i \) and \( q_j \) are in the same transformation matrix, performing the second mixed partial differentiation of the closed loop matrix product gives

\[
\frac{\partial^2 \mathbf{T}(\bar{q})}{\partial q_i \partial q_j} = \mathbf{T}_{12} \mathbf{T}_{23} \cdots \frac{\partial^2 \mathbf{T}(q_i, q_j)}{\partial q_i \partial q_j} \cdots \mathbf{T}_{N-1,N} \mathbf{T}_{N1}. \tag{5.110}
\]

The central difference based formula for partially differentiating this one constituent matrix with respect to \( q_i \) and \( q_j \) is given by

\[
\frac{\partial^2 \mathbf{T}(q_i, q_j)}{\partial q_i \partial q_j} \approx \begin{bmatrix}
\mathbf{T}(q_i + \epsilon, q_j + \delta)_{IJ} & - \mathbf{T}(q_i + \epsilon, q_j - \delta)_{IJ} & - \\
\mathbf{T}(q_i - \epsilon, q_j + \delta)_{IJ} & + \mathbf{T}(q_i - \epsilon, q_j - \delta)_{IJ}
\end{bmatrix}
\]

\[
= \frac{(2\epsilon)(2\delta)}{(2\epsilon)(2\delta)}
\]

\[
\tag{5.111}
\]

The perturbation values used for \( \epsilon \) and \( \delta \) may be chosen as equal or not equal. The matrix yielded by this approximation may then be placed in Equation (5.110) in order to produce an approximation to the second mixed partial of the closed loop transformation product.

5.6 Calculation of Kinematic Properties From the Joint Variable Solutions

Once the displacement, velocity, and acceleration analyses for a particular mechanism have been completed, it is desirable to calculate kinematic properties of the links and points on the links in terms of the ground reference frame. The joint variable information obtained thus far provide relative information between adjacent links. The type of information usually desired is as follows:
• The angular velocities and accelerations of the links relative to ground.

• The locations, velocities, and accelerations of points of interest on the links relative to ground.

5.6.1 Angular Velocities of the Links

A transformation from system $J$ to system $I$ consists of a $3 \times 3$ rotation partition and a $3 \times 1$ displacement partition. The rotation partition is made up of the direction cosines of system $J$'s axes relative to system $I$'s axes. Let a $3 \times 3$ matrix, $C_{JJ}$, be defined using the rotation partition. Since this direction cosine matrix describes the orientation of system $J$ relative to system $I$, it is reasonable that its time derivative would be involved with determining the angular velocity of system $J$ relative to system $I$. Beggs [12] shows

$$I_\Omega_{IJ} = \dot{C}_{IJ} C_{IJ}^T$$

(5.112)

where $I_\Omega_{IJ}$ is the angular velocity matrix of system $J$ relative to system $I$ resolved in $I$ coordinates. Its elements are as follows:

$$I_\Omega_{IJ} = \begin{bmatrix} 0 & -I_\omega_x^{yJ} & I_\omega_y^{yJ} \\ I_\omega_x^{yJ} & 0 & -I_\omega_z^{yJ} \\ -I_\omega_y^{yJ} & I_\omega_z^{yJ} & 0 \end{bmatrix}$$

(5.113)

Hence, the $x, y, z$ components of the angular velocity of system $J$ relative to system $I$ can be calculated using Equation (5.112).

Assuming system $I$ is the ground frame of a mechanism, the angular velocity of any link $J$ relative to ground may be obtained using

$$I_\Omega_{1J} = \dot{C}_{1J} C_{1J}^T.$$ 

(5.114)
Since the direction cosine matrix is synonymous with the rotation partition of the link transformation matrices, the angular velocity of any link relative to ground may be obtained by equating

\[ ^1\Omega_{1J} = \text{rotation partition of } [\dot{T}_{1J} \dot{T}_{1J}^T]. \]  

(5.115)

If link 1 is ground, it obviously has no angular velocity. Hence, link 2 is the first link in the mechanism of any angular velocity interest. Since the displacement and velocity analyses are assumed to be completed, both \( T_{12} \) and \( \dot{T}_{12} \) are easily calculated. \( T_{12} \) is defined as

\[ T(\vec{f})_{12} = S_1 P(\vec{\dot{f}})_{12} \]  

(5.116)

where \( S_1 \) is the constant shape matrix for link 1, and \( P(\vec{\dot{f}})_{12} \) is the pair matrix connecting links 1 and 2. The degrees-of-freedom of this joint are represented by \( \vec{f} \). The number of degrees-of-freedom possessed by this joint is represented by \( n_f \).

All of them are known from the displacement analysis. The time derivative of \( T_{12} \) is performed using the chain rule since it is a function of the \( n_f \) time varying joint variables:

\[ \dot{T}(\vec{f})_{12} = \sum_{i=1}^{n_f} \frac{\partial T(\vec{f})_{12}}{\partial \dot{f}_i} \dot{f}_i. \]  

(5.117)

The partial derivatives are approximated using the central difference formula, and the joint rates, \( \dot{f}_i \), are known from the velocity analysis. The matrices obtained from Equations (5.117) and (5.116) are then used according to Equation (5.115) in order to get the angular velocity components of link 2 relative to link 1.

Proceeding to link 3, \( T_{13} \) may be written as

\[ T_{13} = T_{12} T_{23} \]  

(5.118)
and therefore,

\[ \dot{T}_{13} = \dot{T}_{12}T_{23} + T_{12}\dot{T}_{23}. \]  \hspace{1cm} (5.119)

Since \( T_{12} \) and \( \dot{T}_{12} \) have already been determined, not much additional effort is required to calculate \( T_{13} \) and \( \dot{T}_{13} \). The transformation \( T_{23} \) is calculated according to its "shape-pair" matrix definition, and \( \dot{T}_{23} \) is calculated using the chain rule and central difference approximation. This allows \( T_{13} \) and \( \dot{T}_{13} \) to be evaluated which in turn will lead to the angular velocity of link 3 relative to link 1.

A recursive process is evident for determining successive link angular velocities relative to ground (link 1). The following process begins with link 2 and proceeds through the last link, \( N \):

\[ ^1\Omega_{1J} = \text{rotation partition of } [ \dot{T}_{1J} T_{1J}^T ] \]  \hspace{1cm} (5.120)

where

\[ T_{1J} = T_{1,J-1}T_{J-1,J} \]  \hspace{1cm} (5.121)

\[ \dot{T}_{1J} = \dot{T}_{1,J-1}T_{J-1,J} + T_{1,J-1}\dot{T}_{J-1,J} \]  \hspace{1cm} (5.122)

\[ T_{J-1,J} = S_{J-1}P_{J-1,J} \]  \hspace{1cm} (5.123)

\[ \dot{T}_{J-1,J} \approx S_{J-1} \sum_{i=1}^{n_f} \frac{P(f_i + \epsilon)_{J-1,J} - P(f_i - \epsilon)_{J-1,J}}{2\epsilon} \dot{f}_i \]  \hspace{1cm} (5.124)

Note in the above process, \( S_{J-1} \) is the shape matrix for link \( J - 1 \); \( P_{J-1,J} \) is the pairing element matrix between links \( J - 1 \) and \( J \); \( \dot{f} \) is the vector of the \( n_f \) joint variables found in \( P_{J-1} \); and \( T_{1,1} = I \) and \( \dot{T}_{1,1} = 0 \).

The angular velocity development above produces the components of the link angular velocities relative to system 1. It is important to understand exactly which system is "system 1." It is obvious that system 1 is attached to link 1, however,
referring back to Figure 22, there are three systems (so far) associated with a given link. Systems \( u \) and \( x \) (short for systems \( uvw \) and \( xyz \) respectively) are the pairing element systems. Their locations and orientations on a link are based on the joints used. The link definition system, \( l \), is the system in which systems \( u \) and \( x \) are defined. The indirect method of establishing the shape matrix for a given link uses two points on the \( z \) axis and one point on the \( x \) axis for each of the pairing element systems. The coordinates of these points are defined in terms of system \( l \). As such, system \( l \) is usually placed in a physically significant location and orientation on a link.

The transformation \( \mathbf{T}_{IJ} \) was defined as the transformation from system \( u \) on link \( J \) to system \( u \) on link \( I \). Hence, the angular velocities, as developed above, will be resolved into the \( u \) system of link 1. Rather than have the angular velocity components resolved into a pairing element system, it is more useful to have them resolved into system \( l \) of link 1. The link definition system can be positioned and oriented on a link wherever it is convenient and informative. The placement of the pair element matrices, however, is very structured due to the pair matrix specifications for the available joints. Fortunately, the information needed to transform the angular velocity components into system \( l \) has already been calculated.

When developing the link shape matrices using the indirect method (see page 79), the transformation from system \( u \) to system \( l \), \( \mathbf{T}_{lu} \), was calculated for each link. The rotation partition of this transformation will perform a change of basis from system \( u \) to system \( l \). Hence, writing the \( 3 \times 3 \) rotation partition as \( \mathbf{R}_{lu} \), the angular velocity components can be resolved into the definition system, \( l \), of link.
5.6.2 Angular Accelerations of the Links

A matrix expression for the angular acceleration of the links relative to ground is obtained by differentiating Equation (5.114)

\[ ^1A_{1J} = ^1\dot{\Omega}_{1J} = \dot{C}_{1J} C_{1J}^T + \dot{C}_{1J} C_{1J}^T. \]  

(5.126)

where

\[ ^1A_{J} = \begin{bmatrix} 0 & -\alpha_{\gamma J} & \alpha_{\alpha J} \\ \alpha_{\alpha J} & 0 & -\alpha_{\gamma J} \\ -\alpha_{\gamma J} & \alpha_{\alpha J} & 0 \end{bmatrix}. \]  

(5.127)

As with the angular velocity development, the angular acceleration matrix can be obtained using

\[ ^1A_{JJ} = \text{rotation partition of } [\ddot{T}_{1J} T_{1J}^T + \ddot{T}_{1J} T_{1J}^T]. \]  

(5.128)

The \( T_{1J} \) and \( \ddot{T}_{1J} \) matrices were formed in the angular velocity calculation. Hence, the only new term here is the \( \ddot{T}_{1J} \) matrix. By differentiating the recursion formula for the first derivative shown in Equation (5.122), a recursion formula can be developed for the second derivative:

\[ \ddot{T}_{1J} = \dddot{T}_{1J} \ddot{T}_{J-1J} + 2\dot{T}_{1J} \dddot{T}_{J-1J} + T_{1J} \dddot{T}_{J-1J} \]  

(5.129)

where

\[ \dddot{T}_{J-1J} = S_{J-1} \dddot{P}_{J-1J}. \]  

(5.130)

\[ = S_{J-1} [\dddot{P}_{\text{tangential}} + \dddot{P}_{\text{normal}} + \dddot{P}_{\text{Coriolis}}] \]  

(5.131)
Note the joint variable velocities, $\dot{x}_i$, and accelerations, $\ddot{x}_i$, used in the summations are known from the closed loop velocity and acceleration analysis. The results of Equation (5.132) can now be used with Equations (5.121), (5.122), (5.123), and (5.124) to develop all of the terms necessary for the right-hand side of Equation (5.128). The off-diagonal elements of the resulting matrix will provide the angular acceleration components of link $J$ relative to link 1 according to Equation (5.128).

As with the angular velocity formulation, the recursion process starts with link 2 and proceeds to the last link, $N$, of the mechanism.

As with the angular velocities, the techniques presented above for the angular accelerations produce components resolved in the $u$ pairing element system of link 1. These may be transformed link definition system of link 1 for a more useful result. The change of basis from system $u$ to system $l$ may be obtained from the
rotation partition of $T_{iu}$. This transformation is originally calculated to produce
the shape matrix. Letting the rotation partition be represented as $R_{iu}$, the angular
accelerations may be resolved into the $I$ system of link 1 using

$$
\begin{bmatrix}
  ^I\alpha_{1J}^x \\
  ^I\alpha_{1J}^y \\
  ^I\alpha_{1J}^z
\end{bmatrix}
= R_{iu} \begin{bmatrix}
  ^u\alpha_{1J}^x \\
  ^u\alpha_{1J}^y \\
  ^u\alpha_{1J}^z
\end{bmatrix}.
$$

(5.133)

5.7 Position, Velocity, and Acceleration of Points of Interest

The position, velocity, and acceleration of points on the links of a mechanism can
be determined very easily. In fact, most of the work necessary to determine this
information has already been performed for the angular velocity and acceleration
calculation.

Recall the basic description of a coordinate transformation matrix, say $T_{IJ}$. It
is used to transform the coordinates of a point defined in system $J$ into coordinates
meaningful in system $I$. Hence, given a point defined in system $J$, its coordinates
in system $I$ are given by

$$
^I\mathbf{r} = T_{IJ}^J \mathbf{r}.
$$

(5.134)

This equation can be differentiated with respect to time to yield a relation for the
velocity of the point relative to system $I$:

$$
^I\mathbf{v} = \dot{^I\mathbf{r}} = \dot{T}_{IJ}^J \mathbf{r} + T_{IJ}^J \dot{\mathbf{r}}.
$$

(5.135)

If it is differentiated once again, a relation for the acceleration of the point is
obtained:

$$
^I\mathbf{a} = \ddot{^I\mathbf{r}} = \ddot{T}_{IJ}^J \mathbf{r} + 2 \dot{T}_{IJ}^J \dot{\mathbf{r}} + T_{IJ}^J \ddot{\mathbf{r}}.
$$

(5.136)
The specification of a point on a link will be done in that link's definition system \( I \). All points will be considered fixed relative to the links on which they are defined. The transformation matrix, \( T_{IJ} \), between links \( I \) and \( J \) is defined as the transformation from the \( u \) coordinate system on link \( J \) to the \( u \) coordinate system on link \( I \). Since points on a given link are specified in system \( I \) of that link, the coordinates of those points must be transformed into system \( u \) of the link before the above position, velocity, and acceleration equations can be used.

As discussed previously, the indirect calculation of the shape matrices requires \( T_{lu} \) to be formed for each link in the mechanism. Its rotation partition can be used to change the basis of a point from system \( u \) to system \( l \). In the situation at hand, however, the basis must be changed from system \( l \) to system \( u \). This operation can be performed by using the transpose of the rotation partition. Thus, given a point defined in system \( l \) of a link, it may be resolved into system \( u \) by

\[
_{u}\mathbf{r} = R_{lu}^{T} _{l}\mathbf{r}
\]

where system \( u \) and system \( l \) are fixed on the same link.

Since the coordinates of a point relative to the link on which it is defined are fixed, the time rates of change of the coordinates relative to that link must be zero. Hence, the first and second time derivatives of \( J\mathbf{r} \) in Equations (5.135) and (5.136) must be zero. Setting system \( I \) to 1 in Equations (5.134), (5.135), and (5.136) then yields

\[
_{1}\mathbf{r} = T_{1J} R_{lu}^{T} _{l}\mathbf{r}
\]

\[
_{1}\mathbf{v} = \dot{T}_{1J} R_{lu}^{T} _{l}\mathbf{r}
\]

\[
_{1}\mathbf{a} = \ddot{T}_{1J} R_{lu}^{T} _{l}\mathbf{r}.
\]
Recalling that $T_{IJ}$ transforms between the $u$ systems of links, the above position, velocity, and acceleration will be resolved in the $u$ system of link 1. They can be resolved into the definition system, $l$, of link 1 using the rotation partition of $T_{i1u1}$, i.e.,

$$
{l_1}^p = R_{i1u1} T_{1J} R_{1JuJ} {l_1}^p
$$

(5.141)

$$
{l_1}^\dot{p} = R_{i1u1} \dot{T}_{1J} R_{1JuJ} {l_1}^p
$$

(5.142)

$$
{l_1}^{\ddot{p}} = R_{i1u1} \ddot{T}_{1J} R_{1JuJ} {l_1}^p
$$

(5.143)

Notice that $T_{1J}$, $\dot{T}_{1J}$, and $\ddot{T}_{1J}$ were all calculated for the angular velocity and acceleration evaluation. Hence, given the "local" coordinates of a point on a link, it is very straightforward to determine the position, velocity, and acceleration of that point relative to the definition system of link 1.
CHAPTER VI

Dynamic Force Analysis

The kinematic analysis as presented in this dissertation is based on specified displacement, velocity, and acceleration inputs. If the links of the mechanism have mass, or they have external loads applied to them, it will naturally take a certain amount of input force to generate the specified kinematic inputs. The force analysis of a mechanism, as developed in this dissertation, is concerned with determining the required input force as well as the reactions experienced at each of the joints. Knowledge of the required input force is useful for selecting or designing a power source for the mechanism. The joint reactions are needed to adequately design the links and joints for strength.

This dissertation will consider the force analysis of spatial mechanisms for both frictionless and friction joints. The friction is modeled using Coulomb friction. The system of equations used to solve the force problem come from dynamic force equilibrium principles and form joint force constraint equations. The manner in which the joint force constraint equations are developed determines whether friction is neglected or considered.

As will be seen, the joint force constraints for the frictionless problem will be linear in the unknown joint forces. The friction problem will yield constraint equations nonlinear in the joint forces. Consequently, the frictionless force analysis can
be solved using direct linear equation solvers. A nonlinear solution, however, must be employed when solving the friction analysis. As with the kinematic analyses, singular value decomposition will be used for both types of force analysis. SVD is used directly for the linear frictionless analysis. The nonlinear friction analysis will be done using Newton's method for which SVD is iteratively used in the convergence process. The motivation for the use of SVD is, again, based on its robustness in dealing with deficient systems of equations.

6.1 Dynamic Force Equilibrium

Consider a body with known kinematic properties relative to an inertial frame \([87]\). It has an affixed coordinate system, \(G\), whose origin is the body's center of mass and whose axes are aligned with the body's principal axes. This system is called the kinetic reference frame. Newton's Second Law states:

\[
\sum \mathcal{G} \mathbf{f}_\text{ext} = m \mathbf{a}_G
\]  

\[
\sum \mathcal{G} \mathbf{m}_\text{ext} + \sum \mathcal{G} \mathbf{r} \times \mathcal{G} \mathbf{f}_\text{ext} = \mathbf{I}_G \mathbf{a} + \mathcal{G} \mathbf{\omega} \times \mathcal{G} \mathbf{\omega}
\]

where

- \(\mathcal{G} \mathbf{f}\) = externally applied forces resolved in system \(G\)
- \(m\) = mass of the body
- \(\mathbf{a}_G\) = acceleration of center of mass, resolved in system \(G\), relative to an inertial frame
- \(\mathcal{G} \mathbf{m}\) = externally applied moments resolved in system \(G\)
- \(\mathcal{G} \mathbf{r}\) = position vector, resolved in system \(G\), from center of mass to externally applied force
\( I_{G_G} \) = diagonal matrix of principal moments of inertia

\( \alpha \) = angular velocity, resolved in system \( G \), of the body relative to an inertial frame

\( \alpha \) = angular acceleration, resolved in system \( G \), of the body relative to an inertial frame

Assume this body represents a link in a mechanism whose kinematic analysis is complete. The results from the kinematic analysis provide information relative to the inertial (i.e., ground) frame but resolved into the coordinates of a system embedded in link 1. Hence, the kinematic quantities needed for the above equations must resolved into the kinetic frame, \( G \).

Let \( L \) represent the change of basis from system \( G \) to system 1. The inverse of \( L \) will allow the change of basis from system 1 to system \( G \). As \( L \) is orthogonal, its transpose may be used. Therefore,

\[
\begin{align*}
\alpha &= L^T \omega \\
\alpha &= L^T \alpha
\end{align*}
\]

where

\( \omega \) = the angular velocity of the link relative to ground and resolved in a system embedded in link 1

\( \alpha \) = the angular acceleration of the link relative to ground and resolved in a system embedded in link 1

Note it is possible to determine \( L \) using information from the displacement analysis. Equation (6.2) now becomes

\[
\sum G \vec{m}_{ext} + \sum G \vec{F}_{ext} = I_{G_G} L^T \alpha + L^T \omega \times I_{G_G} L^T \omega
\]

Equations (6.1) and (6.5), which are written in terms of vectors resolved in the kinetic frame, can be written in terms of the ground frame by multiplying through
with \( L \) to yield:

\[
\sum 1f_{ext}^i = m^1\ddot{a}_G \\
\sum 1\dot{m}_{ext}^i + \sum 1\tau \times 1f_{ext}^i = L I_{GG} L^T 1\ddot{\alpha} + 1\dot{\omega} \times L I_{GG} L^T 1\ddot{\omega}.
\] (6.7)

Letting \( I_{G1} = L I_{GG} L^T \), Equation (6.7) may be written as

\[
\sum 1\dot{m}_{ext}^i + \sum 1\tau \times 1f_{ext}^i = I_{G1} 1\ddot{\alpha} + 1\dot{\omega} \times I_{G1} 1\ddot{\omega}.
\] (6.8)

Equations (6.6) and (6.8) are useful forms in that the angular velocity and angular acceleration are resolved into the same system for which they are solved in the kinematic analysis.

Consider the links shown in Figure 35. System \( G \) is assumed aligned with the principal axes and placed at the link center of mass. Systems \( u \) and \( x \) represent the pairing element coordinate systems. In general, the joint bearing forces are not always actually applied at the pairing element coordinate system locations. This is the case when surface contact joints are used. The \( x \) and \( u + 1 \) coordinate systems for such a joint are the systems in which the geometries of the contacting surfaces are defined—the actual contact between the links will occur away from the pairing element coordinate systems.

In order to sum the forces and moments for a general situation, the force state at system \( u + 1 \) of link \( i + 1 \) must be simulated by an equivalent force state at the point of contact with link \( i \). The links in Figure 35 have been shown as separated, but are actually in contact at points \( A \) and \( B \). Given a force state at \( u + 1 \), an equivalent state at \( B \) can be formed using

\[
1f_B^i = 1f_{u+1}^i \\
1\dot{m}_B^i = 1\dot{m}_{u+1}^i - 1\tau_{B/u+1}^i \times 1f_{u+1}^i.
\] (6.9) (6.10)
Figure 35: General joint force configuration between two links.
The force at $A$ is then given as

$$1\vec{f}_A = -1\vec{f}_{u+1}$$  \hspace{1cm} (6.11)

$$1\vec{m}_A = -1\vec{m}_{u+1} + 1\vec{f}_{B/u+1} \times 1\vec{f}_{u+1}.$$  \hspace{1cm} (6.12)

Summing forces and moments about $G$ on link $i$ yields

$$1\vec{f}_u - 1\vec{f}_{u+1} + \sum 1\vec{f}_{ext} = m^1\vec{A}_G$$  \hspace{1cm} (6.13)

$$\left( 1\vec{m}_u - 1\vec{m}_{u+1} + 1\vec{r}_{B/u+1} \times 1\vec{f}_{u+1} \times \right) = I_G 1\vec{\alpha} + 1\vec{\omega} \times I_G 1\vec{\omega}.$$  \hspace{1cm} (6.14)

Note the terms $1\vec{r}_{B/u+1} \times 1\vec{f}_{u+1}$ and $1\vec{r}_{A/G} \times (-1\vec{f}_{u+1})$ may be combined as

$$1\vec{r}_{B/u+1} \times 1\vec{f}_{u+1} + 1\vec{r}_{A/G} \times (-1\vec{f}_{u+1}) = -1\vec{r}_{u+1/G} \times 1\vec{f}_{u+1}$$  \hspace{1cm} (6.15)

since $1\vec{r}_{B/A} = \vec{0}$. Hence, Equation (6.14) may be written as

$$\left( 1\vec{m}_u - 1\vec{m}_{u+1} + 1\vec{r}_{u+1/g} \times 1\vec{f}_{u+1} \times \right) = I_G 1\vec{\alpha} + 1\vec{\omega} \times I_G 1\vec{\omega}.$$  \hspace{1cm} (6.16)

The unknown joint forces and moments are more easily dealt with when resolved into the $u$ coordinate systems. Hence, these forces and moments may be expressed using

$$1\vec{f}_u = C_{1u} u \vec{f}_u$$  \hspace{1cm} (6.17)

$$1\vec{f}_{u+1} = C_{1,u+1} u+1 \vec{f}_{u+1}$$  \hspace{1cm} (6.18)

$$1\vec{m}_u = C_{1u} u \vec{m}_u$$  \hspace{1cm} (6.19)

$$1\vec{m}_{u+1} = C_{1,u+1} u+1 \vec{m}_{u+1}.$$  \hspace{1cm} (6.20)

Note the change of basis transformations are available from the rotation partitions of transformation matrices developed during the displacement analysis.
Equations (6.13) and (6.20) then become

\[
\begin{align*}
C_{1u}^{\dot{u}} f_u - C_{1,u+1}^{\ddot{f}_{u+1}} + \sum f_{ext}^{\dot{f}_{ext}} &= m_1 \ddot{A}_G \quad (6.21) \\
C_{1u}^{\dot{u}} \vec{m}_u - C_{1,u+1}^{\ddot{m}_{u+1} + u+1} + C_{1u}(u \vec{r}_{u/G} \times u \vec{f}_u) - C_{1,u+1}(u+1 \vec{r}_{u+1/G} \times u+1 \vec{f}_{u+1}) + \\
\sum \dot{m}_{ext} + \sum \vec{r} \times \dot{f}_{ext} &= I_{G1}^{\dot{a}} + \omega \times I_{G1}^{\dot{a}}. \quad (6.22)
\end{align*}
\]

The unknowns involved with Equations (6.21) and (6.22) are the joint bearing forces and moments. The angular velocity and acceleration of the link is known from the kinematic analysis. The change of basis (i.e., rotation) transformations are also available from the kinematic analysis. They are functions of the displacement solution. The external forces and moments are given information. The position vectors from the link center of mass to the joint locations are specified link properties as are the link inertia and mass. Placing all of the known terms of Equations (6.21) and (6.22) on their right-hand sides yields

\[
C_{1u}^{\dot{u}} f_u - C_{1,u+1}^{\dot{u}+1} f_{u+1} = m_1 \ddot{A}_G - \sum f_{ext}^{\dot{f}_{ext}} \quad (6.23)
\]

and

\[
\begin{align*}
\begin{pmatrix}
C_{1u}^{\dot{u}} \vec{m}_u - C_{1,u+1}^{\ddot{m}_{u+1} + u+1} + C_{1u}(u \vec{r}_{u/G} \times u \vec{f}_u) - C_{1,u+1}(u+1 \vec{r}_{u+1/G} \times u+1 \vec{f}_{u+1}) + \\
\sum \dot{m}_{ext} + \sum \vec{r} \times \dot{f}_{ext}
\end{pmatrix}
\end{align*}
\]

\[
= \begin{pmatrix}
I_{G1}^{\dot{a}} + \omega \times I_{G1}^{\dot{a}} \\
\sum \dot{m}_{ext} + \sum \vec{r} \times \dot{f}_{ext}
\end{pmatrix}. \quad (6.24)
\]
Equations (6.23) and (6.24) represent six linear equations in twelve unknowns. These unknowns consist of the components of the joint bearing forces $u\tilde{f}_u$, $v\tilde{m}_u$, $u+1\tilde{f}_{u+1}$, and $u+1\tilde{m}_{u+1}$.

It is easily verified that

$$C(\tilde{r} \times \tilde{f}) = CR\tilde{f}$$

(6.25)

where

$$\tilde{r} = (r^x, r^y, r^z)$$

(6.26)

$$R = \begin{bmatrix} 0 & -r^y & r^y \\ r^y & 0 & -r^z \\ -r^y & r^z & 0 \end{bmatrix}$$

(6.27)

Hence, Equations (6.23) and (6.24) may be compactly written as

$$\begin{bmatrix} C_{1u} & 0 & -C_{1,u+1} & 0 \\ C_{1u} & R_u & -C_{1,u+1}R_{u+1} & -C_{1,u+1} \end{bmatrix} \begin{bmatrix} u\tilde{f}_u \\ v\tilde{m}_u \\ u+1\tilde{f}_{u+1} \\ u+1\tilde{m}_{u+1} \end{bmatrix} = \begin{bmatrix} 1\tilde{F}_{\text{inertia}} - 1\tilde{F}_{\text{ext}} \\ 1\tilde{M}_{\text{inertia}} - 1\tilde{M}_{\text{ext}} \end{bmatrix}$$

(6.28)

where

$$1\tilde{F}_{\text{inertia}} = m\tilde{\alpha}$$

(6.29)

$$1\tilde{F}_{\text{ext}} = \sum 1\tilde{f}_{\text{ext}}$$

(6.30)

$$1\tilde{M}_{\text{inertia}} = I_{G1}\tilde{\alpha} + \tilde{\omega} \times I_{G1}\tilde{\omega}$$

(6.31)

$$1\tilde{M}_{\text{ext}} = \sum 1\tilde{m}_{\text{ext}} + \sum 1\tilde{r} \times 1\tilde{f}_{\text{ext}}.$$  

(6.32)

Equations (6.28) may be abbreviated further by writing

$$[Z_u - Z_{u+1}] \begin{bmatrix} f\tilde{m}_u \\ f\tilde{m}_{u+1} \end{bmatrix} = F\tilde{M}_u$$

(6.33)
where

\[
Z_u = \begin{bmatrix}
C_{1u} & 0 \\
C_{1u}R_u & C_{1u}
\end{bmatrix} \tag{6.34}
\]

\[
Z_{u+1} = \begin{bmatrix}
C_{1,u+1} & 0 \\
C_{1,u+1}R_{u+1} & C_{1,u+1}
\end{bmatrix} \tag{6.35}
\]

\[
f\overset{\rightarrow}{m}_u = \begin{bmatrix}
\overset{\rightarrow}{f}_u \\
\overset{\rightarrow}{m}_u
\end{bmatrix} \tag{6.36}
\]

\[
f\overset{\rightarrow}{m}_{u+1} = \begin{bmatrix}
\overset{\rightarrow}{f}_{u+1} \\
\overset{\rightarrow}{m}_{u+1}
\end{bmatrix} \tag{6.37}
\]

\[
F\overset{\rightarrow}{M}_u = \begin{bmatrix}
\overset{\rightarrow}{F}_{\text{inertia}} - \overset{\rightarrow}{F}_{\text{ext}} \\
\overset{\rightarrow}{M}_{\text{inertia}} - \overset{\rightarrow}{M}_{\text{ext}}
\end{bmatrix} \tag{6.38}
\]

Note, \(f\overset{\rightarrow}{m}_u\) can be described as the bearing force/moment applied by link \(u-1\) to link \(u\) (resolved in the \(u\) system of link \(u\)). Hence, \(f\overset{\rightarrow}{m}_2\) represents the force/moment applied by link 1 on link 2. By the way, \(f\overset{\rightarrow}{m}_1\) implies the force/moment applied by link \(n\) to link 1 where \(n\) is the number of links in the mechanism. Additionally, \(F\overset{\rightarrow}{M}_u\) represents the effect of the inertia and external forces/moments acting on link \(u\). Their components are resolved into the inertial system.

Useful dynamic force equilibrium equations may be written for every link in a mechanism except for the ground link (the inertial frame). Using the abbreviated form of Newton's Second Law shown in Equation (6.33), the following system of equations is obtained:

\[
\begin{bmatrix}
Z_2 & -Z_3 \\
\vdots & \ddots \\
Z_n & -Z_1
\end{bmatrix}
\begin{bmatrix}
f\overset{\rightarrow}{m}_2 \\
f\overset{\rightarrow}{m}_3 \\
\vdots \\
f\overset{\rightarrow}{m}_n
\end{bmatrix} = \begin{bmatrix}
F\overset{\rightarrow}{M}_2 \\
F\overset{\rightarrow}{M}_3 \\
\vdots \\
F\overset{\rightarrow}{M}_n
\end{bmatrix} \tag{6.39}
\]
The unknowns in this system of equations are the joint bearing forces, \( \mathbf{f}_m \). The system "Z" matrix is \( 6(n - 1) \times 6n \), the system "\( \mathbf{f}_m \)" vector is \( 6n \times 1 \), and the system "\( \mathbf{F}_M \)" vector is \( 6(n - 1) \times 1 \). Hence, the force equilibrium equations provide \( 6(n - 1) \) linear equations in the \( 6n \) unknown joint bearing forces. Obviously, six more independent equations are (or at least appear to be) needed in order to solve for the joint forces.

### 6.2 Joint Force Constraint Equations

Given a joint described by \( n_f \) degrees-of-freedom, it is possible to write \( n_f \) joint bearing force equations related to those degrees-of-freedom. For instance, a typical ideal model for the revolute joint has one degree-of-freedom in the form of a rotation about the revolute's \( z \)-axis. If the assumption is made that this is a frictionless joint, then the revolute can transfer no torque about its rotation axis. Hence, for an ideal, frictionless revolute joint,

\[
\mathbf{m}_{\text{revolute}} = 0 \tag{6.40}
\]

where \( \mathbf{m}_{\text{revolute}} \) is the moment transferred by the revolute about its \( z \)-axis.

Note that the joint bearing forces/moments, as they are used in the dynamic force equilibrium equations, are always resolved in a link's \( u \) system. The joint force constraint equations are also written in terms of the \( u \) systems. Given two links, \( i - 1 \) and \( i \), connected by a joint, the corresponding joint force constraint equation(s) provide information about the forces applied on link \( i \) by link \( i - 1 \). Convenience of description dictates these forces should be resolved in system \( u \) of link \( i \). Hence, the joint force constraint equations describe some of the components
of $u_i \ddot{f}_i$ and/or $u_i \ddot{m}_i$ (these are written implicitly as $f_i$ and/or $m_i$ in the dynamic equilibrium equations).

The forces obtained from the dynamic force analysis are caused by specified kinematic inputs. Based on the displacement, velocity, and acceleration of the input joint degree-of-freedom, certain dynamic forces are experienced by the mechanism. These dynamic forces (i.e., the joint reactions) are the unknowns of the force solution. However, an additional unknown appears in the force analysis that might be called the kinetic degree-of-freedom. This kinetic degree of freedom simply represents the force or moment that must be applied along or about the kinematic input degree-of-freedom to produce the specified kinematic inputs (displacement, velocity, and acceleration). This dissertation assumes the kinetic input to a mechanism always corresponds to the kinematic input. Hence, if the kinematic input to a mechanism is through the rotation angle of a revolute joint, the kinetic input will be a moment applied about the revolute axis.

The kinetic input to a mechanism must be added to the joint force constraint equations corresponding to the mechanism's input joint. Again considering a revolute joint, if it were a mechanism's kinematic input degree-of-freedom, the joint force constraint equation is modified to

$$m_{revolute} = m_{input}$$

(6.41)

or

$$m_{revolute} - m_{input} = 0.$$  

(6.42)

Note, since the kinetic input is a new unknown that does not appear in the dynamic force equilibrium system of equations, there are really $6n + 1$ unknowns (for a one
degree-of-freedom mechanism). There are the $6n$ unknown components of the joint bearing forces and moments (i.e., the $\mathbf{f}_i$ and $\mathbf{m}_i$), and there is the unknown kinetic input force/moment. Since, $6n + 1$ unknowns exist, and only $6(n - 1)$ equations are provided by dynamic force equilibrium, the joint force constraints must provide seven additional independent equations if the system is to be uniquely solvable. This is a general statement for all single degree-of-freedom mechanisms subjected to a *three-dimensional* force analysis.

A mechanism made up of binary links will have an equal number of links and joints. Using this information, Grüber's equation (for a spatial mechanism) indicates there must be seven joint degrees-of-freedom in order for the mechanism to have a mobility of one. If there are seven degrees-of-freedom, there will be seven joint force constraint equations available. Hence, a one degree-of-freedom spatial mechanism that obeys Grüber's equation will have the seven additional equations needed to solve for all of the joint reactions and the input force. Many single loop mechanisms (all of the planar ones, for instance) do not have seven joint degrees-of-freedom. The deficiency in the resulting system of equations (i.e., fewer equations than unknowns) can be dealt with using singular value decomposition.

### 6.2.1 Joint Force Constraint Equations: Frictionless

This section describes the joint constraint equations for various frictionless joints. The bearing reaction at a joint is made up of a force vector and a moment vector. These vectors will have components $(f^x, f^y, f^z)$ and $(m^x, m^y, m^z)$, respectively, of which only some will be affected by the joint constraint(s).
**Revolute Joint**  The Revolute Joint provides one constraint equation. For a frictionless joint, no moment can be transmitted about the revolute's axis of rotation. Therefore,

\[ m^* = 0 \quad (6.43) \]

**Prismatic Joint**  The Prismatic Joint provides one constraint equation. For a frictionless joint, no force can be transmitted along the prismatic's sliding axis. Therefore,

\[ f^* = 0 \quad (6.44) \]

**Helical Joint**  The Helical Joint provides one constraint equation. For a frictionless joint, the moment about the helical axis is related to the force along that same axis by

\[ m^* - f^* r_i \tan \alpha = 0 \quad \text{or} \quad f^* - \frac{m^*}{r_i \tan \alpha} = 0 \quad (6.45) \]

where \( r_i \) is the pitch radius of the thread, and \( \alpha \) is the helix angle. The above relations are after those of a power screw with no friction. [75]

**Cylindrical Joint**  The Cylindrical Joint provides two constraint equations. For a frictionless joint, no moment can be transmitted along the cylindrical axis, nor can a force be transmitted along this same axis. Therefore,

\[ f^* = 0 \quad (6.46) \]

\[ m^* = 0 \quad (6.47) \]
Planar Joint  The Planar Joint provides three constraint equations. For a frictionless joint, no force can be transmitted in the plane of the joint, nor can any moment be transmitted normal to the plane of the joint. Therefore,

\[ f^x = 0 \]  \hspace{1cm} (6.48)  
\[ f^y = 0 \]  \hspace{1cm} (6.49)  
\[ m^z = 0 \]  \hspace{1cm} (6.50)  

Spherical Joint  The Spherical Joint provides three constraint equations. For a frictionless joint, no moment can be transmitted about any of the pairing element's three axes. Therefore,

\[ f^x = 0 \]  \hspace{1cm} (6.51)  
\[ f^y = 0 \]  \hspace{1cm} (6.52)  
\[ m^z = 0 \]  \hspace{1cm} (6.53)  

Surface Contact Joint  The Contact Joint provides 5 constraint equations. For a frictionless joint, no force can be transmitted in the tangent plane at the point of contact, nor can any moment be transmitted about the floating system's three axes. However, these observations must be "converted" to their equivalent meanings in the \( u \) system of the contact joint.

Recall, a pairing element matrix connecting link \( i \) and \( i - 1 \) is a transformation from system \( u_i \) to system \( x_{i-1} \). For a surface contact joint, the pairing element matrix is constructed from three intermediate transformations relating the surface definition systems and their floating systems at the point of contact:

\[ P_{i-1,i} = T_{\text{defn, float}} T_{\text{float, float}} T_{\text{float, defn}} \]  \hspace{1cm} (6.54)
\[ = T_{x_{i-1},\text{float}_{i-1}} T_{\text{float}_{i-1},\text{float}} T_{\text{float}_{i-1},u} \quad (6.55) \]

For ease of description, let systems \( \text{float}_{i-1} \) and \( \text{float}_i \) be represented by \( A \) and \( B \) respectively. Also, the subscripts will be dropped on systems \( x_{i-1} \) and \( u_i \). Therefore,

\[ P_{i-1,i} = T_{x,A} T_{A,B} T_{B,u} \quad (6.56) \]

In floating system \( B \), for a frictionless joint:

\[ B f^x = 0 \quad (6.57) \]
\[ B f^y = 0 \quad (6.58) \]
\[ B m^x = 0 \quad (6.59) \]
\[ B m^y = 0 \quad (6.60) \]
\[ B m^z = 0. \quad (6.61) \]

These constraints may be converted to constraints in system \( u \) by placing an equivalent load at the origin of \( u \) that is resolved in \( u \) coordinates. An equivalent force and moment state is developed as follows:

\[ B \vec{f}_O = B \vec{f}_{OB} \quad (6.62) \]
\[ B \vec{m}_O = B \vec{r}_{OB/O} \times B \vec{f}_{OB} + B \vec{m}_{OB} \quad (6.63) \]

where \( O_u \) and \( O_B \) are the origins of systems \( u \) and \( B \) respectively. From Equations (6.57)-(6.61):

\[ B \vec{f}_{OB} = \begin{cases} 0 \\ 0 \\ B f^z = ? \end{cases} \quad B \vec{m}_{OB} = \begin{cases} 0 \\ 0 \end{cases} \quad (6.64) \]

Equations (6.62) and (6.63) may then be written as

\[ B \vec{f}_O = \begin{cases} 0 \\ 0 \\ B f^z = ? \end{cases} \]  \( (6.65) \)
If $C_{B,u}$ represents the change of basis transformation from system $u$ to $B$, Equation (6.65) may be written as

$$C_{B,u}^u f_{O_u} = \begin{pmatrix} 0 \\ 0 \\ B f_s = ? \end{pmatrix}$$

(6.67)

The change of basis transformation $C_{B,u}$ is obtained from the rotation partition of $T_{B,u}$ (which is the “floating-to-definition” transformation, $T_{float,defn}$, for the second surface of the contact pair). Therefore, two joint force constraint equations in system $u$ are given by

$$t_{B,u}^{1,1} u f_{O_u} + t_{B,u}^{1,2} u f_{O_u} + t_{B,u}^{1,3} u f_{O_u} = 0$$

(6.68)

$$t_{B,u}^{2,1} u f_{O_u} + t_{B,u}^{2,2} u f_{O_u} + t_{B,u}^{2,3} u f_{O_u} = 0$$

(6.69)

Now consider Equation (6.66) but resolved in system $u$ coordinates

$$u \vec{m}_{O_u} = u \vec{r}_{O_B/O_u} \times u \vec{f}_{O_B}$$

(6.70)

If Equation (6.62) is resolved in $u$ coordinates then

$$u \vec{f}_{O_u} = u \vec{f}_{O_B}$$

(6.71)

can be used in (6.70) to yield

$$u \vec{m}_{O_u} = u \vec{r}_{O_B/O_u} \times u \vec{f}_{O_u}$$

(6.72)

The displacement partition of $T_{u,B}$ will give $u \vec{r}_{O_B/O_u}$. The matrix $T_{B,u}$ is, therefore, inverted. Representing the displacement partition of $T_{B,u}^{-1}$ as $(r^x, r^y, r^z)$, Equation (6.72) may be written as

$$u \vec{m}_{O_u} - (r^x, r^y, r^z) \times u \vec{f}_{O_u} = 0$$

(6.73)
or
\[
\begin{align*}
\begin{bmatrix} u_m^0 \cr u_m^0 \cr u_m^0 \end{bmatrix} - \begin{bmatrix} r^v u f_{Dv} - r^z u f_{Dz} \\
 r^z u f_{Dz} - r^x u f_{Dx} \\
 r^x u f_{Dx} - r^v u f_{Dv} \end{bmatrix} &= \begin{bmatrix} 0 \\
 0 \\
 0 \end{bmatrix}.
\end{align*}
\tag{6.74}
\]

Using Equations (6.68), (6.69), and (6.74), the five constraint equations resolved in the \( u \) system of a frictionless surface contact joint may be presented as

\[
\begin{align*}
t_{B,u}^{1,1} f^x + t_{B,u}^{1,2} f^y + t_{B,u}^{1,3} f^z &= 0 \tag{6.75} \\
t_{B,u}^{2,1} f^x + t_{B,u}^{2,2} f^y + t_{B,u}^{2,3} f^z &= 0 \tag{6.76} \\
 m^x - r^y f^x + r^z f^y &= 0 \tag{6.77} \\
 m^y - r^x f^y + r^z f^x &= 0 \tag{6.78} \\
 m^z - r^x f^z + r^y f^x &= 0 \tag{6.79}
\end{align*}
\]

where \( t_{B,u}^{ij} \) are the \( i, j \) elements of \( T_{B,u} \) and \( (r^x, r^y, r^z) \) represent the displacement partition elements of \( T_{B,u}^{-1} \).

### 6.2.2 Joint Force Constraint Equations: Coulomb Friction

A common friction model used in force analyses is called Coulomb or "dry" friction.[89, 70, 64] This model states the friction force between two sliding bodies is proportional to the normal force between the surfaces in contact. The proportionality constant is called the kinetic coefficient of friction. A Coulomb friction force relation may be written as

\[
f_f = \mu f_n \tag{6.80}
\]

where \( f_f \) is the friction force, \( \mu \) is the kinetic coefficient of friction, and \( f_n \) is the normal force between the sliding surfaces. This force will be in a direction that

The joint force constraint equations with Coulomb friction are given below. As might be expected, the constraint equations are more involved when dealing with friction in the joints. The constraint equations will be seen as nonlinear in the joint reaction forces.

**Revolute Joint** The revolute joint with Coulomb friction provides one constraint equation in the form of a friction moment about the revolute axis. The revolute's pin is assumed to be slightly smaller than the bearing with which it mates. As Figure 36 depicts (in an exaggerated fashion), the infinitesimal clearance allows a normal force, \( N \), to be established at a point of contact. The contact point location will be a function of the loading.

Since, \( f_f = \mu f_n \) due to Coulomb friction, the resultant force (in the plane normal to the joint axis) is inclined by the angle \( \phi \) where

\[
\tan \phi = \frac{f_f}{f_n} = \mu. \tag{6.81}
\]

A moment is now exerted by the resultant force because it does not pass through the joint center. The magnitude of the moment is given by [89]

\[
m_f = r_f F \tag{6.82}
\]

where

\[
r_f = r \sin \phi = \frac{r \mu}{\sqrt{1 + \mu^2}}. \tag{6.83}
\]

The circle of radius \( r_f \) is called the friction circle. The dimension \( r \) represents the pin diameter.
The resultant force can be inclined by $\phi$ on either side of the normal. The correct side is the one creating a moment such that it opposes the relative rotation of the links. Since the procedures developed within this dissertation resolve all forces into the $u$ systems of the links, the friction moment as applied to link $i$ by link $i-1$ must be determined. With this in mind, if the time rate of change of the revolute joint variable is positive, the friction moment as seen by link $i$ will be negative. The normal force consists of the $x$ and $y$ bearing reactions. Hence, the friction moment caused the normal forces becomes

$$m_{normal\ forces}^{*} = -\text{sign}(\dot{q}) \frac{r\mu}{\sqrt{1 + \mu^2}} \sqrt{(f_x)^2 + (f_y)^2} \quad (6.84)$$
where

\[ \text{sign}(\cdot) = \begin{cases} -1 & \text{if its argument is negative}, \\ +1 & \text{if its argument is positive}, \\ 0 & \text{if its argument is zero} \end{cases} \]

\[ \dot{q} = \text{first time derivative of revolute joint variable} \]

\[ r = \text{the revolute pin diameter} \]

Due to the fact that this joint will be used with spatial mechanisms, an additional contribution to the friction moment is necessary. An axial force will, in general, also create a friction moment about the revolute axis. The revolute must have some type of "thrust" surface or bearing to prevent axial motion in the joint. If the thrust area is represented by two circular surfaces rotating upon each other, the friction moment may be approximated by

\[ m^2_{\text{thrust force}} = -\text{sign}(\dot{q}) r_c \mu f^z \quad (6.85) \]

where \( r_c \) is the effective "collar" radius (usually taken to be the midpoint radius) of the thrust surfaces. This expression is borrowed from that of a thrust bearing in a power screw[75].

The complete joint force constraint equation may now be written as

\[ m^2 = m^2_{\text{normal forces}} + m^2_{\text{thrust force}} \quad (6.86) \]

\[ = -\text{sign}(\dot{q}) \frac{r \mu_1}{1 + \mu_1^2} \sqrt{(f^x)^2 + (f^y)^2} - \text{sign}(\dot{q}) r_c \mu_2 f^z \quad (6.87) \]

or

\[ \text{sign}(\dot{q}) \frac{r \mu_1}{1 + \mu_1^2} \sqrt{(f^x)^2 + (f^y)^2} + \text{sign}(\dot{q}) r_c \mu_2 f^z + m^2 = 0. \quad (6.88) \]

Note two different coefficients of friction are used. One represents the pin-journal friction and the other represents the thrust surface friction. Two distinct coefficients allow different materials or conditions to be modeled.
Prismatic Joint  The prismatic joint provides one joint force constraint equation in the form of a friction force along the sliding axis. In the joint’s coordinate system, the forces normal to the sliding are the $x$ and $y$ forces. The sliding friction force is along the $z$-axis. The sliding friction force will be directed so as to oppose the relative sliding velocity. Hence, using the Coulomb friction model,

$$f^z = -\text{sign}(q) \mu \sqrt{(f_x)^2 + (f_y)^2}$$

(6.89)

or

$$\text{sign}(q) \mu \sqrt{(f_x)^2 + (f_y)^2} + f^z = 0$$

(6.90)

Helical Joint  The helical joint provides one joint force equation relating the axial friction moment and the axial friction force. The relations used are taken directly from the analysis of a power screw.[75] Considering just axial loading, the friction torque is

$$m^z = f^z \left( r_i \frac{\tan \alpha + \frac{\mu_1}{\cos \theta_n}}{1 - \frac{\mu_1}{\cos \theta_n} \tan \alpha} + r_c \mu_2 \right)$$

(6.91)

where

- $r_i =$ pitch radius of the thread
- $\alpha =$ helix angle of thread calculated at the pitch radius
- $\theta_n =$ related to thread angle, $\theta$, by $\tan \theta_n = \tan \theta \cos \alpha$
- $\mu_1 =$ coefficient of thread friction
- $r_c =$ midpoint radius of thrust collar
- $\mu_2 =$ coefficient of thrust collar friction.

Note that a radial load on this joint would increase the normal force on the threads thereby increasing the friction moment. An attempt will not be made here to develop an expression involving the contribution to the friction moment from radial loading (i.e., $f^x$ and $f^y$). However, its possible effect should be considered.
Cylindrical Joint  The cylindrical joint provides two joint force equation in terms of an axial friction moment and an axial sliding force. The joint constraints may be borrowed from the revolute and prismatic constraints directly. However, the cylindrical joint has no axial force resistance. Thus, the contribution to the friction moment caused by axial force in the revolute joint's moment constraint is omitted. The Coulomb friction joint constraints equations are then,

\[
\text{sign}(\dot{q}) \frac{r \mu}{\sqrt{1 + \mu^2}} \sqrt{(f_x)^2 + (f_y)^2 + m^2} = 0
\]

\[
\text{sign}(\dot{q}) \mu \sqrt{(f_x)^2 + (f_y)^2 + f^z} = 0
\]

Planar Joint  The planar joint provides three joint force constraint equations in terms of two sliding force components and a moment. The friction force will align itself in a direction opposite to the relative sliding velocity of the joint. The planar joint has two translational degrees-of-freedom in its plane of motion along the \(x\) and \(y\) axes. The vectorial combination of their first derivatives represents the total relative velocity between the mating planar surfaces. Hence, the sliding friction force will be aligned with this total relative velocity vector but in the opposite direction. The magnitude of the friction force will be \(\mu f^z\) where the \(z\)-axis is perpendicular to the planar surfaces. Breaking the friction force into \(x\) and \(y\) components yields

\[
f_x = -\mu f^z \frac{\dot{q}_1}{\sqrt{(\dot{q}_1)^2 + (\dot{q}_2)^2}}
\]

\[
f_y = -\mu f^z \frac{\dot{q}_2}{\sqrt{(\dot{q}_1)^2 + (\dot{q}_2)^2}}
\]

where \(\dot{q}_1\), and \(\dot{q}_2\) are the time rates of change of the joint translational degrees-of-freedom in the \(x\) and \(y\) directions.
A friction moment will also be experienced by the joint due to relative rotation between the planar surfaces in contact. This friction moment will be a function of $f^z$. A very simplified expression can be used in which one of the planar surfaces is assumed to be a circular disk. When it rotates relative to the other joint surface, a friction moment will be created much in the same way as would a thrust bearing. Using an effective collar radius, the friction moment is given as

$$m^z = -\text{sign}(\dot{q}_3) r_c \mu f^z$$  \hspace{1cm} (6.96)

where $\dot{q}_3$ is the first time derivative of the planar joint's rotational degree-of-freedom, and $r_c$ is an effective collar radius for the planes in contact. The collar radius is an approximation supplied by the analyst.

The constraint equations may be written as

$$f^x + \mu f^z \frac{\dot{q}_1}{\sqrt{(\dot{q}_1)^2 + (\dot{q}_2)^2}} = 0$$  \hspace{1cm} (6.97)

$$f^y + \mu f^z \frac{\dot{q}_2}{\sqrt{(\dot{q}_1)^2 + (\dot{q}_2)^2}} = 0$$  \hspace{1cm} (6.98)

$$\text{sign}(\dot{q}_3) r_c \mu f^z + m^z = 0$$  \hspace{1cm} (6.99)

**Spherical Joint** The spherical joint provides three constraint equations in terms of friction moments about the three axes of the joint's $u$ system. The friction model of the spherical joint considers the joint as a sort of instantaneous revolute joint in order to produce meaningful friction information.

Consider two links $i-1$ and $i$ joined by a spherical joint. Figure 37 shows the bearing forces acting on the "ball" of the spherical joint. At any instant in time, the motion permitted by the joint will appear to be a pure rotation about the instantaneous screw axis. The instantaneous screw can be determined from the
Figure 37: Bearing forces acting on the "ball" of a spherical joint.
relative angular velocities of the two links. At any instant, the friction force should be tangent to the spherical surface in a plane normal to the screw axis (i.e., it is tangent to a circle of constant latitude as link \( i \) rotates relative to link \( i - 1 \) about the screw axis). Hence, at any instant, the joint looks something like a revolute with an axis aligned with the screw axis.

The effective pin radius will be a function of the line of action of the joint bearing force. If the force is in the plane of the "equator," the effective pin radius will be the sphere radius. As the force is applied nearer the "poles" of the sphere, the radius will tend toward zero. The goal of the following discussion is then to determine the instant screw axis of link \( i \) relative to link \( i - 1 \), determine the line of action of the bearing force, calculate the friction moment about the screw axis, and then resolve this moment in the \( u \) system of the joint.

The spherical joint friction model will be developed for the spherical pair matrix that rotates system \( (xyz)_{i-1} \) into system \( (uvw)_{i} \) using (many other implementations of a spherical joint exist)

- A rotation of system \( (xyz)_{i-1} \) by \( \alpha \) about \( x_{i-1} \) to produce system \( (xyz)_{a} \).
- A rotation of system \( (xyz)_{a} \) by \( \beta \) about \( z_{a} \) to produce system \( (xyz)_{b} \).
- A rotation of system \( (xyz)_{b} \) by \( \gamma \) about \( x_{b} \) to produce system \( (uvw)_{i} \).

The rotation angles \( \alpha, \beta, \) and \( \gamma \) are the joint variables for this joint. The velocity solution for a given position will yield \( \dot{\alpha}, \dot{\beta}, \) and \( \dot{\gamma} \). These may be used to determine the angular velocity of system \( (uvw)_{i} \) relative to system \( (xyz)_{i-1} \). In equation form, the angular velocity of \( (uvw)_{i} \) relative to \( (xyz)_{i-1} \) resolved in system \( (uvw)_{i} \),
coordinates may be expressed as

\[ u \mathbf{\hat{\omega}}_{u/a} = u \mathbf{\hat{\omega}}_{u/b} + u \mathbf{\hat{\omega}}_{b/a} + u \mathbf{\hat{\omega}}_{a/s} \]  

\[ = \begin{bmatrix} \dot{\gamma} \\ 0 \\ \hat{\beta} \end{bmatrix} + C_{u,b} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + C_{u,a} \begin{bmatrix} \dot{\alpha} \\ 0 \\ 0 \end{bmatrix} \]  

\[ (6.100) \]

\[ (6.101) \]

where \( C_{k,l} \) represents a change of basis from system \( l \) to \( k \). The change of basis transformations may be obtained using common rotation matrices. Note the use of negative rotation angles. The rotations are being used to "go up the chain" of links where all other uses in this dissertation have been used to "go down the chain" of links.

\[ C_{u,b} = R(-\gamma)_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & \sin \gamma \\ 0 & -\sin \gamma & \cos \gamma \end{bmatrix} \]  

\[ (6.102) \]

\[ C_{u,a} = R(-\gamma)_x R(-\beta)_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & \sin \gamma \\ 0 & -\sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\[ (6.103) \]

The angular velocity may then be expressed as

\[ u \mathbf{\hat{\omega}}_{u/a} = \begin{bmatrix} \dot{\omega}^x \\ \dot{\omega}^y \\ \dot{\omega}^z \end{bmatrix} = \begin{bmatrix} \dot{\gamma} + \dot{\alpha} \cos \beta \\ \beta \sin \gamma - \dot{\alpha} \cos \gamma \sin \beta \\ \beta \cos \gamma + \dot{\alpha} \sin \gamma \sin \beta \end{bmatrix} \]

\[ (6.104) \]

The displacement and velocity analyses yield values for \( \alpha, \beta, \gamma, \dot{\alpha}, \dot{\beta}, \text{ and } \dot{\gamma} \).

The instantaneous screw axis is along the angular velocity vector. Hence, the direction cosines for the screw axis are given by

\[ \hat{s} = \begin{bmatrix} \lambda^x \\ \lambda^y \\ \lambda^z \end{bmatrix} = \frac{1}{\sqrt{(\omega^x)^2 + (\omega^y)^2 + (\omega^z)^2}} \begin{bmatrix} \omega^x \\ \omega^y \\ \omega^z \end{bmatrix} \]

\[ (6.105) \]

It is about this screw axis that, at this instant, the spherical joint behaves as a revolute. This screw axis is resolved into system \((uvw)_i\).
It will be convenient to form a coordinate system with its z-axis along the screw axis. This will be called system $S$. Hence, the $z_S$ axis will have direction cosines $(\lambda^x, \lambda^y, \lambda^z)$. The $x_S$ axis will be formed by crossing $z_S$ into $z_u$, and the $y_S$ axis is obtained by crossing $z_S$ into $x_S$:

$$
\hat{z}_S = (\lambda^x, \lambda^y, \lambda^z) \quad (6.106)
$$

$$
\hat{x}_S = \frac{\hat{z}_S \times \hat{z}_u}{|\hat{z}_S \times \hat{z}_u|} \quad (6.107)
$$

$$
= \frac{(\lambda^x, \lambda^y, \lambda^z) \times (0, 0, 1)}{|(\lambda^x, \lambda^y, \lambda^z) \times (0, 0, 1)|} \quad (6.108)
$$

$$
= \frac{(\lambda^y, -\lambda^x, 0)}{\sqrt{(\lambda^y)^2 + (\lambda^x)^2}} \quad (6.109)
$$

$$
\hat{y}_S = \hat{z}_S \times \hat{x}_S \quad (6.110)
$$

$$
= \left[ \frac{\lambda^x \lambda^x, \lambda^y \lambda^z, -(\lambda^y)^2 + (\lambda^x)^2} {\sqrt{(\lambda^y)^2 + (\lambda^x)^2}} \right] \quad (6.111)
$$

A change of basis transformation from system $u$ to system $S$ may now be written as

$$
C_{S,u} = \frac{1}{\sqrt{(\lambda^x)^2 + (\lambda^y)^2}} \begin{bmatrix}
\lambda^y & -\lambda^x & 0 \\
\lambda^x \lambda^x & \lambda^y \lambda^z & -(\lambda^y)^2 + (\lambda^x)^2 \\
\lambda^x \sqrt{(\lambda^x)^2 + (\lambda^y)^2} & \lambda^y \sqrt{(\lambda^x)^2 + (\lambda^y)^2} & \lambda^x \sqrt{(\lambda^x)^2 + (\lambda^y)^2}
\end{bmatrix} \quad (6.112)
$$

If a bearing force state, $f^x, f^y, f^z$, relative to system $u$ of link $i$ is given for this joint, it will be useful to break it down into its surface normal and frictional components. This is most easily done by resolving the given force state into the screw system using $C_{S,u}$ from above:

$$
S_{\vec{F}} = C_{S,u} \, u_{\vec{F}} \quad (6.113)
$$
Figure 38 shows a slice of the spherical joint in terms of the screw-based system looking down the instant screw axis. The force, \( \vec{F} \) has been resolved into this screw system, and its components are shown in the upper right-hand corner of the figure. The \( z \) component, \( Sf^z \), is normal to the plane of the page. The vectors represented by double lines, \( \vec{\Rightarrow} \), have no \( z \) component in the screw aligned system, and therefore, are seen in true length in the figure.

Notice the bearing force, \( \vec{F} \), has been conceptually resolved into a friction and normal component (no actual values have been calculated yet). The components are by definition perpendicular to each other. Note also, it is possible to express \( \vec{F} \) two valid ways in terms of normal and frictional components. Views \( A_1 \) and \( B_1 \) of Figure 38 demonstrate the two possibilities. The one producing a friction component opposing the angular velocity is the desired configuration.

The following equations are useful

\[
\begin{align*}
  f_f &= \mu f_n \\
  \vec{F} &= \vec{f}_n + \vec{f}_f = \vec{f}^x + \vec{f}^y + \vec{f}^z \\
  f_n &= \sqrt{f^2 - f_f^2}
\end{align*}
\]

They yield a relation between the magnitudes of the bearing force and the friction force:

\[
f_f = \frac{\mu F}{\sqrt{1 + \mu^2}}
\]

where \( F = \sqrt{(f^x)^2 + (f^y)^2 + (f^z)^2} \). The components may be in any system.

From Views \( A_1 \) and \( B_1 \) of Figure 38, it is evident \( Sf^z \) cannot contribute to \( f_f \).
Note: "Double lined" vectors have no 'Z' component in the screw axis aligned system.

Figure 38: A view looking down the instant screw axis of a sphere sectioned at the plane where the bearing loads are applied.
since they are perpendicular to each other. Hence, Views $A_3$ and $B_3$ show how the friction force is related graphically to $Sf^x$ and $Sf^y$. The direction cosines of the friction force in the screw system are $(\cos \theta, \sin \theta, 0)$, where $\theta$ is measured from the screw system's $x$-axis. The contributions from $Sf^x$ and $Sf^y$ to the friction force are obtained by projecting them (using the dot product) into the direction of $f_f$. Therefore, the magnitude of the friction force is

$$f_f = |Sf^x \cos \theta + Sf^y \sin \theta|.$$  

(6.119)

Equating (6.118) and (6.119) yields

$$Sf^x \cos \theta + Sf^y \sin \theta = \frac{\mu F}{\sqrt{1 + \mu^2}}.$$  

(6.120)

Using the following half angle identities

$$\cos \theta = \frac{1 - \tan^2 \frac{\theta}{2}}{1 + \tan^2 \frac{\theta}{2}} \quad \text{and} \quad \sin \theta = \frac{2 \tan \frac{\theta}{2}}{1 + \tan^2 \frac{\theta}{2}},$$  

(6.121)

the direction of the friction force is given by

$$\theta = 2 \tan^{-1} \left( \frac{Sf^y \pm \sqrt{(Sf^x)^2 + (Sf^y)^2 - \frac{\mu^2 F^2}{1 + \mu^2}}}{\frac{\mu F}{\sqrt{1 + \mu^2}} + Sf^x} \right).$$  

(6.122)

The friction force in system $S$ is now obtained using

$$Sf_f = |Sf^x \cos \theta + Sf^y \sin \theta|(\cos \theta, \sin \theta, 0).$$  

(6.123)

As expected, there are two possible solutions.

For the scenario depicted in Figure 38, the solution shown in View $B_2$ is the correct one. The friction force obviously causes a moment in the opposite direction.
of the angular velocity. The correct solution to use for the friction force can be determined by satisfying the following:

\[ S \vec{f}_n \times S \vec{f}_f \cdot \hat{s} > 0 \]  

(6.124)

This may be rewritten as

\[ (S \vec{F} - S \vec{f}_f) \times S \vec{f}_f \cdot \hat{s} > 0 \]  

(6.125)

\[ \left\{ \begin{array}{c}
(S F^x - S f^x_f), (S F^y - S f^y_f), (S F^z - S f^z_f) \\
(S f^x_f, S f^y_f, S f^z_f)
\end{array} \right\} \cdot (0,0,1) > 0 \]  

(6.126)

or finally as

\[ (S F^x - S f^x_f) S f^y_f - (S F^y - S f^y_f) S f^x_f > 0. \]  

(6.127)

The components of \( S \vec{F} \) are given, and the components of \( S \vec{f}_f \) (for both solutions) are calculated using Equation (6.123).

With the correct solution for \( S \vec{f}_f \) (i.e., the one satisfying Equation (6.127)), the friction force is then transformed back into the \( u \) system:

\[ u \vec{f}_f = C_{u,S} S \vec{f}_f \]  

(6.128)

\[ = C_{u,S} S \vec{f}_f. \]  

(6.129)

Now that the friction force is known in system \( u \), the normal force may be calculated using

\[ u \vec{f}_n = u \vec{F} - u \vec{f}_f. \]  

(6.130)

The normal force must go through the point of contact between the surfaces in the joint (assuming infinitesimal clearance). It must also go through the center of the joint. Hence, a vector, with magnitude equal to the sphere radius, in the opposite direction of \( u \vec{f}_n \) will provide a position vector from the screw axis to the point of contact in the joint. This vector is shown as \( \vec{r}_p \) in Figure 37.
Let, the direction cosines of the normal force be given by

\[
(d^x, d^y, d^z) = \frac{(u f_n^x, u f_n^y, u f_n^z)}{\sqrt{(u f_n^x)^2 + (u f_n^y)^2 + (u f_n^z)^2}}.
\]  

(6.131)

Then \(\bar{r}_p\) is given by

\[
\bar{r}_p = r(-d^x, -d^y, -d^z)
\]

(6.132)

where \(r\) is the radius of the sphere.

Finally, the friction moment due to a bearing load \(u\bar{F}\) may be given by

\[
u m_f = \bar{r}_p \times u\bar{F}
\]

(6.133)

or

\[
m^x = r^y_p f^x - r^x_p f^y
\]

(6.134)

\[
m^y = r^z_p f^x - r^x_p f^z
\]

(6.135)

\[
m^z = r^z_p f^y - r^y_p f^z
\]

(6.136)

For the uses within this dissertation, these will be rewritten as

\[
r^y_p f^x - r^x_p f^y - m^x = 0
\]

(6.137)

\[
r^z_p f^x - r^x_p f^z - m^y = 0
\]

(6.138)

\[
r^z_p f^y - r^y_p f^z - m^z = 0
\]

(6.139)

It is seen in the end, the friction moments are simply functions of the bearing force loads and the point of contact established by the loads. The point of contact due to a given force state, is determined essentially by the coefficient of friction (albeit an involved procedure).
Surface Contact Joint The surface contact joint provides five joint force constraint equations. Recall once again the nature of the surface contact joint's pair matrix. It joins system \((xyz)_{i-1}\) of one link to system \((uvw)_{i}\) of the next link using three intermediate transformations. These transformations relate the surface definition systems and their floating systems at the point of contact:

\[
P_{i-1,i} = T_{\text{defn,float}} T_{\text{float,float}} T_{\text{float,defn}}
\]

\[
= T_{x_{i-1},float_{i-1}} T_{float_{i-1},float_{i}} T_{float_{i},u_{i}}.
\]

(6.140)

(6.141)

For ease of description, let systems \(float_{i-1}\) and \(float_{i}\) be represented by \(A\) and \(B\) respectively. Also, the subscripts will be dropped on systems \(x_{i-1}\) and \(u_{i}\). Therefore,

\[
P_{i-1,i} = T_{x,A} T_{A,B} T_{B,u}\]

(6.142)

As with all other joints, the bearing reactions for this joint are expressed in system \(u\) of link \(i\). However, a number of force relationships can be stated about the forces occurring at the point of surface contact. This information can be written in terms of the forces at \(u\). Hence, the constraint equations are developed in terms of loading at system \(u\).

Consider the forces at the point of contact resolved in system \(B\) (the floating system on link \(i\)'s surface). There will be a friction force equal to

\[
B f_f = \mu |B f_s|
\]

(6.143)

where \(B f_s\) is the normal force at the point of contact. The direction of the friction force will be such that it opposes the relative sliding of the surfaces in contact.

The velocity analysis can easily determine the velocity of the contact point on
link $i - 1$ relative to the contact point on link $i$. The displacement partition of

$$T_{1A} = T_{12} T_{13} \cdots T_{i-1, A}$$

or

$$T_{1B} = T_{12} T_{13} \cdots T_{i-1, A} T_{A,B}$$

will yield the position of the coincident contact points relative to ground.\(^1\) The "displacement" partitions of $T_{1A}$ and $T_{1B}$ will yield the different velocities of the contact points relative to ground. The velocity of the contact point, say $a$, on link $i - 1$ relative to the contact point, say $b$, on link $i$ is obtained by

$$v_{a/b}^1 = v_a^1 - v_b^1. \quad (6.144)$$

This may be transformed into system $B$ using

$$B v_{a/b}^1 = C_T^{T_B} v_{a/b}^1 \quad (6.145)$$

where $C_T^{T_B}$ may be obtained by transposing the rotation partition of $T_{1B}$. All of the transformations and their derivatives used above are available from the displacement and velocity analyses.

Due to the nature of surface contact, and the manner in which the floating system $B$ is defined on the surface of link $i$, the relative sliding velocity will only have $x$ and $y$ components when resolved in system $B$. Hence, the direction of the sliding velocity of $a$ relative to $b$ in system $B$ is

$$\theta_v = \tan^{-1} \frac{B v_{a/b}^y}{B v_{a/b}^x}. \quad (6.146)$$

\(^1\)The displacement partitions of $T_{1A}$ and $T_{1B}$ will be the same because $T_{A,B}$ has a null displacement partition. This is proper since the contact points are in fact coincident.
Therefore, the direction of the friction force in system \( B \) is

\[ \theta_f = -\theta_v. \] (6.147)

Consequently, the friction force components in system \( B \) as applied to link \( i \) are

\[ B f^x_f = \mu |B f^x| \cos \theta_f \] (6.148)
\[ B f^y_f = \mu |B f^x| \sin \theta_f. \] (6.149)

Note as with the frictionless case, no moments will exist at the point of surface contact assuming contact truly takes place at a point having no dimension in the mathematical sense. Hence,

\[ B m^x = B m^y = B m^z = 0 \] (6.150)

Equations (6.148)–(6.150) now need to be expressed in terms of forces at the origin of system \( u \) in link \( i \). Given a force state at point \( b \) on link \( i \), an equivalent load at the origin of \( u \) is found using

\[ B \vec{F}_o - B \vec{F}_b = 0 \] (6.151)

\[ B \vec{m}_o + B \vec{r}_{ob} \times B \vec{F}_o - B \vec{m}_b = 0. \] (6.152)

Equation (6.151) can be rewritten as

\[ C_{B,u} \vec{F}_o - B \vec{F}_b = 0. \] (6.153)

This yields

\[ t_{B,u}^{1,1} f^x + t_{B,u}^{1,2} f^y + t_{B,u}^{1,3} f^z = B f^x \] (6.154)
\[ = \mu |B f^x| \cos \theta_f \] (6.155)
\[ t_{B,u}^{2,1} f^x + t_{B,u}^{2,2} f^y + t_{B,u}^{2,3} f^z = B f^y \]
\[ = \mu |B f^z| \sin \theta_f \]  
\[ (6.157) \]
\[ t_{B,u}^{3,1} f^x + t_{B,u}^{3,2} f^y + t_{B,u}^{3,3} f^z = B f^z. \]  
\[ (6.158) \]

Equation 6.158 can be used in (6.155) and (6.158) to yield two force constraint equations solely in terms of forces at \( u \) on link \( i \):

\[ t_{B,u}^{1,1} f^x + t_{B,u}^{1,2} f^y + t_{B,u}^{1,3} f^z - \mu |t_{B,u}^{2,1} f^x + t_{B,u}^{2,2} f^y + t_{B,u}^{2,3} f^z| \cos \theta_f = 0 \]
\[ (6.159) \]
\[ t_{B,u}^{3,1} f^x + t_{B,u}^{3,2} f^y + t_{B,u}^{3,3} f^z - \mu |t_{B,u}^{3,1} f^x + t_{B,u}^{3,2} f^y + t_{B,u}^{3,3} f^z| \sin \theta_f = 0 \]
\[ (6.160) \]

Equations (6.150) and (6.152) yield

\[ B \vec{m}_{O_s} + B \vec{r}_{O_s}/O_B \times B \vec{F}_{O_s} = 0. \]  
\[ (6.161) \]

This may also be written in system \( u \) as

\[ u \vec{m}_{O_s} + u \vec{r}_{O_s}/O_B \times u \vec{F}_{O_s} = 0. \]  
\[ (6.162) \]

An identical form for this equation is

\[ u \vec{m}_{O_s} - u \vec{r}_{O_B}/O_s \times u \vec{F}_{O_s} = 0. \]  
\[ (6.163) \]

where the subscripts were reversed on the position vector in the cross product. This equation is identical to the equivalent moment equation in the frictionless surface joint development (see Equation (6.72)). Hence, following the development for the frictionless case, three more constraint equations are

\[ m^x - r^y f^z + r^z f^y = 0 \]  
\[ (6.164) \]
\[ m^y - r^z f^x + r^x f^z = 0 \]  
\[ (6.165) \]
\[ m^z - r^x f^y + r^y f^x = 0 \]  
\[ (6.166) \]
where \( t^{ij}_{B,u} \) are the \( i,j \) elements of \( T_{B,u} \) and \((r^x, r^y, r^z)\) represent the displacement partition elements of \( T^{-1}_{B,u} \).

### 6.3 Solutions to the Force System of Equations

Sections 6.1 and 6.2 developed equations based on dynamic force equilibrium and joint force constraints, respectively. The goal is to develop enough independent equations so that the joint bearing forces and the mechanism input force can be calculated. These forces are caused by the prescribed input displacement, velocity, and acceleration.

As previously discussed, a closed loop, single degree-of-freedom mechanism with \( n \) links will have \( 6n \) unknown joint bearing forces. The unknown input force increases the unknowns to \( 6n + 1 \). The dynamic force equilibrium equations provide \( 6(n - 1) \) independent equations linear in the unknowns. The joint constraint equations, one for each joint degree-of-freedom in the mechanism, provide additional equations in terms of the unknown joint forces. If the number of constraint equations is equal to seven, a unique force solution is possible. The existence of more than or less than seven joint constraints results in an over or underconstrained force system, respectively.

If joint friction is neglected, the joint constraint equations are linear in the unknown joint forces. Hence, the constraint equations are simply augmented to the dynamic force equilibrium system of equations. A linear solution procedure may then be used. If friction is considered, the constraint equations are nonlinear in the unknown joint forces. Hence, the combined system of equilibrium and constraint equations must be solved with a nonlinear technique. The approaches
for both of these approaches is presented below.

**6.3.1 Solution to Linear Frictionless Force Analysis**

The dynamic equilibrium equations are shown in an abbreviated form in Equation (6.39). The system "Z" matrix is solely a function of link geometry and mechanism configuration (displacement). This set of equations is obviously linear in the unknown joint forces. The frictionless joint force constraint equations may be augmented directly to this system.

If a given joint \( k \) has \( k \) constraint equations, they may be written as

\[
\begin{align*}
\alpha_{11}f_x + \alpha_{12}f_y + \alpha_{13}f_z + \alpha_{14}m_x + \alpha_{15}m_y + \alpha_{16}m_z + c_1 f_{m_{\text{input}}} &= 0 \\
\alpha_{21}f_x + \alpha_{22}f_y + \alpha_{23}f_z + \alpha_{24}m_x + \alpha_{25}m_y + \alpha_{26}m_z + c_2 f_{m_{\text{input}}} &= 0 \\
\vdots & \quad \vdots \\
\alpha_{k1}f_x + \alpha_{k2}f_y + \alpha_{k3}f_z + \alpha_{k4}m_x + \alpha_{k5}m_y + \alpha_{k6}m_z + c_k f_{m_{\text{input}}} &= 0
\end{align*}
\]

or

\[
A \hat{f}_m + \bar{c} f_{m_{\text{input}}} = \vec{0} \tag{6.168}
\]

where

- \( A \) = the \( k \times 6 \) matrix of coefficients for the linear force constraint equations for a given joint
- \( k \) = the number of constraint equations for a given joint
- \( \hat{f}_m \) = the \( 6 \times 1 \) vector of unknown bearing forces
- \( \bar{c} \) = the \( k \times 1 \) vector of coefficients for the input force for each constraint equation; if the degree-of-freedom corresponding to a particular constraint equation, say \( i \), is the input, \( c^i \) will be \(-1\). Otherwise, it is zero.
- \( f_{m_{\text{input}}} \) = the unknown input force or moment acting along or about the input degree-of-freedom.

Equations of the form of (6.168) will exist for each joint in the mechanism. For a given joint \( i \), its system of constraint equations may be written as

\[
A_i \hat{f}_m + \bar{c}_i f_{m_{\text{input}}} = \vec{0} \tag{6.169}
\]
For a revolute joint, the $A$ matrix would be (from Equation (6.43)):

$$A_{\text{revolute}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.170)$$

A cylindrical joint (from Equations (6.46) and (6.47)) would have as its $A$ matrix

$$A_{\text{cylindrical}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.171)$$

Adding the joint force constraint equations to the matrix form of the displacement equations involves both “row” and “column” additions to the system “$Z$” matrix. Specifically, Equation (6.39) will become

$$\begin{bmatrix} Z_2 & -Z_3 & \cdots & -Z_1 & 0 & 0 & \cdots & 0 & 0 \\ Z_3 & -Z_4 & \cdots & 0 & 0 & \cdots & 0 & 0 & \end{bmatrix} \begin{bmatrix} \ddot{\vec{z}}_2 \\ \ddot{\vec{z}}_3 \\ \vdots \\ \ddot{\vec{z}}_n \\ \ddot{\vec{z}}_1 \end{bmatrix} = \begin{bmatrix} \vec{f}_{m_2} \\ \vec{f}_{m_3} \\ \vdots \\ \vec{f}_{m_n} \\ \vec{f}_{m_1} \\ \vdots \\ \vec{f}_{m_{\text{input}}} \end{bmatrix} \begin{bmatrix} F_2^M \\ F_3^M \\ \vdots \\ F_n^M \end{bmatrix} \quad (6.172)$$

Let the left-hand side coefficient matrix be abbreviated as $C_F$.

The system “$Z$” partition is $6(n - 1) \times 6n$, the system “$A$” partition is $k \times 6n$. Also note the last column of the $C_F$ matrix; this partition represents the effect of the input force. This column will contain only one nonzero element corresponding to the input degree-of-freedom. This is based on the assumption only single degree-of-freedom mechanisms are analyzed and the kinetic inputs act along or about the kinematic inputs. The coefficient matrix, $C_F$, is $[6(n - 1) + k] \times [6n + 1]$, the unknown force vector is $(6n + 1) \times 1$, and the known right-hand side vector is $[6(n - 1) + k] \times 1$. Hence, this entire system of equations represents $6(n - 1) + k$.
equations in $6n + 1$ for a one degree-of-freedom mechanism where $n$ is the number of links and $k$ is the number of joint degrees-of-freedom.

As discussed previously, the number of independent equations must equal the number of unknowns to have a unique force solution. Thus, seven joint constraint equations must be available for any mechanism analyzed in a three-dimensional sense:

$$6(n - 1) + k = 6n + 1$$

$$6n - 6 + k = 6n + 1$$

$$k = 7$$

Since planar mechanisms will have fewer than seven joint constraint equations, and mechanisms with idle degrees-of-freedom will have more than seven joint constraint equations, the force system of equations may not always be square. Hence, a straightforward solution may not be available using the standard linear equation solvers. This problem can be alleviated using singular value decomposition.

The basic procedure for a SVD force solution is identical to the procedure for the velocity or acceleration analysis as these are linear systems too. This procedure is outlined in Table 9.

6.3.2 Solution to Nonlinear Friction Force Analysis

The friction force analysis is obviously nonlinear due to the joint force constraint equations obtained when Coulomb friction is included. The constraint equations are nonlinear in terms of the joint bearing forces. The solution approach in this case will be similar to that for the nonlinear displacement solution.
Formulation and Solution of Frictionless Force Analysis

1. Evaluate the Z matrices for each of the links according to Equation (6.34). In (6.34), $C_{1u}$ is the rotation partition of $T_{1u}$, and $^uR_u$ is a matrix using the components of the position of the origin of system $u$ relative to the link's center of mass. The layout of $^uR_u$ is shown in (6.27).

2. Evaluate the inertia and external force vector, $F_M$ for each link as per Equation (6.38).

3. Evaluate the A matrices for each joint in the mechanism. The elements of the A matrices come directly from the coefficients in the constraint equation for a given type of joint.

4. Select the kinetic input degree-of-freedom to be solved for by setting the corresponding component in one of the $c$'s to $-1$. All other components of the $c$'s are zero.

5. Assemble the left-hand side matrix, $C_F$, using the Z's, A's, and c's.

6. (a) If there are less than seven joint constraint equations, $C_F$ will have more columns than rows. Make the $C_F$ matrix square by adding $7-k$ rows of zero to it where $k$ equals the number of joint constraint equations. The same number of zeros must also be augmented to the right-hand side vector. If the $k \geq 7$, use $C_F$ as is.

(b) Determine the SVD of $C_F = U W V^T$.

(c) If any $w_j$'s (the diagonal elements of $W$) are equal to zero, the corresponding columns, $j$, of $V$ represent the nullspace basis vectors for the frictionless force solution.

7. After setting the $1/w_j$'s to zero for all $w_j = 0$, calculate the joint bearing forces and the input force:

$$
\begin{bmatrix}
    f_{m_2} \\
    f_{m_3} \\
    \vdots \\
    f_{m_1} \\
    f_{m_{input}}
\end{bmatrix}
= -V \begin{bmatrix}
    \text{diag}(1/w_j)
\end{bmatrix} U^T
\begin{bmatrix}
    \tilde{F}_M_2 \\
    \tilde{F}_M_3 \\
    \vdots \\
    \tilde{F}_M_n \\
    \tilde{\delta}
\end{bmatrix}
$$

(6.173)
If a frictionless force analysis has been performed on a mechanism, it is reasonable for this solution to provide a good starting point for a nonlinear force analysis. Hence, with a good estimate for the friction solution from the frictionless analysis, Newton's method may be reliably used for the friction problem. As outlined in Section 5.3.1, Newton's method linearizes the system of equations about a point assumed to be close to the solution of the nonlinear system. The linearized solution then produces corrections to the estimated nonlinear solution such that the estimate is made closer to the actual nonlinear answer. Obviously, this is an iterative procedure.

In the case of the friction analysis, the equations available are of the same type used for the frictionless analysis. Dynamic force equilibrium and joint force constraint equations are used. The equilibrium equations will be linear in the unknown joint bearing forces whether or not friction is considered. However, the joint force constraint equations become nonlinear in the unknowns as friction is introduced to the joints.

The solution procedure is similar to that used for the Newton's method displacement analysis. The Jacobian of the system of equations to be solved is formed and evaluated at the current estimate to the nonlinear solution. The Jacobian is (conceptually) inverted and then used to multiply the "right-hand side" vector of the system of equations. This produces a correction vector, which when added to the current nonlinear solution estimate, brings the solution closer to the actual answer.

The Jacobian for the friction force analysis will be made up of two portions. One will come from the equilibrium equations, and the other will come from the
joint force constraint equations. Since the force equilibrium equations are linear in the joint forces, the Jacobian for this portion will simply be the system "Z" matrix shown in Equation (6.39).

The portion of the Jacobian due to the nonlinear joint force constraint equations, however, will need to be formed by taking the partial derivatives of the friction force constraint equations. These partial derivatives are then evaluated at the current solution estimate to construct the Jacobian. Let the i-th friction joint force constraint equations be represented by \( g(f) = 0 \). Technically speaking, a given constraint equation can be considered as being a function of all the joint bearing forces in a mechanism. However, most of the terms (all but six at most) will be zero. Hence, the Jacobian will be filled in by the constraint equations similar to the way in which the A matrices filled the left-hand side matrix in the linear solution development (i.e., diagonal sub-matrices).

For a given joint with \( d_f \) degrees-of-freedom, it will have \( d_f \) constraint equations whose contributions to the Jacobian will be rows of the form

\[
J(g)_{ij} = \begin{bmatrix}
0 & 0 & \cdots & \frac{\partial g_i}{\partial f_j^x} & \frac{\partial g_i}{\partial f_j^y} & \frac{\partial g_i}{\partial f_j^z} & \frac{\partial g_i}{\partial m_j^x} & \frac{\partial g_i}{\partial m_j^y} & \frac{\partial g_i}{\partial m_j^z} & \cdots & 0 & 0 & c_i
\end{bmatrix}
\]

(6.174)

where \( i \) goes from 1 to \( d_f \), and \( j \) is the joint containing the degree-of-freedom governed by this constraint. \( c_i \) will be \(-1\) if the kinetic input is applied to the degree-of-freedom represented by this constraint. There are \( 6n + 1 \) terms in this contribution to the Jacobian. Let all of the rows contributed to the Jacobian by a given joint be represented by \( G_j \). Then the \( d_f \) rows contributed by (6.174) may be abbreviated by

\[
\begin{bmatrix}
\vec{0} & \vec{0} & \cdots & G_j & \cdots & \vec{0} & \vec{0} & c_j
\end{bmatrix}
\]

(6.175)
As a result, the entire Jacobian will be given by

\[
J = \begin{bmatrix}
Z_2 & -Z_3 & 0 \\
Z_3 & -Z_4 & 0 \\
& \ddots & \ddots \\
& & Z_n & -Z_1
\end{bmatrix}
\]

(6.176)

The Newton's method procedure then solves the following system for the correction vector to the estimated force solution

\[
J \begin{bmatrix}
\Delta f_{m_2}^* \\
\Delta f_{m_3}^* \\
\vdots \\
\Delta f_{m_n}^* \\
\Delta f_{m_1}^* \\
\Delta f_{m_{input}}^*
\end{bmatrix} = - \begin{bmatrix}
\vec{F}_{0\text{equilibrium}} \\
\vec{F}_{0\text{constraint}}
\end{bmatrix}
\]

(6.177)

where

\[
\vec{F}_{0\text{equilibrium}} = \text{the residual from the force equilibrium equations at the estimated solution; } 6(n - 1) \times 1
\]

\[
\vec{F}_{0\text{constraint}} = \text{the residual from the joint force constraint equations at the estimated solution; } k \times 1
\]

Note the force equilibrium equations are rearranged such that at a correct solution, they evaluate to zero. In other words, Equation (6.39) is rearranged as

\[
\begin{bmatrix}
Z_2 & -Z_3 & 0 \\
Z_3 & -Z_4 & 0 \\
& \ddots & \ddots \\
& & Z_n & -Z_1
\end{bmatrix} \begin{bmatrix}
f_{m_2}^* \\
f_{m_3}^* \\
\vdots \\
f_{m_n}^* \\
f_{m_1}^*
\end{bmatrix} - \begin{bmatrix}
\vec{F}_2 \\
\vec{F}_3 \\
\vdots \\
\vec{F}_n
\end{bmatrix} = \vec{F}_{\text{equilibrium}}
\]

(6.178)
The vector $\mathbf{F_{equilibrium}}$ is simply the result of (6.178) evaluated at the estimated solution. Note, the constraint equations as developed for both the friction and non-friction problems are already written to evaluate to zero when they are satisfied. Hence $\mathbf{F_{constraint}}$ is the result of evaluating the joint force constraint equations at the estimated solution.

As with the displacement analysis, the Jacobian with the friction force analysis is not guaranteed to be invertible. Over and underconstrained mechanisms will produce Jacobians that will not be invertible. Hence, singular value decomposition will be used here as well to solve the linearized system of equations in the Newton's method approach. Table 10 describes the friction solution process. Note the partial derivatives of the joint force constraint equations may be performed analytically or numerically. The simple lower pair joints are easily differentiated analytically. However, the spherical joint and the surface contact joint constraints necessitate a numerical approach. The central difference approximation was used in these cases.
Table 10: General procedure for Coulomb friction force analysis.

**Formulation and Solution of Friction Force Analysis**

1. Use frictionless analysis solution for initial solution estimate.

2. Evaluate the $Z$ matrices and the inertia and external force vector, $F^T M$, for each of the links according to Equations (6.34) and (6.38).

3. Set $m = 0$; this will be the iteration counter.

4. Calculate the $F^T_{\text{equilibrium}}^{(m)}$ and $F^T_{\text{constraint}}^{(m)}$ residuals. If they are zero within an acceptable tolerance, solution has been found.

5. Evaluate the $G^{(m)}$ matrices for each joint in the mechanism. These constitute the contribution by the joint constraint equations to the Jacobian.

6. Assemble the Jacobian, $J^{(m)}$, using the $Z$'s, $G^{(m)}$'s, and $\partial$s.

7. (a) If there are less than seven joint constraint equations, $J^{(m)}$ will have more columns than rows. Make $J^{(m)}$ square by adding $7 - k$ rows of zero to it where $k$ equals the number of joint constraint equations. The same number of zeros must also be augmented to the right-hand side vector. If the $k \geq 7$, use $J^{(m)}$ as is.

   (b) Determine the SVD of $J^{(m)} = U W V^T$.

   (c) If any $w_j$'s (the diagonal elements of $W$) are equal to zero, the corresponding columns, $j$, of $V$ represent the nullspace basis vectors for the friction force solution.

8. After setting the $1/w_j$'s to zero for all $w_j = 0$, calculate the correction vector for the joint bearing forces and the input force:

   $$\Delta f^T_{m}^{(m)} = -V \ [\text{diag} (1/w_j)] \ U^T F^T_{\text{constraint}}^{(m)}$$  \hspace{1cm} (6.179)

9. Calculate the new solution estimate using $f^T_{m}^{(m+1)} = f^T_{m}^{(m)} + \Delta f^T_{m}^{(m)}$

10. Increment $m$ and repeat steps 4–10 until convergence to a solution is obtained, the maximum allowable iterations have been exceeded, or the solution begins to diverge.
CHAPTER VII

Example Mechanism Analyses

A computer program was written to verify the concepts developed within this course of research. The program reads a data file describing a mechanism. The file is currently created through the use of a text editor. However, the ultimate goal would be to interactively construct a mechanism on the screen of a CRT using graphics oriented techniques. The Sheth and Uicker approach to mechanism modeling is well suited to this approach due to its building block nature.

The mechanism examples presented are designed to demonstrate and validate the analysis approach discussed throughout this dissertation. The definition of a mechanism’s structure is based on the relative orientation and location of its pairing element coordinate systems as well as the joints used to interconnect those systems. One of the mechanism’s joint degrees-of-freedom is selected as the input. The displacement, velocity, and acceleration of this input may be specified as a function of time. The raw output of the kinematic portion of the program consists of the zero-th, first, and second time derivatives of the dependent joint degrees-of-freedom. This information may be used to ascertain kinematic information about any point or body in the mechanism.

The program also performs the force analysis of mechanisms. External and inertia loads are considered as well as the effects of Coulomb friction in the joints.
The results obtained from the force solution include the input force required to achieve specified kinematic motion as well as the joint bearing reactions seen by the "ends" of the links. The internal bearing forces are resolved in both the link local coordinate systems and the mechanism ground system.

Appendix B gives a rough estimation of the typical execution times required by the analysis program developed for this dissertation. A discussion of the computer hardware and programming environment is also presented.

7.1 Kinematic Analysis of a Planar RRRR Mechanism

A planar four-bar mechanism is shown in Figure 39. A closed form kinematic solution for this mechanism is easily obtained using a vector loop equation approach [89, 70, 53]. Figures 40–42 show the displacement, velocity, and acceleration results for $\theta_4$ from such a closed form solution.

Modeling this mechanism using the Sheth-Uicker approach requires the establishment of the pairing element coordinate systems. Figure 39 also shows an exploded view of the links with the pairing element systems defined. The revolute joints require the $w$ and $z$ axis of adjoining links to be aligned with the revolute axis. They have been all selected as positive out of the plane of the page. The $u$ and $x$ axes are usually aligned with significant feature of the links. In this particular case, the $u$ and $x$ axes are aligned with the link centerlines. The joint angles $\theta_1, \theta_2, \theta_3,$ and $\theta_4$ are shown as the angle from $x_i$ to $u_{i+1}$. This is consistent with the pairing element definition for a revolute joint.

With the coordinate systems as shown, it is possible to determine the shape matrices for the links by inspection. However, a more computer oriented approach
Figure 39: A planar four-bar mechanism.
Angular Displacement of RRRR Output Link
Closed Form Solution

\[ \theta_4 \text{ (radians)} \]

Input Crank Displacement (Radians)

Figure 40: Closed form displacement results, \( \theta_4 \), for planar RRRR.

Angular Velocity of RRRR Output Link
Closed Form Solution

\[ \omega_{4,1} \text{ (rad/sec)} \]

Input Crank Displacement (Radians)

Figure 41: Closed form velocity results, \( \omega_{4,1} \), for planar RRRR.
is to use the indirect method outlined in Table 2. With this method, strategic points on the axes of the pairing element systems are specified relative to link definition systems. For a given link, two points on the \( w \) axis are specified with the direction from the first one to the second defined as positive \( w \). A third point on the positive \( u \) axis is also specified. The same technique is used on the \( z \) and \( x \) axes of the same link. These six coordinates, defined in the link definition system, will uniquely define the link's shape matrix. For this particular example, the \((uvw)\) systems have been chosen to serve as the link definition systems for each link. This is done without loss of generality.

Using the \((uvw)\) system of each link as a definition system, the points determining the shape matrices are:
The program determines the link shape matrices, $S_i$, to be:

$$
S_1 = \begin{bmatrix}
-1.0 & 0.0 & 0.0 & -6.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

$$
S_2 = \begin{bmatrix}
-1.0 & 0.0 & 0.0 & -1.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

$$
S_3 = \begin{bmatrix}
-1.0 & 0.0 & 0.0 & -5.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

$$
S_4 = \begin{bmatrix}
-1.0 & 0.0 & 0.0 & -4.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

With the shape matrices defined, specifying the joints connecting them will completely define the mechanism. In this case, all joints are chosen to be revolute joints.

Using the above shape matrices with the revolute pair matrices, the program is able to formulate both the optimization and the Newton’s method based displacement analyses. The pair matrices are functions of the joint degrees-of-freedom. The displacement analysis seeks values for the joint degrees-of-freedom that will
close the mechanism. Either the optimization technique or the Newton's method approach may be used. The optimization procedure is usually invoked for the very first position to be solved. Although it is slower than the Newton method solution, it is fairly insensitive to the quality of the solution estimate. Hence, the optimization approach is most often used to obtain closure for the mechanism. After a closed configuration has been found, the next position to be analyzed can use the previous solution as a starting point. As long as the difference in input position is less than 15 degrees, the previous solution is usually sufficiently close to allow a successful, rapidly convergent Newton's method solution.

The results of the displacement, velocity, and acceleration analysis using the techniques presented in this dissertation are shown in Figures 43-44. Comparing the angular velocity and acceleration shows no difference compared to the closed form solution. The displacement appears different due to different references from which $\theta_4$ is measured.

Note the RRRR mechanism is overconstrained in the spatial sense. Grübler's equation gives its mobility as $-2$. However, since all of its revolute axes are parallel, it is able to move. The use of singular value decomposition has allowed the overconstrained system of equations to be solved without incident. Its mathematical formulation within the program proceeds as though it were a general spatial mechanism. No special case actions were necessary.

7.2 Kinematic Analysis of a "Planar" RSSR Mechanism

The RRRR mechanism of the previous section is now converted to an RSSR mechanism. The revolute joints on the coupler link are simply replaced by spherical
Figure 43: Matrix based solution displacement results, $\theta_4$, for planar RRRR.

Figure 44: Matrix based solution velocity results, $\omega_{4,1}$, for planar RRRR.
joints. No changes in pairing element coordinate systems are necessary (although this is not always the case). The program is simply "informed" that link 2 and link 3 are to be joined by a spherical joint. It is similarly instructed to join links 3 and 4 by a spherical joint. As discussed previously, there are a multitude of pair matrices applicable to modeling spherical joints. The particular implementation used here is chosen as the "$z$-Rotation/$z$-Rotation/$z$-Rotation" formulation.

The kinematic behavior of this RSSR should be essentially identical to the behavior of the previously analyzed RRRR mechanism. The only major difference is the existence of an idle degree-of-freedom about the axis of the coupler. The revolute joints attached to ground will still restrict the overall motion of the links to the plane. The traditional kinematic analysis procedure for this mechanism
Angular Displacement of RSSR Output Link
Iterative Matrix Solution

$\theta_4$ (radians)

Input Crank Displacement (Radians)

Figure 46: Matrix based solution displacement results, $\theta_4$, for quasi planar RSSR.

would involve the artificial constraint of the idle degree-of-freedom in order to make the kinematics solvable. The idle degree-of-freedom indicates a mathematically underconstrained system of equations describe this mechanism. The suppression of the idle degree-of-freedom is not necessary with the analysis procedures outlined in this dissertation.

Figures 46–48 show the angular displacement, velocity, and acceleration of the output link. They are identical to the results obtained with the closed form and iterative solutions for the RRRR mechanism. An additional piece of information, however, is obtained when solving the RSSR mechanism. Since it is an underconstrained mechanism, the mathematical concept of a nullspace solution must apply. Using the technique of singular value decomposition, the nullspace
Figure 47: Matrix based solution velocity results, $\omega_{4,1}$, for quasi planar RSSR.

Figure 48: Matrix based solution acceleration results, $\alpha_{4,1}$, for quasi planar RSSR.
basis vectors are automatically determined during the solution process if any exist. The RSSR mechanism is underconstrained by one equation, hence, one nullspace basis vector must exist. For linear problems, the nullspace basis vectors may be added in any linear combination to the "particular" solution to yield another valid solution.

During the solution process, the unknown joint variables in a mechanism are placed in an "unknown" vector. They will appear in the order found when traversing the mechanism loop. For the RSSR mechanism, this would be

\[
\begin{align*}
Q_1 & = \{ \theta_2, \gamma_2, \alpha_3, \beta_3, \gamma_3, \theta_4 \} \\
Q_2 & = \{ \phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7 \}
\end{align*}
\]

Hence, it is evident the system will be underconstrained—seven unknowns exist for which only six (at most) equations are available to determine them. The SVD approach for this mechanism is found to yield a nullspace basis vector that is identical for each position analyzed. The nullspace basis vector, \( \hat{Q}_i \), is found always equal to

\[
\hat{Q}_1 = \{ 0, 0, 0.70710, 0.70710, 0, 0, 0 \}^T.
\]

Although the displacement analysis is nonlinear, it is clear that the axes about which \( \gamma_2 \) and \( \alpha_3 \) rotate are collinear. Thus this nullspace vector may in fact be multiplied by any constant and added to the "particular" solution obtained. The so-called particular solution for the RSSR mechanism is identical to the solution for a similar RRRRR mechanism. In essence, the SVD method of analysis has not
only solved a deficient system of equations for a meaningful answer, but it has also
detected the idle degree-of-freedom and "explains" exactly how it is manifested.
Again, this is done automatically through the SVD process without any special case
approaches being taken. It should be emphasized the analyst does not need to do
anything special or insightful (like suppress a degree-of-freedom) if a mechanism
is underconstrained. In fact, it is not even necessary to recognize a mechanism
as being underconstrained—SVD will announce it as so through the existence of
nullspace vectors.

A nullspace vector is also obtained when performing the velocity and accelera­
tion analyses on the RSSR mechanism. This stands to reason as there will be seven
unknown time derivatives in both cases. The nullspace vectors obtained are ident­
tical to the one produced by the displacement analysis. This is reasonable since
the coupler link can spin about its axis as a function of time at any rate without
affecting the rest of the mechanism. The particular solution to the velocity and
acceleration analyses will show the idle degree-of-freedom with first and second
time rates of change equal to zero.

7.3 Kinematic Analysis of a Spatial Slider Crank Mech­
anism

Sandor and Erdman [22] analyze the RSSP spatial slider crank mechanism shown
in Figure 49. They do this using a matrix based approach. The links and joints
are described in terms of translation and rotation matrices. Using the dimensions
shown in Figure 49, and a crank velocity of 6 radians/sec, they produce the kine­
matic results for the slider as shown in Figure 50. This mechanism suffers from the
Figure 49: Spatial slider crank analyzed by Sandor and Erdman.[22]
Figure 50: Displacement, velocity, and acceleration of the RSSP slider as obtained by Sandor and Erdman.[22]
same condition as the RSSR mechanism—it has an idle degree-of-freedom along the connecting rod. The analysis put forth by Sandor and Erdman required that one of the spherical joints be modeled as a \( U \)-joint containing only two degrees-of-freedom. This reduced the number of unknowns to six thereby making the system of equations determinate.

Consider the individual links with pairing element coordinate systems as shown in Figure 51. Note in this case, local link definition systems have been used that are separate from the \((uvw)\) or \((xyz)\) systems. Using the dimensions shown, the coordinates of the points required to define the shape matrices are

<table>
<thead>
<tr>
<th>Link</th>
<th>System</th>
<th>Point 1</th>
<th>Point 2</th>
<th>Point 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((uvw))</td>
<td>((0,0,0))</td>
<td>((0,0,1))</td>
<td>((1,0,0))</td>
</tr>
<tr>
<td></td>
<td>((xyz))</td>
<td>((4,0,2))</td>
<td>((4,0,3))</td>
<td>((4,1,2))</td>
</tr>
<tr>
<td>2</td>
<td>((uvw))</td>
<td>((0,1.5,0))</td>
<td>((1,1.5,0))</td>
<td>((0,-1.5,0))</td>
</tr>
<tr>
<td></td>
<td>((xyz))</td>
<td>((-0.5,0))</td>
<td>((-0.5,1))</td>
<td>((-0.6,0))</td>
</tr>
<tr>
<td>3</td>
<td>((uvw))</td>
<td>((-5,0,0))</td>
<td>((-5,0,1))</td>
<td>((-6,0,0))</td>
</tr>
<tr>
<td></td>
<td>((xyz))</td>
<td>((5,0,0))</td>
<td>((5,0,1))</td>
<td>((6,0,0))</td>
</tr>
<tr>
<td>4</td>
<td>((uvw))</td>
<td>((1,0,0))</td>
<td>((1,1,0))</td>
<td>((2,0,0))</td>
</tr>
<tr>
<td></td>
<td>((xyz))</td>
<td>((0,0,0))</td>
<td>((0,0,1))</td>
<td>((1,0,0))</td>
</tr>
</tbody>
</table>

The results obtained using the techniques outlined in this dissertation are shown in Figures 52–54. They are identical to those obtained by Sandor and Erdman. The graphs appear as inverted due to differences in reference systems from which measurements are taken. A nullspace basis vector similar to the one obtained in the RSSR analysis is produced by SVD without any special treatment.

7.4 Bent-Shaft Universal Joint—A Surface Contact Mechanism

Beggs[12] presents a kinematic analysis of a bent-shaft universal joint. This mechanism is a low cost means of transmitting rotary motion between two nonparallel
Figure 51: Pairing elements coordinate systems for RSSP mechanism.
Figure 52: Matrix based solution displacement results for the RSSP slider.

Figure 53: Matrix based solution velocity results for the RSSP slider.
(skew in general) shafts. A pictorial representation was shown in Figure 7. Although in reality this mechanism consists of two revolutes joints and one surface contact joint, it may be modeled kinematically using an RCRCR mechanism. As a further verification of the techniques presented in this dissertation, kinematic analyses of the bent-shaft universal joint have been performed on both the “Revolute–Surface–Revolute” and the RCRCR representations. The results are shown to be identical. A force analysis has also been performed. Only the surface contact representation of the mechanism is valid for the force analysis.

7.4.1 Bent-Shaft Kinematic Analysis Using a Surface Contact Joint

A bent-shaft universal joint is shown in Figure 55. It consists of three links; link 1
Figure 55: Bent-shaft universal joint mechanism example.
is the frame, link 2 is the input, and link 3 is the output. The output shaft is at an angle of 15 degrees relative to the input shaft. In this particular example, the input and output shaft centerlines intersect. However, this in general is not necessary. The diameters of the two contacting shafts are both 0.25 inches. The angular velocity of the input link is 3.14159 radians/second, and the angular acceleration is zero.

Figure 56 shows each of the links with their pairing element coordinate systems. These systems are consistent with the joints used in the mechanism. The locations and orientations of the systems used, however, are not the only ones possible. For the systems presented, the points used to define the shape matrices are:

<table>
<thead>
<tr>
<th>Link</th>
<th>System</th>
<th>Point 1</th>
<th>Point 2</th>
<th>Point 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(uvw)</td>
<td>(0.534976,0,-4.96248)</td>
<td>(0.793795,0,-5.92841)</td>
<td>(-0.430950,0,-5.22130)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>(uvw)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0,0,-2)</td>
<td>(0,1,-2)</td>
<td>(0,0,-3)</td>
</tr>
<tr>
<td>3</td>
<td>(uvw)</td>
<td>(0,1.5,-1)</td>
<td>(0,1.5,-2)</td>
<td>(-1,1.5,-1)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
</tbody>
</table>

The resulting link shape matrices are

\[
S_1 = \begin{bmatrix}
-0.96592582630 & 0 & -0.25881904508 & -0.76763808979 \\
0 & 1 & 0 & 0 \\
0.25881904508 & 0 & -0.96592582630 & -0.49318516525 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
S_2 = \begin{bmatrix}
0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
-1 & 0 & 0 & -2 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
S_3 = \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & 1 & 0 & -1.5 \\
0 & 0 & -1 & -1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

The angular displacement, velocity, and acceleration of the output link are shown in Figures 57-59. The velocity and acceleration results clearly demonstr-
Figure 56: Pairing element coordinate systems for the bent-shaft universal joint links.
Angular Displacement of Bent-Shaft Output Link
Surface Contact Model

\[ \theta_3 \quad \text{(radians)} \]

Figure 57: Output shaft displacement results for the bent-shaft universal joint using a surface contact joint.

strate the fluctuation in the output relative to the input. Figure 60 shows the displacement of the contact point relative to the surface definition systems for the two cylindrical surfaces. The excursion of the contact points indicates the length of the contacting shafts (relative to the surface definition systems) necessary to maintain contact throughout the motion of the mechanism.

7.4.2 Kinematic Analysis Using Equivalent Lower Pair Joints

The bent-shaft universal joint can also be modeled kinematically using lower pair joints. This is possible since the contacting surfaces are very regular—the common normal between the contacting cylindrical surfaces will always go through the point
of contact. The length of the common normal will always equal the sum of the shaft radii. General surfaces in contact will not be representable using lower pair equivalents.

A schematic representation of the bent-shaft universal joint is shown in Figure 61 using lower pair joints. The surface contact joint is essentially modeled using two cylindrical joints and one revolute joint. This is consistent with the known five degree-of-freedom behavior of surface contact joints. The cylindrical joints provide the degrees-of-freedom necessary to allow the movement of the point of contact on the surfaces of the contacting cylinders. The revolute joint is used at the point of contact. It is analogous to the “floating-to-floating” transformation used in the mathematical development of the surface contact joint presented in

Figure 58: Output shaft velocity results for the bent-shaft universal joint using a surface contact joint.
Figure 59: Output shaft acceleration results for the bent-shaft universal joint using a surface contact joint.

Chapter III. It allows the "twisting" of one surface relative to the other about their common normal at the point of contact.

The pairing element coordinate systems for the lower pair equivalent mechanism are also shown in Figure 61. As with the surface contact model for this mechanism, the locations and orientations of the pairing element systems used are not the only ones possible. The points used to describe the shape matrix are as follows:
Figure 60: Displacement of the contact point on the input and output shafts of the bent-shaft universal joint using a surface contact joint.

<table>
<thead>
<tr>
<th>Link</th>
<th>System</th>
<th>Point 1</th>
<th>Point 2</th>
<th>Point 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(uvw)</td>
<td>(0.534976,0,-4.96248)</td>
<td>(0.793795,0,-5.92841)</td>
<td>(-0.430950,0,-5.22130)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>(uvw)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0,0,-2)</td>
<td>(0,1,-2)</td>
<td>(0,0,-3)</td>
</tr>
<tr>
<td>3</td>
<td>(uvw)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0.125,0,0)</td>
<td>(1.125,0,0)</td>
<td>(0.125,0,1)</td>
</tr>
<tr>
<td>4</td>
<td>(uvw)</td>
<td>(0.125,0,0)</td>
<td>(-0.875,0,0)</td>
<td>(0.125,1,0)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>5</td>
<td>(uvw)</td>
<td>(0.15,-1)</td>
<td>(0.15,-2)</td>
<td>(-1,1.5,-1)</td>
</tr>
<tr>
<td></td>
<td>(xyz)</td>
<td>(0,0,0)</td>
<td>(0,0,1)</td>
<td>(1,0,0)</td>
</tr>
</tbody>
</table>

The resulting link shape matrices are

\[
S_1 = \begin{bmatrix}
-0.96592582630 & 0 & -0.25881904508 & -0.76763808979 \\
0 & 1 & 0 & 0 \\
0.25881904508 & 0 & -0.96592582630 & -0.49318516525 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Figure 61: Equivalent lower pair model for the bent-shaft universal joint
\[ S_2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ S_3 = \begin{bmatrix} 0 & 0 & 1 & 1.25 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ S_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1.25 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ S_5 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1.5 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

The angular displacement, velocity, and acceleration of the output shaft obtained by modeling the bent-shaft with lower pair joints are presented in Figures 62–64. The results are identical with those produced by the surface contact model.

### 7.4.3 Force Analysis

A valid force analysis of the bent-shaft universal joint can be performed only using the surface contact model. Although the lower pair equivalents can be used to model the kinematics of the mechanism, they will not yield force relations equivalent to the actual surface contact situation. A torque was placed on the output shaft equal to 100 in-lb. The input torque required to produce the specified input angular velocity of 3.14159 radians/sec is shown in Figure 65. The links were modeled as massless. Both frictionless and friction cases were studied. Coefficients of friction were taken to be 0.1 for all three of the joints. The pin radii for the revolute joints were taken to be 0.125 inches.
Angular Displacement of Bent-Shaft Output Link
Equivalent Lower Pair Model

\[ \theta_s \quad \text{(radians)} \]

Input Crank Displacement (Radians)

Figure 62: Output shaft displacement results for the bent-shaft universal joint using lower pair joints.

7.5 Swash Plate with Spherical Follower—A Surface Contact Mechanism

Another mechanism containing a surface contact joint is demonstrated in the swash plate mechanism of Figure 66. The exploded view shows the pairing element systems used. The mechanism analyzed has dimensions \( a = 1.45 \) inches, \( b = 2.60 \) inches, and \( c = 2.35 \) inches. The radius of the spherical follower is 0.625 inches. The angle, \( \alpha \), of the swash plate is 15 degrees. A load is placed on the follower equal to 10 pounds along the follower axis such that the follower is pressed against the swash plate.

The points on the pairing element axes used to construct the shape matrices
Figure 63: Output shaft velocity results for the bent-shaft universal joint using lower pair joints.

The odd looking values of 0.258819 and 0.965926 come from \( \sin \alpha \) and \( \cos \alpha \). The resulting shape matrices are

\[
S_1 = \begin{bmatrix}
-1.0 & 0.0 & 0.0 & 1.45 \\
0.0 & 1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & -1.0 & 2.60 \\
0.0 & 0.0 & 0.0 & 1.0 \\
\end{bmatrix}
\]
Figure 64: Output shaft acceleration results for the bent-shaft universal joint using lower pair joints.

Figure 65: Required input shaft torque for the bent-shaft universal joint
Figure 66: A swash plate mechanism with spherical follower.
Figure 67: Matrix based solution displacement results for the swash plate mechanism.

\[
S_2 = \begin{bmatrix}
0.96592584920 & 0.0 & 0.25881895959 & 0.0 \\
0.0 & 1.0 & 0.0 & 0.0 \\
-0.25881895959 & 0.0 & 0.96592584920 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
S_3 = \begin{bmatrix}
1.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & -2.35 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Using an input angular velocity for the swash plate of \(3.14159\) radians per second, and an angular acceleration of 0, the displacement, velocity, and acceleration of the follower is determined as shown in Figures 67–69.

The swash plate torque required to produce this motion based on the 10 pounds of follower loading is shown in Figure 70. The torque has been calculated for a totally frictionless mechanism as well as one in which friction is found in each
Figure 68: Matrix based solution velocity results for the swash plate mechanism.

Figure 69: Matrix based solution acceleration results for the swash plate mechanism.
Figure 70: Input torque required for frictionless and friction cases of the swash plate mechanism.

joint. The friction case uses a uniform coefficient of friction for all joints as \(0.1\). The revolute pin radius (i.e., the axis of the swash plate) is taken as \(0.125\) inches.

7.6 A Simple Verification Test for the Spherical Joint Friction Model

Wilson [89] analyzes a planar slider crank with Coulomb friction in all of its joints. The mechanism is shown in Figure 71. The pairing element coordinate systems are also shown in an exploded view of the links. The points on the pairing element axes used to determine the link shape matrices are
Figure 71: A slider crank mechanism analyzed with Coulomb friction
The shape matrices are then calculated by the program to be

\[
\begin{align*}
S_1 &= \begin{bmatrix}
0.0 & 0.0 & 1.0 & 0.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
1.0 & 0.0 & 0.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix} \\
S_2 &= \begin{bmatrix}
-1.0 & 0.0 & 0.0 & 30.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix} \\
S_3 &= \begin{bmatrix}
-1.0 & 0.0 & 0.0 & 70.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix} \\
S_4 &= \begin{bmatrix}
0.0 & 0.0 & 1.0 & 0.0 \\
0.0 & -1.0 & 0.0 & 0.0 \\
1.0 & 0.0 & 0.0 & 0.0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

If a load of 40N is applied to the slider along its sliding axis, and the crank is turning clockwise, Wilson finds the corresponding crankshaft torque for the frictionless case to be -1119 Nmm when the crank angle is 45 degrees. When the coefficients of friction for all of the joints are specified as equal to 0.1, and the pin radii for the revolute joints are 10mm, Wilson finds the torque is increased to -1273 Nmm.

The program written for this dissertation finds the torque for the frictionless case to be -1118.3595020 Nmm and for the friction case to be -1274.0438211 Nmm.
when the crank angle is 45 degrees. The agreement is obviously very good.

The revolute joint attaching the coupler to the crank was then replaced by a spherical joint. An “x-rotation/z-rotation/x-rotation” implementation was used. Recall, a friction model was developed for this particular description of a spherical joint. For the frictionless case, with a spherical joint in place, yielded a required input torque of -1118.3594845 Nmm. For the friction case, with the coefficients of friction equal to 0.1, and the sphere and revolute radii equal to 10mm, the input torque is found to be -1274.0437997 Nmm. This agrees exactly (within numerical tolerances) with the case where all the rotary joints were revolutes. This is an especially encouraging result. It indicates the friction model for the spherical joint is performing as designed—the friction model treats the joint as a revolute whose axis is aligned with the instant screw axis between the links joined by the spherical pair. For the planar situation, the internal bearing forces will act in the plane of the “equator”, and hence, the effective pin radius is simply the sphere radius. Thus, for the case at hand, the spherical joint should behave as a revolute whose axis is perpendicular to the plane of motion and whose radius is simply the sphere radius. The results actually obtained suggest this is in fact the case.
CHAPTER VIII

Conclusion

8.1 Summary

It is felt the key contributions put forth by this dissertation are as follows:

- A robust means for solving under and overconstrained systems of kinematic equations has been found using singular value decomposition. This is very useful for mechanisms possessing idle degrees-of-freedom or dead center positions. It also allows “special mobility” mechanisms such as planar mechanisms or maverick mechanisms (like the Bennett mechanism) to be analyzed as general spatial mechanisms.

- A general, three-dimensional surface contact joint based on transformation matrix techniques was developed. This is a five degree-of-freedom joint. Both the kinematic and force characteristics of this joint have been presented.

- A friction model for the spherical joint was developed. This allows the friction characteristics of a spherical joint to be accurately considered. Lower pair equivalents for the spherical joint do not exhibit the same joint bearing forces as the actual “ball and socket” in a spherical joint.

The concepts presented in this dissertation represent a generalized method for the kinematic and force analysis of closed single loop, one degree-of-freedom
mechanisms. There are various numerical pathologies that may affect mechanism analysis. The ability to deal with under and overconstrained mechanisms is important since these types of mechanisms are quite common and useful. The whole class of planar mechanisms is overconstrained, and yet these are probably the most commonly used mechanisms.

If it is desirable to computerize the analysis of mechanisms, the availability of a unified analysis approach makes for an easier programming task and results in cleaner code. Furthermore, a general approach reduces the amount of experience and intuition required on the part of the analyst when performing a computer assisted analysis. The procedures discussed within this work develop a general approach to mechanism analysis that is readily programmable and thought to be more robust than previous procedures.

A procedure developed by Sheth and Uicker was used to model the mechanisms. It is a matrix based approach that separates the link dimensions from joint dimensions in a mechanism. This makes for a very building block oriented modeling procedure. As a result it was possible to develop a joint model for general three dimensional surface contact. This is a five degree-of-freedom joint. The contact surfaces may be well defined analytical surfaces or made up of parametric surface patches.

Perhaps one of the most interesting facets of this research is the application of singular value decomposition to the kinematic analysis of a mechanism. It is a relatively straightforward process that provides extremely useful information for deficient systems of equations. Overconstrained systems of equations are solved by SVD to yield a least squares solution to the system. In the case of a closable
mechanism, the overconstrained system of equations is consistent and the "least squares" solution will represent an exact solution. For underconstrained mechanisms, no unique solution exists since there are more unknowns than independent equations. Traditional methods of analysis are not able to solve such systems of equations without artificially reducing the number of unknowns. This requires some experience and insight on the part of the analyst.

The SVD solution to an underconstrained system of linear equations actually provides the infinite number of solutions possible by obtaining a "particular" solution and the system's nullspace basis vectors. The nullspace vectors may be added in any linear combination to the particular solution to get another valid solution. Even though the displacement system of equations is nonlinear for mechanisms, the concept of the nullspace is still useful. SVD is employed in the linearized portion of the Newton's method procedure. Newton's method is perhaps the most common means of solving a nonlinear system of equations. The nullspace vectors obtained from the Newton's method solution will "detect" rotational or translation axes of motion that happen to line up in a mechanism. Even though the displacement equations are nonlinear in the unknown joint variables, there is a bit of "linearity" involved when axes line up. If two rotational axes are collinear, a rotation about one axis can be offset by an equal but opposite rotation about the other. The SVD process has been shown to be very useful in describing this type of situation in a mechanism. In essence, the nullspace solution produced by a displacement analysis is a clear and meaningful indication of idle degrees-of-freedom within a mechanism. Furthermore, SVD "sees past" the idle degrees-of-freedom and still determines the correct kinematic behavior of the rest of the mechanism.
Singular value decomposition is equally useful in the force analysis of a mechanism. Again, treating all mechanism as spatial causes problems with under and overconstrained systems of force equations. Planar mechanisms for instance are indeterminate when a spatial force analysis is performed on them. The out of plane loads cannot be determined. The in plane loads are calculable if the force equations involving the out of plane loads are discarded. SVD will in a sense ignore the out of plane force equations automatically in the case of a planar mechanism. The associated system of force equations is simply underconstrained which SVD has no problem dealing with. It will solve those equations that are independent and provide nullspace basis vectors corresponding to the unknowns which do not have a unique value given the available system of equations.

The use of SVD yields a fairly robust solution procedure. Since it does not rely on inverting a coefficient matrix, situations such as “zero pivot elements” or ill conditioned matrices do not affect it. Hence, some sort of quantitative solution is always available. Experience has always shown this solution to be of direct and useful physical significance in terms of the mechanism’s motion or force state (i.e., idle degrees-of-freedom, dead-center positions, partial force indeterminacy, etc.). SVD is also useful in those situations where a system of equations is nearly deficient as might happen when a mechanism is approaching a singular position. Here too, it will detect the impending problem and perhaps yield an extra nullspace basis vector, but it will not simply give up with an “ill conditioned matrix” error.

To enhance the robustness of the SVD approach to mechanism analysis, an optimization technique was used to solve the mechanism in its first position. The optimization technique is found to be fairly insensitive to the initial guess for
the joint variable values that close a mechanism. Although quite a bit slower than the Newton's method approach, it does provide a high degree of robustness in converging to a closure solution. After one position is found, the successive positions can use the previous one as an initial guess assuming they are not too far apart (less than 15 degrees for an angular degree-of-freedom seems to work well).

The force analysis described is relatively straightforward. The formulation is relatively traditional. It uses dynamic equilibrium for each of the links other than ground, and obtains additional force equations from constraints due to the joints used. The major difference in this research is, again, that SVD has been used to solve the system of equations. This virtually eliminates the possibility of a numerical problem preventing a quantitative solution.

In order to effectively model friction in the joints, a friction model was developed for the spherical joint. The models for the other joints are relatively straightforward. Pin friction in a revolute joint is treated in most undergraduate kinematics text books. Slider friction as well as surface contact friction is very easily described in terms of the normal force, the coefficient of friction, and the relative velocities of the surfaces in contact. Friction in a helical joint can be expressed in terms of power screw relations found in most machine design text books. The model developed within this dissertation for the spherical joint with friction assumes the joint acts as an instantaneous revolute whose axis is aligned with the instant screw axis between the two links connected by the spherical joint. The radius of the instantaneous revolute is a function of the point of contact between the spherical pair surfaces.
8.2 Recommendations

The approaches outlined in this research are very amenable to computerized implementation. The technique currently used to define a mechanism (via a text edited data file) is cumbersome and prone to error. It would be highly desirable to provide an interactive graphics interface with which an analyst could visualize the mechanism as it is being defined. This should be a relatively straightforward process using the Sheth and Uicker approach to mechanism modeling. The ability to determine the link shape matrices by selecting points on the axes of the paring element coordinate systems will greatly facilitate this process.

Furthermore, since the link and joint dimensions and characteristics are considered as separate entities in the solution process (i.e., shape and pair matrices), it is now feasible to consider complex interactions in the joints. Joint clearances might now be more easily addressed. In this case, joints with clearances have increased numbers of degrees-of-freedom. Hence, ideally single degree-of-freedom mechanisms will become multi-degree-of-freedom mechanisms. The solutions to these system will be functions of the mechanism loading. As such, the kinematic and force analyses can no longer be performed independently. They will become integral parts of a single iterative process that solves the kinematic problem in a way that is consistent with dynamic force equilibrium.

The separation of link dimensions from joint dimensions also provides a clear manner in which to consider dimensional tolerances in a mechanism. The parameters that define a shape matrix could be made expressible in terms of the manufacturing tolerances on a link. The kinematic analysis might then produce a statistical estimate of the motion characteristics for a population of similar mech-
anisms.
Appendix A

Relations Between the Elements of Rotation Matrices

Rotation matrices as used in this dissertation are orthonormal transformations whose columns can be interpreted as direction cosines of right-handed coordinate systems. Therefore, the following can be said:

- The dot product of any rotation partition column with itself is equal to unity.
- The cross product of any two columns (in cyclical order) must equal the remaining column.

Hence, letting $t^{i,j}$ represent the element in the $i$-th row and $j$-column, the following equations result:

\[
(t^{1,1})^2 + (t^{2,1})^2 + (t^{3,1})^2 = 1 \quad (A.1)
\]
\[
(t^{1,2})^2 + (t^{2,2})^2 + (t^{3,2})^2 = 1 \quad (A.2)
\]
\[
(t^{1,3})^2 + (t^{2,3})^2 + (t^{3,3})^2 = 1 \quad (A.3)
\]

and

\[
t^{2,1} t^{3,2} - t^{3,1} t^{2,2} = t^{1,3} \quad (A.4)
\]
\[
t^{3,1} t^{1,2} - t^{1,1} t^{3,2} = t^{2,3} \quad (A.5)
\]
Equations (A.1)-(A.3) come from the dot products and Equations (A.4)-(A.12) come from the cross products. Note a total of 12 equations have been presented in terms of the nine rotation elements. Therefore, they cannot all be independent—three of the equations must contain redundant information in relation to the others.

The three dot product equations cannot be discarded since they are the only way of restraining the column magnitudes to unity. However, if two elements of a column are determined using cross product information, the third can be obtained by knowing that the sum of their squares must equal one. Thus, a cross product equation for one element of each column can be considered as redundant information. The three equations bracketed above are those that might be chosen as the redundant ones.

As per Sandor and Erdman [66], if \( t^{1,3}, t^{2,1}, \) and \( t^{3,2} \) are set to zero, the following equations are obtained from (A.1)-(A.12):

\[
(t^{1,1})^2 + (t^{3,1})^2 = 1 \tag{A.13}
\]
\[
(t^{1,2})^2 + (t^{2,2})^2 = 1 \tag{A.14}
\]
\[(t^{2,3})^2 + (t^{3,3})^2 = 1 \quad \text{(A.15)}\]

and

\[-t^{3,1}t^{2,2} = 0 \quad \text{(A.16)}\]
\[t^{3,1}t^{1,2} = t^{2,3} \quad \text{(A.17)}\]
\[\left[t^{1,1}t^{2,2} = t^{3,3}\right] \quad \text{(A.18)}\]
\[\left[t^{2,2}t^{3,3} = t^{1,1}\right] \quad \text{(A.19)}\]
\[-t^{1,2}t^{3,3} = 0 \quad \text{(A.20)}\]
\[t^{1,2}t^{2,3} = t^{3,1} \quad \text{(A.21)}\]
\[t^{2,3}t^{3,1} = t^{1,2} \quad \text{(A.22)}\]
\[\left[t^{3,3}t^{1,1} = t^{2,2}\right] \quad \text{(A.23)}\]
\[-t^{2,3}t^{1,1} = 0. \quad \text{(A.24)}\]

Examining Equations (A.17), (A.21), and (A.22) shows they are of the form

\[ab = c \quad \text{(A.25)}\]
\[bc = a \quad \text{(A.26)}\]
\[ca = b. \quad \text{(A.27)}\]

Placing (A.25) in (A.26) for c yields

\[ab^2 = a \quad \text{(A.28)}\]

or

\[a(b^2 - 1) = 0 \quad \text{(A.29)}\]
Hence, \( a = 0 \) and \( b = \pm 1 \) are all separate solutions. If \( a = 0 \), then \( b = c = 0 \). If \( b = +1 \), then \( a = \pm 1 \) and \( c = \mp 1 \) where \( a \) and \( c \) must have the same sign. If \( b = -1 \), then \( a = \pm 1 \) and \( c = \mp 1 \) where \( a \) and \( c \) must have different signs.

Realizing that \( a, b, \) and \( c \) represent the off-diagonal elements that were not specified as zero, the following possibilities arise given \( t^{1,3} = t^{2,1} = t^{3,2} = 0 \):

* The off-diagonal is zero, and the diagonal is nonzero:

\[
\begin{align*}
t^{3,1} &= t^{1,2} = t^{2,3} = 0 \\
t^{1,1} &= \pm 1 \\
t^{2,2} &= \pm 1 \\
t^{3,3} &= \pm 1 
\end{align*}
\]

All three diagonal elements may be \(+1\) or any two may be \(-1\).

* The diagonal is zero, and the off-diagonal is nonzero:

\[
\begin{align*}
t^{1,1} &= t^{2,2} = t^{3,3} = 0 \\
t^{3,1} &= \pm 1 \\
t^{1,2} &= \pm 1 \\
t^{2,3} &= \pm 1 
\end{align*}
\]

All three off-diagonal elements may be \(+1\) or any two may be \(-1\).

Therefore, setting the three off-diagonal elements \( t^{2,1}, t^{3,2}, \) and \( t^{1,3} \) to zero certainly does not require the remaining off-diagonal elements to be zero nor does it require the diagonal elements to be \(+1\). Hence, the knowledge that \( t^{2,1}, t^{3,2}, \) and \( t^{1,3} \) are equal to zero is not a sufficient condition for the rest of the elements in \( T \) to correspond to an identity matrix.
Appendix B

Computer Hardware, Programming Environment, and Analysis Software Execution Speeds

A kinematic and force analysis program was written to demonstrate and verify the techniques developed in this dissertation. The hardware consisted of a Commodore A1000 (otherwise known as an Amiga(tm)) personal computer. This is an MC68000 based computer running at 7.14 MHz. The operating system used was AmigaDOS 1.3. This is a fully implemented, true multitasking operating system. The system used had 2.5Mb of RAM. An MC68881 math chip running at 16MHz was used for the floating point operations. This chip is used as a peripheral device rather than as a true math coprocessor—the MC68000 instruction set does not have the ability to access the instruction set of the MC68881 directly.

An ANSI 77 implementation of FORTRAN for the Commodore A1000 was used as the development language. It is marketed by the Absoft Corporation. The routines necessary to use the MC68881 for floating point operations were custom written since the Absoft compiler does not recognize the math chip when accessed as a peripheral device. All floating point variables and arrays are declared as double precision (i.e., 64 bit quantities) in the analysis program.

The analysis program consists of roughly 12,000 lines of source code. This
includes a liberal amount of comment lines. The source code consumes approximately 550,000 bytes of storage. The compiled code (using the Absoft compiler) uses approximately 330,000 bytes of storage.

The closed form solution to the RRRRR mechanism described in Section 7.1 is solved on the A1000 in approximately 9 seconds. This is the composite time for the displacement, velocity, and acceleration analyses of 73 positions. The iterative matrix approach presented in this dissertation for 73 positions of the RRRR mechanism requires approximately 150 seconds for the displacement analysis, 37 seconds for the velocity analysis, and 100 seconds for the acceleration analysis. Obviously, the iterative method is much slower than a closed form approach. Hence, if a closed form solution is available, it might be desirable to choose the closed form method over an iterative one.

The bent-shaft problem presented in Section 7.4.1 and analyzed in 37 positions using the iterative matrix approached required approximately 180 seconds for the displacement analysis, 56 seconds for the velocity analysis, and 195 seconds for the acceleration analysis. The frictionless force solution for the bent-shaft universal joint required approximately 1,100 seconds for 37 positions. The iterative friction force analysis took approximately 2,200 seconds for 37 positions. Note, however, the friction solution necessitates having a frictionless solution as a starting point. Hence, the execution time for the frictionless solution must be added to the friction solution time to get a true indication of the force analysis execution time when the friction problem is the ultimate goal.

It is evident the solution time for the force solution is considerable compared to the kinematic analysis. This is even the case for the linear, noniterative frictionless
solution. This is due to the number of unknowns in a force analysis. A simple four link mechanism will have 25 unknown forces (i.e., six bearing forces at each joint and one input force). Hence, a system of 25 unknowns in 25 equations must be solved. This is a much bigger problem than the kinematic problem where at most six independent equations are found.
BIBLIOGRAPHY


