Pantomeshes: Kinematics, Synthesis, and Applications of Closed Pantograph-Style Linkage Systems

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Blake Timothy Larson

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Arthur G. Erdman, Adviser

December 2010

Acknowledgments

I would like to thank Professor Art Erdman for his ideas, guidance, and patience throughout this entire process. Together we have designed devices, wrote grant applications, and even ran a few marathons.

I would also like to acknowledge my colleagues who have worked with me on this and other projects. In particular, I would like to thank Deepa, Nathan Knutson, Ryan Buessler, Sam Will, and Megan Kruse. I would also like to acknowledge summer interns Valerie Grant, who coined the phrase "breast cradle," and Rebecca Grove, who did early work on breast stabilization.

Finally, I would like to thank my loving wife Terra for her support, proofreading, and seemingly infinite patience. I could not have done any of this without her.

Dedication

This work is dedicated to my children, Elsa (4) and Ian (1). I have been working on this thesis their entire lives (and, it would seem, a decent portion of my own).

Abstract

This research describes the kinematics, analysis, and synthesis of a pantomesh. A pantomesh is a patchwork assembly of pantograph elements (known elsewhere as scissor pairs or duplets) that obey certain mobility requirements. A pantomesh, as described in this thesis, has scissor-like *elements* connected to one another by spherical joints to allow a wide variety of motions. Previous pantograph-style linkages, such as the Hoberman Sphere, use special geometry restrictions and have elements joined with gussets, thereby limiting the variety of shapes possible.

The thesis begins with examining the kinematics of pantomeshes and their constituent parts. First, the kinematics of the individual pantograph elements are detailed for further use. The mobility of a closed pantomesh is ensured by the mobility of its constituent pantopatches, two-wide by two-high sub-assemblies of pantograph elements that must be mobile themselves for the entire pantomesh to be mobile. A new method for mobility of spatial linkages is presented relating the use of polygonal elements.

Next, two methods for pantomesh synthesis are presented. A graphical method is presented to use a computer-aided design system to create a mobile pantomesh that meets specified requirements. A computational method for synthesis is also presented, using a numerical optimization method to create pantomeshes to certain specifications.

Practical considerations of manufacturing are considered in the discussion of multi-link spherical joints, including past work and new approaches. The new approaches include a compliant multi-link spherical joint and a crossed-tendon system that acts a a spherical joint. Finally, an application is presented: a new linkage which provides radial pressure for the purpose of stabilizing a human breast during cancer-related diagnosis and treatment procedures.

Contents

List of Figures **viii**

List of Tables **xi**

Chapter 1 Introduction **1**

1.1 Definitions and Taxonomy 1

1.2 Previous Work 4

1.2.1 Planar Pantrograph-style Linkages 4

1.2.2 Constant-Curvature Pantomeshes 5

1.2.3 Variable-Curvature Pantomeshes 10

1.3 Scope of Thesis 12

1.4 Outline of the Thesis 13

Chapter 2 Kinematics of Pantograph Elements **16**

2.1 Nomenclature 16

2.2 Straight-Link Pantograph Elements 17

2.2.1 Planar Straight-Link Pantograph Elements 17

2.2.2 Skew, Normal-Pivot Straight-Link Pantograph Elements . . . 20

2.2.3 Skew, Angled-Pivot Straight-Link Pantograph Elements . . . 22

2.3 Bent-Link Pantograph Elements 25
2.3.1 Planar Bent-Link Pantograph Elements 26
2.3.2 Skew, Normal-Pivot Bent-Link Pantograph Elements 29
2.3.3 Skew, Angled-pivot Bent-Link Pantograph Elements 31
2.4 Discussion 36
2.5 Conclusions 37

Chapter 3 Mobility of Pantopatches 38

3.1 Degrees of Freedom 38
3.2 Mobility of Straight-Link Pantopatches 39
3.2.1 Planar Straight-Link Pantopatches 40
3.2.2 Mobility of Skew, Normal-Pivot Straight-Link Pantopatches . 42
3.2.3 Mobility of Skew, Angled-Pivot Straight-Link Pantopatches . 43
3.3 Mobility of Bent-Link Pantopatches 44
3.3.1 Mobility of Planar Bent-Link Pantopatches 44
3.3.2 Mobility of Skew, Normal-Pivot Bent-Link Pantopatches . . . 53
3.3.3 Mobility of Skew, Angled-Pivot Bent-Link Pantopatches . . . 56
3.4 Graphical Mobility Determination 60
3.5 Discussion 61
3.6 Conclusions 62

Chapter 4 A Mobility Formula for Connected Polyhedra 63

4.1 Mobility of Connected Polyhedra 63
4.1.1 Rigid Bodies 64
4.1.2 Spherical and Rotational Joints 65

4.1.3 A Mobility Formula 67
4.1.4 Joints with Translational Motion 68
4.1.5 Adjustment for Remote Spherical Centers 69
4.2 Mobility of Connected Sub-Mechanisms 71
4.3 Examples: Collapsible Polyhedra 74
4.4 Conclusions 78

Chapter 5 Graphical Synthesis of Pantomeshes 79

5.1 Graphical Determination of Mobility 79
5.2 Graphical Systhesis Procedure 80
5.3 Discussion 84
5.4 Conclusions 85

Chapter 6 Numerical Synthesis of Pantomeshes 86

6.1 Shape Synthesis 86
6.2 Example: One Position, Three Point-on-Curve Constraints 88
6.3 Example: Three Rows, Two Positions 91
6.4 Conclusion 96

Chapter 7 Multi-Link Spherical Joints 98

7.1 Background 98
7.2 Rigid (Non-compliant) Multi-Link Spherical Joints 100
7.3 Compliant Multi-Link Spherical Joints 102
7.4 Crossed-Tendon Multi-Link Spherical Joints 106
7.5 Summary 107

Chapter 8 Breast Cradle **109**

8.1 Current State of the Art 110

8.2 Design Approach 110

8.3 Closed Pantomesh Rings 111

8.4 Designed Example 113

8.5 Discussion 115

8.6 Conclusions 115

Chapter 9 Conclusions **116**

Bibliography **117**

Bibliography **118**

Appendix A Nomenclature **122**

Appendix B Numerical Analysis Code for Shape Synthesis **126**

B.1 Introduction 126

B.2 Program Listing 126

List of Figures

1.1 Early drawing of a pantograph. 2

1.2 Lazy tongs. 2

1.3 Progression of pantomesh constituents. 3

1.4 Link and element classification. 5

1.5 Kempe's eight-bar planar pantograph-style linkage. 6

1.6 Camp stool with a four link pair collapsible mechanism. 7

1.7 Deployable antenna mast 8

1.8 Kinematic interpretation of Hoberman's angulated element 8

1.9 Collapsible Hoberman Mini Sphere® toy 9

1.10 Radial collapsing of link pairs 10

1.11 Collapsible chair patent, with angled supports. 11

1.12 Example of umbrella from U.S. Patent #4,193,415 12

1.13 Collapsible, variable-curvature pantomesh dome. 13

1.14 Hoberman's expandable dome. 14

2.1 Bounding quadrilateral for a straight-link pantograph element. . . . 18

2.2 Simple straight-line pantograph elements 18

2.3 Skewed straight-link pantograph element with an normal pivot. . . . 20

2.4 Skewed straight-link pantograph element with an angled pivot. . . . 23

2.5 Coordinate system for calculating the bounding tetrahedron of a skew, angled-pivot straight-link pantoelement. 23

2.6 Bent-link pantograph element. 26

2.7 Skewed bent-link pantograph element with an normal pivot. 30

2.8 Differently-shaped angled-pivot bent links with identical kinematic properties. 32

2.9 Skewed bent-link pantograph element with an angled pivot. 33

2.10 Shared-edge dual tetrahedron model of a skew pantograph element. 36

3.1 Mobility requirements for a straight-link pantograph patch. 40

3.2 Illustration of Carnot's polygon theorem. 61

4.1 Edge-vertex representation of rigid bodies. 64

4.2 Spherical joint between two triangular links 65

4.3 Spatial 2R chain 66

4.4 Spherical 2R chain 66

4.5 Spherical four-bar polyhedral model 67

4.6 Spatial eight-bar with mobility of 2 68

4.7 Basic translational joints 69

4.8 Spherical loop with remote center 70

4.9 Submechanisms having a remote center 72

4.10 Scissor submechanisms 73

4.11 Three scissor linkages with a common center 73

4.12 Hoberman Mini Sphere® 74

4.13 Hoberman-style scissor element 75

4.14 Collapsible cube 76

4.15 Collapsible truncated icosahedron 77

5.1 Graphical synthesis — the first two columns. 80

5.2 Graphical synthesis — the $(n-1)^{th}$ column. 82

5.3 Graphical synthesis — the final column. 84

6.1 Solution for a two-row, six column straight-line pantograph mesh. . 91

6.2 Superimposed pantograph elements used to find synthesized pivot. . 93

6.3 Solution for two specified shapes. 96

7.1 Existing multiple collocated spherical joints. 99

7.2 Existing compliant joints suitable for pantomeshes. 100

7.3 Rigid multi-link spherical joint. 100

7.4 Pantomesh with rigid multi-link spherical mechanism. 102

7.5 Double-sided compliant multi-link spherical joint. 103

7.6 One-sided compliant multi-link spherical joint. 104

7.7 Crossed-tendon spherical joint. 106

8.1 An example of bilateral compression plates. 111

8.2 Proposed breast stabilization device. 111

8.3 Single layer pantograph mesh breast cradle. 112

8.4 Schematic profile representation of a breast. 113

8.5 Schematic of a curved link with a set pivot ratio. 115

List of Tables

1.1 Sections detailing the six types of pantograph elements 4

1.2 Comparison of previous work. 6

2.1 Comparison of pantograph elements 36

3.1 Equal-angle configurations for bent-link pantopatch mobility. 52

3.2 Kite and parallelogram configurations for bent-link pantopatch mobility. 53

4.1 Translation from polyhedral parameters to polygonal mobility model. 78

6.1 Shape synthesis variables. 87

6.2 Shape synthesis equations. 88

7.1 Relative compliance in several degrees of freedom for chosen designs. 105

7.2 Comparison of multi-link spherical joints. 107

Chapter 1

Introduction

The pantograph has a long history in kinematics; it is one of the oldest multi-link examples of displacement magnification. Early uses were for transcription, then for collapsible structures, and even for children's toys. Many pantograph-style linkages have been developed using multiple pantograph-like units that obey certain geometric restrictions, but have yet to be described in a general fashion. This thesis will define the *pantomesh*, an assemblage of multiply-connected pantograph elements, using the most basic of requirements: mobility of the assembled links. But, first, some definitions:

1.1 Definitions and Taxonomy

A *pantograph* is an assembly of links often used for copying or scaling drawings[1]. Typically, four links are arranged in a parallelogram fashion as shown in Figure 1.1 [2]. A *pantograph element* may be part of this pantograph; here it is defined as a pair of two links joined by a central pivot joint. This pair of links is also referred to as a *scissor pair* or *duplet* in related literature [3]. This thesis defines a *pantomesh* as an articulated surface created by connecting pantograph elements in a patchwork fashion by spherical joints, using principles of pantograph construction and exhibiting single degree-of-freedom pantograph-like motion.

A discussion of rigid-body kinematics should begin with a review of the terminology. A *linkage* contains links connected by joints such as pins or sliders. In

[1]The term *pantograph* is also used for electrical power transmission from overhead lines for electric trains [1], but that linkage will not be discussed here.

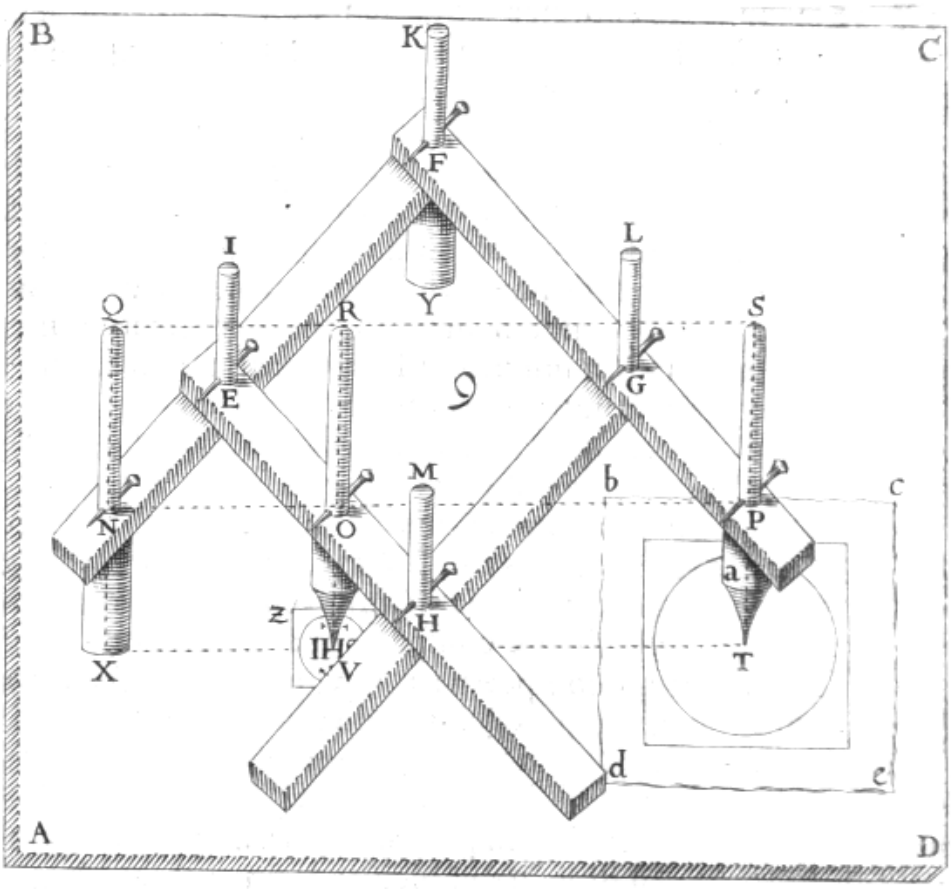


Figure 1.1: Early drawing of a pantograph [2].

three-dimensional space, a link's position and orientation may be specified exactly by six parameters: three positional coordinates (e.g. x, y, and z) and three orientation parameters (e.g. θ_x, θ_y, and θ_z). A completely unconstrained link is said to have six *degrees of freedom* or *DOF* [4]. When a link is connected to another with a joint, its motion is constrained (relative to the other link). The concept of degrees of freedom becomes more complex as the number of links increase, as in the the case of lazy tongs (see Figure 1.2).

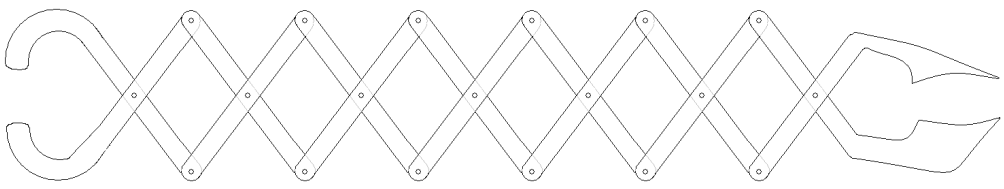
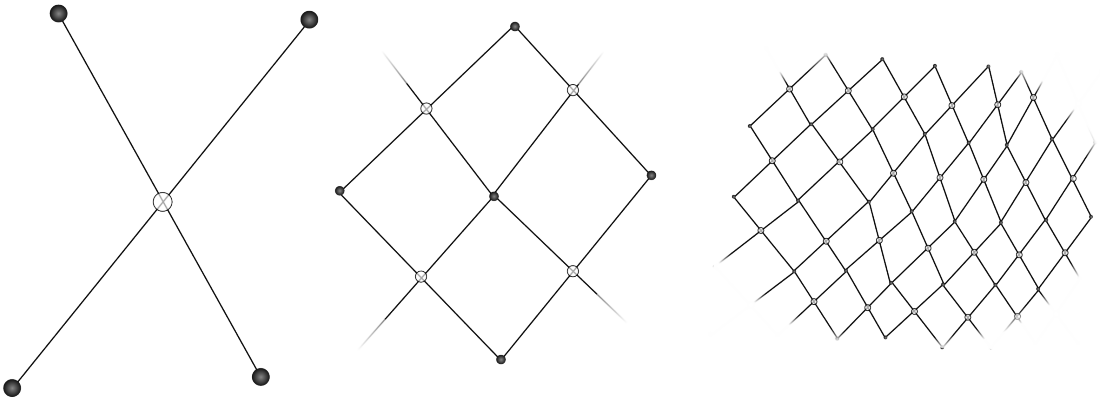


Figure 1.2: Lazy tongs.

The previously-mentioned linkages are composed of *rigid* links and joints. A rigid

link is self-explanatory, but a rigid joint is not immobile; rather it is the result of movement between two separate rigid surfaces. This is in contrast to *compliant* links and joints, which provide mobility by deflection of flexible components [5]. *Spatial linkages*, as compared with planar linkages, have mobility in three dimensions. Naturally, spatial mechanisms may contain any combination of rigid and compliant links and joints. It is through spatial mechanisms that complex three-dimensional motion may be achieved.



(a) Element (b) Patch (c) Mesh

Figure 1.3: Progression of pantomesh constituents.

A pantomesh, as defined here, has the following distinguishing characteristics as compared to previous work discussed in the following section: First, they are built using pantograph elements (see Figure 1.3(a)), which are connected to each other by *spherical* joints. Spherical joints allow three rotational degrees of freedom; a ball-and-socket joint is a commonly-known example of a spherical joint. A *pantograph element* is a pair of two links joined by a central pivot joint. Assuming that each link is also joined to other pantograph elements at both endpoints, each is a *tertiary* link in conventional linkage theory. The spherical joints at the endpoints allow a varying angle between pantograph elements and greater shape-changing during actuation. Secondly, the mobility of the entire pantomesh is calculated using individual *pantopatches* like those shown in Figure 1.3(b), rather than evaluation of the pantomesh as a whole, shown in Figure 1.3(c). Finally, the pantograph elements may contain straight or bent links, planar or skew endpoints, and normal or angled pivots (see Table 1.1 and Figure 1.4).

There are two primary classification schemes for pantograph elements with three types each, as shown in Figure 1.4. First, the elements contain one of two *link types*:

		Straight-Link	Bent-Link
	Planar	§2.2.1, §3.2.1	§2.3.1, §3.3.1
Skew {	**Normal-Pivot**	§2.2.2, §3.2.2	§2.3.2, §3.3.2
	Angled-Pivot	§2.2.3, §3.2.3	§2.3.3, §3.3.3

Table 1.1: Sections detailing the six types of pantograph elements

straight links or bent links. A *straight-link* pantograph element is one where the pivot axis intersects the *link span*, the line between the endpoints for both links. A pair of links with a non-intersecting pivot axis is a *bent-link* pantograph element. Second, a *planar* pantograph element is one where the four endpoints of the links are coplanar; otherwise they are *skew*. This chapter is first divided into straight-link and bent-link elements, and then subdivided into planar and skew configurations for each link type. Skew elements are further divided into normal-pivot and angled-pivot. These many options create a wide design space for many shapes and motions.

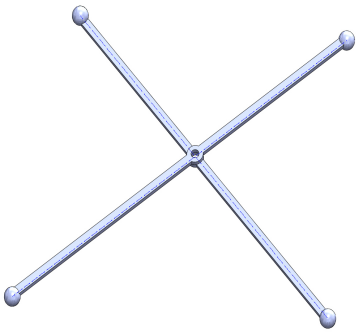
Previous work on pantomeshes has focused on two areas: constant-curvature bent-link pantomeshes and simple straight-link pantomeshes with isosceles trapezoidal elements (see Table 1.2). Those areas, their early planar forbears, and closely related linkages, are detailed in the following section.

1.2 Previous Work

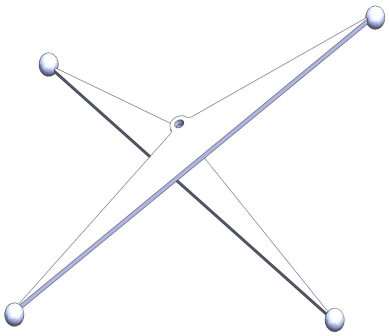
Previously-developed systems of pantograph-style linkages have characteristics that distinguish them from the pantomesh linkage defined above.

1.2.1 Planar Pantrograph-style Linkages

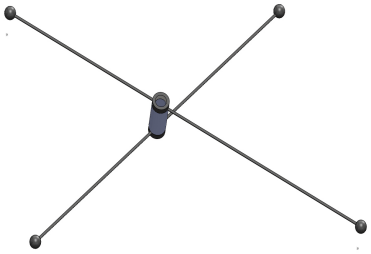
One of the earliest examples of a pantograph-style linkage was developed by Kempe [6] as an examination of mobile eight-bar linkage (see Figure 1.5). The links, all located in the same plane, are connected by pin joints. As discussed in the next sub-section, Hoberman has developed planar systems that are connected to make three-dimensional linkages.



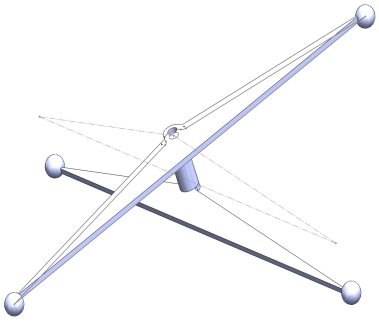
(a) Planar Straight-Link.



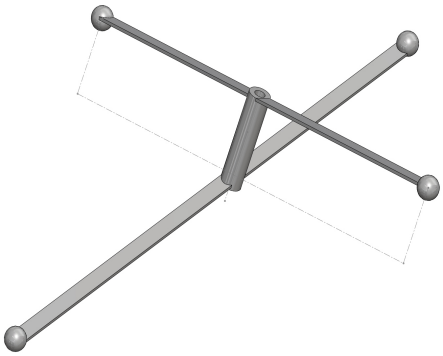
(b) Planar Bent-Link.



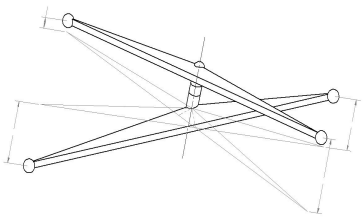
(c) Skew, Normal-Pivot Straight-Link.



(d) Skew, Normal-Pivot Bent-Link.



(e) Skew, Angled-Pivot Straight-Link.



(f) Skew, Angled-Pivot Bent-Link.

Figure 1.4: Link and element classification.

1.2.2 Constant-Curvature Pantomeshes

Close relatives to planar pantomeshes are general constant-curvature pantomeshes. Much like a flat membrane stretched over a ball, these linkages maintain their basic shape during their motion; only a linear scaling is increased. Each of the panto-

Table 1.2: Comparison of previous work.

Description	Dimensions	# Links Min	# Links Max	Topology	Connecting Joint	Mobility Criterion
Kempe [6]	Planar	8	8	bent-link	revolute	
Hoberman's doubly-curved [7]	Compound Planar	4	∞	bent-link	gusset	Half-Angle
GAE [8]	Planar			bent-link	gusset	Parallel-ogram
Wohlhart [9]	Spatial	4	∞	bent-link	gusset	
General Pantomesh	Spatial	4	∞	any	spherical	multiple

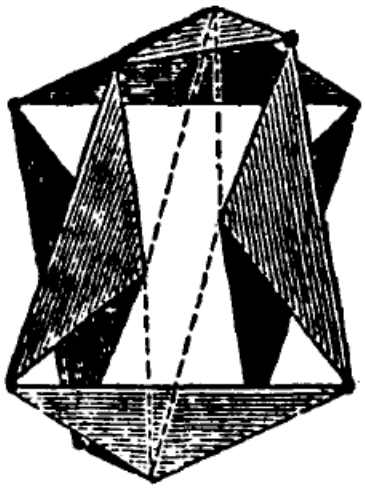


Figure 1.5: Kempe's eight-bar planar pantograph-style linkage [6].

graph elements are connected by a *gusset joint*, or two revolute joints whose axes intersect. Constant-curvature pantomeshes have constant angles between their constituent pantograph elements, and therefore do not require spherical joints between them as variable-curvature pantomeshes do.

United States Patent #244,215 describes a camp stool that is in this configuration (see Figure 1.6), which is the earliest mention of such a system in the U.S. patent literature [10]. The topology of these linkages is simple: each pantograph element

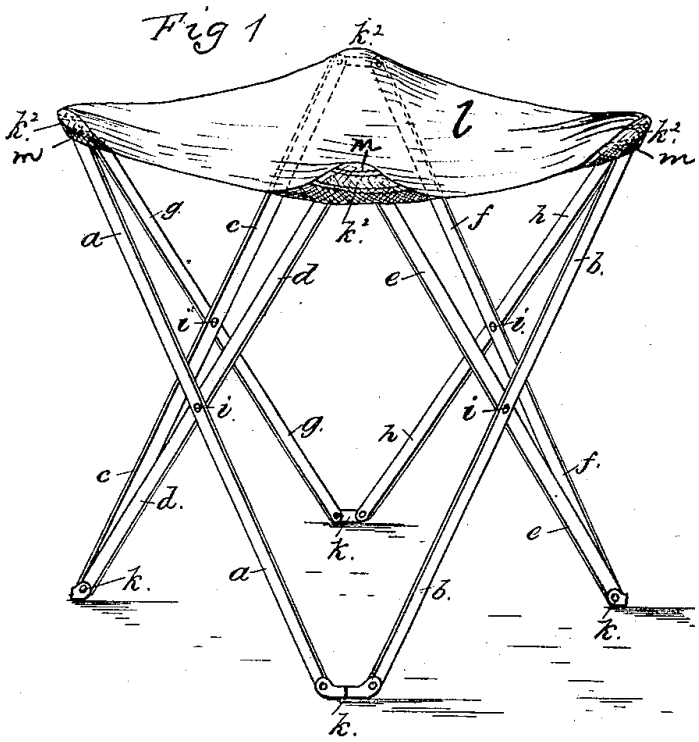


Figure 1.6: Camp stool with a four link pair collapsible mechanism [10].

is joined by a revolute joint in the center of each link. A bracket connects the revolute joints for the bases of each link pair and for the summits of each link pair, until the last link pair connects with the first. All points collapse to the same axis at the same rate, given the midpoint location of the pin joint in each pantograph element. At the same time, the height of the chair grows along this axis.

These scissor-pair units have been stacked to create deployable antenna mechanisms [11, 12] as shown in Figure 1.7. As with the camp stool, the deployable antenna uses a fixed-angle bracket to join pantograph elements at their base and summit. In Figure 1.7(d), the fixed angle is 120° for a three-sided device, which is forced because each scissor pair is the same size, thus creating an equilateral triangle in cross-section. The chair in Figure 1.6 has 90° (when viewed from the top) for a four-sided device. While the size of each scissor-pair is also the same for the chair, the 90° angle between neighboring scissor pairs must be enforced by 90° gusset joints. Otherwise, the cross-sectional shape is a rhombus, allowing opposing gusset angles to be equal and all four totalling 360°.

The antenna stacks units by joining the pin joints of the summit of one level to the base pin joints of the next level, similar to lazy tongs. As opposed to the chair

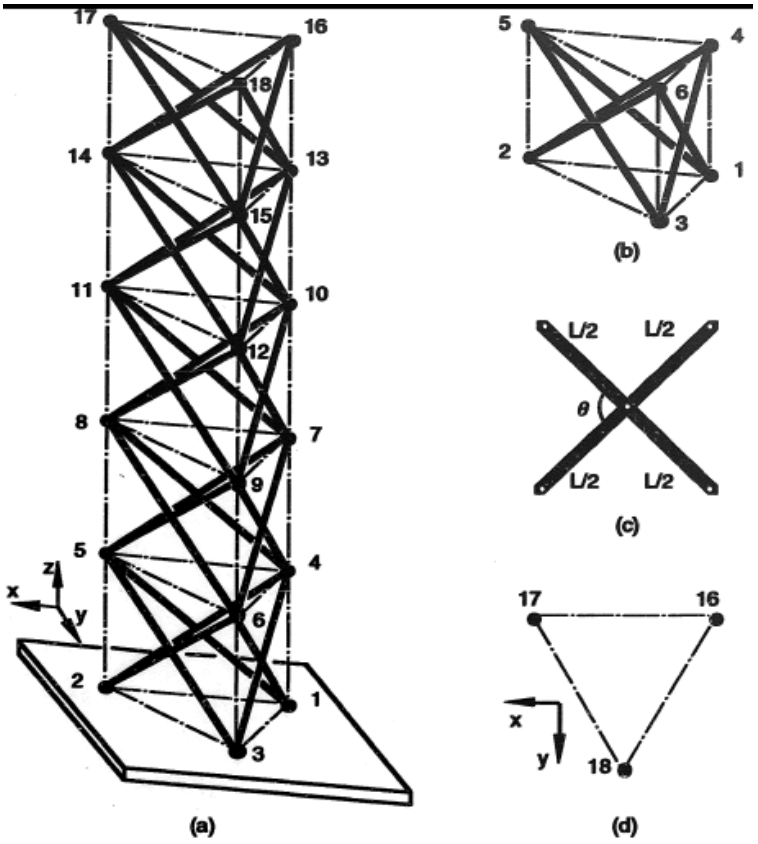
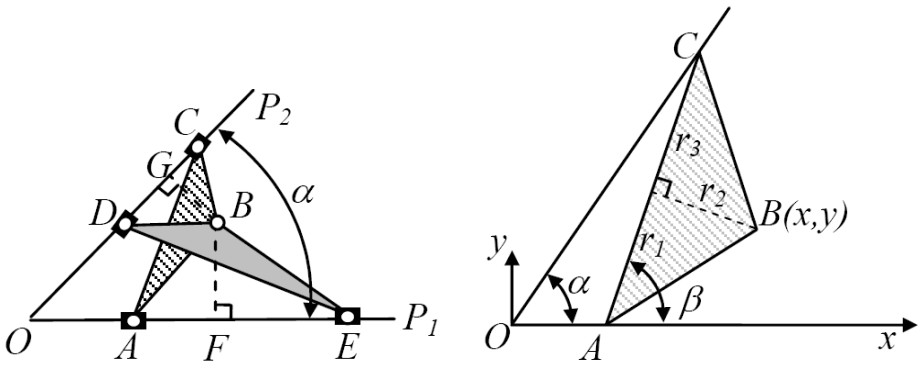


Figure 1.7: Deployable antenna mast: (a) backbone of the antenna; (b) a module; (c) a pantographic element; (d) top view [11].

design, the antenna is intended to collapse downward to a plane when not in use, rather than collapsing radially to an axis.



(a) A pair of general PRRP linkages sharing a common coupler point B. [7, 13]

(b) A PRRP Linkage and its Coupler Point

Figure 1.8: Kinematic interpretations of Hoberman's angulated element.

One special application of planar pantomeshes is the Hoberman Sphere [7] as

shown in Figure 1.9. The Hoberman pantograph element is the most widely-known given its visibility in collapsible toys [7]. This and other fixed-angle pantograph elements were thoroughly examined in a recent paper by Patel and Ananthasuresh [13]. Each pantograph element can be represented as two joined links that slide along prismatic joints at the ends as shown in Figure 1.8. Each pantograph element can be represented as two joined links that slide along prismatic joints at the ends as shown in Figure 1.8. Each link is part of a prismatic-revolute-revolute-prismatic (PRRP) linkage (as shown in Figure 1.8a), and the pantograph element is joined at each links' respective coupler points. The coupler points are labeled as **B** in Figure 1.8 and are equivalent to the pivot of a pantograph element. The angle α remains constant so that the symmetry lines OP_1 and OP_2 remain stationary throughout the linkage.

The collapsible inventions by Hoberman and others [7, 8, 13] are not connected by spherical joints, and therefore are only a special case in this thesis. By maintaining the same angle between opposite sides of the bounding quadrilateral, the angle between pantograph elements is constant during its range of motion, and therefore a bracket with two revolute joints at a fixed angle may be used to connect pantograph elements. Other fixed-angle pantograph elements are possible, as discussed in [13]. The Generalized Angular Elements (GAE), defined by You and Pellegrino [8], also maintain an fixed angle during deployment.

(a) Collapsed Hoberman Sphere®. (b) Expanded Hoberman Sphere®.

Figure 1.9: Collapsible toy based on Hoberman's expandable truss patent [7, 14].

The Hoberman linkage has many links joined in radially-collapsing pantograph

elements that are arranged so that they move toward a common point (see Figure 1.10). In fact, it consists of several planar great circle linkages that are connected at their intersections. This creates a three-dimensional shape with multiple planar pantomeshes. Although the overall linkage is spatial, the combination of planar pantomeshes makes this a *compound* pantomesh, which will not be derived in further detail in this thesis.

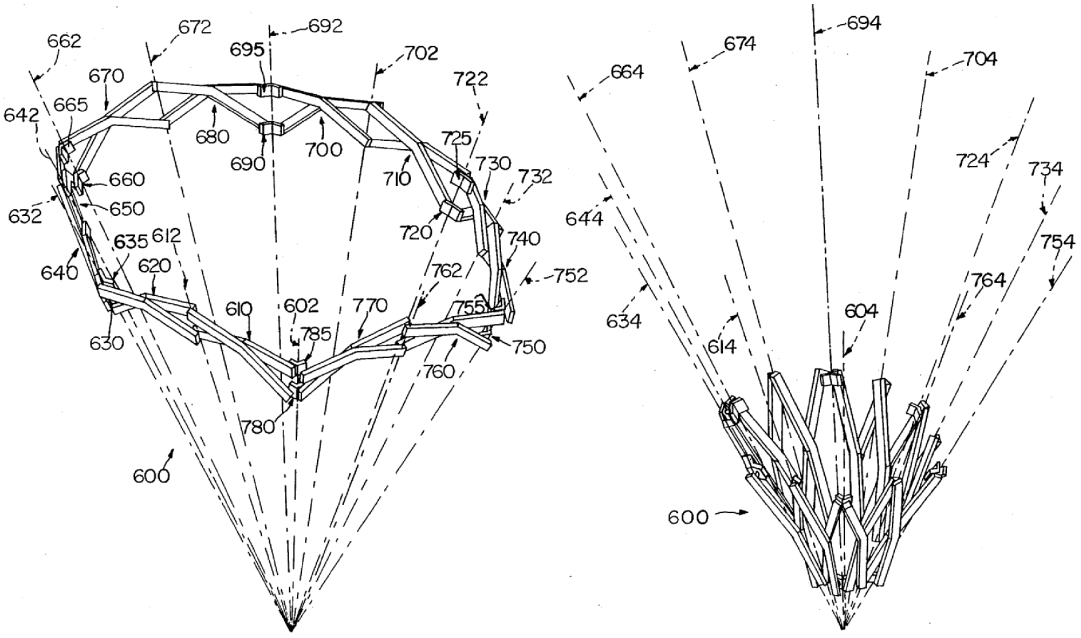
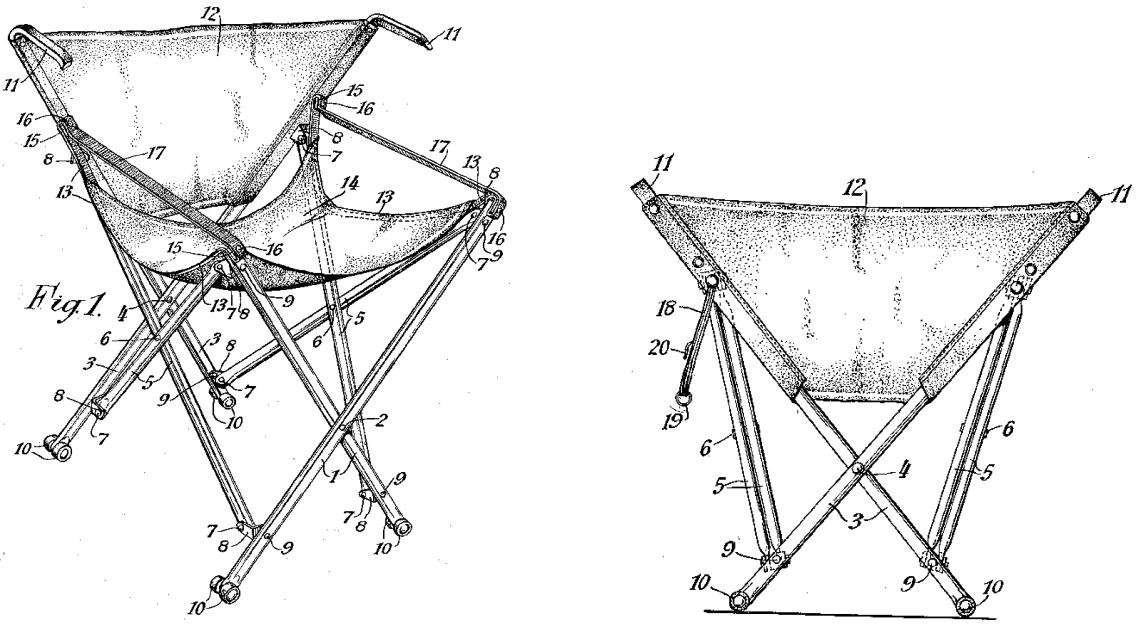


Figure 1.10: Radial collapsing of link pairs [7]. The image on the left is expanded; the image on the right is collapsed.

The notion of a retractable roof using straight-link pantograph elements was first discussed by Pinero [15]. Escrig discovered early uses of expandable systems for space structures [16]. Both of these designs use compound planar pantomeshes. Other dome designs are created using variable-curvature pantomeshes, as described in the next section.

1.2.3 Variable-Curvature Pantomeshes

As compared with constant-curvature pantomeshes, variable-curvature pantomeshes have varying angles between pantograph elements. U.S. Patent #3,124,387 is a similar linkage to the basis camp stool, although the pantograph elements are not joined at their centers (see Figure 1.11) [17]. This system is similar to the topic of this thesis, where offset revolute axes create different rates of collapsing. The kine-



(a) Perspective View (b) Rear View

Figure 1.11: Collapsible chair patent (#3,124,387), with angled supports [17].

matics of this design are not explicitly expressed in the patent literature. In brief, the offset pin joint in the front and rear pantograph elements cause the top of the chair to collapse faster than the bottom. The side pantograph elements appear to have the pin joint at the link midpoints, therefore allowing the front and back of the chair to collapse simultaneously. The primary advantage of the offset pin joint arrangement appears to be that the base of the chair is not as wide as the top of the chair when deployed. This chair design is not in common use, probably because it is less stable than wider-base chairs.

Several umbrellas also appear to use offset-axis pantograph elements [18, 19], although the exact nature of the link dimensions is not specified. Furthermore, Gantes and others have examined deployable dome with similarly constructed elements [9,20]. The umbrella in Figure 1.12 appears to use pantograph elements (**54** and **56**) that have a pivot slightly closer to the top of the umbrella, thereby allowing the outer part to expand faster than the inner part when deployed. However, the patent specification and claims describe pivots at the link midpoints. Additionally, all joints are claimed to be pin joints, but the angle between link pairs varies during deployment.

Both the umbrella and angled-support chair have designs similar to that patented by Atake [21]. Each pantograph element has a perimeter defined by an isosceles

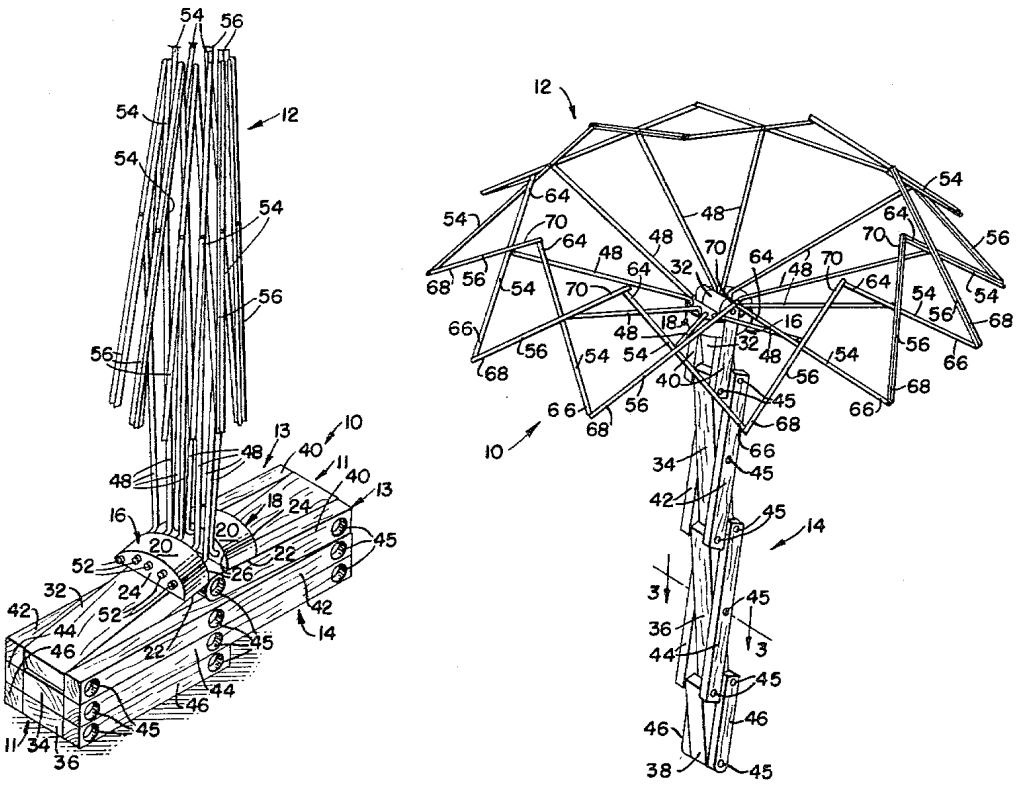


Figure 1.12: Example of umbrella from U.S. Patent #4,193,415 [18].

trapezoid. Because the relative angle of pantograph element planes changes during actuation, they must be connected by multiple revolute joints.

Finally, more dome-style linkages are presented by Hoberman [22]

While all of the previously-mentioned linkages have simliar properties to pantomeshes as defined by this thesis, they lack some or all of the following qualities:

1. Pantograph elements joined by spherical joints.
2. Mobility determined by pantopatch analysis.
3. Ability to offset element links to create *skew* elements, and therefore normal and angled pivots.

1.3 Scope of Thesis

This thesis details single degree-of-freedom spatial linkages defined as pantomeshes. It is the purpose of this thesis to describe a class of mechanisms that can provide

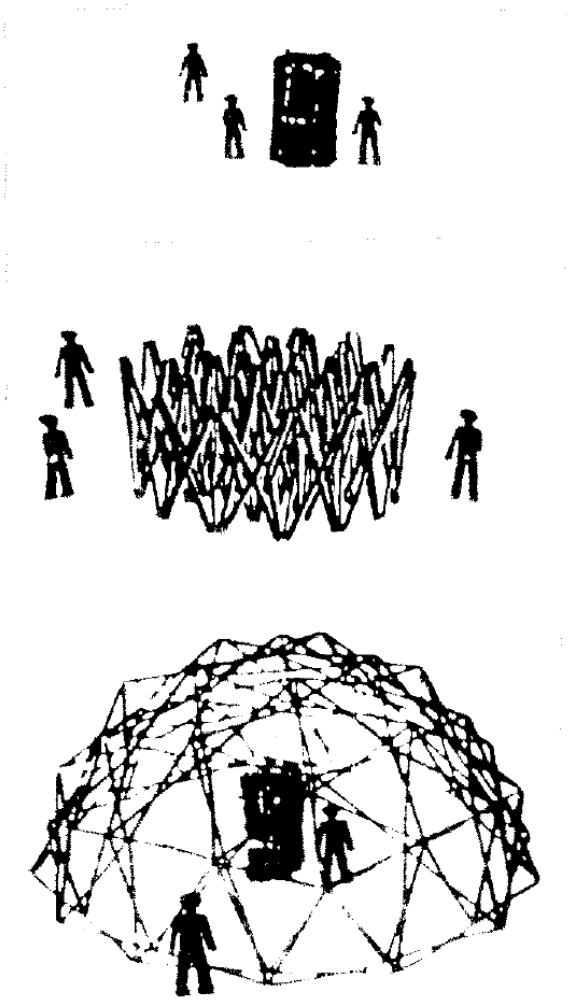


Figure 1.13: Collapsible, variable-curvature pantomesh dome [20].

a meaningful radial force or define a shape in general. There will be additional discussion of the specific case of a breast stabilization device for cancer-related interventions.

1.4 Outline of the Thesis

This thesis contains three main parts:

1. Examining pantomesh properties, starting from kinematics of elements, then to mobility of patches, and finally onto assembled pantomeshes.
2. Synthesis of pantomeshes to fit some desired criteria or shape.
3. Practical design considerations and applications, including multi-link spher-

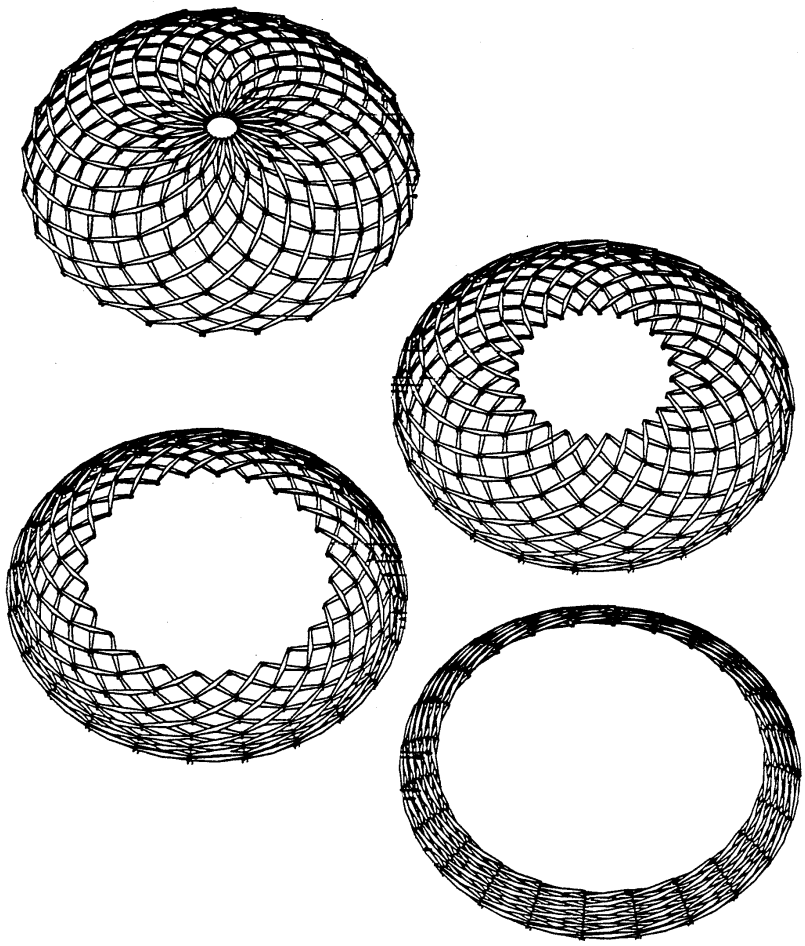


Figure 1.14: Hoberman's expandable dome [22].

ical joints and an example pantomesh designed for an application.

Chapters 2 and 3 detail pantograph elements and their mobility within closed pantomeshes. By properly assembling closed pantomeshes in two or more rows, a single degree-of-freedom linkage is created. Nomenclature introduced in these chapters is summarized in Appendix A.

A unique, graphical method for mobilty of a number of polygonal blocks is presented in Chapter 4.

Two methods for creating new pantomeshes, or *synthesizing* them, are also presented. Chapter 5 describes a method for synthesizing pantomeshes graphically with a computer-aided-design system. This is followed by Chapter 6, which shows a method of synthesizing such linkages numerically. An example program listing is given in Appendix B.

The complexity of a large, multi-element pantomesh would require a significant manufacturing and assembly effort if rigid joints are used. Therefore, a multi-link compliant spherical joint is presented in Chapter 7 for use in a compliant version of a closed pantograph element chain. This compliant linkage could be manufactured with vacuum forming, rapid prototyping, or even some molding techniques.

After closed pantograph element chains have been described, Chapter 8 details the design of a new mechanism to provide radial pressure for the purpose of stabilizing the human breast during cancer-related interventions. Chapter 9 concludes the thesis.

Chapter 2

Kinematics of Pantograph Elements

This chapter describes the basic kinematics of pantomeshes and, in particular, the constituent elements of pantomeshes: the pantograph element. First, nomenclature common to all pantograph elements is discussed, followed by a detailed analysis of all six pantograph element types.

2.1 Nomenclature

As discussed in the introductory chapter, pantograph elements may be classified in several different ways. First, the topology of each link may be either *straight* or *bent*, depending upon the location of the pivot axis relative to the two link endpoints. If the line segment drawn between endpoints (or *link span*) intersects the pivot axis, it is a straight link. Otherwise, it is a bent link. Typically, elements consist of a pair of similar links, and therefore are referred to as either *straight-link elements* or *bent-link elements*.

Pantograph elements may also be classified as either *planar* or *skew*, depending on the position of all four link endpoints. If those endpoints are coplanar, the element is planar; otherwise it is skew.

The pivot axis of a pantograph element may be either *normal* or *angled*, relative to both link spans. If the pivot axis is perpendicular to both link spans, it is a normal pivot; otherwise, it is angled.

Pantograph elements may be assembled in rows and columns; a pantograph element in row A and column 1 is defined by four link endpoints: $\boldsymbol{\mathcal{P}}_{A1}$, $\boldsymbol{\mathcal{P}}_{A2}$, $\boldsymbol{\mathcal{P}}_{B1}$,

and $\boldsymbol{\mathcal{P}}_{B2}$ (see Figure 2.1). Each point is defined by three variables in space: $\boldsymbol{\mathcal{P}}_i = (x_i, y_i, z_i)$. The distance between the endpoints (along the link span) of the *right-handed* link is L_{A1} and the *left-handed* is M_{A1}, as shown. The assignment of link handedness is arbitrary, based on a cylindrical-shaped closed pantomesh. The distance from $\boldsymbol{\mathcal{P}}_{A1}$ to the center pivot is e_{A1}, and from the center pivot to $\boldsymbol{\mathcal{P}}_{B2}$ is f_{A1}. Similarly, the distance from $\boldsymbol{\mathcal{P}}_{A2}$ to the center pivot is g_{A1}, and from the center pivot to $\boldsymbol{\mathcal{P}}_{B1}$ is h_{A1}. In the case of straight-link pantograph elements, $L_{A1} = e_{A1} + f_{A1}$ and $M_{A1} = g_{A1} + h_{A1}$, as shown in Figure 2.1. Furthermore, the exterior dimensions of the pantograph element perimeter, or *bounding quadrilateral*, are defined as $\boldsymbol{j_{A1}}$, $\boldsymbol{k_{A1}}$, $\boldsymbol{j_{B1}}$, and $\boldsymbol{k_{A2}}$; these are the "south," "west," "north," and "east" borders, respectively, as shown in Figure 2.1.

The following sections are arranged according to the topology of the links: straight or bent, planar or skew, and normal or angled pivots (for the skew case). These cases assume that both links are of the same topology, where in practice that may not be. In that case, the element should be analyzed as the most complex version of the combined links.

2.2 Straight-Link Pantograph Elements

A *straight-link* pantograph element is one where the pivot joint is on the line between the endpoints for both links. A diagram of such a pantograph element is shown in Figure 2.1. Straight-link elements may be further classified into planar and skew, and the skew elements can be classifed further yet into normal and angled pivots.

2.2.1 Planar Straight-Link Pantograph Elements

Kinematically speaking, planar pantograph elements consisting of straight links are the most simple. Two nearly trivial examples are shown in Figure 2.2, where the pivot occurs at the midpoint of both link spans, forming rectangular or parallelogram bounding quadrilaterals.

For the general straight-link pantograph element shown in Figure 2.1, the sides of the bounding quadrilateral may be calculated as follows. The left side of this bounding quadrilateral, $\boldsymbol{k_{A1}}$, may be found using the cosine law as shown in Eqn. (2.1). Then $\cos\boldsymbol{\psi_{A1}}$ can be expressed in other terms by again invoking the

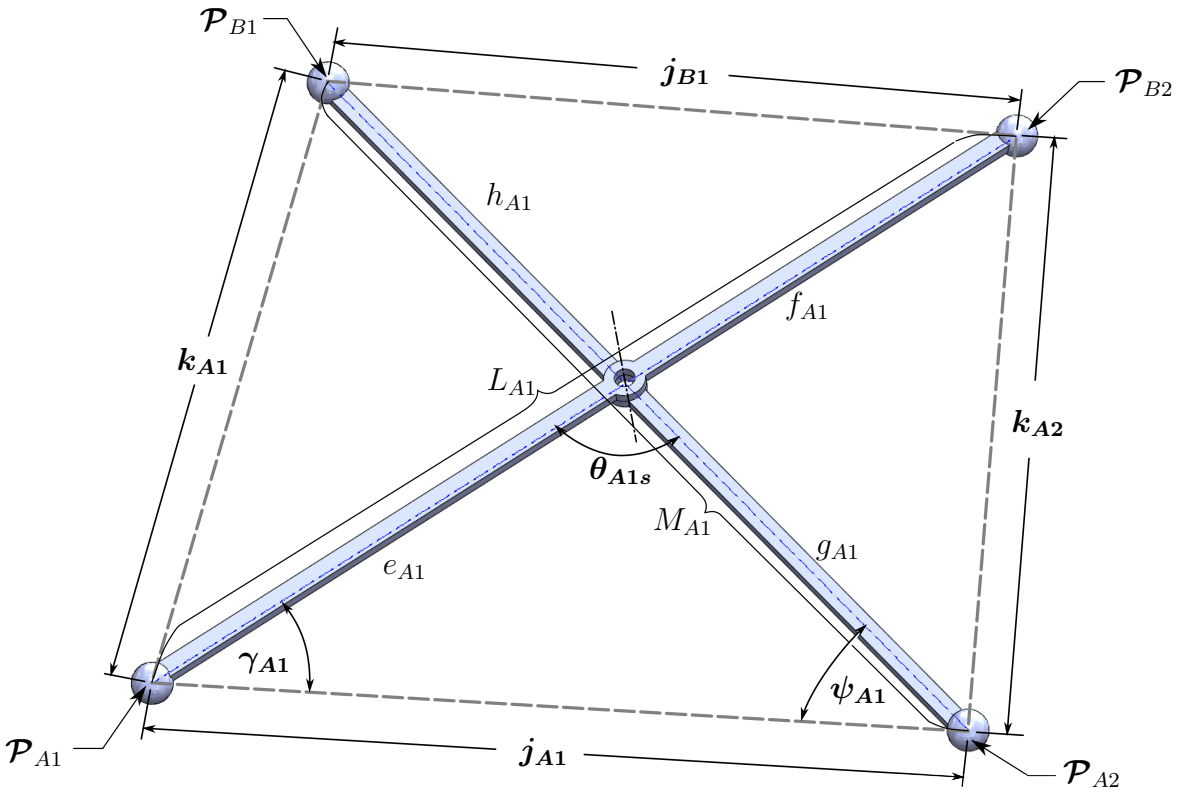
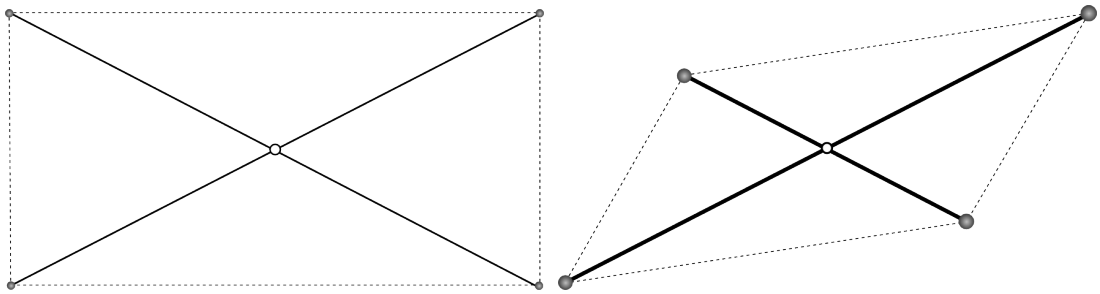


Figure 2.1: Bounding quadrilateral for a straight-link pantograph element.



(a) Rectangular pantograph elements (b) Parallelogram pantograph elements

Figure 2.2: Simple straight-line pantograph elements

cosine law in Eqn. (2.2), thereby obtaining Eqn. (2.3) by substitution.

$$\boldsymbol{k}_{\boldsymbol{A1}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + M_{A1}^2 - 2\boldsymbol{j}_{\boldsymbol{A1}} M_{A1} \cos \boldsymbol{\psi}_{\boldsymbol{A1}} \tag{2.1}$$

$$\cos \boldsymbol{\psi}_{\boldsymbol{A1}} = \frac{\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - e_{A1}^2}{2\boldsymbol{j}_{\boldsymbol{A1}} g_{A1}} \tag{2.2}$$

$$\boldsymbol{k}_{\boldsymbol{A1}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + M_{A1}^2 - \left(\frac{M_{A1}}{g_{A1}}\right)(\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - e_{A1}^2) \tag{2.3}$$

$$\boldsymbol{k}_{\boldsymbol{A1}}^2 = \left(\frac{M_{A1}}{g_{A1}}\right) e_{A1}^2 + M_{A1} h_{A1} - \left(\frac{h_{A1}}{g_{A1}}\right) \boldsymbol{j}_{\boldsymbol{A1}}^2 \tag{2.4}$$

The right side, $\boldsymbol{k_{A2}}$, is found in a manner very similar to $\boldsymbol{k_{A1}}$, so the equation will not be derived here. The final expression, Eqn. (2.5), is similar to Eqn. (2.4) except some variables are switched:

$$\boldsymbol{k_{A2}^2} = L_{A1}f_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right) g_{A1}^2 - \left(\frac{f_{A1}}{e_{A1}}\right) \boldsymbol{j_{A1}^2} \tag{2.5}$$

The top side of the bounding quadrilateral, $\boldsymbol{j_{B1}}$, may be found with a similar solution of the cosine law, only noting that $\boldsymbol{\theta_s}$ is equal to $\boldsymbol{\theta_n}$ on either side of the intersecting diagonals (see Figure 2.1). The final expression is given in Eqn. (2.6).

$$\boldsymbol{j_{B1}^2} = \left(\frac{f_{A1}^2}{e_{A1}^2} - \frac{f_{A1}h_{A1}}{e_{A1}g_{A1}}\right) e_{A1}^2 + \left(\frac{h_{A1}^2}{g_{A1}^2} - \frac{f_{A1}h_{A1}}{e_{A1}g_{A1}}\right) g_{A1}^2 + \left(\frac{f_{A1}h_{A1}}{e_{A1}g_{A1}}\right) \boldsymbol{j_{A1}^2} \tag{2.6}$$

Another useful relation is between opposite sides of the pantograph element bounding quadrilateral, as in Eqn. (2.6) above and also between the left and right sides as in Eqn. (2.7) below. The derivation of this equation is very similar to Eqn. (2.6) and is not shown here.

$$\boldsymbol{k_{A2}^2} = \left(\frac{f_{A1}^2}{e_{A1}^2} - \frac{f_{A1}g_{A1}}{e_{A1}h_{A1}}\right) e_{A1}^2 + \left(1 - \frac{f_{A1}h_{A1}}{e_{A1}g_{A1}}\right) g_{A1}^2 + \left(\frac{f_{A1}g_{A1}}{e_{A1}h_{A1}}\right) \boldsymbol{k_{A1}^2} \tag{2.7}$$

Finally, the relation of $\boldsymbol{j_{B1}}$ to both $\boldsymbol{k_{A1}}$ and $\boldsymbol{k_{A2}}$ is useful in pantopatch mobility calculations in Ch. 3.

$$\boldsymbol{j_{B1}^2} = M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right) f_{A1}^2 - \left(\frac{h_{A1}}{g_{A1}}\right) \boldsymbol{k_{A2}^2} \tag{2.8}$$

$$\boldsymbol{j_{B1}^2} = L_{A1}f_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right) h_{A1}^2 - \left(\frac{f_{A1}}{e_{A1}}\right) \boldsymbol{k_{A1}^2} \tag{2.9}$$

These equations show a simple, linear relationship between the squares of the edge lengths of the bounding quadrilateral (i.e. j^2's and k^2's). These relations make assembled pantomeshes easier to synthesize and analyze. However, there is a drawback to co-planar endpoints: it is difficult to have a wide range of motion for the element, since the ends would run into each other. In practice, this limits the range of motion to about 120°. Some of these issues can be addressed by using skew pantograph elements.

2.2.2 Skew, Normal-Pivot Straight-Link Pantograph Elements

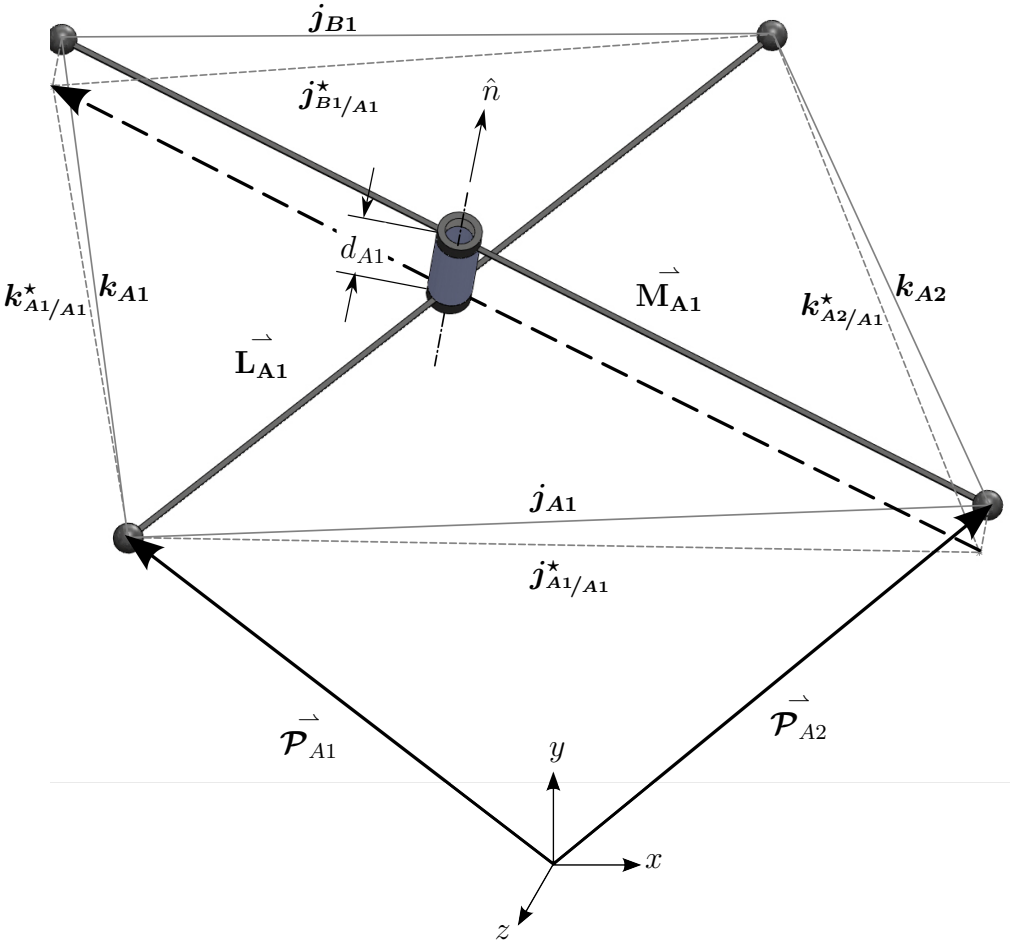


Figure 2.3: Skewed straight-link pantograph element with an normal pivot.

A skew pantograph element still has a *bounding quadrilateral*, however it is a technically a skew quadrilateral (i.e. non-planar). The distance between the two pantograph links is found by measuring the distance needed to *flatten* the skew bounding quadrilateal into a planar quadrilateral. This may be performed by setting the distance d between the non-parallel link span vectors $\vec{\mathbf{L}}_{\mathbf{A1}}$ and $\vec{\mathbf{M}}_{\mathbf{A1}}$ to zero, as shown in Figure 2.3. That distance is first found by finding the normal vector cross product of the two link spans expressed as vectors as shown in Eqn. (2.10), hereby referred to as the *pantograph normal*. This normal vector can be thought of as the normal of two parallel planes that contain $\vec{\mathbf{L}}_{\mathbf{A1}}$ and $\vec{\mathbf{M}}_{\mathbf{A1}}$, respectively. The distance between these parallel planes is the distance between the vectors going from an arbitrary origin to the start of each link vector, $\vec{\boldsymbol{\mathcal{P}}}_{A1}$ and $\vec{\boldsymbol{\mathcal{P}}}_{A2}$ (for $\vec{\mathbf{L}}_{\mathbf{A1}}$ and $\vec{\mathbf{M}}_{\mathbf{A1}}$, respectively) but taken only along the direction of the pantograph normal

vector. This is calculated using the dot product as shown in Eqn. (2.11).

$$\hat{n} = \frac{\overrightarrow{\mathbf{L_{A1}}} \times \overrightarrow{\mathbf{M_{A1}}}}{|\ \overrightarrow{\mathbf{L_{A1}}}\ ||\ \overrightarrow{\mathbf{M_{A1}}}\ |} \tag{2.10}$$

$$d_{A1} = \hat{n_{A1}} \cdot \overrightarrow{\boldsymbol{\mathcal{P}}}_{A1} - \hat{n_{A1}} \cdot \overrightarrow{\boldsymbol{\mathcal{P}}}_{A2} \tag{2.11}$$

For a planar element, $d = 0$.

The edge lengths of the skew bounding quadrilateral may be found by first calculating the edge lengths of the flattened bounding quadrilateral. Since the pantograph normal is perpendicular to both of the links, each true edge is the hypotenuse of the triangle containing the flattened quadrilateral edge length and the pantograph element distance (see Figure 2.3). Therefore, referring to the flattened dimensions with a star (e.g. $\boldsymbol{k^{\star}_{A1/A1}}$) the true edges may be calculated as follows:

$$\boldsymbol{j}^2_{\boldsymbol{A1}} = d^2_{A1} + (\boldsymbol{j}^{\star}_{\boldsymbol{A1/A1}})^2 \tag{2.12}$$

$$\boldsymbol{k}^2_{\boldsymbol{A1}} = d^2_{A1} + (\boldsymbol{k}^{\star}_{\boldsymbol{A1/A1}})^2 \tag{2.13}$$

$$\boldsymbol{k}^2_{\boldsymbol{A2}} = d^2_{A1} + (\boldsymbol{k}^{\star}_{\boldsymbol{A2/A1}})^2 \tag{2.14}$$

$$\boldsymbol{j}^2_{\boldsymbol{B1}} = d^2_{A1} + (\boldsymbol{j}^{\star}_{\boldsymbol{B1/A1}})^2 \tag{2.15}$$

furthermore,

$$(\boldsymbol{k}^{\star}_{\boldsymbol{A2/A1}})^2 = L_{A1} f_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right) g^2_{A1} - \left(\frac{f_{A1}}{e_{A1}}\right) (\boldsymbol{j}^{\star}_{\boldsymbol{A1/A1}})^2 \tag{2.16}$$

$$\boldsymbol{k}^2_{\boldsymbol{A2}} - d^2_{A1} = L_{A1} f_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right) g^2_{A1} - \left(\frac{f_{A1}}{e_{A1}}\right) (\boldsymbol{j}^2_{\boldsymbol{A1}} - d^2_{A1}) \tag{2.17}$$

$$\boldsymbol{k}^2_{\boldsymbol{A2}} = L_{A1} f_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right) g^2_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right) d^2_{A1} - \left(\frac{f_{A1}}{e_{A1}}\right) \boldsymbol{j}^2_{\boldsymbol{A1}} \tag{2.18}$$

$$(\boldsymbol{k}^{\star}_{\boldsymbol{A1/A1}})^2 = M_{A1} h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right) e^2_{A1} - \left(\frac{h_{A1}}{g_{A1}}\right) (\boldsymbol{j}^{\star}_{\boldsymbol{A1/A1}})^2 \tag{2.19}$$

$$\boldsymbol{k}^2_{\boldsymbol{A1}} - d^2_{A1} = M_{A1} h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right) e^2_{A1} - \left(\frac{h_{A1}}{g_{A1}}\right) (\boldsymbol{j}^2_{\boldsymbol{A1}} - d^2_{A1}) \tag{2.20}$$

$$\boldsymbol{k}^2_{\boldsymbol{A1}} = \left(\frac{M_{A1}}{g_{A1}}\right) e^2_{A1} + M_{A1} h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right) d^2_{A1} - \left(\frac{h_{A1}}{g_{A1}}\right) \boldsymbol{j}^2_{\boldsymbol{A1}} \tag{2.21}$$

with the other relations similarly formed.

$$\boldsymbol{k}_{\boldsymbol{A2}}^2 = L_{A1}f_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right)\left(g_{A1}^2 + d_{A1}^2\right) - \left(\frac{f_{A1}}{e_{A1}}\right)\boldsymbol{j}_{\boldsymbol{A1}}^2 \tag{2.22}$$

$$\boldsymbol{j}_{\boldsymbol{B1}}^2 = L_{A1}f_{A1} + \left(\frac{L_{A1}}{e_{A1}}\right)\left(h_{A1}^2 + d_{A1}^2\right) - \left(\frac{f_{A1}}{e_{A1}}\right)\boldsymbol{k}_{\boldsymbol{A1}}^2 \tag{2.23}$$

$$\boldsymbol{j}_{\boldsymbol{B1}}^2 = M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right)\left(f_{A1}^2 + d_{A1}^2\right) - \left(\frac{h_{A1}}{g_{A1}}\right)\boldsymbol{k}_{\boldsymbol{A2}}^2 \tag{2.24}$$

In summary, the skew, normal-pivot straight-link elements are similar to the planar case, except the addition of the constant d terms. This slight increase in complexity yields an advantage mentioned earlier: larger range of motion of elements because the endpoints do not necessarily collide during movement. If more complex motion is required to meet specifications of an assembled pantomesh, the pivot may be angled relative to the pantograph normal, as shown in the following section.

2.2.3 Skew, Angled-Pivot Straight-Link Pantograph Elements

When the pivot of a pantograph element is not perpendicular to one or both link spans, it is *angled* (see Figure 2.4). In addition to the pivot distance d, two additional variables are introduced: ν and ξ, where ν is the angle between the pivot axis and the link vector $\vec{\mathbf{L}}_{\mathbf{A1}}$ and ξ is the angle between the pivot axis and the link vector $\vec{\mathbf{M}}_{\mathbf{A1}}$. For a normal pivot, $\nu = \xi = 90^\circ$.

Using d as the distance between the pivot location on the link, the actual $\boldsymbol{j}_{\boldsymbol{A1}}$ may be calculated as follows for straight links. Consider the coordinate system shown in Figure 2.5 where the origin is where the pivot axis intersects with the right hand link. Furthermore, the $x^\star$-axis is parallel to $\vec{\boldsymbol{j}}^\star_{\boldsymbol{A1}}$ and $\vec{\mathbf{L}}_{\mathbf{A1}}$ is in the $x^\star y^\star$-plane. In this case, the $x^\star$ and $y^\star$ coordinates of the $\vec{\mathbf{d}}$ vector may be expressed both in terms of both pivot angles ν and ξ as measured from $\vec{\mathbf{L}}_{\mathbf{A1}}$ and $\vec{\mathbf{M}}^\star_{\mathbf{A1}}$, respectively.

$$d_{x^\star} = d\cos\nu\cos\boldsymbol{\gamma}^\star + t\sin\boldsymbol{\gamma} = d\cos\xi\cos(180^\circ - \boldsymbol{\psi}^\star) + s\sin(180^\circ - \boldsymbol{\psi}) \tag{2.25}$$

$$d_{y^\star} = d\cos\nu\sin\boldsymbol{\gamma}^\star - t\cos\boldsymbol{\gamma} = d\cos\xi\sin(180^\circ - \boldsymbol{\psi}^\star) - s\cos(180^\circ - \boldsymbol{\psi}) \tag{2.26}$$

The dummy variable t represents the perpendicular distance from $\vec{\mathbf{L}}_{\mathbf{A1}}$, while the dummy variable s represents the distance from $\vec{\mathbf{M}}^\star_{\mathbf{A1}}$. Together, two equations represent two different ways to arrive at the same point in the $x^\star$-$y^\star$ plane. Using

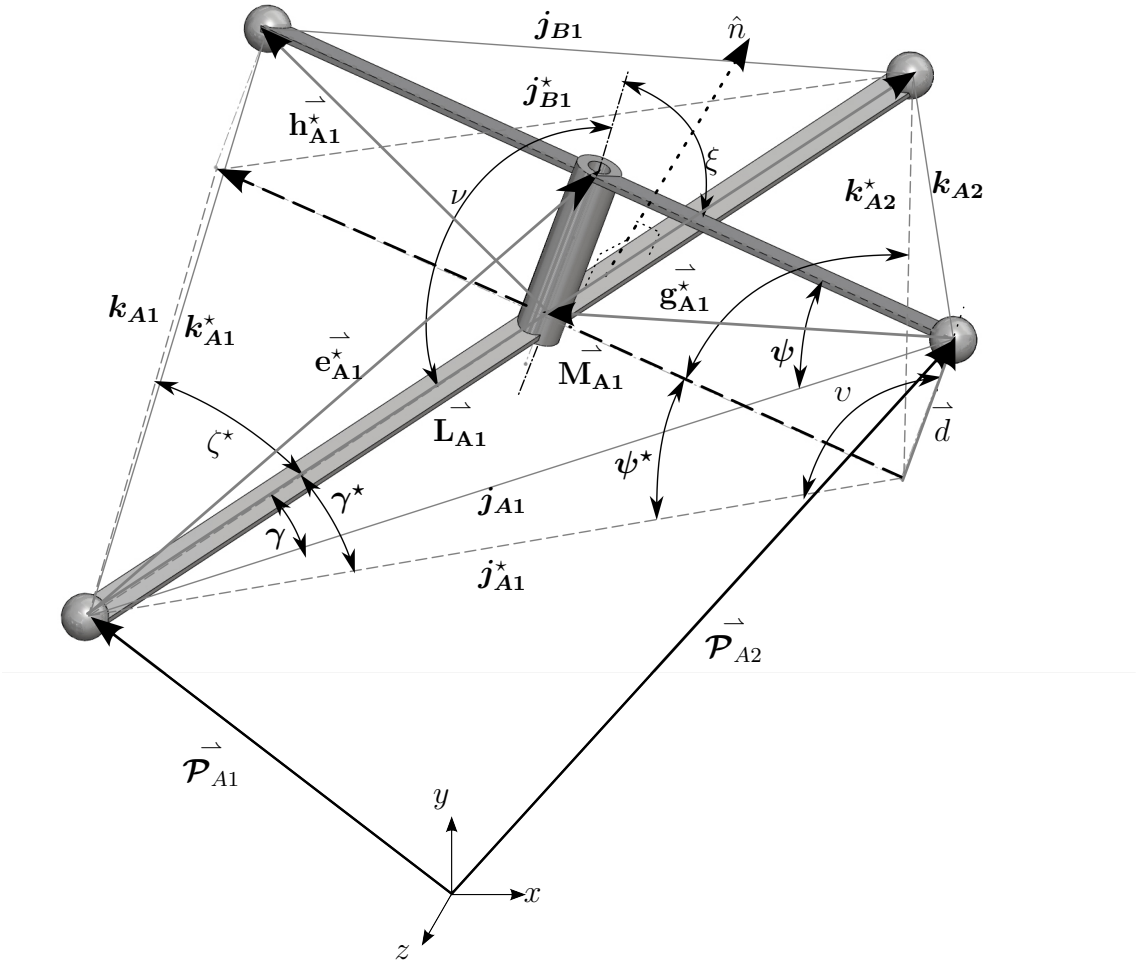


Figure 2.4: Skewed straight-link pantograph element with an angled pivot.

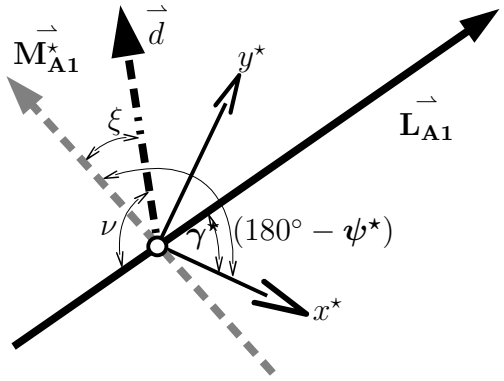


Figure 2.5: Coordinate system for calculating the bounding tetrahedron of a skew, angled-pivot straight-link pantoelement.

substitution, Cramer's rule, or other solution techniques, the dummy variables t or s may be found to obtain d_x. From there, the actual $\boldsymbol{j_{A1}}$ may be found by simply

using the cosine law, substituting $\cos \upsilon = (^{-d_{x^\star}}/_{d})$.

$$\boldsymbol{j}_{\boldsymbol{A1}}^2 = d^2 + \left(\boldsymbol{j}_{\boldsymbol{A1}}^\star\right)^2 - 2d\boldsymbol{j}_{\boldsymbol{A1}}^\star \left(\frac{-d_x^\star}{d}\right) \tag{2.27}$$

$$= d^2 + \left(\boldsymbol{j}_{\boldsymbol{A1}}^\star\right)^2 + 2\boldsymbol{j}_{\boldsymbol{A1}}^\star d_{x^\star} \tag{2.28}$$

Having found $\boldsymbol{j_{A1}}$, the distance $\boldsymbol{k_{A1}}$ may be found using substitutive trigonometry. Using the definition of ψ as the angle between $\vec{\mathbf{j}}_{\boldsymbol{A1}}$ and $\vec{\mathbf{M}}_{\mathbf{A1}}$,

$$\cos \boldsymbol{\psi} = \frac{\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - \left(e_{A1}^\star\right)^2}{2\boldsymbol{j}_{\boldsymbol{A1}} g_{A1}} \tag{2.29}$$

where

$$\left(e_{A1}^\star\right)^2 = d^2 + e_{A1}^2 - 2de_{A1}\cos\nu \tag{2.30}$$

Therefore, solving for $\boldsymbol{k_{A1}}$.

$$\boldsymbol{k}_{\boldsymbol{A1}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + M_{A1}^2 - 2\boldsymbol{j}_{\boldsymbol{A1}} M_{A1} \left(\frac{\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - \left(e_{A1}^\star\right)^2}{2\boldsymbol{j}_{\boldsymbol{A1}} g_{A1}}\right) \tag{2.31}$$

$$= \boldsymbol{j}_{\boldsymbol{A1}}^2 + M_{A1}^2 - \left(\frac{M_{A1}}{g_{A1}}\right)\left(\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - d^2 + e_{A1}^2 - 2de_{A1}\cos\nu\right) \tag{2.32}$$

$\boldsymbol{k_{A2}}$ is found in a similar fashion:

$$\cos \boldsymbol{\gamma}^\star = \frac{\boldsymbol{j}_{\boldsymbol{A1}}^2 + e_{A1}^2 - \left(g_{A1}^\star\right)^2}{2\boldsymbol{j}_{\boldsymbol{A1}} e_{A1}} \tag{2.33}$$

where

$$\left(g_{A1}^\star\right)^2 = d^2 + g_{A1}^2 - 2dg_{A1}\cos\xi \tag{2.34}$$

Therefore, solving for $\boldsymbol{k_{A2}}$,

$$\boldsymbol{k}_{\boldsymbol{A2}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + L_{A1}^2 - 2\boldsymbol{j}_{\boldsymbol{A1}} L_{A1} \left(\frac{\boldsymbol{j}_{\boldsymbol{A1}}^2 + e_{A1}^2 - \left(g_{A1}^\star\right)^2}{2\boldsymbol{j}_{\boldsymbol{A1}} e_{A1}}\right) \tag{2.35}$$

$$= \boldsymbol{j}_{\boldsymbol{A1}}^2 + L_{A1}^2 - \left(\frac{L_{A1}}{e_{A1}}\right)\left(\boldsymbol{j}_{\boldsymbol{A1}}^2 + e_{A1}^2 - d^2 + g_{A1}^2 - 2dg_{A1}\cos\xi\right) \tag{2.36}$$

Finally, the top side $\boldsymbol{j_{B1}}$ may be expressed in terms of $\boldsymbol{k_{A1}}$ using $h_{A1}^\star$, which runs

from the right-hand pivot point on the L_{A1} link to $\mathcal{P}_{B2}$.

$$\cos\zeta^\star = \frac{\boldsymbol{k_{A1}^2} + e_{A1}^2 - \left(h_{A1}^\star\right)^2}{2\boldsymbol{k_{A1}}e_{A1}} \tag{2.37}$$

where

$$\left(h_{A1}^\star\right)^2 = d^2 + h_{A1}^2 - 2dh_{A1}\cos\xi \tag{2.38}$$

Therefore, solving for $\boldsymbol{j_{B1}}$,

$$\boldsymbol{j_{B1}^2} = \boldsymbol{k_{A1}^2} + L_{A1}^2 - 2\boldsymbol{k_{A1}}L_{A1}\left(\frac{\boldsymbol{k_{A1}^2} + e_{A1}^2 - \left(h_{A1}^\star\right)^2}{2\boldsymbol{k_{A1}}e_{A1}}\right) \tag{2.39}$$

$$= L_{A1}^2 - \left(\frac{L_{A1}}{e_{A1}}\right)\left(e_{A1}^2 - d_{A1}^2 + h_{A1}^2 - 2d_{A1}h_{A1}\cos\xi\right) - \left(\frac{f_{A1}}{e_{A1}}\right)\boldsymbol{k_{A1}^2} \tag{2.40}$$

Furthermore, $\boldsymbol{j_{B1}}$ may be expressed in terms of $\boldsymbol{k_{A2}}$:

$$\boldsymbol{j_{B1}^2} = M_{A1}^2 - \left(\frac{M_{A1}}{g_{A1}}\right)\left(g_{A1}^2 - d_{A1}^2 + f_{A1}^2 - 2d_{A1}f_{A1}\cos\nu\right) - \left(\frac{h_{A1}}{g_{A1}}\right)\boldsymbol{k_{A2}^2} \tag{2.41}$$

These results add a number of terms to the equations derived in the normal-pivot and planar sections, giving more flexibility in synthesis of pantomeshes. By adding the $\cos\nu$ and $\cos\xi$ terms, the relations between the bounding quadrilateral edges can differ much more than a simple d^2 terms as in the normal-pivot case. Yet the relation is still linear between the squared edge lengths of the bounding quadrilateral.

Straight-link pantograph elements are a simple, yet highly flexible pair of links. Straight links, when collapsed, take up less space then bent links. Also, all straight-link elements have a linear relation between the squared edge lengths of the bounding quadrilateral, a fact that will become important in determining the mobility of pantomeshes as derived in Ch. 3.

2.3 Bent-Link Pantograph Elements

In contrast to straight-link pantograph elements, bent-link pantograph elements have pivots that do *not* intersect the span between spherical joint endpoints. While pantomeshes of bent-link elements do not collapse to straight lines as straight-link

pantomeshes, the trade-off is greater flexibility in motion. For bent-link pantograph elements, two more variables are introduced: α and β are offset angles for the links as shown in Figure 2.6. Obviously if one of the links is "straight", then one can simple substitute $\alpha = 0$ or $\beta = 0$ as appropriate.

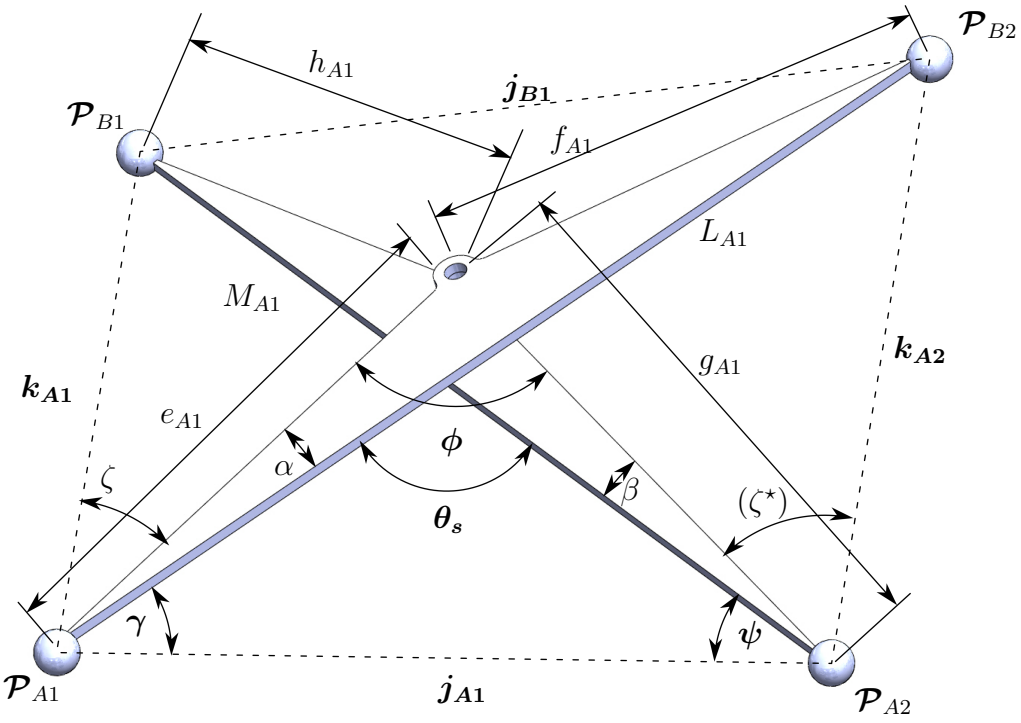


Figure 2.6: Bent-link pantograph element.

This section details both planar and skew bent-link pantograph elements, including normal-pivot and angular-pivot skew elements.

2.3.1 Planar Bent-Link Pantograph Elements

Planar bent-link pantograph elements are slightly more complex than straight-link elements in that the relative lengths of the bounding quadrilateral are no longer constant; there is now a transcendental term. Bent-link pantograph elements are also more commonly used for assembled pantomeshes, with particular examples in toys and movable structures.

A bent-link pantograph element is the most general case of planar pantograph elements, with many variables as shown in Figure 2.6. The length of the left side

may be found using the cosine law as shown in Eqn. (2.42).

$$\boldsymbol{k}_{\boldsymbol{A1}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + M_{A1}^2 - 2\boldsymbol{j}_{\boldsymbol{A1}} M_{A1} \cos \boldsymbol{\psi} \tag{2.42}$$

The unknown term $\boldsymbol{\psi}$ may be expressed in known quantities by using the cosine addition formula (see Eqn. (2.43)).

$$\cos \boldsymbol{\psi} = \cos (\boldsymbol{\psi} + \beta) \cos \beta + \sin (\boldsymbol{\psi} + \beta) \sin \beta \tag{2.43}$$

The unknown $\cos (\boldsymbol{\psi} + \beta)$ is given by the cosine law again (see Eqn. (2.44)), while $\sin (\boldsymbol{\psi} + \beta)$ is given using a trigonometric identity (see Eqn. (2.45)).

$$\cos (\boldsymbol{\psi} + \beta) = \frac{\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - e_{A1}^2}{2\boldsymbol{j}_{\boldsymbol{A1}} g_{A1}} \tag{2.44}$$

$$\sin (\boldsymbol{\psi} + \beta) = \left[1 - \frac{(\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - e_{A1}^2)^2}{4\boldsymbol{j}_{\boldsymbol{A1}}^2 g_{A1}^2}\right]^{1/2} \tag{2.45}$$

By substitution, $\boldsymbol{k}_{\boldsymbol{A1}}$ is given in terms of $\boldsymbol{j}_{\boldsymbol{A1}}$ in Eqn. (2.46).

$$\begin{aligned} \boldsymbol{k}_{\boldsymbol{A1}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + M_{A1}^2 &- (\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - e_{A1}^2) \left(\frac{M_{A1}}{g_{A1}}\right) \cos \beta \\ &- \left[4\boldsymbol{j}_{\boldsymbol{A1}}^2 g_{A1}^2 - (\boldsymbol{j}_{\boldsymbol{A1}}^2 + g_{A1}^2 - e_{A1}^2)^2\right]^{1/2} \left(\frac{M_{A1}}{g_{A1}}\right) \sin \beta \end{aligned} \tag{2.46}$$

Similarly, the right side may be found in terms of $\boldsymbol{j}_{\boldsymbol{A1}}$ as shown in Eqn. (2.51).

$$\boldsymbol{k}_{\boldsymbol{A2}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + L_{A1}^2 - 2\boldsymbol{j}_{\boldsymbol{A1}} L_{A1} \cos \boldsymbol{\gamma} \tag{2.47}$$

$$\cos \boldsymbol{\gamma} = \cos (\boldsymbol{\gamma} + \alpha) \cos \alpha + \sin (\boldsymbol{\gamma} + \alpha) \sin \alpha \tag{2.48}$$

$$\cos (\boldsymbol{\gamma} + \alpha) = \frac{\boldsymbol{j}_{\boldsymbol{A1}}^2 + e_{A1}^2 - g_{A1}^2}{2\boldsymbol{j}_{\boldsymbol{A1}} e_{A1}} \tag{2.49}$$

$$\sin (\boldsymbol{\gamma} + \alpha) = \left[1 - \frac{(\boldsymbol{j}_{\boldsymbol{A1}}^2 + e_{A1}^2 - g_{A1}^2)^2}{4\boldsymbol{j}_{\boldsymbol{A1}}^2 e_{A1}^2}\right]^{1/2} \tag{2.50}$$

$$\begin{aligned} \boldsymbol{k}_{\boldsymbol{A2}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + L_{A1}^2 &- (\boldsymbol{j}_{\boldsymbol{A1}}^2 + e_{A1}^2 - g_{A1}^2) \left(\frac{L_{A1}}{e_{A1}}\right) \cos \alpha \\ &- \left[4\boldsymbol{j}_{\boldsymbol{A1}}^2 e_{A1}^2 - (\boldsymbol{j}_{\boldsymbol{A1}}^2 + e_{A1}^2 - g_{A1}^2)^2\right]^{1/2} \left(\frac{L_{A1}}{e_{A1}}\right) \sin \alpha \end{aligned} \tag{2.51}$$

Finally, $\boldsymbol{j_{B1}}$ is shown in terms of $\boldsymbol{k_{A1}}$ in Eqns. (2.52) through (2.56).

$$\boldsymbol{j}_{\boldsymbol{B1}}^2 = \boldsymbol{k}_{\boldsymbol{A1}}^2 + L_{A1}^2 - 2\boldsymbol{k_{A1}} L_{A1} \cos\zeta \tag{2.52}$$

$$\cos\zeta = \cos(\zeta - \alpha)\cos\alpha - \sin(\zeta - \alpha)\sin\alpha \tag{2.53}$$

$$\cos(\zeta - \alpha) = \frac{\boldsymbol{k}_{\boldsymbol{A1}}^2 + e_{A1}^2 - h_{A1}^2}{2\boldsymbol{k_{A1}} e_{A1}} \tag{2.54}$$

$$\sin(\zeta - \alpha) = \left[1 - \frac{(\boldsymbol{k}_{\boldsymbol{A1}}^2 + e_{A1}^2 - h_{A1}^2)^2}{4\boldsymbol{k}_{\boldsymbol{A1}}^2 e_{A1}^2}\right]^{1/2} \tag{2.55}$$

$$\begin{aligned}\boldsymbol{j}_{\boldsymbol{B1}}^2 = \boldsymbol{k}_{\boldsymbol{A1}}^2 + L_{A1}^2 - \left(\boldsymbol{k}_{\boldsymbol{A1}}^2 + e_{A1}^2 - h_{A1}^2\right)\frac{L_{A1}}{e_{A1}}\cos\alpha \\ + \left[4\boldsymbol{k}_{\boldsymbol{A1}}^2 e_{A1}^2 - (\boldsymbol{k}_{\boldsymbol{A1}}^2 + e_{A1}^2 - h_{A1}^2)^2\right]^{1/2}\frac{L_{A1}}{e_{A1}}\sin\alpha\end{aligned} \tag{2.56}$$

For reference, $\boldsymbol{j_{B1}}$ is also shown in terms of $\boldsymbol{k_{A2}}$:

$$\begin{aligned}\boldsymbol{j}_{\boldsymbol{B1}}^2 = \boldsymbol{k}_{\boldsymbol{A2}}^2 + M_{A1}^2 - \left(\boldsymbol{k}_{\boldsymbol{A2}}^2 + g_{A1}^2 - f_{A1}^2\right)\frac{M_{A1}}{g_{A1}}\cos\beta \\ + \left[4\boldsymbol{k}_{\boldsymbol{A2}}^2 g_{A1}^2 - (\boldsymbol{k}_{\boldsymbol{A2}}^2 + g_{A1}^2 - f_{A1}^2)^2\right]^{1/2}\frac{M_{A1}}{g_{A1}}\end{aligned} \tag{2.57}$$

The top side, $\boldsymbol{j_{B1}}$, may also be placed in terms of $\boldsymbol{j_{A1}}$ by using different formulas for the area of a quadrilateral (see Eqns. (2.58) and (2.59)), where $\boldsymbol{\phi}$ is the angle between the diagonals L_{A1} and M_{A1} [23]. Combining these two equations and solving for $\boldsymbol{j_{B1}}$, Eqn. (2.60) is obtained. Using the cosine addition law as shown in Eqn. (2.64), $\boldsymbol{\phi}$ may be expressed in known quantities that include only $\boldsymbol{j_{A1}}$ as an unknown quantity. The final result in Eqn. (2.67) may be found purely in terms of $\boldsymbol{j_{A1}}$ if Eqn. (2.46) and Eqn. (2.51) are used to substitute for $\boldsymbol{k}_{\boldsymbol{A1}}^2$ and $\boldsymbol{k}_{\boldsymbol{A2}}^2$, respec-

tively.

$$\{\mathbf{Area}\} = \frac{1}{2} L_{A1} M_{A1} \sin \boldsymbol{\phi} \tag{2.58}$$

$$\{\mathbf{Area}\} = \frac{1}{4} (\boldsymbol{k}_{\boldsymbol{A1}}^2 + \boldsymbol{k}_{\boldsymbol{A2}}^2 - \boldsymbol{j}_{\boldsymbol{A1}}^2 - \boldsymbol{j}_{\boldsymbol{B1}}^2) \tan \boldsymbol{\phi} \tag{2.59}$$

$$\boldsymbol{j}_{\boldsymbol{B1}}^2 = L_{A1}^2 + M_{A1}^2 + \boldsymbol{j}_{\boldsymbol{A1}}^2 + \boldsymbol{k}_{\boldsymbol{A1}}^2 + \boldsymbol{k}_{\boldsymbol{A2}}^2 - 2 L_{A1} M_{A1} \cos \boldsymbol{\phi} \tag{2.60}$$

$$\boldsymbol{\theta_s} = 180^\circ - (\boldsymbol{\gamma} + \alpha) - (\boldsymbol{\psi} + \beta) \tag{2.61}$$

$$\boldsymbol{\phi} = 180^\circ - (\boldsymbol{\gamma} + \boldsymbol{\psi}) \tag{2.62}$$

$$\boldsymbol{\phi} = \boldsymbol{\theta_s} + (\alpha + \beta) \tag{2.63}$$

$$\cos \boldsymbol{\phi} = \cos \boldsymbol{\theta_s} \cos (\alpha + \beta) - \sin \boldsymbol{\theta_s} \sin (\alpha + \beta) \tag{2.64}$$

$$\cos \boldsymbol{\theta_s} = \frac{e_{A1}^2 + g_{A1}^2 - \boldsymbol{j}_{\boldsymbol{A1}}^2}{2 e_{A1} g_{A1}} \tag{2.65}$$

$$\sin \boldsymbol{\theta_s} = \left[1 - \frac{(e_{A1}^2 + g_{A1}^2 - \boldsymbol{j}_{\boldsymbol{A1}}^2)^2}{4 e_{A1}^2 g_{A1}^2} \right]^{1/2} \tag{2.66}$$

$$\begin{aligned} \boldsymbol{j}_{\boldsymbol{B1}}^2 = {} & L_{A1}^2 + M_{A1}^2 + \boldsymbol{j}_{\boldsymbol{A1}}^2 + \boldsymbol{k}_{\boldsymbol{A1}}^2 + \boldsymbol{k}_{\boldsymbol{A2}}^2 \\ & - L_{A1} M_{A1} \cos (\alpha + \beta) \frac{e_{A1}^2 + g_{A1}^2 - \boldsymbol{j}_{\boldsymbol{A1}}^2}{e_{A1} g_{A1}} \\ & - L_{A1} M_{A1} \sin (\alpha + \beta) \left[4 - \frac{(e_{A1}^2 + g_{A1}^2 - \boldsymbol{j}_{\boldsymbol{A1}}^2)^2}{4 e_{A1}^2 g_{A1}^2} \right]^{1/2} \end{aligned} \tag{2.67}$$

In contrast with planar straight-link elements, equations of motion for bent-link elements not only contain more variables, the relation between squares of edge-lengths of the bounding quadrilateral (i.e. j^2's and k^2's) are no longer linear. This fact becomes quite important in determining the mobility of closed pantomeshes as discussed in Ch. 3. Also, planar bent-link elements have the same restricted range-of-motion as planar straight-link elements since the endpoints can collide during the element's range of motion. This can be avoided by offsetting the links, forming skew bent-link elements.

2.3.2 Skew, Normal-Pivot Bent-Link Pantograph Elements

Skew, bent-link pantograph elements function much like the planar version, except that there is a distance between the links. Much like the straight-link case in §2.2.2, the calculation of the skew bounding quadrilateral dimensions as shown in Figure 2.7 begins by first calculating the edge lengths of the flattened bounding quadrilateral. Since the pantograph normal is perpendicular to both of the links,

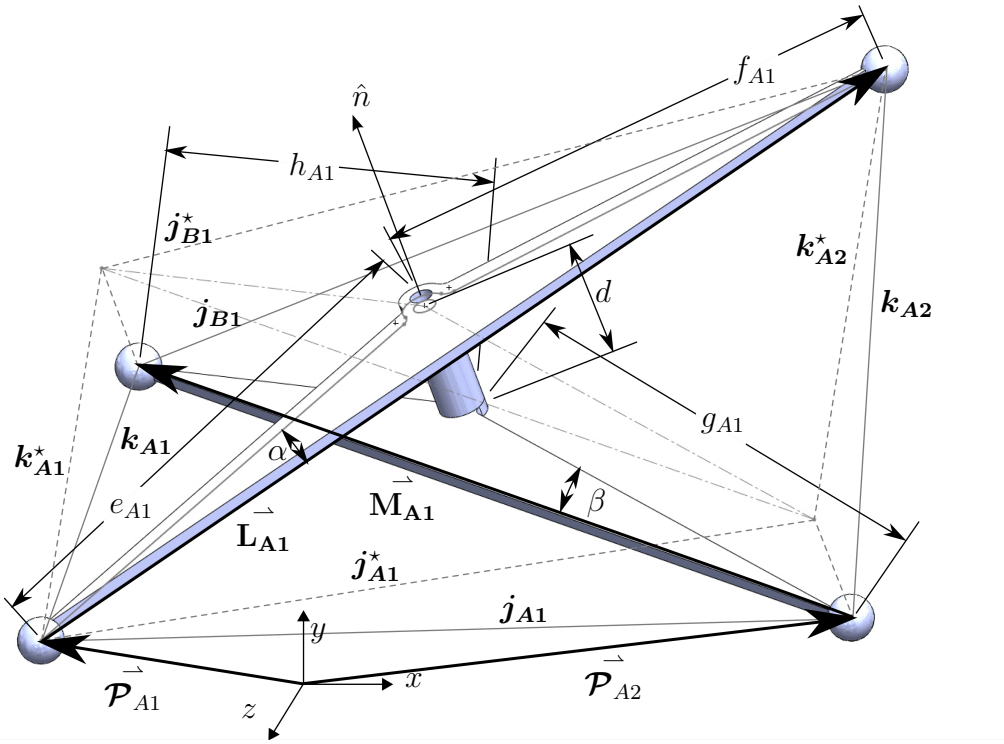


Figure 2.7: Skewed bent-link pantograph element with an normal pivot.

each true edge is the hypotenuse of the triangle containing the flattened quadrilateral edge length and the pantograph element distance. Therefore, referring to the flattened dimensions with a star, as in $\boldsymbol{k}^{\star}_{\boldsymbol{A1}}$, the true edges may be calculated as follows:

$$\boldsymbol{j}^{2}_{\boldsymbol{A1}} = d^{2}_{A1} + (\boldsymbol{j}^{\star}_{\boldsymbol{A1}/\boldsymbol{A1}})^{2} \tag{2.68}$$

$$\boldsymbol{k}^{2}_{\boldsymbol{A1}} = d^{2}_{A1} + (\boldsymbol{k}^{\star}_{\boldsymbol{A1}/\boldsymbol{A1}})^{2} \tag{2.69}$$

$$\boldsymbol{k}^{2}_{\boldsymbol{A2}} = d^{2}_{A1} + (\boldsymbol{k}^{\star}_{\boldsymbol{A2}/\boldsymbol{A1}})^{2} \tag{2.70}$$

$$\boldsymbol{j}^{2}_{\boldsymbol{B1}} = d^{2}_{A1} + (\boldsymbol{j}^{\star}_{\boldsymbol{B1}/\boldsymbol{A1}})^{2} \tag{2.71}$$

where

$$(\boldsymbol{k}^{\star}_{\boldsymbol{A1/A1}})^2 = (\boldsymbol{j}^{\star}_{\boldsymbol{A1/A1}})^2 + M_{A1}^2 - \left((\boldsymbol{j}^{\star}_{\boldsymbol{A1/A1}})^2 + g_{A1}^2 - e_{A1}^2\right)\frac{M_{A1}}{g_{A1}}\cos\beta$$
$$- \left[4(\boldsymbol{j}^{\star}_{\boldsymbol{A1/A1}})^2 g_{A1}^2 - \left((\boldsymbol{j}^{\star}_{\boldsymbol{A1/A1}})^2 + g_{A1}^2 - e_{A1}^2\right)^2\right]^{1/2}\frac{M_{A1}}{g_{A1}}\sin\beta \tag{2.72}$$

$$\boldsymbol{k}_{\boldsymbol{A1}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + M_{A1}^2 - \left(\boldsymbol{j}_{\boldsymbol{A1}}^2 - d_{A1}^2 + g_{A1}^2 - e_{A1}^2\right)\frac{M_{A1}}{g_{A1}}\cos\beta$$
$$- \left[4\left(\boldsymbol{j}_{\boldsymbol{A1}}^2 - d_{A1}^2\right)g_{A1}^2 - \left(\boldsymbol{j}_{\boldsymbol{A1}}^2 - d_{A1}^2 + g_{A1}^2 - e_{A1}^2\right)^2\right]^{1/2}\frac{M_{A1}}{g_{A1}}\sin\beta \tag{2.73}$$

with the other relations similarly formed as shown below:

$$\boldsymbol{k}_{\boldsymbol{A2}}^2 = \boldsymbol{j}_{\boldsymbol{A1}}^2 + L_{A1}^2 - \left(\boldsymbol{j}_{\boldsymbol{A1}}^2 - d_{A1}^2 + e_{A1}^2 - g_{A1}^2\right)\frac{L_{A1}}{e_{A1}}\cos\alpha$$
$$- \left[4\left(\boldsymbol{j}_{\boldsymbol{A1}}^2 - d_{A1}^2\right)e_{A1}^2 - \left(\boldsymbol{j}_{\boldsymbol{A1}}^2 - d_{A1}^2 + e_{A1}^2 - g_{A1}^2\right)^2\right]^{1/2}\frac{L_{A1}}{e_{A1}}\sin\alpha \tag{2.74}$$

$$\boldsymbol{j}_{\boldsymbol{B1}}^2 = \boldsymbol{k}_{\boldsymbol{A2}}^2 + M_{A1}^2 - \left[\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A1}^2\right) + g_{A1}^2 - (f_{A1})^2\right]\left(\frac{M_{A1}}{g_{A1}}\right)\cos\beta_{A1}$$
$$+ \left[4\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A1}^2\right)g_{A1}^2 - \left(\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A1}^2\right) + g_{A1}^2 - (f_{A1})^2\right)^2\right]^{1/2}\left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1} \tag{2.75}$$

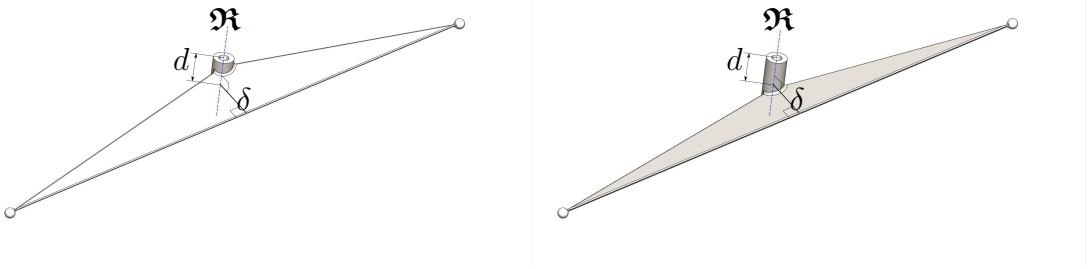
$$\boldsymbol{j}_{\boldsymbol{B2}}^2 = \boldsymbol{k}_{\boldsymbol{A2}}^2 + L_{A2}^2 - \left[\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A2}^2\right) + e_{A2}^2 - (h_{A2})^2\right]\left(\frac{L_{A2}}{e_{A2}}\right)\cos\alpha_{A2}$$
$$+ \left[4\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A2}^2\right)e_{A2}^2 - \left(\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A2}^2\right) + e_{A2}^2 - (h_{A2})^2\right)^2\right]^{1/2}\left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2} \tag{2.76}$$

The results are very similar to the planar case, with a small d^2 term subtracted from the relevant square of the bounding quadrilateral edge length in each equation. For a small increase in complexity, the links can be designed to have complete range of motion. Some special requirements, however, are required for mobility of an assembled pantomesh of skew, angled-pivot bent-link elements, as discussed in Ch. 3.

2.3.3 Skew, Angled-pivot Bent-Link Pantograph Elements

The most complex form of pantograph elements is the skew, angled-pivot bent-link element. By adding two more angular variables to denote the tilting of each element relative to the pivot axis, the very definition of link dimensions must be

examined since there are no measurements that are forced to be perpendicular than with all other cases. Therefore, the approach to deriving motion equations for these pantograph elements is different from the previous approaches, starting with a universal description of link dimensions. Two links are shown in Figure 2.8, with identical link spans but with clear differences in other dimensions and angles. However, they in fact have identical kinematic characteristics based on the shortest distance between the pivot axis and the link span. The use of shortest-distance formulation of the link geometry is to reduce any ambiguity between different links that have identical characteristics, as there is only one shortest distance between two non-parallel lines as shown in Figure 2.8. In this figure, the shapes of the link bodies are different. However, the link span, the angle of the pivot, and the distance d are the same, even though both links have different appearances.



(a) One configuration (b) Another configuration

Figure 2.8: Differently-shaped angled-pivot bent links with identical kinematic properties.

After defining the shortest-distance dimensions of both links, the next step in kinematic analysis is to project both links onto a common plane. From Figure 2.9, the variable ν is the angle[1] between the pivot axis $\mathfrak{R}$ and $\overrightarrow{\mathbf{L_{A1}}}$. The distance δ is the shortest distance between the pivot axis and $\overrightarrow{\mathbf{L_{A1}}}$. As the shortest distance between two skew lines, it is perpendicular to both vectors, leaving only one degree of freedom between them. The distance ξ is also therefore measured between the pivot axis $\mathfrak{R}$ and $\overrightarrow{\mathrm{M}_{\mathbf{A1}}}$.

The solution approach is to project each link shape onto a plane that has the pivot axis as its normal as shown in Figure 2.9. This projection plane also contains the

[1]The angle between two skew lines is equal to the angle between any two intersecting lines that are parallel to the skew lines.

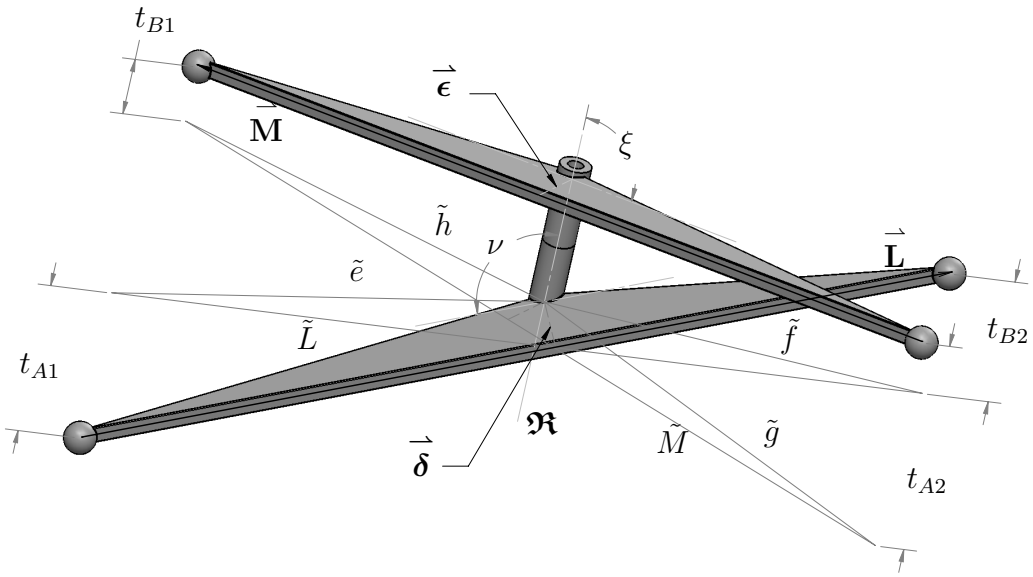


Figure 2.9: Skewed bent-link pantograph element with an angled pivot.

distance vector $\vec{\boldsymbol{\delta}}$. Similar to the normal-pivot case, the left-hand link ($\vec{M}$) is first moved along the pivot axis so that $\vec{\boldsymbol{\epsilon}}$ is in the projection plane. Then the link shape is projected onto the plane. Note that the distance that the left-hand link moves is not the same as the offset distance d, and in fact changes during movement.

To obtain the edge lengths of the element's bounding quadrilateral, the bounding lengths of the projected quadrilateral must be found. The projected lengths are

related as shown in the following equations.

$$\tilde{\boldsymbol{k}}^2_{\boldsymbol{A1/A1}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}} + \tilde{M}^2_{A1} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}} + \tilde{g}^2_{A1} - \tilde{e}^2_{A1}\right)\left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\cos\tilde{\beta}_{A1}$$
$$- \left[4\tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}}\tilde{g}^2_{A1} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}} + \tilde{g}^2_{A1} - \tilde{e}^2_{A1}\right)^2\right]^{1/2}\left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\sin\tilde{\beta} \tag{2.77}$$

$$\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}} + \tilde{L}^2_{A1} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}} + \tilde{e}^2_{A1} - \tilde{g}^2_{A1}\right)\left(\frac{\tilde{L}_{A1}}{\tilde{e}_{A1}}\right)\cos\tilde{\alpha}$$
$$- \left[4\tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}}\tilde{e}^2_{A1} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}} + \tilde{e}^2_{A1} - \tilde{g}^2_{A1}\right)^2\right]^{1/2}\left(\frac{\tilde{L}_{A1}}{\tilde{e}_{A1}}\right)\sin\tilde{\alpha} \tag{2.78}$$

$$\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/A1}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{A1/A1}} + \tilde{L}^2_{A1} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A1/A1}} + \tilde{e}^2_{A1} - \tilde{h}^2_{A1}\right)\left(\frac{\tilde{L}_{A1}}{\tilde{e}_{A1}}\right)\cos\tilde{\alpha}$$
$$+ \left[4\tilde{\boldsymbol{k}}^2_{\boldsymbol{A1/A1}}\tilde{e}^2_{A1} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A1/A1}} + \tilde{e}^2_{A1} - \tilde{h}^2_{A1}\right)^2\right]^{1/2}\left(\frac{\tilde{L}_{A1}}{\tilde{e}_{A1}}\right)\sin\tilde{\alpha} \tag{2.79}$$

$$\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/A1}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + \tilde{M}^2_{A1} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + \tilde{g}^2_{A1} - \tilde{f}^2_{A1}\right)\left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\cos\tilde{\beta}$$
$$+ \left[4\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}}\tilde{g}^2_{A1} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + \tilde{g}^2_{A1} - \tilde{f}^2_{A1}\right)^2\right]^{1/2}\left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\sin\tilde{\beta} \tag{2.80}$$

The constituent components of these projected quantities may be expressed in terms of the actual link geometry:

$$\tilde{L}_{A1} = L_{A1}\sin\nu_{A1} \tag{2.81}$$
$$\tilde{e}^2_{A1} = e^2_{A1}\left(\cos^2\alpha_{A1}\sin^2\nu_{A1} + \sin^2\alpha_{A1}\right) \tag{2.82}$$
$$\sin\tilde{\alpha}_{A1} = \frac{e_{A1}}{\tilde{e}_{A1}}\sin\alpha_{A1} \tag{2.83}$$
$$\tilde{M}_{A1} = M_{A1}\sin\xi_{A1} \tag{2.84}$$
$$\tilde{g}^2_{A1} = g^2_{A1}\left(\cos^2\beta_{A1}\sin^2\xi_{A1} + \sin^2\beta_{A1}\right) \tag{2.85}$$
$$\sin\tilde{\beta}_{A1} = \frac{g_{A1}}{\tilde{g}_{A1}}\sin\beta_{A1} \tag{2.86}$$
$$\tilde{h}^2_{A1} = \tilde{M}^2_{A1} + \tilde{g}^2_{A1} - 2\tilde{M}_{A1}\tilde{g}_{A1}\cos\tilde{\beta}_{A1} \tag{2.87}$$

Finally, the actual lengths of the bounding tetrahedron are simply:

$$\boldsymbol{j}^2_{\boldsymbol{A1}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{A1/A1}} + t^2_{A2/A1} - t^2_{A1/A1} \tag{2.88}$$

$$\boldsymbol{k}^2_{\boldsymbol{A1}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{A1/A1}} + t^2_{B1/A1} - t^2_{A1/A1} \tag{2.89}$$

$$\boldsymbol{j}^2_{\boldsymbol{B1}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/A1}} + t^2_{B2/A1} - t^2_{B1/A1} \tag{2.90}$$

$$\boldsymbol{k}^2_{\boldsymbol{A2}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + t^2_{B2/A1} - t^2_{A2/A1} \tag{2.91}$$

where the translational distances between the projected points and the actual points are:

$$t_{A1/A1} = e_{A1} \cos\alpha_{A1} \cos\nu_{A1} \tag{2.92}$$

$$t_{A2/A1} = g_{A1} \cos\beta_{A1} \cos\xi_{A1} + d_{A1} \tag{2.93}$$

$$t_{B1/A1} = (M_{A1} - g_{A1} \cos\beta_{A1}) \cos\xi_{A1} + d_{A1} \tag{2.94}$$

$$t_{B2/A1} = -(L_{A1} - e_{A1} \cos\alpha_{A1}) \cos\nu_{A1} \tag{2.95}$$

Equations (2.92) through (2.95) can be substituted into equations (2.88) through (2.91), and then into the original equations of motion. Those equations are complex, and will not be expanded here. Edge-length relative velocities necessary for mobility calculations are derived in Chapter 3.

One approach to the visualization and assembly of skew, angled-pivot bent-link pantograph meshes is to use a *shared-edge dual tetrahedron model*, where each pantograph element is represented as a tetrahedron that share a common edge with the other. That common edge acts as the pivot axis (see Figure 2.10).

This approach lends itself to constructing pantograph elements using string-connected joints as described in Chapter 7, with a rigid tubular link for every line segment.

In summary, the most complex pantograph element requires new definitions and approaches to determining mobility. Many parameters are required to define a skew, angled-pivot bent-link pantograph element, thereby allowing flexibility for synthesizing the proper pantomesh for a desired motion. However, this complexity further complicates the mobility requirements and therefore makes synthesis more difficult. The trade-off between the two is up to the designer.

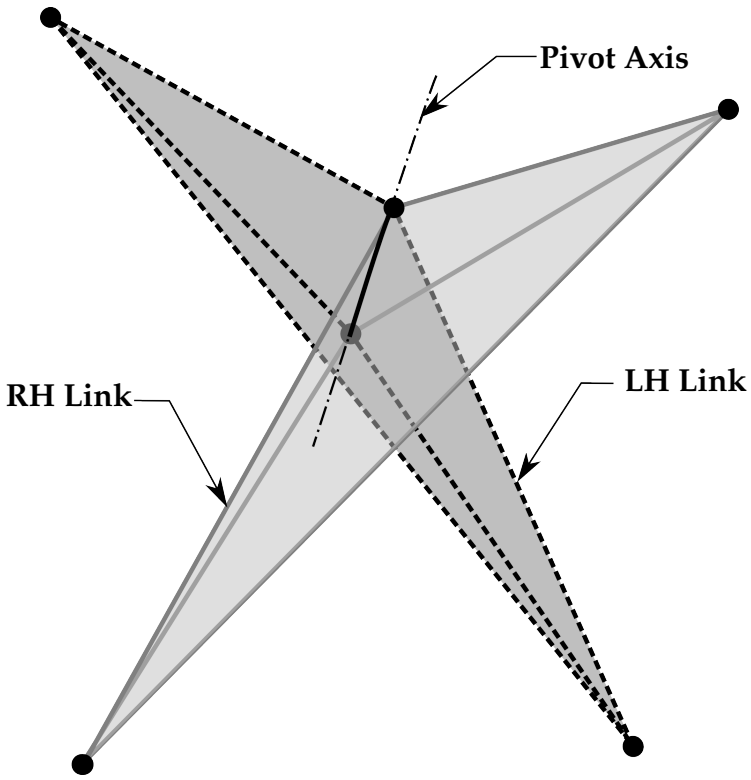


Figure 2.10: Shared-edge dual tetrahedron model of a skew pantograph element.

2.4 Discussion

A summary of the independent variables required to define each type of pantograph element is shown in Table 2.1.

Element Type				Independent Variables
Straight-link	Planar		4	L, M, e, g
	Skew	Normal-Pivot	5	L, M, e, g, d
		Angled-Pivot	7	L, M, e, g, d, ν, ξ
Bent-link	Planar		6	L, M, e, g, α, β
	Skew	Normal-Pivot	7	L, M, e, g, α, β, d
		Angled-Pivot	9	L, M, e, g, α, β, d, ν, ξ

Table 2.1: Comparison of pantograph elements

The number and type of independent variables is important in the synthesis of pantomeshes, as described in Chapter 6.

2.5 Conclusions

This chapter described the basic kinematics of pantograph elements, the basic building blocks of pantomeshes. Six different types of pantograph elements were detailed; those types fall into the following categories: straight-link and bent-link, planar and skew, and, within skew, normal-pivot and angled pivot. The equations developed in this chapter relate the motion of the bounding sides to each other, giving an idea of how pantograph elements connect to each other in a larger pantomesh. These relations are crucial in understanding mobility and for synthesizing pantomeshes.

Chapter 3

Mobility of Pantopatches

This chapter describes the kinematc mobility of *pantopatches*, defined here as an assembly of four pantograph elements, two wide and two high. Pantopatches may be considered the basic *unit of mobility* for pantomeshes. While the basic unit of an assembled pantomesh is the pantograph element, the basic unit of mobility is the pantograph patch. A discussion of pantopatch mobility is presented, both in terms of basic kinematics and partial derivative requirements. Finally, a new description of a graphically-based mobility determination is presented. But first, a brief description of linkage mobility is discussed.

3.1 Degrees of Freedom

Mobility, or the number of degrees of freedom of a mechanism, can be calculated using the Gruebler-Kutzbach equation as shown in Eqn. (3.1) below [24, 25].

$$\mathcal{M} = 6(N-1) - 5f_1 - 4f_2 - 3f_3 - 2f_4 - f_5 \tag{3.1}$$

where $\mathcal{M}$ is the *mobility* or number of degrees of freedom, N is the number of links, f_1 is the number of f_1 joints, or joints that allow one degree of freedom such as revolute or slider joints. Furthermore, f_2 joints allow two DOF, f_3 joints allow three DOF, and so on. Spherical joints are f_3 joints because they allow rotations in three orthogonal axes but restrict any translational motion.

A pantopatch is an assembly of four pantograph elements into an open 2x2 arrangement. The number of links is 8 and the number of revolute (f_1) joints is 4. At

first glance it would appear that there are 9 spherical joints, but there are only 7 because of the following two reasons . First, the four "corners" of the pantopatch are not connected to any other links outside the pantopatch, and are therefore not joints in this analysis. Also, the center spherical joint is in fact three spherical joints that are collocated. Therefore, according to the Gruebler-Kutzbach equation:

$$\mathcal{M} = 6(8-1) - 5(8) - 3(4+3) = 1 \tag{3.2}$$

However, a mobile pantopatch should have two degrees of freedom; the single DOF found in (3.2) is simply the ability to flex the pantograph elements relative to each other. It does not include the ability of the pantograph elements to perform scissor-like actuation. Therefore, additional conditions must be met to create a properly mobile pantopatch for use in a larger assembled pantomesh.

By examining a basic $n \times \underline{m}$ pantomesh, the number of links is $2nm$, the number of revolute (f_1) joints is nm, and the number of spherical joints is $2m + 3(n-1)m$. Therefore, according to the Gruebler-Kutzbach equation:

$$\begin{aligned} \mathcal{M} &= 6(2nm-1) - 5(nm) - 4(2m + 3(n-1)m) \\ &= -5nm + 4m - 6 \end{aligned} \tag{3.3}$$

The mobility of a mesh requires the mobility of its constituent pantopatches. Each interior point is surrounded by four pantograph elements and therefore can define a *pantopatch* (see Figure 3.1). Examining the interior point $\boldsymbol{\mathcal{P}}_{B2}$, the distance $\boldsymbol{k_{B2}}$ may be placed in terms of $\boldsymbol{k_{A2}}$ by two routes as shown with curved arrows in the figure.

3.2 Mobility of Straight-Link Pantopatches

This section describes the mobility determination of straight-link pantopatches. The process is fairly straightforward given the quasi-linear nature of the element kinematics as shown in Ch. 2, but is detailed to show the general procedure for determining mobility of the more-complicated bent-link pantopatches.

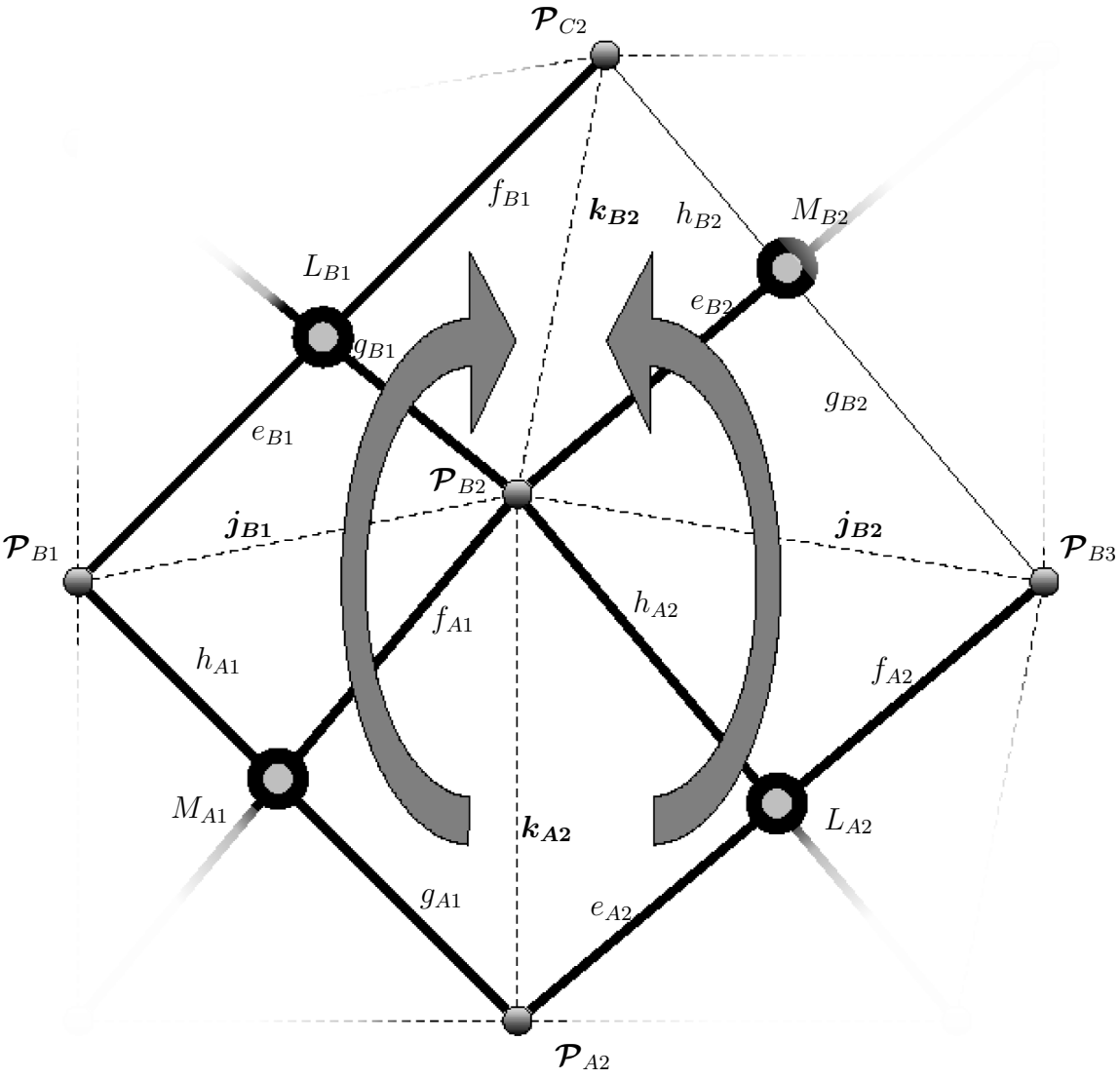


Figure 3.1: Mobility requirements for a straight-link pantograph patch.

3.2.1 Planar Straight-Link Pantopatches

For planar straight-link pantograph patches, Eqn. (3.4) and Eqn. (3.5) detail the same concept, expressing $\boldsymbol{k_{A2}}$ in terms of $\boldsymbol{j_{A1}}$ and $\boldsymbol{j_{B2}}$, respectively. Furthermore,

$\boldsymbol{j_{B1}}$ can be placed in terms of $\boldsymbol{k_{A2}}$ as in Eqn. (3.6), as can $\boldsymbol{j_{B2}}$ in Eqn. (3.7).

$$\boldsymbol{k_{B2}^2} = L_{B1}f_{B1} + \left(\frac{L_{B1}}{e_{B1}}\right) g_{B1}^2 - \left(\frac{f_{B1}}{e_{B1}}\right) \boldsymbol{j_{B1}^2} \tag{3.4}$$

$$\boldsymbol{k_{B2}^2} = M_{B2}h_{B2} + \left(\frac{M_{B2}}{g_{B2}}\right) e_{B2}^2 - \left(\frac{h_{B2}}{g_{B2}}\right) \boldsymbol{j_{B2}^2} \tag{3.5}$$

$$\boldsymbol{j_{B1}^2} = M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right) f_{A1}^2 - \left(\frac{h_{A1}}{g_{A1}}\right) \boldsymbol{k_{A2}^2} \tag{3.6}$$

$$\boldsymbol{j_{B2}^2} = L_{A2}f_{A2} + \left(\frac{L_{A2}}{e_{A2}}\right) h_{A2}^2 - \left(\frac{f_{A2}}{e_{A2}}\right) \boldsymbol{k_{A2}^2} \tag{3.7}$$

Combining these equations, two equations for $\boldsymbol{k_{B2}}$ in terms of $\boldsymbol{k_{A2}}$ are found as shown in Eqn. (3.8) and Eqn. (3.9):

$$\boldsymbol{k_{B2}^2} = L_{B1}f_{B1} + \left(\frac{L_{B1}}{e_{B1}}\right) g_{B1}^2 - \left(\frac{f_{B1}}{e_{B1}}\right) \left[M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right) f_{A1}^2 - \left(\frac{h_{A1}}{g_{A1}}\right) \boldsymbol{k_{A2}^2}\right] \tag{3.8}$$

$$\boldsymbol{k_{B2}^2} = M_{B2}h_{B2} + \left(\frac{M_{B2}}{g_{B2}}\right) e_{B2}^2 - \left(\frac{h_{B2}}{g_{B2}}\right) \left[L_{A2}f_{A2} + \left(\frac{L_{A2}}{e_{A2}}\right) h_{A2}^2 - \left(\frac{f_{A2}}{e_{A2}}\right) \boldsymbol{k_{A2}^2}\right] \tag{3.9}$$

For a pantograph patch to be mobile, $\boldsymbol{k_{B2}}$ must be the same for *both* equations, regardless of how the other variables change. Therefore, the rate of change for $\boldsymbol{k_{B2}}$ must also be the same for either equation. Therefore, taking partial derivatives and thereby eliminating constant link length terms, Eqn. (3.10) and Eqn. (3.11) are obtained. Since all pantograph element bounding edges are squared, the partial derivatives of those squared terms are calculated for simplicity.

$$\frac{d(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})} = \left(\frac{f_{B1}}{e_{B1}}\right)\left(\frac{h_{A1}}{g_{A1}}\right) \tag{3.10}$$

$$\frac{d(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})} = \left(\frac{h_{B2}}{g_{B2}}\right)\left(\frac{f_{A2}}{e_{A2}}\right) \tag{3.11}$$

Since higher-order derivatives are equal to zero, there are no further constraints on mobility. Therefore, the following must be true:

$$\frac{h_{A1}f_{B1}}{g_{A1}e_{B1}} = \frac{f_{A2}h_{B2}}{e_{A2}g_{B2}} \tag{3.12}$$

Subsequently, if given the link dimensions for the $A1$, $A2$, and $B1$ scissor links and having solved for $\left(\frac{h_{B2}}{g_{B2}}\right)$, the unknown terms e_{B2}, M_{B2}, and g_{B2} are related to known terms as shown in Eqn. (3.13). This equation is obtained by combining Eqns. (3.8) and (3.9) and substituting Eqn. (3.12) to eliminate the $\boldsymbol{k^2_{A2}}$ term.

$$M_{B2}h_{B2} + \left(\frac{M_{B2}}{g_{B2}}\right) e^2_{B2} = \left(\frac{h_{A1}e_{A2}f_{B1}}{g_{A1}f_{A2}e_{B1}}\right)\left[L_{A2}f_{A2} + \left(\frac{L_{A2}}{e_{A2}}\right) h^2_{A2}\right] + L_{B1}f_{B1} + \left(\frac{L_{B1}}{e_{B1}}\right) g^2_{B1} - \left(\frac{f_{B1}}{e_{B1}}\right)\left[M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right) f^2_{A1}\right] \quad (3.13)$$

In most cases, either M_{B2} or g_{B2} may be chosen while using the equations to solve for e_{B2}.

Equations (3.12) and (3.13) ensure the mobility of an open straight-link mesh. As with all pantopatch types, these equations must be satisfied for each interior point; that is, each point that is shared by exactly four pantograph elements.

3.2.2 Mobility of Skew, Normal-Pivot Straight-Link Pantopatches

For normal-pivot skew pantograph elements with straight links, mobility is similar to the planar case. As found in §3.2.1, the mobility of a connected mesh is found for each interior point; that is, the common point of four pantograph elements. Mobility is again determined by the relative rate of change of the bounding quadrilaterals of the constituent pantograph elements. The addition of the offset distance d affects the kinematics as shown below (from Ch. 2).

$$\boldsymbol{k^2_{B2}} = L_{B1}f_{B1} + \left(\frac{L_{B1}}{e_{B1}}\right)\left(g^2_{B1} + d^2_{B1}\right) - \left(\frac{f_{B1}}{e_{B1}}\right)\boldsymbol{j^2_{B1}} \quad (3.14)$$

$$\boldsymbol{k^2_{B2}} = M_{B2}h_{B2} + \left(\frac{M_{B2}}{g_{B2}}\right)\left(e^2_{B2} + d^2_{B2}\right) - \left(\frac{h_{B2}}{g_{B2}}\right)\boldsymbol{j^2_{B2}} \quad (3.15)$$

$$\boldsymbol{j^2_{B1}} = M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right)\left(f^2_{A1} + d^2_{A1}\right) - \left(\frac{h_{A1}}{g_{A1}}\right)\boldsymbol{k^2_{A2}} \quad (3.16)$$

$$\boldsymbol{j^2_{B2}} = L_{A2}f_{A2} + \left(\frac{L_{A2}}{e_{A2}}\right)\left(h^2_{A2} + d^2_{A2}\right) - \left(\frac{f_{A2}}{e_{A2}}\right)\boldsymbol{k^2_{A2}} \quad (3.17)$$

Again, the relative velocities of these measurement are needed to ensure mobility. The first derivative solution is identical to Eqn. (3.12), since the d term is constant. All that remains is to solve for the remaining unknowns, finding $\boldsymbol{k_{B2}}$ by both

"paths":

$$\boldsymbol{k}^2_{\boldsymbol{B2}} = L_{B1}f_{B1} + \left(\frac{L_{B1}}{e_{B1}}\right)\left(g^2_{B1} + d^2_{B1}\right) - \left(\frac{f_{B1}}{e_{B1}}\right)\left[M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right)\left(f^2_{A1} + d^2_{A1}\right) - \left(\frac{h_{A1}}{g_{A1}}\right)\boldsymbol{k}^2_{\boldsymbol{A2}}\right] \quad (3.18)$$

$$\boldsymbol{k}^2_{\boldsymbol{B2}} = M_{B2}h_{B2} + \left(\frac{M_{B2}}{g_{B2}}\right)\left(e^2_{B2} + d^2_{B2}\right) - \left(\frac{h_{B2}}{g_{B2}}\right)\left[L_{A2}f_{A2} + \left(\frac{L_{A2}}{e_{A2}}\right)\left(h^2_{A2} + d^2_{A2}\right) - \left(\frac{f_{A2}}{e_{A2}}\right)\boldsymbol{k}^2_{\boldsymbol{A2}}\right] \quad (3.19)$$

As with the planar straight-link case, Eqn. (3.12) is used to substitute for $\left(\frac{h_{B2}}{g_{B2}}\right)$ and to eliminate the $\boldsymbol{k}^2_{\boldsymbol{A2}}$ term. Furthermore, the terms from the $B2$ pantograph element are collected so they may be related.

$$\begin{aligned} M_{B2}h_{B2} + \left(\frac{M_{B2}}{g_{B2}}\right)\left(e^2_{B2} + d^2_{B2}\right) &= \left(\frac{h_{A1}e_{A2}f_{B1}}{g_{A1}f_{A2}e_{B1}}\right)\left[L_{A2}f_{A2} + \left(\frac{L_{A2}}{e_{A2}}\right)\left(h^2_{A2} + d^2_{A2}\right)\right] \\ &\quad + L_{B1}f_{B1} + \left(\frac{L_{B1}}{e_{B1}}\right)\left(g^2_{B1} + d^2_{B1}\right) \\ &\quad - \left(\frac{f_{B1}}{e_{B1}}\right)\left[M_{A1}h_{A1} + \left(\frac{M_{A1}}{g_{A1}}\right)\left(f^2_{A1} + d^2_{A1}\right)\right] \end{aligned} \quad (3.20)$$

While slightly more complicated than the planar, it is shown that simply skewing a pantograph element does not greatly affect the motion or mobility of a pantopatch, as long as the pivot axis is perpendicular to both link spans.

3.2.3 Mobility of Skew, Angled-Pivot Straight-Link Pantopatches

When the pivot axis is no longer perpendicular to link spans (yet still intersects them), additional terms are considered. Proceeding in a manner similar to the

normal-pivot case, the basic relations are repeated from Ch. 2:

$$\boldsymbol{k}_{\boldsymbol{B2}}^2 = L_{B1}^2 + \left(\frac{L_{B1}}{e_{B1}}\right)\left(d_{B1}^2 + g_{B1}^2 - 2d_{B1}g_{B1}\cos\xi_{B1} - e_{B1}^2\right) - \left(\frac{f_{B1}}{e_{B1}}\right)\boldsymbol{j}_{\boldsymbol{B1}}^2 \tag{3.21}$$

$$\boldsymbol{k}_{\boldsymbol{B2}}^2 = M_{B2}^2 + \left(\frac{M_{B2}}{g_{B2}}\right)\left(d_{B2}^2 + e_{B2}^2 - 2d_{B2}e_{B2}\cos\nu_{B2} - g_{B2}^2\right) - \left(\frac{h_{B2}}{g_{B2}}\right)\boldsymbol{j}_{\boldsymbol{B2}}^2 \tag{3.22}$$

$$\boldsymbol{j}_{\boldsymbol{B1}}^2 = M_{A1}^2 + \left(\frac{M_{A1}}{g_{A1}}\right)\left(d_{A1}^2 + f_{A1}^2 - 2d_{A1}f_{A1}\cos\nu_{A1} - g_{A1}^2\right) - \left(\frac{h_{A1}}{g_{A1}}\right)\boldsymbol{k}_{\boldsymbol{A2}}^2 \tag{3.23}$$

$$\boldsymbol{j}_{\boldsymbol{B2}}^2 = L_{A2}^2 + \left(\frac{L_{A2}}{e_{A2}}\right)\left(d_{A2}^2 + h_{A2}^2 - 2d_{A2}h_{A2}\cos\xi_{A2} - e_{A2}^2\right) - \left(\frac{f_{A2}}{e_{A2}}\right)\boldsymbol{k}_{\boldsymbol{A2}}^2 \tag{3.24}$$

The first derivatives of relative motion are equal to Eqn. (3.12), as with the other straight-link pantopatches. This leaves the following two approaches to finding the edge length $\boldsymbol{k}_{\boldsymbol{B2}}^2$:

$$\begin{aligned}\boldsymbol{k}_{\boldsymbol{B2}}^2 = {} & L_{B1}^2 + \left(\frac{L_{B1}}{e_{B1}}\right)\left(d_{B1}^2 + g_{B1}^2 - 2d_{B1}g_{B1}\cos\xi_{B1} - e_{B1}^2\right) \\ & - \left(\frac{f_{B1}}{e_{B1}}\right)\left[M_{A1}^2 + \left(\frac{M_{A1}}{g_{A1}}\right)\left(d_{A1}^2 + f_{A1}^2 - 2d_{A1}f_{A1}\cos\nu_{A1} - g_{A1}^2\right) - \left(\frac{h_{A1}}{g_{A1}}\right)\boldsymbol{k}_{\boldsymbol{A2}}^2\right]\end{aligned} \tag{3.25}$$

$$\begin{aligned}\boldsymbol{k}_{\boldsymbol{B2}}^2 = {} & M_{B2}^2 + \left(\frac{M_{B2}}{g_{B2}}\right)\left(d_{B2}^2 + e_{B2}^2 - 2d_{B2}e_{B2}\cos\nu_{B2} - g_{B2}^2\right) \\ & - \left(\frac{h_{B2}}{g_{B2}}\right)\left[L_{A2}^2 + \left(\frac{L_{A2}}{e_{A2}}\right)\left(d_{A2}^2 + h_{A2}^2 - 2d_{A2}h_{A2}\cos\xi_{A2} - e_{A2}^2\right) - \left(\frac{f_{A2}}{e_{A2}}\right)\boldsymbol{k}_{\boldsymbol{A2}}^2\right]\end{aligned} \tag{3.26}$$

Mobility is ensured when these two equations are solved simultaneously.

The straight-link pantopatches all have one mobility requirement in common: Eqn. (3.12). Additionally, the link lengths must all follow relation between $\boldsymbol{k}_{\boldsymbol{B2}}$ and $\boldsymbol{k}_{\boldsymbol{A2}}$ following both the "left" and "right" paths. The approach is similar in bent-link pantopatches, except that there are more requirements.

3.3 Mobility of Bent-Link Pantopatches

3.3.1 Mobility of Planar Bent-Link Pantopatches

The approach for ensuring mobility of bent-link pantomeshes is similar to that of straight-link meshes, except that the equations are generally more complex and the higher-order derivatives are not necessarily equal to zero. As before, $\boldsymbol{k}_{\boldsymbol{B2}}$ may be

expressed in terms of $\boldsymbol{j_{B1}}$ and $\boldsymbol{j_{B2}}$ as shown in Eqns. (3.27) and (3.28), respectively.

$$\begin{aligned}\boldsymbol{k_{B2}^2} = \boldsymbol{j_{B1}^2} + L_{B1}^2 - (\boldsymbol{j_{B1}^2} + e_{B1}^2 - g_{B1}^2)\left(\frac{L_{B1}}{e_{B1}}\right)\cos\alpha_{B1} \\ - \left[4\boldsymbol{j_{B1}^2}e_{B1}^2 - (\boldsymbol{j_{B1}^2} + e_{B1}^2 - g_{B1}^2)^2\right]^{1/2}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \qquad (3.27)\end{aligned}$$

$$\begin{aligned}\boldsymbol{k_{B2}^2} = \boldsymbol{j_{B2}^2} + M_{B2}^2 - (\boldsymbol{j_{B2}^2} + g_{B2}^2 - e_{B2}^2)\left(\frac{M_{B2}}{g_{B2}}\right)\cos\beta_{B2} \\ - \left[4\boldsymbol{j_{B2}^2}g_{B2}^2 - (\boldsymbol{j_{B2}^2} + g_{B2}^2 - e_{B2}^2)^2\right]^{1/2}\left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2} \qquad (3.28)\end{aligned}$$

Furthermore, $\boldsymbol{j_{A1}}$ and $\boldsymbol{j_{B1}}$ may be expressed in terms of $\boldsymbol{k_{A2}}$.

$$\begin{aligned}\boldsymbol{j_{B1}^2} = \boldsymbol{k_{A2}^2} + M_{A1}^2 - \left[\boldsymbol{k_{A2}^2} + g_{A1}^2 - f_{A1}^2\right]\left(\frac{M_{A1}}{g_{A1}}\right)\cos\beta_{A1} \\ + \left[4\boldsymbol{k_{A2}^2}g_{A1}^2 - \left(\boldsymbol{k_{A2}^2} + g_{A1}^2 - (f_{A1})^2\right)^2\right]^{1/2}\left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1} \qquad (3.29)\end{aligned}$$

$$\begin{aligned}\boldsymbol{j_{B2}^2} = \boldsymbol{k_{A2}^2} + L_{A2}^2 - \left[\boldsymbol{k_{A2}^2} + e_{A2}^2 - (h_{A2})^2\right]\left(\frac{L_{A2}}{e_{A2}}\right)\cos\alpha_{A2} \\ + \left[4\boldsymbol{k_{A2}^2}e_{A2}^2 - \left(\boldsymbol{k_{A2}^2} + e_{A2}^2 - h_{A2}^2\right)^2\right]^{1/2}\left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2} \qquad (3.30)\end{aligned}$$

To ensure mobility, the change in length of $\boldsymbol{k_{B2}}$ relative to $\boldsymbol{k_{A2}}$ must be the same whether calculated via the pantographs of $A1$ and $B1$ or the pantographs of $A2$ and $B2$, the same as calculating with either $\boldsymbol{j_{B1}}$ or $\boldsymbol{j_{B2}}$.

$$\frac{d(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})} = \left(\frac{d(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B1}^2})}\right)\left(\frac{d(\boldsymbol{j_{B1}^2})}{d(\boldsymbol{k_{A2}^2})}\right) = \left(\frac{d(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B2}^2})}\right)\left(\frac{d(\boldsymbol{j_{B2}^2})}{d(\boldsymbol{k_{A2}^2})}\right) \qquad (3.31)$$

These terms are as follows:

$$\frac{d(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B1}}^2)} = 1 - \left(\frac{L_{B1}}{e_{B1}}\right)\cos\alpha_{B1} - \left(\frac{1}{2}\right)\frac{(-2\boldsymbol{j}_{\boldsymbol{B1}}^2 + 2e_{B1}^2 + 2g_{B1}^2)\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1}}{[-\boldsymbol{j}_{\boldsymbol{B1}}^4 - e_{B1}^4 - g_{B1}^4 + 2\boldsymbol{j}_{\boldsymbol{B1}}^2 e_{B1}^2 + 2\boldsymbol{j}_{\boldsymbol{B1}}^2 g_{B1}^2 + 2e_{B1}^2 g_{B1}^2]^{1/2}} \tag{3.32}$$

$$\frac{d(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B2}}^2)} = 1 - \left(\frac{M_{B2}}{g_{B2}}\right)\cos\beta_{B2} - \left(\frac{1}{2}\right)\frac{(-2\boldsymbol{j}_{\boldsymbol{B2}}^2 + 2g_{B2}^2 + 2e_{B2}^2)\left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2}}{[-\boldsymbol{j}_{\boldsymbol{B2}}^4 - g_{B2}^4 - e_{B2}^4 + 2\boldsymbol{j}_{\boldsymbol{B2}}^2 g_{B2}^2 + 2\boldsymbol{j}_{\boldsymbol{B2}}^2 e_{B2}^2 + 2g_{B2}^2 e_{B2}^2]^{1/2}} \tag{3.33}$$

which simplifies to

$$\frac{d(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B1}}^2)} = 1 - \left(\frac{L_{B1}}{e_{B1}}\right)\cos\alpha_{B1} + \frac{(\boldsymbol{j}_{\boldsymbol{B1}}^2 - e_{B1}^2 - g_{B1}^2)\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1}}{[4\boldsymbol{j}_{\boldsymbol{B1}}^2 e_{B1}^2 - (\boldsymbol{j}_{\boldsymbol{B1}}^2 + e_{B1}^2 - g_{B1}^2)^2]^{1/2}} \tag{3.34}$$

$$\frac{d(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B2}}^2)} = 1 - \left(\frac{M_{B2}}{g_{B2}}\right)\cos\beta_{B2} + \frac{(\boldsymbol{j}_{\boldsymbol{B2}}^2 - g_{B2}^2 - e_{B2}^2)\left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2}}{[4\boldsymbol{j}_{\boldsymbol{B2}}^2 g_{B2}^2 - (\boldsymbol{j}_{\boldsymbol{B2}}^2 + g_{B2}^2 - e_{B2}^2)^2]^{1/2}} \tag{3.35}$$

The other terms are similar:

$$\frac{d(\boldsymbol{j}_{\boldsymbol{B1}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)} = 1 - \left(\frac{M_{A1}}{g_{A1}}\right)\cos\beta_{A1} - \frac{(\boldsymbol{k}_{\boldsymbol{A2}}^2 - g_{A1}^2 - f_{A1})\left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1}}{\left[4\boldsymbol{k}_{\boldsymbol{A2}}^2 g_{A1}^2 - (\boldsymbol{k}_{\boldsymbol{A2}}^2 + g_{A1}^2 - (f_{A1})^2)^2\right]^{1/2}} \tag{3.36}$$

$$\frac{d(\boldsymbol{j}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)} = 1 - \left(\frac{L_{A2}}{e_{A2}}\right)\cos\alpha_{A2} - \frac{(\boldsymbol{k}_{\boldsymbol{A2}}^2 - e_{A2}^2 - h_{A2})\left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2}}{\left[4\boldsymbol{k}_{\boldsymbol{A2}}^2 e_{A2}^2 - (\boldsymbol{k}_{\boldsymbol{A2}}^2 + e_{A2}^2 - (h_{A2})^2)^2\right]^{1/2}} \tag{3.37}$$

By inspection, it is clear that further derivatives will not be equal to zero as in the straight-link cases. Therefore, it is beneficial to express these terms differently, by

noting the following identities:

$$\cos\boldsymbol{\theta_{A1e}} = \frac{-\left(\boldsymbol{k_{A2}^2} - g_{A1}^2 - f_{A1}^2\right)}{2g_{A1}f_{A1}} \tag{3.38}$$

$$\sin\boldsymbol{\theta_{A1e}} = \frac{\left[4\boldsymbol{k_{A2}^2}g_{A1} - \left(\boldsymbol{k_{A2}^2} + g_{A1}^2 - f_{A1}^2\right)^2\right]^{1/2}}{2g_{A1}f_{A1}} \tag{3.39}$$

$$\cos\boldsymbol{\theta_{A2w}} = \frac{-\left(\boldsymbol{k_{A2}^2} - e_{A2}^2 - h_{A2}^2\right)}{2e_{A2}h_{A2}} \tag{3.40}$$

$$\sin\boldsymbol{\theta_{A2w}} = \frac{\left[4\boldsymbol{k_{A2}^2}e_{A2} - \left(\boldsymbol{k_{A2}^2} + e_{A2}^2 - h_{A2}^2\right)^2\right]^{1/2}}{2e_{A2}h_{A2}} \tag{3.41}$$

$$\cos\boldsymbol{\theta_{B1s}} = \frac{-\left(\boldsymbol{j_{B1}^2} - e_{B1}^2 - g_{B1}^2\right)}{2e_{B1}g_{B1}} \tag{3.42}$$

$$\sin\boldsymbol{\theta_{B1s}} = \frac{\left[4\boldsymbol{j_{B1}^2}e_{B1}^2 - \left(\boldsymbol{j_{B1}^2} + e_{B1}^2 - g_{B1}^2\right)^2\right]^{1/2}}{2e_{B1}g_{B1}} \tag{3.43}$$

$$\cos\boldsymbol{\theta_{B2s}} = \frac{-\left(\boldsymbol{j_{B2}^2} - g_{B2}^2 - e_{B2}^2\right)}{2e_{B2}g_{B2}} \tag{3.44}$$

$$\sin\boldsymbol{\theta_{B2s}} = \frac{\left[4\boldsymbol{j_{B2}^2}g_{B2}^2 - \left(\boldsymbol{j_{B2}^2} + g_{B2}^2 - e_{B2}^2\right)^2\right]^{1/2}}{2e_{B2}g_{B2}} \tag{3.45}$$

Therefore, the first derivatives may be expressed as follows:

$$\frac{d(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B1}^2})} = 1 - \left(\frac{L_{B1}}{e_{B1}}\right)\cos\alpha_{B1} + \left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1}\cot\boldsymbol{\theta_{B1s}} \tag{3.46}$$

$$\frac{d(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B2}^2})} = 1 - \left(\frac{M_{B2}}{g_{B2}}\right)\cos\beta_{B2} + \left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2}\cot\boldsymbol{\theta_{B2s}} \tag{3.47}$$

$$\frac{d(\boldsymbol{j_{B1}^2})}{d(\boldsymbol{k_{A2}^2})} = 1 - \left(\frac{M_{A1}}{g_{A1}}\right)\cos\beta_{A1} - \left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1}\cot\boldsymbol{\theta_{A1e}} \tag{3.48}$$

$$\frac{d(\boldsymbol{j_{B2}^2})}{d(\boldsymbol{k_{A2}^2})} = 1 - \left(\frac{L_{A2}}{e_{A2}}\right)\cos\alpha_{A2} - \left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2}\cot\boldsymbol{\theta_{A2w}} \tag{3.49}$$

The calculations will now continue onto higher-order derivatives. Multiple levels of differentiation will be required to establish a pattern and create a general solution to the mobility of a pantopatch. All derivatives must also obey these general equations of motion (following Eqn. (3.31)), where n is the derivative order:

$$\frac{d^n(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})^n} = \left(\frac{d^n(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B1}^2})^n}\right)\left(\frac{d^n(\boldsymbol{j_{B1}^2})}{d(\boldsymbol{k_{A2}^2})^n}\right) = \left(\frac{d^n(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B2}^2})^n}\right)\left(\frac{d^n(\boldsymbol{j_{B2}^2})}{d(\boldsymbol{k_{A2}^2})^n}\right) \tag{3.50}$$

Therefore, the second-order derivatives are as follows:

$$\frac{d^2(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B1}}^2)^2} = \left(\frac{L_{B1}}{e_{B1}}\right) \frac{\sin\alpha_{B1}}{2e_{B1}g_{B1}\sin^3\boldsymbol{\theta}_{\boldsymbol{B1s}}} \tag{3.51}$$

$$\frac{d^2(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B2}}^2)^2} = \left(\frac{M_{B2}}{g_{B2}}\right) \frac{\sin\beta_{B2}}{2g_{B2}e_{B2}\sin^3\boldsymbol{\theta}_{\boldsymbol{B2s}}} \tag{3.52}$$

$$\frac{d^2(\boldsymbol{j}_{\boldsymbol{B1}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)^2} = \left(\frac{M_{A1}}{g_{A1}}\right) \frac{-\sin\beta_{A1}}{2g_{A1}f_{A1}\sin^3\boldsymbol{\theta}_{\boldsymbol{A1e}}} \tag{3.53}$$

$$\frac{d^2(\boldsymbol{j}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)^2} = \left(\frac{L_{A2}}{e_{A2}}\right) \frac{-\sin\alpha_{A2}}{2e_{A2}h_{A2}\sin^3\boldsymbol{\theta}_{\boldsymbol{A2w}}} \tag{3.54}$$

Now combining with the second-order version of Eqn. (3.50), we get the following relation:

$$\begin{aligned}\frac{d^2(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)^2} &= \left(\frac{L_{B1}M_{A1}}{e_{B1}g_{A1}}\right) \frac{-\sin\alpha_{B1}\sin\beta_{A1}}{4e_{B1}g_{B1}g_{A1}f_{A1}\sin^3\boldsymbol{\theta}_{\boldsymbol{B1s}}\sin^3\boldsymbol{\theta}_{\boldsymbol{A1e}}} \\ &= \left(\frac{M_{B2}L_{A2}}{g_{B2}e_{A2}}\right) \frac{-\sin\beta_{B2}\sin\alpha_{A2}}{4g_{B2}e_{B2}e_{A2}h_{A2}\sin^3\boldsymbol{\theta}_{\boldsymbol{B2s}}\sin^3(\boldsymbol{\theta}_{\boldsymbol{A1w}})}\end{aligned} \tag{3.55}$$

$$\frac{\sin^3\boldsymbol{\theta}_{\boldsymbol{B1s}}\sin^3\boldsymbol{\theta}_{\boldsymbol{A1e}}}{\sin^3\boldsymbol{\theta}_{\boldsymbol{B2s}}\sin^3(\boldsymbol{\theta}_{\boldsymbol{A1w}})} = \frac{L_{B1}M_{A1}g_{B2}^2e_{B2}^2e_{A2}h_{A2}\sin\alpha_{B1}\sin\beta_{A1}}{M_{B2}L_{A2}e_{B1}^2g_{B1}^2g_{A1}f_{A1}\sin\beta_{B2}\sin\alpha_{A2}} \tag{3.56}$$

The approach for third-order derivatives is identical:

$$\frac{d^3(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B1}}^2)^3} = \frac{-3\cos\boldsymbol{\theta}_{\boldsymbol{B1s}}}{4e_{B1}^2g_{B1}^2\sin^5\boldsymbol{\theta}_{\boldsymbol{B1s}}} \left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.57}$$

$$\frac{d^3(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B2}}^2)^3} = \frac{-3\cos\boldsymbol{\theta}_{\boldsymbol{B2s}}}{4g_{B2}^2e_{B2}^2\sin^5\boldsymbol{\theta}_{\boldsymbol{B2s}}} \left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2} \tag{3.58}$$

$$\frac{d^3(\boldsymbol{j}_{\boldsymbol{B1}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)^3} = \frac{3\cos\boldsymbol{\theta}_{\boldsymbol{A1e}}}{4g_{A1}^2f_{A1}^2\sin^5\boldsymbol{\theta}_{\boldsymbol{A1e}}} \left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1} \tag{3.59}$$

$$\frac{d^3(\boldsymbol{j}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)^3} = \frac{3\cos\boldsymbol{\theta}_{\boldsymbol{A2w}}}{4e_{A2}^2h_{A2}^2\sin^5\boldsymbol{\theta}_{\boldsymbol{A2w}}} \left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2} \tag{3.60}$$

And, to obey motion continuity:

$$\begin{aligned}\frac{d^3(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{k}_{\boldsymbol{A2}}^2)^3} &= \frac{-9\cos\boldsymbol{\theta}_{\boldsymbol{B1s}}\cos\boldsymbol{\theta}_{\boldsymbol{A1e}}}{16e_{B1}^2g_{B1}^2g_{A1}^2f_{A1}^2\sin^5\boldsymbol{\theta}_{\boldsymbol{B1s}}\sin^5\boldsymbol{\theta}_{\boldsymbol{A1e}}} \left(\frac{L_{B1}M_{A1}}{e_{B1}g_{A1}}\right)\sin\alpha_{B1}\sin\beta_{A1} \\ &= \frac{-9\cos\boldsymbol{\theta}_{\boldsymbol{B2s}}\cos\boldsymbol{\theta}_{\boldsymbol{A2w}}}{16g_{B2}^2e_{B2}^2e_{A2}^2h_{A2}^2\sin^5\boldsymbol{\theta}_{\boldsymbol{B2s}}\sin^5\boldsymbol{\theta}_{\boldsymbol{A2w}}} \left(\frac{M_{B2}L_{A2}}{g_{B2}e_{A2}}\right)\sin\beta_{B2}\sin\alpha_{A2}\end{aligned} \tag{3.61}$$

$$\frac{\sin^5\boldsymbol{\theta_{B1s}}\sin^5\boldsymbol{\theta_{A1e}}\cos\boldsymbol{\theta_{B2s}}\cos\boldsymbol{\theta_{A2w}}}{\sin^5\boldsymbol{\theta_{B2s}}\sin^5\boldsymbol{\theta_{A2w}}\cos\boldsymbol{\theta_{B1s}}\cos\boldsymbol{\theta_{A1e}}} = \frac{L_{B1}M_{A1}g_{B2}^3e_{B2}^3e_{A2}^2h_{A2}^2\sin\alpha_{B1}\sin\beta_{A1}}{M_{B2}L_{A2}e_{B1}^3g_{B1}^3g_{A1}^2f_{A1}^2\sin\beta_{B2}\sin\alpha_{A2}} \tag{3.62}$$

The mobility equations concerning the second and third derivatives (Eqns. (3.56) and (3.62)) may be combined to reduce terms:

$$\frac{\sin^2\boldsymbol{\theta_{B1s}}\sin^2\boldsymbol{\theta_{A1e}}\cos\boldsymbol{\theta_{B2s}}\cos\boldsymbol{\theta_{A2w}}}{\sin^2\boldsymbol{\theta_{B2s}}\sin^2\boldsymbol{\theta_{A2w}}\cos\boldsymbol{\theta_{B1s}}\cos\boldsymbol{\theta_{A1e}}} = \frac{g_{B2}e_{B2}e_{A2}h_{A2}}{e_{B1}g_{B1}g_{A1}f_{A1}} \tag{3.63}$$

Moving on, the fourth-order derivatives deliver a term important for resolution of these sets of equations:

$$\frac{d^4(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})^4} = \frac{3+12\cos\boldsymbol{\theta_{B1s}}}{8e_{B1}^3g_{B1}^3\sin^7\boldsymbol{\theta_{B1s}}}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.64}$$

$$\frac{d^4(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B2}^2})^4} = \frac{3+12\cos\boldsymbol{\theta_{B2s}}}{8g_{B2}^3e_{B2}^3\sin^7\boldsymbol{\theta_{B2s}}}\left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2} \tag{3.65}$$

$$\frac{d^4(\boldsymbol{j_{B1}^2})}{d(\boldsymbol{k_{A2}^2})^4} = \frac{-3-12\cos\boldsymbol{\theta_{A1e}}}{8g_{A1}^3f_{A1}^3\sin^7\boldsymbol{\theta_{A1e}}}\left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1} \tag{3.66}$$

$$\frac{d^4(\boldsymbol{j_{B2}^3})}{d(\boldsymbol{k_{A2}^2})^4} = \frac{-3-12\cos\boldsymbol{\theta_{A2w}}}{8e_{A2}^3h_{A2}^3\sin^7\boldsymbol{\theta_{A2w}}}\left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2} \tag{3.67}$$

This leads to:

$$\frac{d^4(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})^4} = \frac{(3+12\cos\boldsymbol{\theta_{B1s}})(-3-12\cos\boldsymbol{\theta_{A1e}})}{64e_{B1}^3g_{B1}^3g_{A1}^3f_{A1}^3\sin^7\boldsymbol{\theta_{B1s}}\sin^7\boldsymbol{\theta_{A1e}}}\left(\frac{L_{B1}M_{A1}}{e_{B1}g_{A1}}\right)\sin\alpha_{B1}\sin\beta_{A1} \tag{3.68}$$

$$= \frac{(3+12\cos\boldsymbol{\theta_{B2s}})(-3-12\cos\boldsymbol{\theta_{A2w}})}{64g_{B2}^3e_{B2}^3e_{A2}^3h_{A2}^3\sin^7\boldsymbol{\theta_{B2s}}\sin^7\boldsymbol{\theta_{A2w}}}\left(\frac{M_{B2}L_{A2}}{g_{B2}e_{A2}}\right)\sin\beta_{B2}\sin\alpha_{A2} \tag{3.69}$$

$$\frac{\sin^7\boldsymbol{\theta_{B2s}}\sin^7\boldsymbol{\theta_{A2w}}(1+4\cos\boldsymbol{\theta_{B2s}})(1+4\cos\boldsymbol{\theta_{A2w}})}{\sin^7\boldsymbol{\theta_{B1s}}\sin^7\boldsymbol{\theta_{A1e}}(1+4\cos\boldsymbol{\theta_{B1s}})(1+4\cos\boldsymbol{\theta_{A1e}})}$$
$$= \frac{L_{B1}M_{A1}g_{B2}^4e_{B2}^4e_{A2}^3h_{A2}^3\sin\alpha_{B1}\sin\beta_{A1}}{M_{B2}L_{A2}e_{B1}^4g_{B1}^4g_{A1}^3f_{A1}^3\sin\beta_{B2}\sin\alpha_{A2}} \tag{3.70}$$

As before, this result may be combined with the third-order derivative equation from Eqn. (3.62) to simplify.

$$\frac{\sin^2\boldsymbol{\theta_{B2s}}\sin^2\boldsymbol{\theta_{A2w}}(1+4\cos\boldsymbol{\theta_{B2s}})(1+4\cos\boldsymbol{\theta_{A1e}})}{\sin^2\boldsymbol{\theta_{B1s}}\sin^2\boldsymbol{\theta_{A1e}}(1+4\cos\boldsymbol{\theta_{B1s}})(1+4\cos\boldsymbol{\theta_{A1e}})} = \frac{g_{B2}e_{B2}e_{A2}h_{A2}}{e_{B1}g_{B1}g_{A1}f_{A1}} \tag{3.71}$$

Furthermore, this can be reduced with the already-combined Eqn. (3.63) to yield

the following:

$$\frac{(1+4\cos\boldsymbol{\theta_{B2s}})(1+4\cos\boldsymbol{\theta_{A2w}})}{(1+4\cos\boldsymbol{\theta_{B1s}})(1+4\cos\boldsymbol{\theta_{A1e}})}=\frac{\cos\boldsymbol{\theta_{B2s}}\cos\boldsymbol{\theta_{A2w}}}{\cos\boldsymbol{\theta_{B1s}}\cos\boldsymbol{\theta_{A1e}}} \tag{3.72}$$

or

$$(\sec\boldsymbol{\theta_{B2s}}+4)\,(\sec\boldsymbol{\theta_{A2w}}+4)=(\sec\boldsymbol{\theta_{B1s}}+4)\,(\sec\boldsymbol{\theta_{A1e}}+4) \tag{3.73}$$

Finally, the fifth-order derivatives are found to be:

$$\frac{d^5(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})^5}=\frac{-45\cos\boldsymbol{\theta_{B1s}}-60\cos^3\boldsymbol{\theta_{B1s}}}{16e_{B1}^4g_{B1}^4\sin^9\boldsymbol{\theta_{B1s}}}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.74}$$

$$\frac{d^5(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{j_{B2}^2})^5}=\frac{-45\cos\boldsymbol{\theta_{B2s}}-60\cos^3\boldsymbol{\theta_{B2s}}}{16g_{B2}^4e_{B2}^4\sin^9\boldsymbol{\theta_{B2s}}}\left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2} \tag{3.75}$$

$$\frac{d^5(\boldsymbol{j_{B1}^2})}{d(\boldsymbol{k_{A2}^2})^5}=\frac{45\cos\boldsymbol{\theta_{A1e}}+60\cos^3\boldsymbol{\theta_{A1e}}}{16g_{A1}^4f_{A1}^4\sin^9\boldsymbol{\theta_{A1e}}}\left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1} \tag{3.76}$$

$$\frac{d^5(\boldsymbol{j_{B2}^3})}{d(\boldsymbol{k_{A2}^2})^5}=\frac{45\cos\boldsymbol{\theta_{A2w}}+60\cos^3\boldsymbol{\theta_{A2w}}}{16e_{A2}^4h_{A2}^4\sin^9\boldsymbol{\theta_{A2w}}}\left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2} \tag{3.77}$$

Therefore, for fifth-order mobility to be maintained, the following must be true:

$$\frac{d^5(\boldsymbol{k_{B2}^2})}{d(\boldsymbol{k_{A2}^2})^5} \tag{3.78}$$

$$=\frac{(-45\cos\boldsymbol{\theta_{B1s}}-60\cos^3\boldsymbol{\theta_{B1s}})(45\cos\boldsymbol{\theta_{A1e}}+60\cos^3\boldsymbol{\theta_{A1e}})}{256e_{B1}^4g_{B1}^4g_{A1}^4f_{A1}^4\sin^9\boldsymbol{\theta_{B1s}}\sin^9\boldsymbol{\theta_{A1e}}}\left(\frac{L_{B1}M_{A1}}{e_{B1}g_{A1}}\right)\sin\alpha_{B1}\sin\beta_{A1} \tag{3.79}$$

$$=\frac{(-45\cos\boldsymbol{\theta_{B2s}}-60\cos^3\boldsymbol{\theta_{B2s}})(45\cos\boldsymbol{\theta_{A2w}}+60\cos^3\boldsymbol{\theta_{A2w}})}{256g_{B2}^4e_{B2}^4e_{A2}^4h_{A2}^4\sin^9\boldsymbol{\theta_{B2s}}\sin^9\boldsymbol{\theta_{A2w}}}\left(\frac{M_{B2}L_{A2}}{g_{B2}e_{A2}}\right)\sin\beta_{B2}\sin\alpha_{A2} \tag{3.80}$$

$$\frac{\sin^9\boldsymbol{\theta_{B2s}}\sin^9\boldsymbol{\theta_{A2w}}\cos\boldsymbol{\theta_{B2s}}(3+4\cos^2\boldsymbol{\theta_{B2s}})\cos\boldsymbol{\theta_{A2w}}(3+4\cos^2\boldsymbol{\theta_{A2w}})}{\sin^9\boldsymbol{\theta_{B1s}}\sin^9\boldsymbol{\theta_{A1e}}\cos\boldsymbol{\theta_{B1s}}(3+4\cos^2\boldsymbol{\theta_{B1s}})\cos\boldsymbol{\theta_{A1e}}(3+4\cos^2\boldsymbol{\theta_{A1e}})}$$
$$=\frac{L_{B1}M_{A1}g_{B2}^5e_{B2}^5e_{A2}^4h_{A2}^4\sin\alpha_{B1}\sin\beta_{A1}}{M_{B2}L_{A2}e_{B1}^5g_{B1}^5g_{A1}^4f_{A1}^4\sin\beta_{B2}\sin\alpha_{A2}} \tag{3.81}$$

As before, we can combine this with the fourth-order derivative solution from

Eqn. (3.70) to obtain:

$$\frac{\sin^2\boldsymbol{\theta_{B2s}}\sin^2\boldsymbol{\theta_{A2w}}\cos\boldsymbol{\theta_{B2s}}\cos\boldsymbol{\theta_{A2w}}(3+4\cos^2\boldsymbol{\theta_{B2s}})(3+4\cos^2\boldsymbol{\theta_{A2w}})}{\sin^2\boldsymbol{\theta_{B1s}}\sin^2\boldsymbol{\theta_{A1e}}\cos\boldsymbol{\theta_{B1s}}\cos\boldsymbol{\theta_{A1e}}(3+4\cos^2\boldsymbol{\theta_{B1s}})(3+4\cos^2\boldsymbol{\theta_{A1e}})} = \frac{g_{B2}e_{B2}e_{A2}h_{A2}}{e_{B1}g_{B1}g_{A1}f_{A1}} \tag{3.82}$$

Finally, this result may be combined with Eqn. (3.63) and rearranged to get the following:

$$(3+4\cos^2\boldsymbol{\theta_{B2s}})(3+4\cos^2\boldsymbol{\theta_{A2w}}) = (3+4\cos^2\boldsymbol{\theta_{B1s}})(3+4\cos^2\boldsymbol{\theta_{A1e}}) \tag{3.83}$$

Now, using the previous equation and Eqn. (3.73)

$$\cos\boldsymbol{\theta_{B2s}} = \cos\boldsymbol{\theta_{A2w}} \qquad \text{and} \qquad \cos\boldsymbol{\theta_{B1s}} = \cos\boldsymbol{\theta_{A1e}} \tag{3.84}$$

or

$$\cos\boldsymbol{\theta_{B2s}} = \cos\boldsymbol{\theta_{A1e}} \qquad \text{and} \qquad \cos\boldsymbol{\theta_{B1s}} = \cos\boldsymbol{\theta_{A2w}} \tag{3.85}$$

However, only Eqn. (3.84) is appropriate, since the relevant equation pair must be valid for the entire range of motion. To explain further, imagine the scissor motion of a bent-link pantopatch — whenever $\boldsymbol{\theta_{A1e}}$ increases, $\boldsymbol{\theta_{B1s}}$ must then decrease. Furthermore, the other angles must obey the relations as well. To summarize:

$$\boldsymbol{\theta_{A1e}} = \boldsymbol{\theta_{A2w}} \tag{3.86}$$

$$\boldsymbol{\theta_{A1n}} = \boldsymbol{\theta_{B1s}} \tag{3.87}$$

$$\boldsymbol{\theta_{A2n}} = \boldsymbol{\theta_{B2s}} \tag{3.88}$$

$$\boldsymbol{\theta_{B1e}} = \boldsymbol{\theta_{B2w}} \tag{3.89}$$

In other words, the angles *opposite* in the pantopatch must be equal. To force equality on opposite pivot intersection angles during the entire range of motion, either opposite sides or adjacent pairs must be equal in length. To see why this is true, consider the following cosine-law relation:

$$\boldsymbol{k_{A2}^2} = g_{A1}^2 + f_{A1}^2 - 2g_{A1}f_{A1}\cos\boldsymbol{\theta_{A1e}} = e_{A2}^2 + h_{A2}^2 - e_{A2}h_{A2}\cos\boldsymbol{\theta_{A2w}} \tag{3.90}$$

After recognizing the equal angles, we can collect terms:

$$g_{A1}^2 + f_{A1}^2 - e_{A2}^2 - h_{A2}^2 = 2\left(g_{A1}f_{A1} - e_{A2}h_{A2}\right)\cos\boldsymbol{\theta_{A1e}} \tag{3.91}$$

By inspection, it can be seen that Eqn. (3.91) sets a constant term equal to a variable term (with $\boldsymbol{\theta_{A1e}}$). This is only possible if both of the constant terms are equal to zero. After rearranging, this leads to the following relations:

$$g_{A1}^2 + f_{A1}^2 = e_{A2}^2 + h_{A2}^2 \tag{3.92}$$

$$g_{A1} f_{A1} = e_{A2} h_{A2} \tag{3.93}$$

Therefore the only two viable solutions are:

$$g_{A1} = e_{A2} \quad \text{and} \quad f_{A1} = h_{A2} \tag{3.94}$$

or

$$g_{A1} = h_{A2} \quad \text{and} \quad f_{A1} = e_{A2} \tag{3.95}$$

These requirements are the same for all four equal-angle requirements (Eqns. (3.86) through (3.89)). These relations, each with four link lengths, force two different kinds of quadrilaterals as their shape, specifically either parallelogram or kite. These exact relations are shown in Table 3.1. Assembled into a 2x2 pantopatch

Location	Parallelogram	Kite
"south"	$g_{A1} = h_{A2}$ $f_{A1} = e_{A2}$	$g_{A1} = e_{A2}$ $f_{A1} = h_{A2}$
"west"	$f_{A1} = e_{B1}$ $h_{A1} = g_{B1}$	$f_{A1} = g_{B1}$ $h_{A1} = e_{B1}$
"east"	$f_{A2} = e_{B2}$ $h_{A2} = g_{B2}$	$f_{A2} = g_{B2}$ $h_{A2} = e_{B2}$
"north"	$g_{B1} = h_{B2}$ $f_{B1} = e_{B2}$	$g_{B1} = e_{B2}$ $f_{B1} = h_{B2}$

Table 3.1: Equal-angle configurations for bent-link pantopatch mobility.

and denoting a parallelogram or kite as P or K, respectively, the following different bent-link pantopatch types are possible: 4P, 3PK, 2(KP), 2K2P, and 4K. Type 3KP is not among these since it is actually a 4K (or perhaps a 3KR, where R is a rhombus). In a 3KP pantopatch, any parallelogram would have all equal sides, thus being a rhombus (which is both a parallelogram and a kite). There are also different configurations, depending on which quadrilateral is assigned a shape, for some of the types as shown in Table 3.2, with directional labels for relational positions of the quadrilaterals.

Finally, there are additional constraints put upon the angle of pantopatches. Since

Type	4P	3PK				2(KP)		2K2P				4K
W Quad	P	K	P	P	P	K	P	K	P	P	K	K
S Quad	P	P	K	P	P	P	K	K	K	P	P	K
E Quad	P	P	P	K	P	K	P	P	K	K	P	K
N Quad	P	P	P	P	K	P	K	P	P	K	K	K

Table 3.2: Kite and parallelogram configurations for bent-link pantopatch mobility.

all angles around a pivot equal 360°, it can be easily seen that the following relations are true:

$$\hat{\beta}_{A1} + \boldsymbol{\theta_{A1e}} + \boldsymbol{\theta_{A1n}} = 360^\circ \tag{3.96}$$

$$\hat{\alpha}_{A2} + \boldsymbol{\theta_{A2w}} + \boldsymbol{\theta_{A2n}} = 360^\circ \tag{3.97}$$

$$\hat{\alpha}_{B1} + \boldsymbol{\theta_{B1s}} + \boldsymbol{\theta_{B1e}} = 360^\circ \tag{3.98}$$

$$\hat{\beta}_{B2} + \boldsymbol{\theta_{B2s}} + \boldsymbol{\theta_{B2w}} = 360^\circ \tag{3.99}$$

Combining these equations with the four equal-angle requirements from Eqns. (3.86) through (3.89), the following *must* be true:

$$\hat{\beta}_{A1} - \hat{\alpha}_{A2} + \hat{\alpha}_{B1} - \hat{\beta}_{B2} = 0^\circ \tag{3.100}$$

This result concludes the mobility requirements for planar bent-link pantopatches. Despite the long derivation, the result is quite straightforward: the equal-angle requirements satisfy all mobility requirements beyond the first derivative of motion. Put in other terms: if the link bend angles ($\hat{\alpha}$s and $\hat{\beta}$s) equal 180°, then the equal-angle requirements of bent-link mobility are trivially solved.

As mentioned in Chapter 2, the bent-link pantopatch gives more flexibility in design than the planar straight-link, but the range of motion can be problematic. Therefore, the next sections investigate skew bent-link pantopatches, which have greater range of motion.

3.3.2 Mobility of Skew, Normal-Pivot Bent-Link Pantopatches

By adding an offset distance between the links, a skew pantopatch is created. Skew bent-link pantograph elements with normal pivots are solved in a similar fashion

as the planar pantograph elements, only adding the d term.

$$\boldsymbol{k}_{\boldsymbol{B2}}^2 = \boldsymbol{j}_{\boldsymbol{B1}}^2 + L_{B1}^2 - \left(\left(\boldsymbol{j}_{\boldsymbol{B1}}^2 - d_{B1}^2\right) + e_{B1}^2 - g_{B1}^2\right)\left(\frac{L_{B1}}{e_{B1}}\right)\cos\alpha_{B1} - \left[4\left(\boldsymbol{j}_{\boldsymbol{B1}}^2 - d_{B1}^2\right)e_{B1}^2 - \left(\left(\boldsymbol{j}_{\boldsymbol{B1}}^2 - d_{B1}^2\right) + e_{B1}^2 - g_{B1}^2\right)^2\right]^{1/2}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.101}$$

$$\boldsymbol{k}_{\boldsymbol{B2}}^2 = \boldsymbol{j}_{\boldsymbol{B2}}^2 + M_{B2}^2 - \left(\left(\boldsymbol{j}_{\boldsymbol{B2}}^2 - d_{B2}^2\right) + g_{B2}^2 - e_{B2}^2\right)\left(\frac{M_{B2}}{g_{B2}}\right)\cos\beta_{B2} - \left[4\left(\boldsymbol{j}_{\boldsymbol{B2}}^2 - d_{B2}^2\right)g_{B2}^2 - \left(\left(\boldsymbol{j}_{\boldsymbol{B2}}^2 - d_{B2}^2\right) + g_{B2}^2 - e_{B2}^2\right)^2\right]^{1/2}\left(\frac{M_{B2}}{g_{B2}}\right)\sin\beta_{B2} \tag{3.102}$$

$$\boldsymbol{j}_{\boldsymbol{B1}}^2 = \boldsymbol{k}_{\boldsymbol{A2}}^2 + M_{A1}^2 - \left[\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A1}^2\right) + g_{A1}^2 - f_{A1}^2\right]\left(\frac{M_{A1}}{g_{A1}}\right)\cos\beta_{A1} + \left[4\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A1}^2\right)g_{A1}^2 - \left(\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A1}^2\right) + g_{A1}^2 - (f_{A1})^2\right)^2\right]^{1/2}\left(\frac{M_{A1}}{g_{A1}}\right)\sin\beta_{A1} \tag{3.103}$$

$$\boldsymbol{j}_{\boldsymbol{B2}}^2 = \boldsymbol{k}_{\boldsymbol{A2}}^2 + L_{A2}^2 - \left[\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A2}^2\right) + e_{A2}^2 - h_{A2}^2\right]\left(\frac{L_{A2}}{e_{A2}}\right)\cos\alpha_{A2} + \left[4\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A2}^2\right)e_{A2}^2 - \left(\left(\boldsymbol{k}_{\boldsymbol{A2}}^2 - d_{A2}^2\right) + e_{A2}^2 - h_{A2}^2\right)^2\right]^{1/2}\left(\frac{L_{A2}}{e_{A2}}\right)\sin\alpha_{A2} \tag{3.104}$$

In the end, these equations will require the same process of mobility determination as the planar version:

$$\frac{d(\boldsymbol{k}_{\boldsymbol{B2}}^2)}{d(\boldsymbol{j}_{\boldsymbol{B1}}^2)} = 1 - \left(\frac{L_{B1}}{e_{B1}}\right)\cos\alpha_{B1} - \frac{\left(\left(\boldsymbol{j}_{\boldsymbol{B1}}^2 - d_{B1}^2\right) - e_{B1}^2 - g_{B1}^2\right)}{\left[4\left(\boldsymbol{j}_{\boldsymbol{B1}}^2 - d_{B1}^2\right)e_{B1}^2 - \left(\left(\boldsymbol{j}_{\boldsymbol{B1}}^2 - d_{B1}^2\right) + e_{B1}^2 - g_{B1}^2\right)^2\right]^{1/2}}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.105}$$

The "flattened" version of the pivot angles are used to solve in a manner that

echoes the planar bent-link pantopatches of §3.3.

$$\cos(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}}) = \frac{-\left((\boldsymbol{j}^2_{\boldsymbol{B1}} - d^2_{B1}) + e^2_{B1} - g^2_{B1}\right)}{2e_{B1}g_{B1}} \tag{3.106}$$

$$\sin(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}}) = \frac{\left[4\left(\boldsymbol{j}^2_{\boldsymbol{B1}} - d^2_{B1}\right)e^2_{B1} - \left((\boldsymbol{j}^2_{\boldsymbol{B1}} - d^2_{B1}) + e^2_{B1} - g^2_{B1}\right)^2\right]^{1/2}}{2e_{B1}g_{B1}} \tag{3.107}$$

$$\tag{3.108}$$

therefore,

$$\frac{d(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})} = 1 - \left(\frac{L_{B1}}{e_{B1}}\right)\cos\alpha_{B1} + \cot(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.109}$$

furthermore,

$$\frac{d^2(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^2} = \left(\frac{L_{B1}}{e_{B1}}\right)\frac{\sin\alpha_{B1}}{2e_{B1}g_{B1}\sin^3(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})} \tag{3.110}$$

$$\frac{d^3(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^3} = \frac{-3\cos(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})}{4e^2_{B1}g^2_{B1}\sin^5(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.111}$$

$$\frac{d^4(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^4} = \frac{3 + 12\cos^2(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})}{8e^3_{B1}g^3_{B1}\sin^7(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.112}$$

$$\frac{d^5(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^5} = \frac{-45\cos(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}}) - 60\cos^3(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})}{16e^4_{B1}g^4_{B1}\sin^9(\boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}})}\left(\frac{L_{B1}}{e_{B1}}\right)\sin\alpha_{B1} \tag{3.113}$$

and therefore, to achieve mobility, opposite shifted pivot angles must be equal as with the planar version:

$$\boldsymbol{\theta}^{\star}_{\boldsymbol{A1e}} = \boldsymbol{\theta}^{\star}_{\boldsymbol{A2w}} \tag{3.114}$$

$$\boldsymbol{\theta}^{\star}_{\boldsymbol{A1n}} = \boldsymbol{\theta}^{\star}_{\boldsymbol{B1s}} \tag{3.115}$$

$$\boldsymbol{\theta}^{\star}_{\boldsymbol{A2n}} = \boldsymbol{\theta}^{\star}_{\boldsymbol{B2s}} \tag{3.116}$$

$$\boldsymbol{\theta}^{\star}_{\boldsymbol{B1e}} = \boldsymbol{\theta}^{\star}_{\boldsymbol{B2w}} \tag{3.117}$$

As before, there are also requirements in the bend angle of each link.

$$\hat{\beta}_{A1} - \hat{\alpha}_{A2} + \hat{\alpha}_{B1} - \hat{\beta}_{B2} = 0^{\circ} \tag{3.118}$$

However, the solution for link lengths is different because $\boldsymbol{k}^{\star}_{\boldsymbol{B2/B1}} \neq \boldsymbol{k}^{\star}_{\boldsymbol{B2/B2}}$ if $d_{B1} \neq$

d_{B2}. Therefore, the angle equality must be achieved through alternate methods:

$$\begin{aligned}\boldsymbol{k}_{\boldsymbol{A2}}^2 &= (\boldsymbol{k}_{\boldsymbol{A2/A1}}^{\star})^2 + d_{A1}^2 = g_{A1}^2 + f_{A1}^2 - 2g_{A1}f_{A1}\cos\boldsymbol{\theta}_{\boldsymbol{A1e}}^{\star} + d_{A1}^2 \\ &= (\boldsymbol{k}_{\boldsymbol{A2/A2}}^{\star})^2 + d_{A2}^2 = e_{A2}^2 + h_{A2}^2 - 2e_{A2}h_{A2}\cos\boldsymbol{\theta}_{\boldsymbol{A2w}}^{\star} + d_{A2}^2 \end{aligned} \tag{3.119}$$

To simultaneously solve these equations, we can again substitute $\cos\boldsymbol{\theta}_{\boldsymbol{A2e}}^{\star} = \cos\boldsymbol{\theta}_{\boldsymbol{A2w}}^{\star}$. Therefore, for the "south" part of the pantopatch:

$$e_{A2}h_{A2} = g_{A1}f_{A1} \tag{3.120}$$

$$e_{A2}^2 + h_{A2}^2 + d_{A2}^2 = g_{A1}^2 + f_{A1}^2 + d_{A1}^2 \tag{3.121}$$

The result is not the parallelogram and kite configuration of the planar case, since $\boldsymbol{k}_{\boldsymbol{A2/A1}}^{\star} \neq \boldsymbol{k}_{\boldsymbol{A1/A1}}^{\star}$ if $d_{A1} \neq d_{A2}$. However, there are two solutions to each set of equations so that, in the equations above, e_{A2} and h_{B2} can be switched as with the planar case. The requirements for the rest of the pantograph patch are as follows:

$$e_{B1}g_{B1} = h_{A1}f_{A1} \tag{3.122}$$

$$e_{B1}^2 + g_{B1}^2 + d_{B1}^2 = h_{A1}^2 + f_{A1}^2 + d_{A1}^2 \tag{3.123}$$

$$g_{B2}e_{B2} = f_{A2}h_{A2} \tag{3.124}$$

$$g_{B2}^2 + e_{B2}^2 + d_{B2}^2 = f_{A2}^2 + h_{A2}^2 + d_{A2}^2 \tag{3.125}$$

$$e_{B2}h_{B2} = g_{B1}f_{B1} \tag{3.126}$$

$$e_{B2}^2 + h_{B2}^2 + d_{B2}^2 = g_{B1}^2 + f_{B1}^2 + d_{B1}^2 \tag{3.127}$$

These equations do not easily converge on a solution as simple as the planar case. The assurance of mobility for skew, normal-pivot bent-link pantopatches is made easiest when the offset distance d is equal for all elements in the pantopatch (and, therefore the entire pantomesh). In practice this is also practical — a constant distance may be easier to model and manufacture. However, there may be situations where different ds are needed for flexibility in obtaining desired motion characteristics. For the ultimate flexibility, the designer may choose to use angled-pivot pantopatches.

3.3.3 Mobility of Skew, Angled-Pivot Bent-Link Pantopatches

The mobility of skew, angled-pivot bent-link pantopatches is the most complicated of the six pantopatch types. Not only have eight variables been added to the anal-

ysis (νs and ξs), but also the shortest-distance formulation of the elements must be obeyed (see §2.3.3). The procedure for analyzing the mobility of skew, angled-pivot bent-link pantograph patches is similar to the other bent-link configurations. The first step is to define the additional relevant projected distances:

$$\boldsymbol{k}^2_{\boldsymbol{A2}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + t^2_{B2/A1} - t^2_{A2/A1} \qquad \text{from (2.91)}$$

$$\boldsymbol{k}^2_{\boldsymbol{A2}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A2}} + t^2_{B2/A2} - t^2_{A2/A2} \qquad (3.128)$$

$$\boldsymbol{j}^2_{\boldsymbol{B1}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/A1}} + t^2_{B2/A1} - t^2_{B1/A1} \qquad \text{from (2.90)}$$

$$\boldsymbol{j}^2_{\boldsymbol{B1}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B1}} + t^2_{B2/B1} - t^2_{B1/B1} \qquad (3.129)$$

$$\boldsymbol{k}^2_{\boldsymbol{B2}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{B2/B1}} + t^2_{C2/B1} - t^2_{B2/B1} \qquad (3.130)$$

$$\boldsymbol{k}^2_{\boldsymbol{B2}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{B2/B2}} + t^2_{C2/B2} - t^2_{B2/B2} \qquad (3.131)$$

$$\boldsymbol{j}^2_{\boldsymbol{B2}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{B2/A2}} + t^2_{B3/A2} - t^2_{B2/A2} \qquad (3.132)$$

$$\boldsymbol{j}^2_{\boldsymbol{B2}} = \tilde{\boldsymbol{j}}^2_{\boldsymbol{B2/B2}} + t^2_{B3/B2} - t^2_{B2/B2} \qquad (3.133)$$

Using these terms, the planar bent versions of the motion equations can be used, with appropriate substitution of the projected figures:

$$\begin{aligned}\boldsymbol{k}^2_{\boldsymbol{B2}} = {} & \tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B1}} + t^2_{C2/B1} - t^2_{B2/B1} + \tilde{L}^2_{B1} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B1}} + \tilde{e}^2_{B1} - \tilde{g}^2_{B1}\right)\left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right)\cos\tilde{\alpha}_{B1} \\ & - \left[4\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B1}}\tilde{e}^2_{B1} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B1}} + \tilde{e}^2_{B1} - \tilde{g}^2_{B1}\right)^2\right]^{1/2}\left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right)\sin\tilde{\alpha}_{B1} \qquad (3.134)\end{aligned}$$

$$\begin{aligned}\boldsymbol{k}^2_{\boldsymbol{B2}} = {} & \tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B2}} + t^2_{C2/B2} - t^2_{B3/B2} + \tilde{M}^2_{B2} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B2}} + \tilde{g}^2_{B2} - \tilde{e}^2_{B2}\right)\left(\frac{\tilde{M}_{B2}}{\tilde{g}_{B2}}\right)\cos\tilde{\beta}_{B2} \\ & - \left[4\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B2}}\tilde{g}^2_{B2} - \left(\tilde{\boldsymbol{j}}^2_{\boldsymbol{B1/B2}} + \tilde{g}^2_{B2} - \tilde{e}^2_{B2}\right)^2\right]^{1/2}\left(\frac{\tilde{M}_{B2}}{\tilde{g}_{B2}}\right)\sin\tilde{\beta}_{B2} \qquad (3.135)\end{aligned}$$

$$\begin{aligned}\boldsymbol{j}^2_{\boldsymbol{B1}} = {} & \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + t^2_{B2/A1} - t^2_{B1/A1} + \tilde{M}^2_{A1} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + \tilde{g}^2_{A1} - \tilde{f}^2_{A1}\right)\left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\cos\tilde{\beta}_{A1} \\ & - \left[4\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}}\tilde{g}^2_{A1} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + \tilde{g}^2_{A1} - \tilde{f}^2_{A1}\right)^2\right]^{1/2}\left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\sin\tilde{\beta}_{A1} \qquad (3.136)\end{aligned}$$

$$\begin{aligned}\boldsymbol{j}^2_{\boldsymbol{B2}} = {} & \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A2}} + t^2_{B3/A2} - t^2_{A2/A2} + \tilde{L}^2_{A2} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A2}} + \tilde{e}^2_{A2} - \tilde{h}^2_{A2}\right)\left(\frac{\tilde{L}_{A2}}{\tilde{e}_{A2}}\right)\cos\tilde{\alpha}_{A2} \\ & + \left[4\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A2}}\tilde{e}^2_{A2} - \left(\tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A2}} + \tilde{e}^2_{A2} - \tilde{h}^2_{A2}\right)^2\right]^{1/2}\left(\frac{\tilde{L}_{A2}}{\tilde{e}_{A2}}\right)\sin\tilde{\alpha}_{A2} \qquad (3.137)\end{aligned}$$

The solution is simplified for the derivatives because the constant terms are eliminated. Therefore, the following hold:

$$\frac{d(\boldsymbol{k_{B2}})}{d(\boldsymbol{j_{B1}})} = \frac{d\left(\tilde{\boldsymbol{k}}_{\boldsymbol{B2/B1}}\right)}{d\left(\tilde{\boldsymbol{j}}_{\boldsymbol{B1/B1}}\right)} = 1 - \left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right)\cos\tilde{\alpha}_{B1} - \frac{\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}}}{\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}}}\left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right)\sin\tilde{\alpha}_{B1} \tag{3.138}$$

$$\frac{d(\boldsymbol{k_{B2}})}{d(\boldsymbol{j_{B2}})} = \frac{d\left(\tilde{\boldsymbol{k}}_{\boldsymbol{B2/B2}}\right)}{d\left(\tilde{\boldsymbol{j}}_{\boldsymbol{B2/B2}}\right)} = 1 - \left(\frac{\tilde{M}_{B2}}{\tilde{g}_{B2}}\right)\cos\tilde{\beta}_{B2} - \frac{\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{B2s}}}{\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{B2s}}}\left(\frac{\tilde{M}_{B2}}{\tilde{g}_{B2}}\right)\sin\tilde{\beta}_{B2} \tag{3.139}$$

$$\frac{d(\boldsymbol{j_{B1}})}{d(\boldsymbol{k_{A2}})} = \frac{d\left(\tilde{\boldsymbol{j}}_{\boldsymbol{B1/A1}}\right)}{d\left(\tilde{\boldsymbol{k}}_{\boldsymbol{A2/A1}}\right)} = 1 - \left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\cos\tilde{\beta} + \frac{\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1e}}}{\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1e}}}\left(\frac{\tilde{M}_{A1}}{\tilde{g}_{A1}}\right)\sin\tilde{\beta} \tag{3.140}$$

$$\frac{d(\boldsymbol{j_{B2}})}{d(\boldsymbol{k_{A2}})} = \frac{d\left(\tilde{\boldsymbol{j}}_{\boldsymbol{B2/A2}}\right)}{d\left(\tilde{\boldsymbol{k}}_{\boldsymbol{A2/A2}}\right)} = 1 - \left(\frac{\tilde{L}_{A2}}{\tilde{e}_{A2}}\right)\cos\tilde{\alpha}_{A2} + \frac{\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{A2w}}}{\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{A2w}}}\left(\frac{\tilde{L}_{A2}}{\tilde{e}_{A2}}\right)\sin\tilde{\alpha}_{A2} \tag{3.141}$$

where

$$\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1e}} = \frac{\tilde{\boldsymbol{k}}^{2}_{\boldsymbol{A2/A1}} - \tilde{g}^{2}_{A1} - \tilde{f}^{2}_{A1}}{-2\tilde{g}_{A1}\tilde{f}_{A1}} \tag{3.142}$$

$$\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1e}} = \frac{\left[4\tilde{\boldsymbol{k}}^{2}_{\boldsymbol{A2/A1}}\tilde{e}^{2}_{A2} - \left(\tilde{\boldsymbol{k}}^{2}_{\boldsymbol{A2/A1}} + \tilde{e}^{2}_{A2} - \tilde{h}^{2}_{A2}\right)^{2}\right]^{1/2}}{2\tilde{g}_{A1}\tilde{f}_{A1}} \tag{3.143}$$

$$\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{A2w}} = \frac{\left(\tilde{\boldsymbol{k}}_{\boldsymbol{A2/A2}}\right)^{2} - \tilde{e}^{2}_{A2} - \tilde{h}^{2}_{A2}}{-2\tilde{e}_{A2}\tilde{h}_{A2}} \tag{3.144}$$

$$\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{A2w}} = \frac{\left[4\tilde{\boldsymbol{k}}^{2}_{\boldsymbol{A2/A2}}\tilde{e}^{2}_{A2} - \left(\tilde{\boldsymbol{k}}^{2}_{\boldsymbol{A2/A2}} + \tilde{e}^{2}_{A2} - \tilde{h}^{2}_{A2}\right)^{2}\right]^{1/2}}{8\tilde{e}_{A2}\tilde{h}_{A2}} \tag{3.145}$$

$$\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}} = \frac{\tilde{\boldsymbol{j}}^{2}_{\boldsymbol{B1/B1}} - \tilde{e}^{2}_{B1} - \tilde{g}^{2}_{B1}}{-2\tilde{e}_{B1}\tilde{g}_{B1}} \tag{3.146}$$

$$\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}} = \frac{\left[4\tilde{\boldsymbol{j}}^{2}_{\boldsymbol{B1/B1}}\tilde{e}^{2}_{B1} - (\tilde{\boldsymbol{j}}^{2}_{\boldsymbol{B1/B1}} + \tilde{e}^{2}_{B1} - \tilde{g}^{2}_{B1})^{2}\right]^{1/2}}{8\tilde{e}_{B1}\tilde{g}_{B1}} \tag{3.147}$$

$$\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{B2s}} = \frac{\tilde{\boldsymbol{j}}^{2}_{\boldsymbol{B2/B2}} - \tilde{g}^{2}_{B2} - \tilde{e}^{2}_{B2}}{-2\tilde{g}_{B2}\tilde{e}_{B2}} \tag{3.148}$$

$$\sin\tilde{\boldsymbol{\theta}}_{\boldsymbol{B2s}} = \frac{\left[4\tilde{\boldsymbol{j}}^{2}_{\boldsymbol{B2/B2}}\tilde{g}^{2}_{B2} - (\tilde{\boldsymbol{j}}^{2}_{\boldsymbol{B2/B2}} + \tilde{g}^{2}_{B2} - \tilde{e}^{2}_{B2})^{2}\right]^{1/2}}{8\tilde{g}_{B2}\tilde{e}_{B2}} \tag{3.149}$$

The higher-order derivatives of motion are similar to those of the planar and normal-

pivot skew pantopatches, only the projected angles are presented. Therefore:

$$\frac{d^2(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^2} = \left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right) \frac{\sin\tilde{\alpha}_{B1}}{2\tilde{e}_{B1}\tilde{g}_{B1}\sin^3(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}})} \tag{3.150}$$

$$\frac{d^3(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^3} = \frac{-3\cos(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}})}{4\tilde{e}^2_{B1}\tilde{g}^2_{B1}\sin^5(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}})} \left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right) \sin\tilde{\alpha}_{B1} \tag{3.151}$$

$$\frac{d^4(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^4} = \frac{3+12\cos^2(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}})}{8\tilde{e}^3_{B1}\tilde{g}^3_{B1}\sin^7(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}})} \left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right) \sin\tilde{\alpha}_{B1} \tag{3.152}$$

$$\frac{d^5(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})^5} = \frac{-45\cos(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}}) - 60\cos^3(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}})}{16\tilde{e}^4_{B1}\tilde{g}^4_{B1}\sin^9(\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}})} \left(\frac{\tilde{L}_{B1}}{\tilde{e}_{B1}}\right) \sin\tilde{\alpha}_{B1} \tag{3.153}$$

What follows is that the opposite angles of the projected links must always be equal to achieve mobility:

$$\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1e}} = \tilde{\boldsymbol{\theta}}_{\boldsymbol{A2w}} \tag{3.154}$$

$$\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1n}} = \tilde{\boldsymbol{\theta}}_{\boldsymbol{B1s}} \tag{3.155}$$

$$\tilde{\boldsymbol{\theta}}_{\boldsymbol{A2n}} = \tilde{\boldsymbol{\theta}}_{\boldsymbol{B2s}} \tag{3.156}$$

$$\tilde{\boldsymbol{\theta}}_{\boldsymbol{B1e}} = \tilde{\boldsymbol{\theta}}_{\boldsymbol{B2w}} \tag{3.157}$$

Furthermore, this requires the projected bent-link angle at each pivot to equal the following:

$$\tilde{\hat{\beta}}_{A1} - \tilde{\hat{\alpha}}_{A2} + \tilde{\hat{\alpha}}_{B1} - \tilde{\hat{\beta}}_{B2} = 0^\circ \tag{3.158}$$

So as with the normal-pivot solution, the result is not the parallelogram and kite configuration of the planar case. For example, for $\boldsymbol{k}^2_{\boldsymbol{A2}}$:

$$\boldsymbol{k}^2_{\boldsymbol{A2}} = \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A1}} + t^2_{B2/A1} - t^2_{A2/A1} \quad = g^2_{A1} + f^2_{A1} - 2g_{A1}f_{A1}\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1e}} + t^2_{B2/A1} - t^2_{A2/A1} \tag{3.159}$$

$$= \tilde{\boldsymbol{k}}^2_{\boldsymbol{A2/A2}} + t^2_{B2/A2} - t^2_{A2/A2} \quad = e^2_{A2} + h^2_{A2} - 2e_{A2}h_{A2}\cos\tilde{\boldsymbol{\theta}}_{\boldsymbol{A2w}} + t^2_{B2/A2} - t^2_{A2/A2} \tag{3.160}$$

$$\tag{3.161}$$

Therefore, after substituting $\tilde{\boldsymbol{\theta}}_{\boldsymbol{A1e}} = \tilde{\boldsymbol{\theta}}_{\boldsymbol{A2w}}$ as in the other bent-link cases, we get the

following:

$$e_{A2}h_{A2} = g_{A1}f_{A1} \tag{3.162}$$

$$e_{A2}^2 + h_{A2}^2 + t_{B2/A2}^2 - t_{A2/A2}^2 = g_{A1}^2 + f_{A1}^2 + t_{B2/A1}^2 - t_{A2/A1}^2 \tag{3.163}$$

$$e_{B1}g_{B1} = h_{A1}f_{A1} \tag{3.164}$$

$$e_{B1}^2 + g_{B1}^2 + t_{B2/B1}^2 - t_{B1/B1}^2 = h_{A1}^2 + f_{A1}^2 + t_{B2/A1}^2 - t_{B1/A1}^2 \tag{3.165}$$

$$g_{B2}e_{B2} = f_{A2}h_{A2} \tag{3.166}$$

$$g_{B2}^2 + e_{B2}^2 + t_{B3/B2}^2 - t_{B2/B2}^2 = f_{A2}^2 + h_{A2}^2 + t_{B3/A2}^2 - t_{B2/A2}^2 \tag{3.167}$$

$$e_{B2}h_{B2} = g_{B1}f_{B1} \tag{3.168}$$

$$e_{B2}^2 + g_{B2}^2 + t_{C2/B2}^2 - t_{B2/B2}^2 = g_{B1}^2 + e_{B1}^2 + t_{C2/B1}^2 - t_{B2/B1}^2 \tag{3.169}$$

Much like the normal-pivot case, there are no simple parallelogram / kite rules to ease implementation of those rules. However, that does add flexibility to the design of the assembled pantomesh.

In summary, the requirements for mobility of a skew, angled-pivot bent-link pantopatch follow a similar, if more complicated, path as the other bent-link pantopatches.

3.4 Graphical Mobility Determination

The equation-based mobility calculations of the previous sections have value in computationally synthesizing and analyzing pantomeshes. However, there are also benefits to graphical construction of many mechanisms [26], and pantomeshes are no exception. Instead of paper-based 2D graphical constructions, 3D graphical constructions usually benefit from computer-aided solid modeling packages.

Fortunately, the mobility equations derived for straight-link pantopatches may be used for graphical construction. An illustration of the planar, straight-link case is shown in Figure 3.2, with $\boldsymbol{\mathcal{P}}_{B1}$, $\boldsymbol{\mathcal{P}}_{B2}$, and $\boldsymbol{\mathcal{P}}_{B3}$ shown in front of a plane while $\boldsymbol{\mathcal{P}}_{A2}$ and $\boldsymbol{\mathcal{P}}_{C2}$ are behind it. As a case of Carnot's polygon theorem, when a skew quadrilateral $ABCD$ is cut by a plane, the intersection points $PQRS$ occur such that the ratios are related as follows [27–29]:

$$\frac{|PB|}{|AP|}\frac{|QC|}{|BQ|} = \frac{|CR|}{|RD|}\frac{|DS|}{|SA|} \tag{3.170}$$

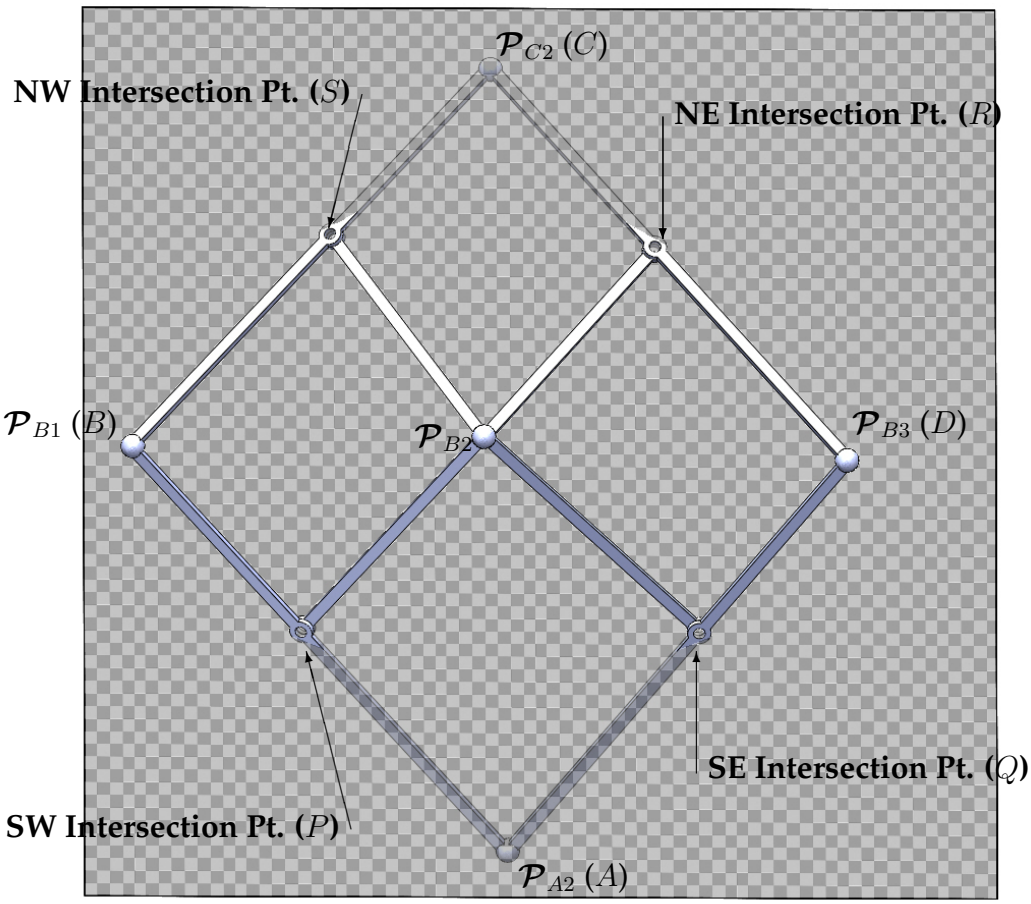


Figure 3.2: Illustration of Carnot's polygon theorem. A single plane (translucent checkered) cuts through all four intersection points.

This result is identical to the the planar straight-link case in Eqn. (3.12), as well as satisfying the basic geometric constraints of the all pantograph patches. In the bent-link cases, these ratios are measured at the intersection of the pantograph element diagonals rather than the actual location of the pivots.

For skew pantograph patches, the diagonal point along each link is measured where the intersecting link diagonal plane (i.e. the plane containing the intersecting diagonal and parallel to the pivot axis).

3.5 Discussion

The requirements for pantomesh mobility presented in this chapter, despite being new derivations, are related to earlier discoveries in this field. For instance, the 4P (all parallelogram) configuration of planar bent-link pantopatches is the basis of

the planar pantomesh made of Generalized Angular Elements (GAE) as described in [8]. Later, the same approach is mentioned in [9] as a possible solution to ensure mobility. Neither of these examples investigate the kite-based solutions, either partially or totally.

3.6 Conclusions

This chapter detailed the geometric requirements for pantopatches, and therefore pantomeshes, to be mobile. In addition to the detailed derivations presented for all six pantopatch types, a short description of a geometric construction procedure was presented. Another approach to mobility, also based on graphical techniques, is presented in the following chapter.

Chapter 4

A Mobility Formula for Connected Polyhedra[1]

In order to synthesize a new mechanism or analyze a given system, it is necessary to determine the degrees of freedom of the system. At the outset of designing a mechanism or robot for a given task, the number of degrees of freedom of the desired device is specified. The designer wishes to explore various kinematic topologies to achieve the desired functionality. In this "type synthesis" phase, it is convenient to employ a simple formula, based on the number of links and the joint types connecting them, to predict the mobility of various linkage topologies under consideration. The mobility is equal to the actual degrees of freedom when the link geometries are within the assumptions of the accounting method. However, the degrees of freedom can exceed this calculated mobility when the links meet special conditions. The purpose of this chapter is to introduce a new mobility formulation that, compared to existing approaches, more easily captures certain special configurations of links. This formulation thereby provides a valid mobility count for more classes of mechanisms.

4.1 Mobility of Connected Polyhedra

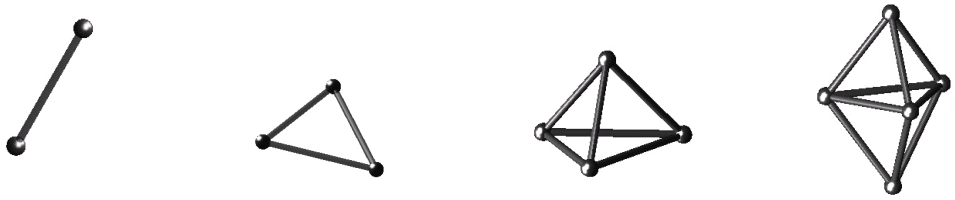
For any mechanism built with lower-order pair joints, excluding helical joints, the relevant features of each link are most often points, lines, or planes. A link may be naturally modeled as a polyhedron whose vertices, edges, and faces include

[1]Portions of this chapter were originally presented at the 2007 ASME Mechanisms and Robotics Conference [30]

the salient points, lines, and planes. A joint then constrains certain features of one polyhedron to lie on, or be equal to, certain features of another polyhedron. This leads to a mobility formula based on a simple count of vertices and edges. The simplest form accounts for spherical sub-mechanisms by insisting that one of the vertices be placed at the spherical center. A modified formula relaxes this condition by introducing an adjustment for remote spherical centers, which are exhibited in pantomeshes. This modification is duplicated in a similar adjustment that arises when links represented by polyhedra are replaced by planar sub-mechanisms having special properties.

4.1.1 Rigid Bodies

The defining characteristic of a rigid body is that the distance between any two points of the body is always preserved. If freely independent, each point would have three degrees of freedom, its (x, y, z) coordinates. When the distance between two points is constrained, there is one fewer degree of freedom. If each point is considered a *vertex* and each fixed distance an *edge*, the degrees of freedom of a free-floating rigid body with V vertices and E edges is always $3V - E$. The rigid links for $V = 2, 3, 4, 5$ are illustrated in Fig. 4.1 and described below.



(a) $V = 2$ (b) $V = 3$ (c) $V = 4$ (d) $V = 5$

Figure 4.1: Edge-vertex representation of rigid bodies.

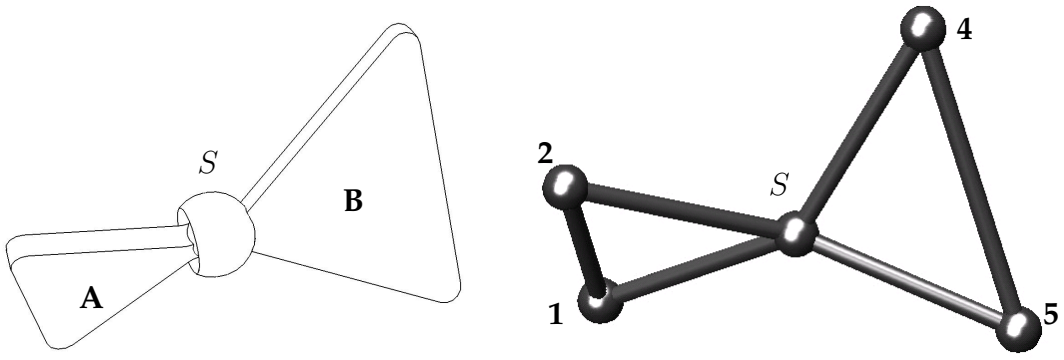
The simplest rigid body of interest consists of just two feature points connected by an edge of constant length (see Fig. 4.1a). If free to float in three-dimensional-space, such a body has five degrees of freedom, since each point has three coordinates and the edge between them imposes one distance constraint: $2 * 3 - 1 = 5$. In this manner, we *do not count* the rotation around the edge through the points that would be evident in a real link, which must have some thickness. An edge, in this polyhedral notation, is merely the fixed distance between points or vertices. If we

wish to include rotation about an edge in our count, an off-axis feature point must be introduced.

For three feature points, we have a triangle, whose three vertices and three edges result in six degrees of freedom: $3 * 3 - 3 = 6$ (see Fig. 4.1b). Each feature point beyond the third should be introduced with three new edges joining it to other points of the polyhedron, thus leaving it with six degrees of freedom. Hence a polyhedron with four feature points is modeled as a tetrahedron, having six edges (see Fig. 4.1c); a polyhedron with 5 feature points must have 9 edges (see Fig. 4.1d); and so on.

4.1.2 Spherical and Rotational Joints

A spherical joint, also known as a ball-and-socket joint, constrains the centers of the respective spherical surfaces of two links to be equal. In the representation of links as polyhedra, this simply means that two bodies share a common vertex. This is illustrated in Fig. 4.2. In Fig. 4.2a, two triangular-shaped links, labeled "**A**" and "**B**", are joined by a spherical joint at vertex S. Figure 4.2b represents the polyhedral model, two triangles that share vertex S. A minimum of three vertices are required to represent each of these links; otherwise, one of the three rotational DOF of the spherical joint would be indeterminate. That is, if point **2** did not exist, then rotation about the **1**-S cannot be determined because edges have only one dimension.



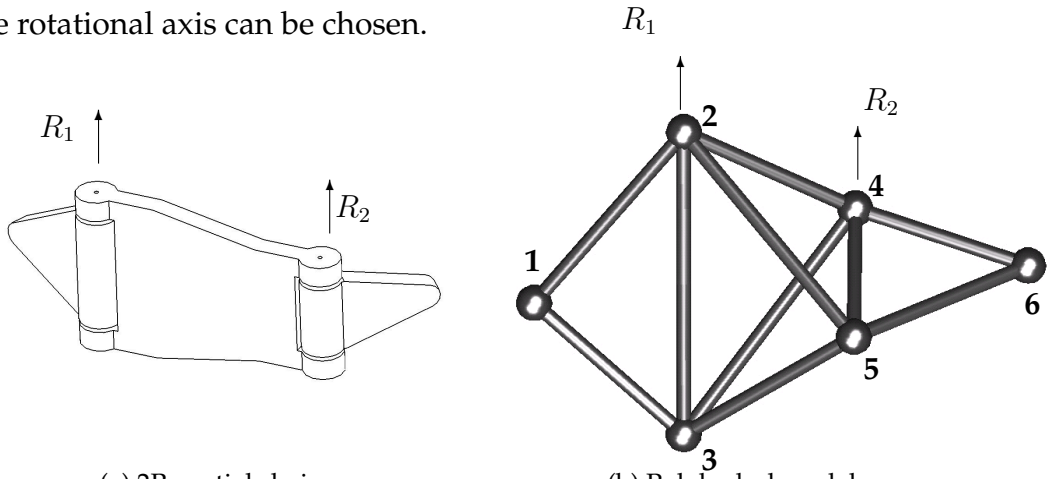
(a) Spherical joint, two links

(b) Polyhedral model ($V = 5, E = 6, \mathcal{M} = 3$)

Figure 4.2: Spherical joint between two triangular links

A rotational joint is most simply represented in the polyhedral model as a pair of

distinct points shared by two polyhedra. The rotation axis of the joint is the edge through the vertices. The location of the vertices is not unique, as any two points on the rotational axis can be chosen.



(a) 2R spatial chain

(b) Polyhedral model ($V = 6, E = 10, \mathcal{M} = 2$)

Figure 4.3: Spatial 2R chain

Figures 4.3 shows an example of an open $2R$ chain. Figure 4.3a shows three links joined at two revolute joints; the general spatial case where joints R_1 and R_2 are skew. Figure 4.3b shows a polyhedral model of that configuration, and because the rotation axes are skew the middle link is a tetrahedron. The edge **2-3** represents the R_1 axis, and edge **4-5** represents the R_2 axis.

A spherical sub-mechanism occurs when several rotational joints meet in a common point. In Fig. 4.4, joints R_1 and R_2 intersect at vertex **1**, and the middle link

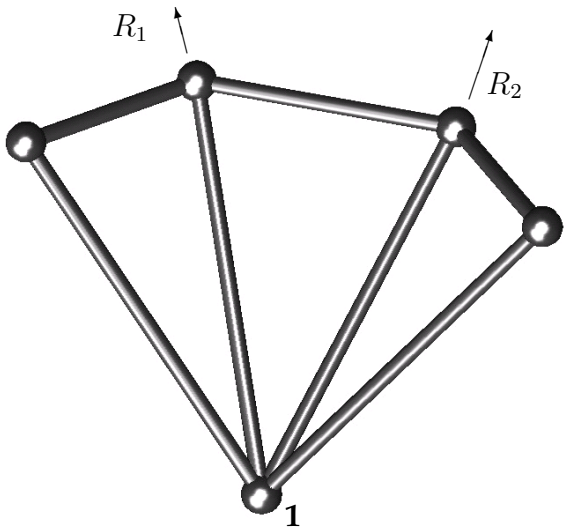
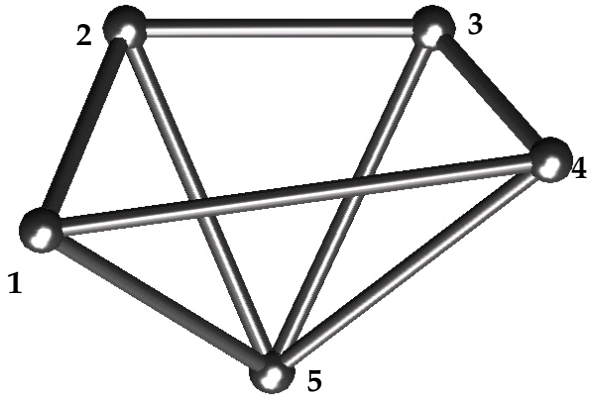


Figure 4.4: Spherical 2R chain

becomes a triangle. To properly account for this, we simply select the common point as one of the two points representing each of the intersecting axes. For parallel joints, the common point is at infinity. For the purpose of mobility counting, it is convenient to draw such points as if they are finite, although other notations could be devised. Figure 4.5 shows a closed-loop 4R spherical four-bar, where four triangles all share a single vertex, **5**, with each other as well as each edge with a neighbor. Note that the four-sided loop formed by vertices **1-2-3-4-1** is *not* a rigid body. A solid tetrahedron would require two additional edges that would join opposing vertices **1-3** and **2-4**, as in Fig. 4.1c, thereby reducing the mobility by 1.



$V = 5, E = 8, \mathcal{M} = 1$

Figure 4.5: Spherical four-bar polyhedral model

4.1.3 A Mobility Formula

Using the ideas of the previous subsections, one can model any mechanism having only spherical and rotational links as a collection of vertices and edges. Accordingly, the mobility $\mathcal{M}$ of the mechanism is given as

$$\mathcal{M} = 3V - E - 6 \tag{4.1}$$

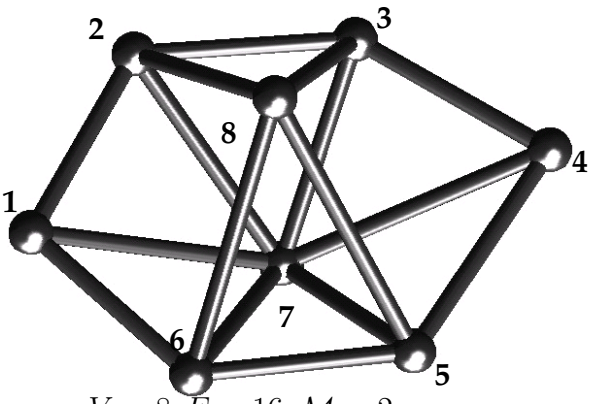
This equation is in contrast with the common Grübler-Kutzbach formula Eqn. (3.1) and the special case for spherical mechanisms Eqn. (4.2).

$$\mathcal{M} = 3(N-1) - 2f_1, \quad \text{(spherical or planar)} \tag{4.2}$$

As in the first relations of Eqns. (3.1) and (4.2), we subtract 6 to fix one link and count only internal degrees of freedom. Using this formula for the mechanisms of

Figs. 4.2, 4.3, and 4.5 gives the correct mobility in all cases. Unlike Eqns. (3.1) and (4.2), Eqn. (4.1) holds for both spatial and spherical mechanisms.

Furthermore, Eqn. (4.1) holds for mechanisms where Eqn. (3.1) does not hold and Eqn. (4.2) does not apply. Such mechanisms mix spherical sub-mechanisms with spherical joints. An example is shown in Fig. 4.6, having six triangles in a spherical 6R loop defined by vertices **1** through **7**. There is also a coupling between two opposing triangle faces **2-3-7** and **5-6-7** via two triangles **2-3-8** and **5-6-8** joined at spherical joint **8**. The two loops in this mechanism, **1-2-8-6-1** and **3-4-5-8-3**, have 2 degrees of freedom for which Eqn. (4.1) gives the correct result, $\mathcal{M} = 2$. In contrast, Eqn. (3.1) gives $\mathcal{M} = -1$.



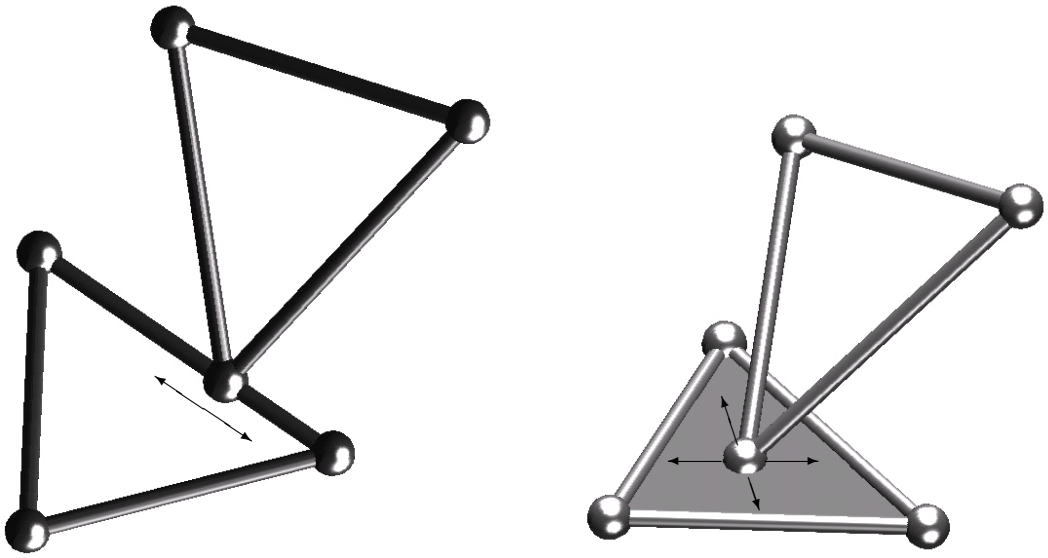
$V = 8, E = 16, \mathcal{M} = 2$

Figure 4.6: Spatial eight-bar with mobility of 2

4.1.4 Joints with Translational Motion

So far spherical and rotational joints have been discussed; both only involve rotations. Among the lower-order pairs, prismatic, cylindrical, and planar joints all involve translation, independent of rotation. These can easily be incorporated into the polyhedral model. Helical joints, which have rotation and translation coupled by the pitch of the screw, are not as easily accommodated, and will not be examined here.

Polyhedra naturally contain linear elements: edges are line segments and faces are planar pieces. More generally, we may consider any two vertices of a polyhedron to define the line through them and similarly any three vertices define a plane. Accordingly, we may constrain a vertex of one polyhedron to lie on a line or plane of another polyhedron. Figure 4.7 illustrates each of these basic linear constraints. A point-to-line constraint costs two degrees of freedom (the point retains just one



(a) Point-to-line joint (b) Point-to-plane joint

Figure 4.7: Basic translational joints

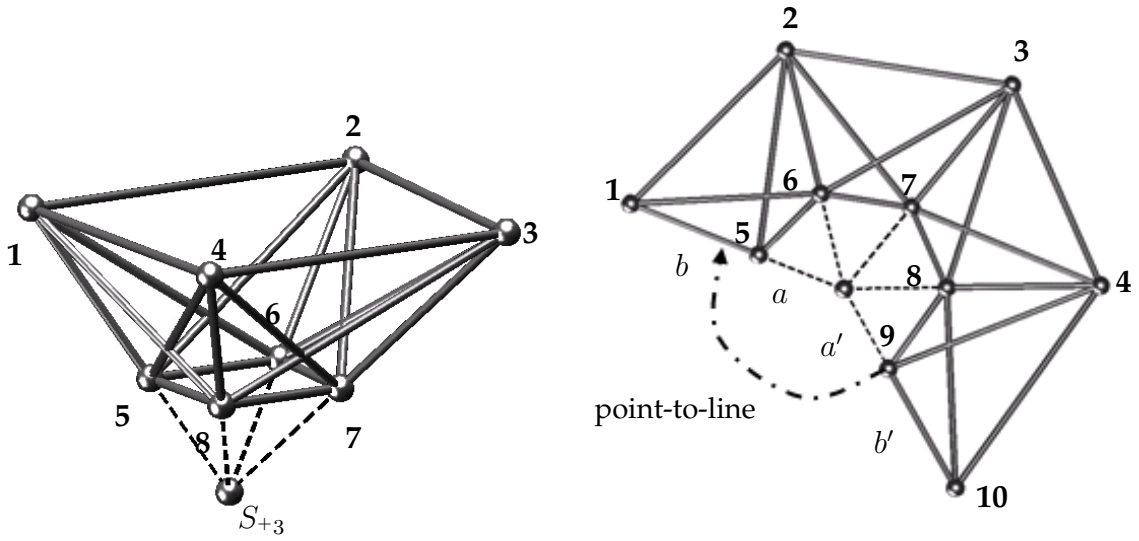
degree of freedom of motion along the line) whereas a point-to-plane constraint costs just one degree of freedom. Consequently, letting L be the number of vertex-to-line constraints and P be the number of point-to-plane constraints, the mobility formula becomes

$$\mathcal{M} = 3V - E - P - 2L - 6 \tag{4.3}$$

One can combine several basic linear constraints to construct kinematic equivalents of the lower-order pairs. Constraining two vertices of polyhedron A to lie on the same line of polyhedron B is equivalent to a cylindrical joint between A and B. Constraining three points of one to lie on a common plane of the other is equivalent to plane-plane contact between the polyhedra. Finally, a prismatic link may be modeled by constraining two vertices of A to a line of B and constraining a third point of A to a plane of B that contains the line. In this way, we generalize the polyhedral model to include rotational, prismatic, cylindrical, planar, and spherical joints.

4.1.5 Adjustment for Remote Spherical Centers

The correctness of Eqns. (4.1) and (4.3) in the presence of spherical sub-mechanisms depends on the choice of the sphere center as a vertex for each of the rotational



(a) Closed-loop spherical four-bar (b) Open-loop spherical four-bar

Figure 4.8: Spherical loop with remote center

joints around the loop. Although vertices may be placed at any point along the rotation axis, at times it may be convenient not collocate edge vertices to illustrate a spherical sub-mechanism as shown in Figs. 4.4 and 4.5. For example, the spherical four-bar of four triangles in Fig. 4.5 may be represented by the four tetrahedra in Fig. 4.8a. Indeed, this is the only way a spherical sub-mechanism can be specified using just vertices and edges without additional notation. However, suppose we begin with a spherical loop and mark two arbitrary points on each joint axis distinct from the spherical center. Without somehow noting that the joints intersect, we will obtain a lower mobility consistent with a spatial mechanism in which the joints do not intersect. Suppose then, that we simply note that the axes intersect in a remote center, as in Fig. 4.8a. To compute mobility correctly, we can start with an open-loop spherical four-bar as shown in Fig. 4.8b. To close this loop around the spherical center, it is merely sufficient to use a single point-to-line constraint, i.e. vertex **9** onto edge **1-5**. The *cost* of this constraint, in terms of mobility, is -2. This can be expressed as $\Delta M_{true} = -2$. But when the distances satisfy $a = a', b = b'$, four vertices are reduced to two and two edges combine to one (by eliminating vertices **9** and **10**) . The apparent cost to close the loop is two vertices and one edge, i.e., $\Delta M_{apparent} = -5$. The common spherical center therefore adds $\Delta M_{true} - \Delta M_{apparent} = 3$ freedoms to the system, which must be included in the formulation. To keep track of this, we mark the remote center as type S_{+3}, and letting S_{+3} denote the total number of such centers, the adjusted mobility formula

becomes

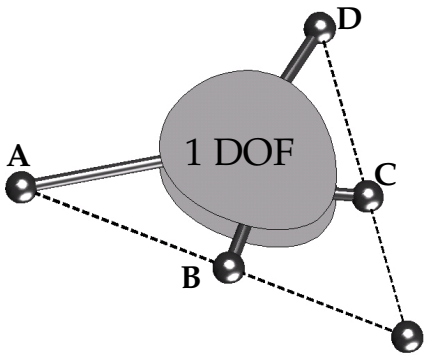
$$\mathcal{M} = 3V + 3S_{+3} - E - P - 2L - 6 \tag{4.4}$$

We could, of course, do without remote centers and instead force a common vertex as in Fig. 4.5. However, the S_{+3} terms are included as a convenience and in anticipation of additional adjustment factors needed for other types of mechanisms discussed below.

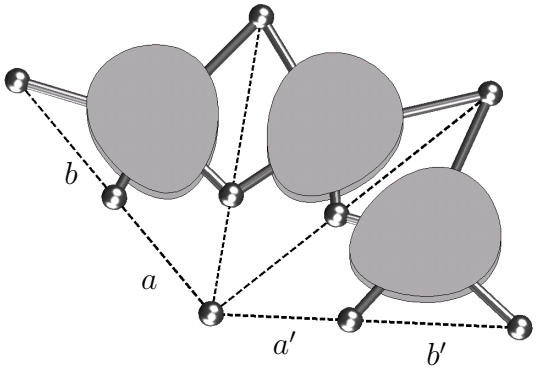
4.2 Mobility of Connected Sub-Mechanisms

In the previous section, we considered mechanisms formed from polyhedral links connected by joints between points, lines, and planes. Now, suppose we join several such mechanisms together. The mobility for a compound mechanism formed from m sub-mechanisms is the sum of the mobilities for the sub-mechanisms minus the number of constraints imposed to connect them plus $6m-6$, where this last factor recognizes that rigid-body motion of the sub-mechanisms now contributes to the mobility of the whole. This is no different than just counting up the mobility of the whole mechanism using Eqn. (4.4) unless some kind of extra relations exist between the sub-mechanisms that would require an adjustment similar to the remote center adjustment discussed previously. In this section, we consider one such class of mechanism that has a particular implementation using planar scissors sub-mechanisms.

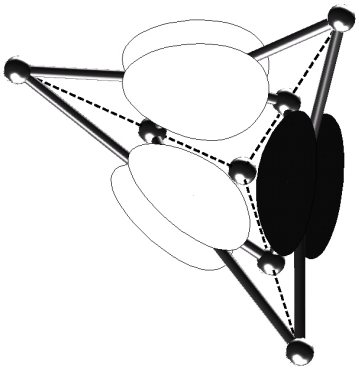
As an example of a sub-mechanism, consider a one-degree-of-freedom mechanism that has four output vertices, illustrated schematically in Fig. 4.9a. Assume that the motion of the mechanism is such that the four vertices are always coplanar. Therefore the lines through any two pairs of vertices, say line AB and line CD, intersect. Further, assume there are several such mechanisms connected sequentially as in Fig. 4.9b whose motion characteristics are such that all the indicated lines intersect in a common point. Then, we know to check the possibility that closing a loop around this common point to form the mechanism of Fig. 4.9c might, as in the case of remote spherical centers, produce a disparity between the true number of constraints required to close the loop and the apparent number implied by counting vertices. Indeed, if the whole chain of sub-mechanisms is constructed such that the output distances equal the input distances, i.e., if $a = a', b = b'$, then the true cost of closing the loop is $\Delta M_{true} - 2$, whereas the apparent cost is two vertices, i.e., $\Delta M_{apparent} = -6$. Therefore, a correct mobility count requires an adjustment of +4.



(a) Planar submechanism



(b) Open chain of three planar submechanisms

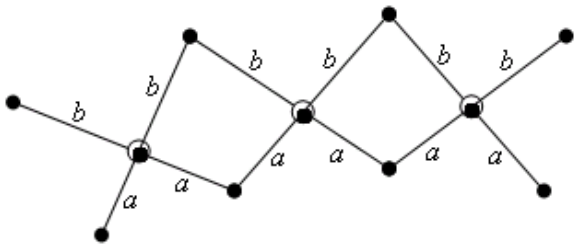


(c) Closed chain of three planar submechanisms

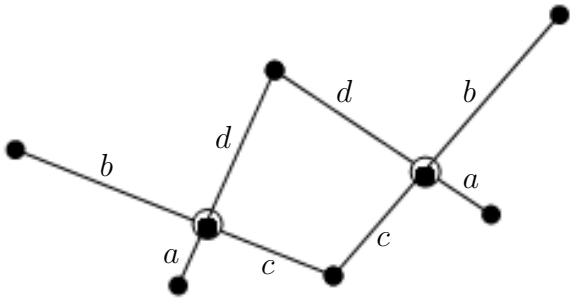
Figure 4.9: Submechanisms having a remote center

Accordingly, we mark the remote center in this case as type S_{+4}.

Single degree-of-freedom sub-mechanisms such as scissor-pairs can arise in many ways. One is that each of the sub-mechanisms in the series is an identical symmetric scissors linkage (see Fig. 4.10a). If the number of sub-mechanisms is even, the conditions are met by identical asymmetrical scissors attached in mirror-image



(a) Chain of three symmetric scissors



(b) Flipped pair of symmetric scissors

Figure 4.10: Scissor submechanisms

pairs (see Fig. 4.10b).

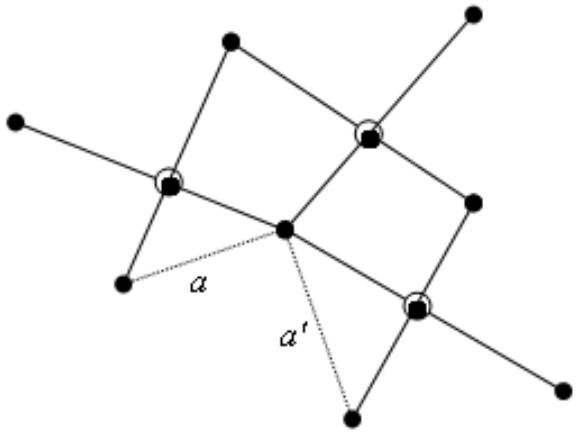


Figure 4.11: Three scissor linkages with a common center

A related type of closure is illustrated in Fig. 4.11, where a series of sub-mechanisms are connected all sharing a vertex at a common center. For this configuration, if the design of the sub-mechanisms is such that $a = a'$, the true cost of closing the loop is again $\Delta M_{true} = -2$, whereas the apparent cost is one vertex, i.e., $\Delta M_{apparent} = -3$, hence the closure requires an adjustment of +1. This may be indicated on a diagram of the mechanism as a vertex of type S_{+1}. Similar to above, the condition for a S_{+1} vertex is met by an even number of scissors linkages connected in mirror-

image pairs.

We may account for all such special arrangements by letting $S^{+}3$ denote the total number of vertices of that type. Accordingly, the mobility formula becomes

$$\mathcal{M} = 3V + \sum_{k} kS_{+k} - E - P - 2L - 6 \tag{4.5}$$

This count includes the internal vertices, edges, etc., of the sub-mechanisms. One can summarize the internal constraints of each sub-mechanism for the purpose of counting the mobility of the whole. For example, a scissors mechanism has four exposed vertices which contribute 12 DOF to the mobility, but its true contribution is its internal mobility of 1 plus 6 degrees of freedom of rigid-body motion, i.e., 7 total. Accordingly, the internal constraints must add up to -5. So, if we count X scissors sub-mechanisms and their exposed vertices, ignoring their internal vertices and edges, the mobility formula becomes

$$\mathcal{M} = 3V + \sum_{k} kS_{+k} - E - P - 2L - 5X - 6 \tag{4.6}$$

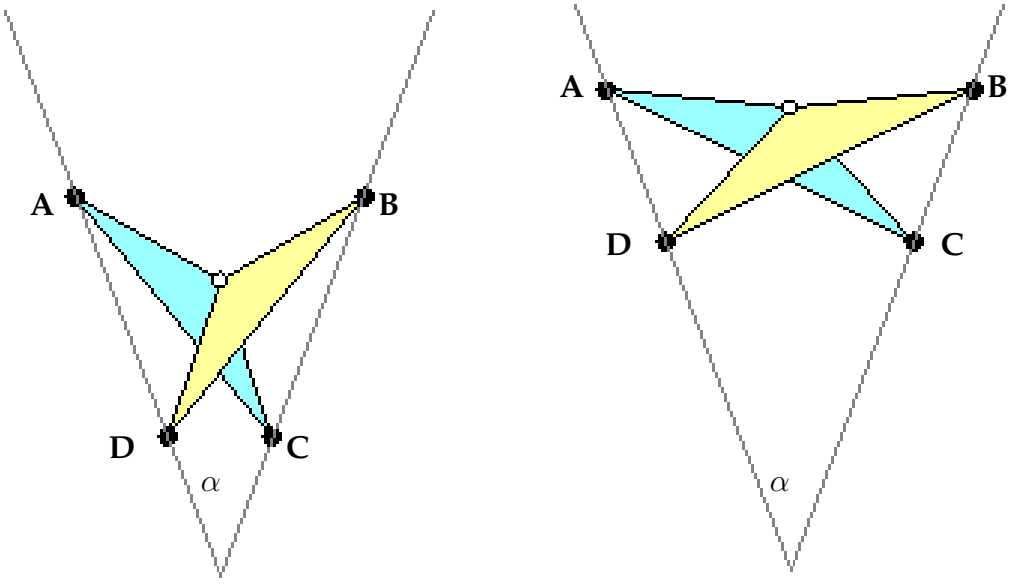
4.3 Examples: Collapsible Polyhedra

(a) Collapsed shape (b) Expanded shape

Figure 4.12: .
Hoberman Mini Sphere®. [14]

The Hoberman Sphere is a children's toy that performs a dramatic radial expansion

from a spiny collapsed shape to a large spherical shape with a single degree of freedom (see Fig. 4.12). The key element of the Hoberman Sphere is a scissor pair designed such that the angle α defined by its exterior points **A**, **B**, **C**, and **D** remains the same as shown in Fig. 4.13. Such an element may take several forms [13], and may even consist of multiple scissor pairs joined in series. We will restrict ourselves to single scissor pairs, knowing that any planar, single DOF system that maintains the same angle α could be used. The Hoberman-style elements may



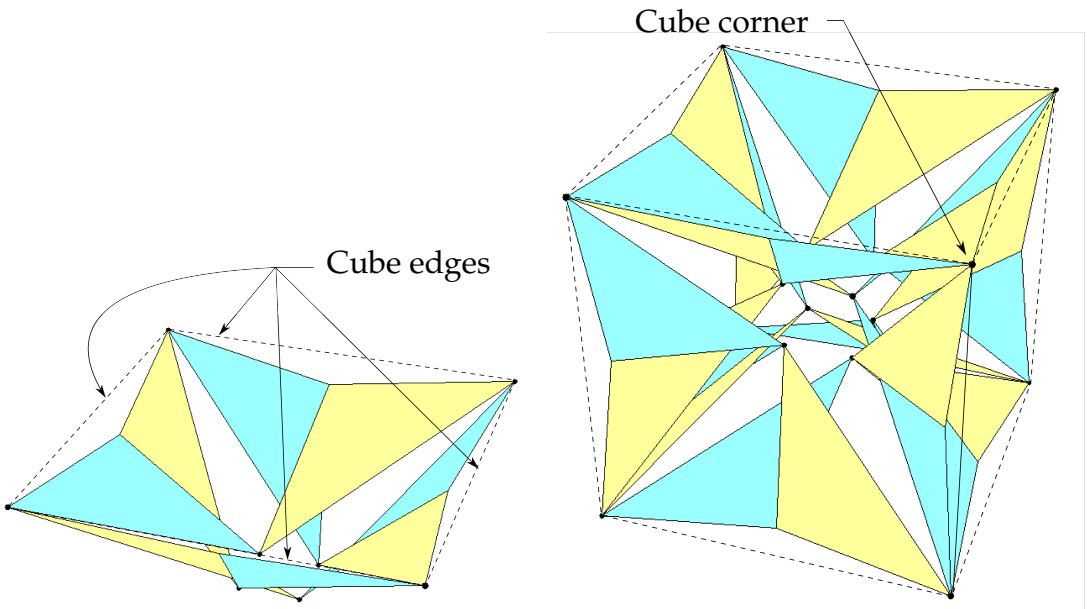
(a) Collapsed element (b) Expanded element

Figure 4.13: Hoberman-style scissor element

be connected by spherical joints as shown in Fig. 4.10 to form various collapsible shapes. We will show that certain arrangements of these elements, connected by spherical joints, have a single degree of freedom as correctly determined by the polyhedral-vertex-edge method.

A cube is a regular polyhedron with 8 vertices, 12 edges, and 6 faces. To convert this into a Hoberman-style collapsible object, we replace each edge with a single DOF sub-mechanism, such as a scissor pair or chain of scissor pairs, that collapse to the same point. Figure 4.14a shows how a four-edged cube face can be replaced by four scissor pairs. The geometry of the scissor pair limits the range of motion of the mechanism; we can replace each single scissor pair with multiply-connected scissor pairs joined by revolute joints as long as the external vertices remain copla-

nar and maintain the same angle.



(a) Four cube edges replaced by scissor pairs (b) Assembled collapsible cube

Figure 4.14: Collapsible cube

Just as three edges meet at each cube vertex, three sub-mechanisms are joined at a "corner" of a collapsible cube (see Fig. 4.14b). The difference, however, is that the sub-mechanisms meet at two vertices for each corner. Each cube face translates to a closed loop of sub-mechanisms. In order for each sub-mechanism to collapse, the pseudo-edges for each scissor pair must meet at the same point. In order to have all 6 loops collapse to the same point, we must employ the equal-angle conditions detailed in 3. Without those conditions, the mechanism would not assemble.

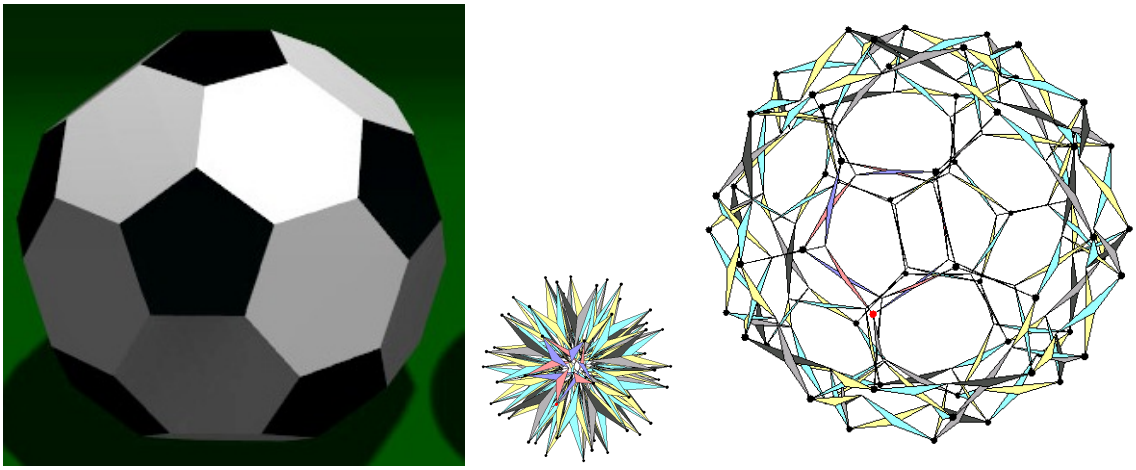
The mobility of the collapsible cube is calculated using Eqn. (4.6) as follows: There are 16 vertices (an "outer" and "inner" vertex for each cube vertex) so $V = 16$. Each polyhedral edge is replaced by a single DOF scissor pair, therefore $X = 12$. Furthermore, there are six loops that each share a common collapse point. However, there is also an additional "free" DOF obtained when closing the final loop during assembly, so $S_{+4} = 1$ and therefore $S_{+3} = (6 - 1)$. Therefore, the mobility is calculated as shown in Eqn. (4.7) below:

$$\mathcal{M}_t extitcube = 3(16) - 5(12) + 3(6 - 1) + 4(1) - 6 = 1 \tag{4.7}$$

Similar results are obtained for other collapsible shapes. A tetrahedron has 4 vertices, 6 edges, and 4 faces (loops). Therefore, the mobility of a collapsible tetrahedron is:

$$\mathcal{M}_{tetrahedron} = 3(2(4)) - 5(6) + 3(4-1) + 4(1) - 6 = 1 \tag{4.8}$$

A more complicated example is that of a collapsible truncated icosahedron, similar to a soccer ball (football) or C_{60} Buckminster Fullerene (see Fig. 4.15). Such shapes have twelve pentagonal faces and twenty hexagonal faces.



(a) Truncated icosahedron (b) Collapsed (c) Expanded

Figure 4.15: Collapsible truncated icosahedron

The Hoberman sphere comes in two main shapes: the larger spheres are shaped like a icosidodecahedron, and the mini sphere is shaped like a deltoidal icositetrahedron. Both of these polyhedra differ from the previous one mentioned — some or all vertices have *four* edges rather than three, which require an additional DOF when closing the loop. This changes the number of S_{+4}'s to equal the number of 4-edge vertices plus one. The number of S_{+3}'s is equal to the number of faces less the number of S_{+4}'s. The icosidodecahedron has 32 faces, 60 edges, and 30 vertices, all of which have four edges. The deltoidal icositetrahedron has 24 faces, 48 edges, and 26 vertices, of which 8 are 3-edge and 18 are 4-edge. The mobility calculations for both collapsible shapes are shown in Eqns. (4.9) and (4.10).

$$\mathcal{M}_{icosidodecahedron} = 3(2(30)) - 5(60) + 3(1) + 4(30+1) - 6 = 1 \tag{4.9}$$

$$\mathcal{M}_{deltoidal\ icositetrahedron} = 3(2(26)) - 5(48) + 3(5) + 4(18+1) - 6 = 1 \tag{4.10}$$

In general, the number of S_{+k}'s is related to the number edges that meet at each vertex. This can be verified with the tetrakis hexahedron and pentakis dodecahedron. The tetrakis hexahedron has 24 faces, 36 edges, and 14 vertices, of which 6 are 4-edge and 8 are 6-edge. The pentakis dodecahedron has 60 faces, 90 edges, and 32 vertices, of which 12 are 5-edge and 20 are 6-edge. As with the earlier examples, the number of S_{+4}'s equals the number of 4-edge vertices plus one. Each 5-edge vertex counts as a S_{+5}, and each 6-edge vertex counts as a S_{+6}. All remaining faces count as S_{+3}'s. Therefore, the mobility calculations for both shapes are:

$$\mathcal{M}_{\textit{tetrakis hexahedron}} = 3(2(14)) - 5(36) - 3(9) + 4(6+1) - 6(8) - 6 = 1 \tag{4.11}$$

$$\mathcal{M}_{\textit{pentakis dodecahedron}} = 3(2(32)) - 5(90) - 3(27) + 4(1) - 5(12) - 6(20) - 6 = 1 \tag{4.12}$$

Polyhedron Parameter	**Polygonal Mobility Model**
v vertices	$V = 2v$
v_4 4-edge vertices	$S_{+4} = v_4 + 1$
v_5 5-edge vertices	$S_{+5} = v_5$
v_6 6-edge vertices	$S_{+6} = v_6$
v_k k-edge vertices	$S_{+k} = v_k$
f faces	$S_{+4} = f - \sum_{k=3}^{n} S_{+k}$
e edges	$X = e$

Table 4.1: Translation from polyhedral parameters to polygonal mobility model.

In general, the faces, edges, and vertices of a polyhedron may be translated into the polygonal mobility model as shown in Table 4.1. These guidelines may be used to convert any convex polyhedron into a collapsible linkage.

4.4 Conclusions

This chapter has presented alternatives to the traditional Grübler-Kutzbach mobility formula. In the case of just spherical and rotational joints, the new formula (Eqn (4.1)) is based on a simple count of the vertices and edges in a polyhedral model of the assembled links. This single formula applies equally well to spatial, spherical, and planar linkages. We also show how to adapt the formula to include prismatic, cylindrical, and planar joints, and to correctly count the mobility of certain compound spatial mechanisms formed by connecting special planar linkages. This last class of mechanisms includes some constructions built with scissors linkages that may prove useful in medical devices.

Chapter 5

Graphical Synthesis of Pantomeshes

Mechanism or linkage *synthesis* is defined as the creation of a linkage to meet certain specifications [26]. While synthesis is often done using explicit analytical techniques, graphical techniques can also be used and have several advantages. Graphical synthesis allows for a more intuitive dimensional synthesis procedure as compared with analytical techniques, which may require computational solution techniques for complex systems. Although graphical methods have been rarely used in three-dimensional constructions in the past, it is now possible with modern computer-aided-design software to create spatial linkages graphically.

This chapter presents a procedure for synthesizing an arbitrary two-row pantomesh using three-dimensional solid models.

5.1 Graphical Determination of Mobility

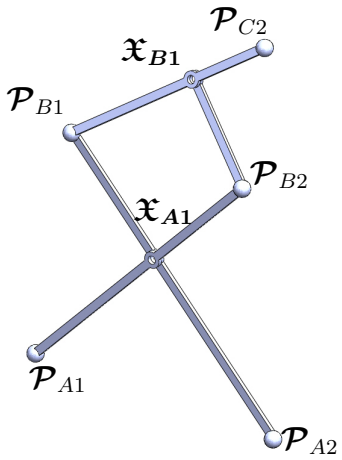
As discussed in Chapter 3, the mobility of each 2x2 pantopatch in a pantomesh is required for the mobility of the entire pantomesh. The mobility of a planar, straight-link pantopatch is given as:

$$\frac{h_{A1} f_{B1}}{g_{A1} e_{B1}} = \frac{f_{A2} h_{B2}}{e_{A2} g_{B2}} \qquad \text{from (3.12)}$$

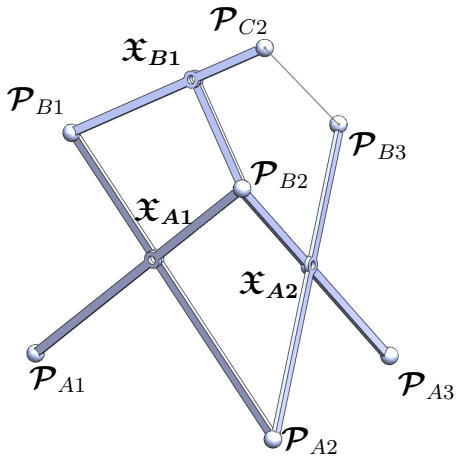
Furthermore, this mobility may be ensured by a graphical construction of Carnot's Law as discussed in §3.4. To restate, if all of the pivot points for all for pantograph elements in the pantopatch are co-planar, then the pantopatch is mobile. This idea

can be extended to constructing an entire pantomesh graphically, using a solid-modeling program.

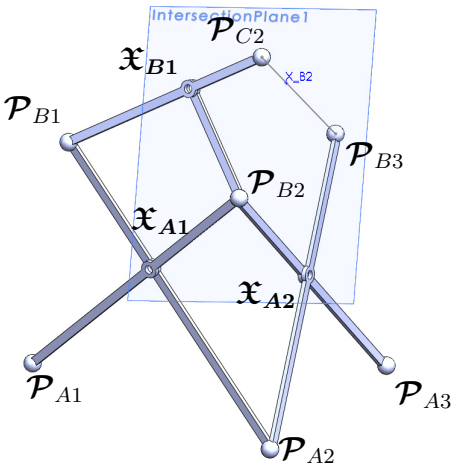
5.2 Graphical Systhesis Procedure



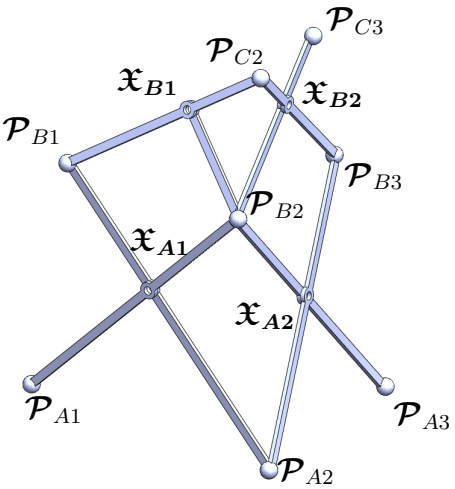
(a) First column except for h_{B1}



(b) Addition of pantograph element $A2$



(c) $\mathfrak{X}_{B2}$ determined by mobility requirements



(d) Columns 1 and 2

Figure 5.1: Graphical synthesis of a closed 2x4 pantomesh — the first two columns.

The construction of two-row, n-column ($2 \times n$) pantomesh begins with the selection

of parameters for the first two columns as shown in Figure 5.1d. All of the link dimensions for the first column in shown Figure 5.1a can be chosen by the designer except for $\boldsymbol{\mathcal{P}}_{C1}$, which will be constructed at when the pantomesh is closed in the final steps.

The next pantograph element, in the first row and second column ($A2$), is also chosen entirely by the designer. These choices include the location of element $A2$ in relation to $A1$. Once element $A2$ is fixed, the left-handed link $\overrightarrow{M}_{B2}$ is determined as the line between $\boldsymbol{\mathcal{P}}_{B3}$ and $\boldsymbol{\mathcal{P}}_{C2}$ as shown in Figure 5.1b.

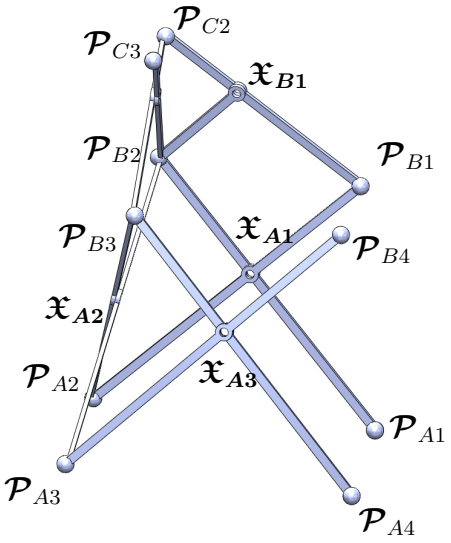
The next step in construction is to find the pivot point $\boldsymbol{\mathfrak{X}_{B2}}$ by using Carnot's law to ensure mobility of the pantopatch. First, draw a plane containing $\boldsymbol{\mathfrak{X}_{A1}}$, $\boldsymbol{\mathfrak{X}_{B1}}$, and $\boldsymbol{\mathfrak{X}_{A2}}$ as shown in Figure 5.1c. From Eqn. (3.12) and Carnot's law, mobility of the pantopatch is guaranteed when $\boldsymbol{\mathfrak{X}_{B2}}$ is on the same plane as $\boldsymbol{\mathfrak{X}_{A1}}$, $\boldsymbol{\mathfrak{X}_{B1}}$, and $\boldsymbol{\mathfrak{X}_{A2}}$. Since planar, straight-link elements are being used, the pivot point $\boldsymbol{\mathfrak{X}_{B2}}$ must be at the intersection between $\overrightarrow{M}_{B2}$ and the constructed plane as shown in Figure 5.1d.

The final point of the second column, $\boldsymbol{\mathcal{P}}_{C2}$, may be chosen by the designer, as long as it is located along the line through $\boldsymbol{\mathcal{P}}_{B2}$ and $\boldsymbol{\mathfrak{X}_{B2}}$ (see Figure 5.1d).

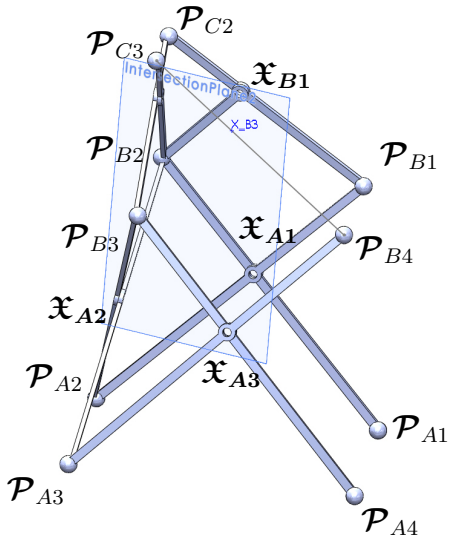
It should be noted that the process could be modified slightly to first define the $B2$ pantograph element, then use the same techniques to find the $A2$ element and select $\boldsymbol{\mathcal{P}}_{A2}$.

The synthesis of all columns between 2 and $(n-1)$ are done in this fashion. To summarize, for each pantomesh column j from 2 to $(n-1)$:

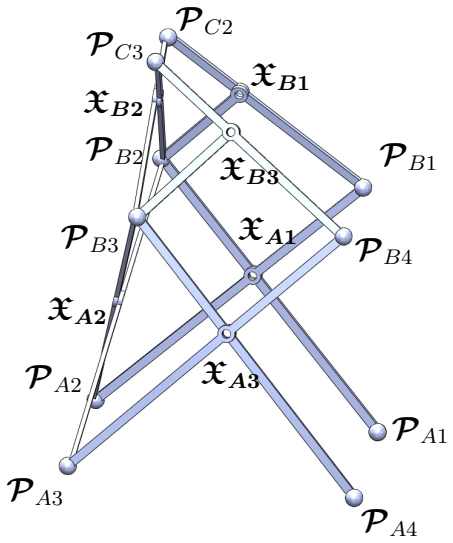
1. Specify pantograph element Ai. This will define $\overrightarrow{M}_{Bi}$.
2. Construct pantograph element Bi using the diagonal intersection planarity mobility requirement discussed in §3.4 to locate $\boldsymbol{\mathfrak{X}_{Bi}}$. To do this, first draw a plane containing $\boldsymbol{\mathfrak{X}_{A(i-1)}}$, $\boldsymbol{\mathfrak{X}_{B(i-1)}}$, and $\boldsymbol{\mathfrak{X}_{Ai}}$. Since $\overrightarrow{M}_{Bi}$ is given with the construction of pantograph element Ai, $\boldsymbol{\mathfrak{X}_{Bi}}$ is found at the intersection between the left-hand diagonal and the mobility plane (see Figure 5.1c).
3. $\boldsymbol{\mathcal{P}}_{C(i+1)}$ may be arbitrarily chosen (except for $\boldsymbol{\mathcal{P}}_{Cn}$ in the $(n-1)^{th}$ column), as long as it is located on the line that goes through $\boldsymbol{\mathcal{P}}_{Bi}$ and $\boldsymbol{\mathfrak{X}_{Bi}}$ (see Figure 5.1d).



(a) Addition of pantograph element $A3$



(b) $\mathfrak{X}_{B3}$ determined by mobility requirements



(c) Columns 1, 2, and 3

Figure 5.2: Graphical synthesis of a closed 2x4 pantomesh — the $(n-1)^{th}$ column.

To construct the $(n-1)^{th}$ column, relationships with the first column must be considered. So when $\mathcal{P}_{Bn}$ is chosen, the location of $\mathcal{P}_{An}$ is restricted to be at the intersection of two planes: the plane containing $\mathcal{P}_{A(n-1)}$, $\mathcal{P}_{B(n-1)}$, and $\mathcal{P}_{Bn}$, and the plane containing points $\mathcal{P}_{Bn}$, $\mathcal{P}_{A1}$, and $\mathcal{P}_{B1}$. This is because all pantograph elements, including $A(n-1)$ and An, are planar for this construction. Once $\mathcal{P}_{An}$ is

chosen, the pivot point $\boldsymbol{\mathfrak{X}_{An}}$ is found as shown in Figure 5.2a.

After pantograph element $A(n-1)$ has been defined, the rest of the construction proceeds in a fashion similar to the previous columns. $\boldsymbol{\mathcal{P}}_{Bn}$ and $\boldsymbol{\mathcal{P}}_{C(n-1)}$ define $\overrightarrow{M_{B(n-1)}}$, and Carnot's law may be used to find $\boldsymbol{\mathfrak{X}_{B(n-1)}}$ as shown in Figure 5.2b.

Much like $\boldsymbol{\mathcal{P}}_{C1}$ in the first column, $\boldsymbol{\mathcal{P}}_{Cn}$ must remain undefined (see Figure 5.2c) until the next steps, when the pantomesh is closed by connecting columns n and 1.

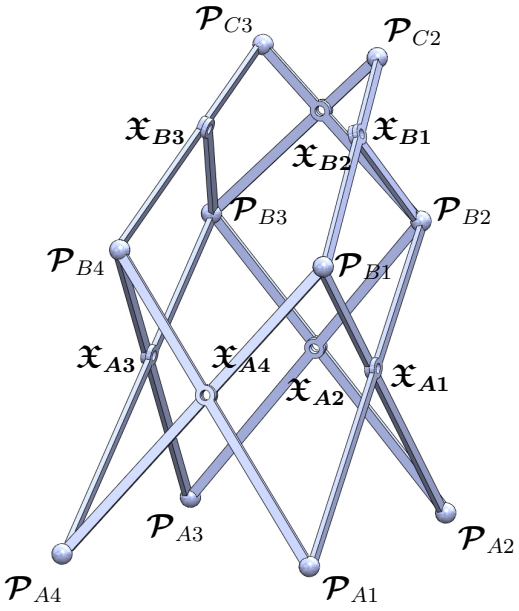
Construction of the final column begins with drawing the links of element An, which is already defined by the existing specifications of $\boldsymbol{\mathcal{P}}_{An}$, $\boldsymbol{\mathcal{P}}_{A1}$, $\boldsymbol{\mathcal{P}}_{Bn}$, and $\boldsymbol{\mathcal{P}}_{B1}$ as shown in Figure 5.3a.

The location of pivot point $\boldsymbol{\mathfrak{X}_{Bn}}$ must satisfy two pantopatch mobility requirements, and therefore must lie on the intersection of two planes. The first contains $\boldsymbol{\mathfrak{X}_{A(n-1)}}$, $\boldsymbol{\mathfrak{X}_{B(n-1)}}$, and $\boldsymbol{\mathfrak{X}_{An}}$; and the second contains $\boldsymbol{\mathfrak{X}_{A1}}$, $\boldsymbol{\mathfrak{X}_{B1}}$, and $\boldsymbol{\mathfrak{X}_{An}}$ as shown in Figure 5.3b.

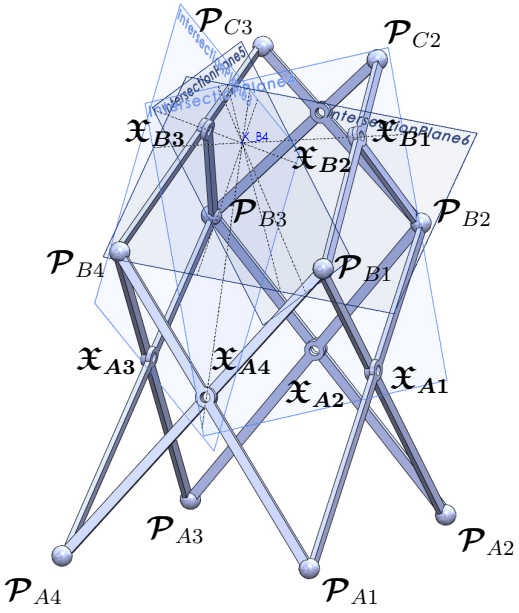
Furthermore, since the links are straight and planar, the location of the pivot is again restricted. $\boldsymbol{\mathfrak{X}_{Bn}}$ is located on $\overrightarrow{L_{Bn}}$, which intersects $\overrightarrow{M_{B1}}$ at $\boldsymbol{\mathcal{P}}_{C1}$. That means that $\boldsymbol{\mathfrak{X}_{Bn}}$ is on a plane that contains those two links, which must contain $\boldsymbol{\mathcal{P}}_{B4}$, $\boldsymbol{\mathcal{P}}_{B2}$, and $\boldsymbol{\mathfrak{X}_{B1}}$. The same can be said for $\overrightarrow{M_{Bn}}$, which is in the plane defined by $\boldsymbol{\mathcal{P}}_{B1}$, $\boldsymbol{\mathcal{P}}_{B(n-1)}$, and $\boldsymbol{\mathfrak{X}_{B(n-1)}}$ (see Figure 5.3b).

Pivot point $\boldsymbol{\mathfrak{X}_{Bn}}$ is on the point where any three of these four planes meet as shown in Figure 5.3b. Therefore, the plane containing the rest of pantograph element Bn, including $\boldsymbol{\mathcal{P}}_{Cn}$ and $\boldsymbol{\mathcal{P}}_{C1}$, may be defined by $\boldsymbol{\mathcal{P}}_{Bn}$, $\boldsymbol{\mathcal{P}}_{B1}$, and $\boldsymbol{\mathfrak{X}_{Bn}}$. $\boldsymbol{\mathcal{P}}_{Cn}$ is located at the intersection between three planes: the plane containing element $B(n-1)$, the plane containing element Bn, and the plane containing defined by $\boldsymbol{\mathcal{P}}_{B1}$, $\boldsymbol{\mathcal{P}}_{B(n-1)}$, and $\boldsymbol{\mathfrak{X}_{B(n-1)}}$ described above. $\boldsymbol{\mathcal{P}}_{C1}$ is similarly found using: the plane containing element $B(n-1)$, the plane containing element Bn, and the plane containing $\boldsymbol{\mathcal{P}}_{B4}$, $\boldsymbol{\mathcal{P}}_{B2}$, and $\boldsymbol{\mathfrak{X}_{B1}}$. These points are shown in Figure 5.3c.

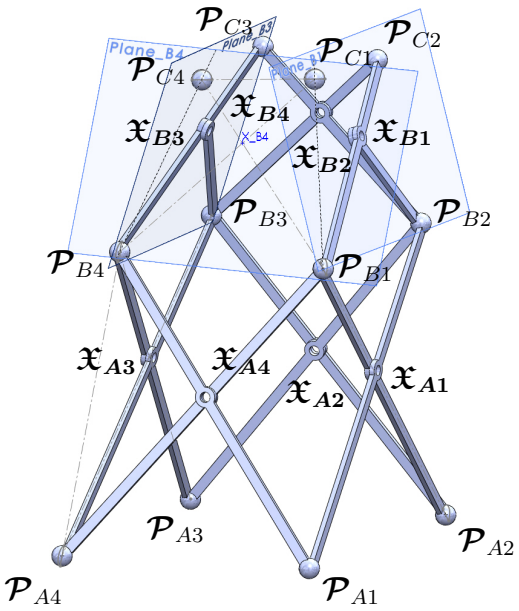
Finally, the rest of the link segments (L_{A1}, M_{B1}, $g_{A(n-1)}$, and $f_{B(n-1)}$) are completed to close the pantomesh (see Figure 5.3d). The final, assembled pantomesh is fully mobile with a single degree of freedom.



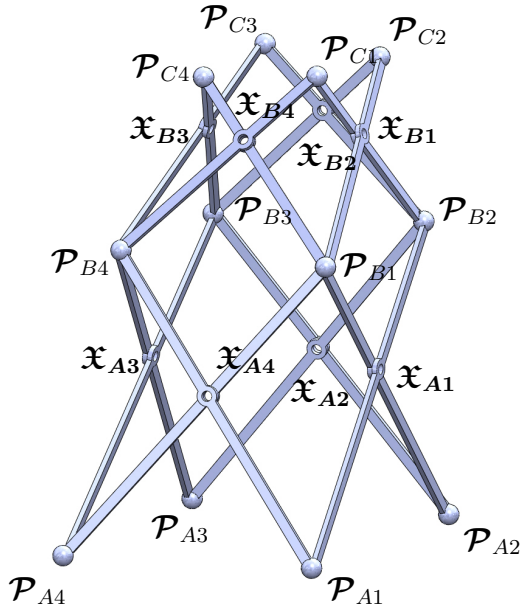
(a) Pantograph element $A4$ is given by existing geometry



(b) $\mathfrak{X}_{B4}$ determined by mobility requirements



(c) $\mathcal{P}_{C4}$ and $\mathcal{P}_{C1}$ determined by geometry



(d) Completed mobile pantomesh

Figure 5.3: Graphical synthesis of a closed 2x4 pantomesh — the final column.

5.3 Discussion

Graphical synthesis as presented in the planar, straight-link example above illuminates several important properties of synthesis in general. The first is the availability of "free choices" in synthesizing the pantomesh. Almost all of the bottom two

rows of points is freely chosen except for one ($\boldsymbol{\mathcal{P}}_{An}$ in the example). $\boldsymbol{\mathcal{P}}_{C2}$ is likewise freely chosen. The remaining C-level points have only one free choice along a line through a pivot point, up to $\boldsymbol{\mathcal{P}}_{C(n-1)}$. The final points, $\boldsymbol{\mathcal{P}}_{Cn}$ and $\boldsymbol{\mathcal{P}}_{C1}$, are determined by the mobility requirements. Naturally, more complicated element types have more free choices. Details of the exact number of free choices are provided in Chapter 6.

While not discussed in this chapter, this procedure could be duplicated for multi-row pantomeshes. After the first two rows are constructed, more rows may be added as long as each pantopatch obeys the planar pivot point criteria for mobility.

5.4 Conclusions

This chapter presents a method for synthesizing pantomeshes using graphical techniques. A computer-aided-design system is used to assemble links column-by-column, obeying pantopatch mobility requirements to create a two-row by four-column pantomesh is synthesized.

Chapter 6

Numerical Synthesis of Pantomeshes

The synthesis of pantomeshes can be achieved by providing geometric specifications (or shape specifications) and solving the mobility equations presented in Chapter 3. Depending on the design task, a pantomesh may be designed to approximate a certain shape, while maintaining a certain element type (i.e. straight-link, planar, etc.). As opposed to the graphical techniques presented in Chapter 5, this chapter will briefly describe a general procedure for analytical synthesis of pantomeshes. Since there are many ways to specify desired pantomesh characteristics, the main focus will be on two examples. The first example is a simple, planar straight-link pantomesh that will match one shape. The second example is skew, normal-pivot bent-link pantomesh that goes through two different shapes during its range of motion.

6.1 Shape Synthesis

The first step in numerical synthesis is to determine the number of independent variables and, therefore, the number of equations necessary to obtain a solution. For a planar column-closed mesh with m rows and n columns, the total number of unknown variables is $4mn$ for straight link pantographs and $6mn$ for bent link pantographs. This is easily calculated: each straight link has two variables, its length and the location of the revolute joint. For bent links, an extra variable is required to define an angle.

To synthesize a mesh based on constraints, basic geometric rules may be followed as shown in Table 6.1. First is the set of variables, starting with the link dimensions

that define the links themselves. The link dimensions do not change throughout the motion of the mesh. In general, any geometric specifications translate into a position in space – three coordinate variables for each pantograph element endpoint. These points may also be solved for different positions or *shape configurations*, the total number of shapes being s. A shape configuration is analogous to a precision point in standard kinematic synthesis [26, p. 491]. Obviously, the points in space are related to the link dimensions. Those relations are defined in the synthesis equations.

Variables	Straight-Link Pantograph	Bent-Link Pantograph
Link Lengths & Ratios (L, M, e, g, etc.)	$4mn$	$6mn$
Pantograph End Points (x, y, z)	$3(m+1)ns$	$3(m+1)ns$

Table 6.1: Shape synthesis variables.

To have a solvable system, the number of equations (see Table 6.1) must equal the number of variables. These equations can be considered in two categories: link equations (those that relate the links in space) and shape equations (those that constraint points to certain geometric features). The first of the link equations relate the link dimensions to the pantograph endpoints using basic distance equations. There are four equations for each pantoelement for a total of $4mns$.

Next, the planarity of each pantograph element must be established if the element is planar by setting the distance between link spans for each element to equal zero. Similarly, if the element is skew, then a fixed offset distance d would be used instead of a zero distance. There is only one of these equations for each pantoelement for a total of mns.

Finally, the mobility of the mesh is verified by ensuring that every pantopatch satisfies the mobility requirements detailed in Chapter 3. There are $(m-1)n$ pantomeshes for a pantomesh closed along the columns.

The other set of equations involves the actual shape specifications. A shape can be defined in a number of different ways: several constraints may be placed on points, perhaps in different shape configurations. Examples include: point-on-curve constraints, consisting of two equations per point, and point-on-surface constraints, which consist of one equation per point. Points may also be fixed to pre-defined points in space, thus requiring an equation for each dimension.

Geometry Equations		Constraint Equations	
Link Dimensions	$4mns$	**Fixed point**	$3p$
Pantograph Planarity	mns	**Point-on-curve**	$2p$
Mobility	$(m-1)n$	**Point-on-surface**	p

Table 6.2: Shape synthesis equations.

These equations can be arranged as constraints to be solved for an entire system, with the goal of finding the "best fit" for the specified requirements. One option for determining the best fit is to use the conjugate gradient search method [31]. This algorithm is as follows:

1. **Guess** the values of t_i for all $3n$ points that satisfy the geometric constraints.
2. **Calculate** and sum the sources of error e, such as d which should be zero if they are planar, and the difference from "perfect mobility" for each pantograph. Other errors may be added to coerce the solver to stay within certain boundaries, to avoid overlap, etc.
3. **Simulate** the gradient of the error relative to guesses, i.e. $\frac{\partial e}{\partial t_i}$.
4. **Search** along the negative gradient (*steepest descent*) for the lowest total error.
5. **Calculate** the error and its gradient.
6. **Calculate** the search direction based on the current and previous gradient.
7. **Search** along the search direction for the lowest total error.
8. **Compare** the error value to the lowest acceptable value and/or the lowest acceptable change in value.
9. **If** the criteria has not be satisfied, go back to step 5.

These steps are best detailed by example. The following example is a planar, straight-link pantomesh sythesized to fit within a pre-set shape.

6.2 Example: One Position, Three Point-on-Curve Constraints

The simplest form of numerical synthesis is a two-row, planar, straight-link pantomesh. Given three curves in space, a pantomesh may be synthesized so that the

endpoints of each pantograph lie on those curves. Using the terms from Tables 6.1 and 6.2, the number of equations and variables is matched exactly for this two-row, six-column mesh. There are 17 variables and 17 equations for each column, for a total of 102 each.

In this example, the three curves are ellipses located in parallel planes. These curves define the desired endpoint locations for the $\boldsymbol{\mathcal{P}}_{Ai}$'s, $\boldsymbol{\mathcal{P}}_{Bi}$'s, and $\boldsymbol{\mathcal{P}}_{Ci}$'s. The equations for these ellispes are represented using a parametric variable, t, to follow point-on-curve constraints as shown in Eqn. (6.1).

$$\begin{aligned}
&x_A = p_A \cos t; && y_A = q_A \sin t; && z_A = 0 && \\
&x_B = p_B \cos t; && y_B = q_B \sin t; && z_B = z_B && \text{(const)} \\
&x_C = p_C \cos t; && y_C = q_C \sin t; && z_C = z_C && \text{(const)}
\end{aligned} \tag{6.1}$$

The links are closed across the columns, meaning that the final, Nth column is connected to the first column. This means that

$$\boldsymbol{\mathcal{P}}_{A(N+1)} = \boldsymbol{\mathcal{P}}_{A1} \tag{6.2}$$

$$\boldsymbol{\mathcal{P}}_{B(N+1)} = \boldsymbol{\mathcal{P}}_{B1} \tag{6.3}$$

$$\boldsymbol{\mathcal{P}}_{C(N+1)} = \boldsymbol{\mathcal{P}}_{C1} \tag{6.4}$$

Therefore, each point is associated with a row and a parametric term t for each point. This yields a precise position in space defined by Eqn. (6.1). The parametric term is used as the independent variable for the optimization.

Given the coordinates of each link endpoint, the link lengths of the synthesized pantomesh are calculated with a basic distance formula:

$$L_{i,j}^2 = (x_{i+1,j+1} - x_{i,j})^2 + (y_{i+1,j+1} - y_{i,j})^2 + (z_{i+1,j+1} - z_{i,j})^2 \tag{6.5}$$

$$M_{i,j}^2 = (x_{i,j+1} - x_{i+1,j})^2 + (y_{i,j+1} - y_{i+1,j})^2 + (z_{i,j+1} - z_{i+1,j})^2 \tag{6.6}$$

The remainder of the variables are measured against their desired values. For the distance between links d (see Figure 2.3 on p. 20), calculated as shown in Eqn. (2.10) and Eqn. (2.11), the desired value is zero for each pantograph elment to force pla-

narity.

$$\hat{n} = \frac{\vec{\mathbf{L}}_{\mathbf{A1}} \times \vec{\mathbf{M}}_{\mathbf{A1}}}{|\ \vec{\mathbf{L}}_{\mathbf{A1}}\ ||\ \vec{\mathbf{M}}_{\mathbf{A1}}\ |} \qquad \text{from (2.10)}$$

$$d_{A1} = \hat{n_{A1}} \cdot \vec{\boldsymbol{\mathcal{P}}}_{A1} - \hat{n_{A1}} \cdot \vec{\boldsymbol{\mathcal{P}}}_{A2} \qquad \text{from (2.11)}$$

$$\mathcal{E}_{d,A1} = d_{A1} \qquad (6.7)$$

The "flattened" equations above may be used to calculate the the other link dimensions, namely e, f, g, and h. First, a new pantograph bottom distance $\vec{j}_{\text{trans}}$ is calculated by transposing the vector $\vec{\boldsymbol{\mathcal{P}}}_{A2}$ the distance d (which may not be zero for this set of t's) as shown in Eqn. (6.8). The subsequent equations calculate ratios of link dimensions using vector algebra.

$$\vec{j}_{\text{trans}} = \vec{\boldsymbol{\mathcal{P}}}_{A2} + d\hat{\mathbf{n}} - \vec{\boldsymbol{\mathcal{P}}}_{A1} \qquad (6.8)$$

$$\frac{f}{e} = \frac{\|\vec{L} \times \vec{M}\|}{\left(\vec{j}_{\text{trans}} \times \vec{M}\right) \cdot \left(\vec{L} \times \vec{M}\right) - 1} \qquad (6.9)$$

$$e = \frac{L}{1 + \frac{f}{e}} \qquad (6.10)$$

$$f = L - e \qquad (6.11)$$

$$\frac{h}{g} = \frac{\|\vec{M} \times \vec{L}\|}{\left(-\vec{j}_{\text{trans}} \times \vec{L}\right) \cdot \left(\vec{M} \times \vec{L}\right) - 1} g \qquad = \frac{M}{1 + \frac{h}{g}} \qquad (6.12)$$

$$h = M - g \qquad (6.13)$$

After the link dimensions are calculated, the *mobility error* is calculated using the mobility criteria from Eqn. (3.12), rearranged to obtain zero error for a mobile linkage:

$$\mathcal{E}_{\mathcal{M},A1} = \frac{f_{B1} h_{A1}}{e_{B1} g_{A1}} - \frac{f_{B2} h_{A2}}{e_{B2} g_{A2}} \qquad (6.14)$$

The total error of the system, represented by the root-mean-square of all component errors, then becomes the factor to be minimized in the optimization algorithm. Successive iterations involve minimizing the error by altering values for the t's representing pantograph endpoints on each curve.

A two-row, six-column example is presented in Figure 6.1. The three ellipses are

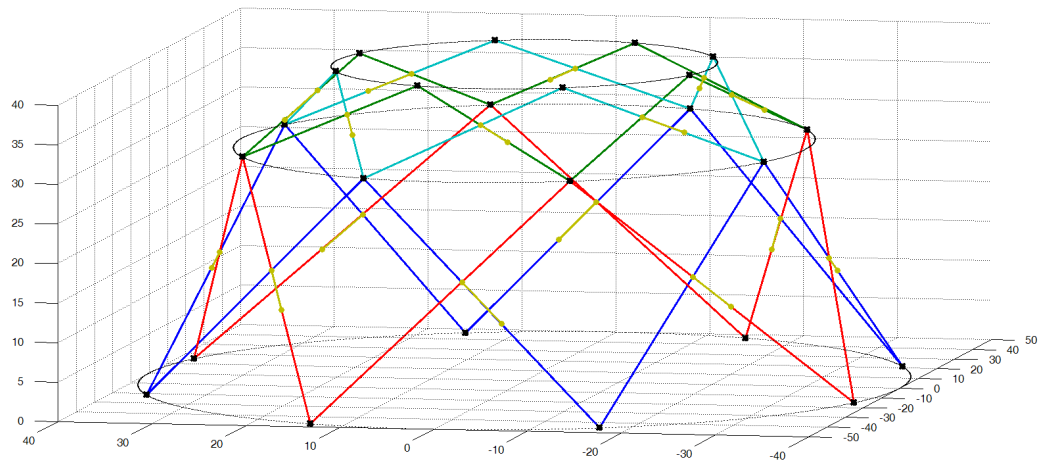


Figure 6.1: Solution for a two-row, six column straight-line pantograph mesh.

curves that the link endpoints are arbitrarily specified to lie on. The straight lines and points represent the links themselves. Visually it can be surmised that the endpoints lie on their respective curves with very little error.

6.3 Example: Three Rows, Two Positions

The previous example works well for single-position specifications, but since the point of a pantomesh is to have a linkage in *motion*, an example for two positions is presented. The synthesis process is similar to the previous section, except for the manner of point constraints, the use of skew, bent-link elements, and multiple positions. In this example, we will use point-on-surface constraints for the link endpoints. Therefore, we must first define the surface.

Each point (x, y, z) in space is defined as a point on a surface. One method for defining a surface is a NURBS (non-uniform rational B-spline) surface [32]. Those points are calculated as a function of parametric variables u and v:

$$x(u,v) = \sum_{i=1,j=1}^{m,n} (x_{patch(i,j)} B_i(u) B_j(v)) \tag{6.15}$$

$$y(u,v) = \sum_{i=1,j=1}^{m,n} (y_{patch(i,j)} B_i(u) B_j(v)) \tag{6.16}$$

$$z(u,v) = \sum_{i=1,j=1}^{m,n} (z_{patch(i,j)} B_i(u) B_j(v)) \tag{6.17}$$

where $x_{patch(i,j)}, y_{patch(i,j)}$, and $z_{patch(i,j)}$ are control points specified at the outset to define the shape of the NURBS surface. The terms $B_i(u)$ and $B_j(v)$ are *Bernstein polynomial functions*, which are used to weight the control points appropriately for that position on the patch. These functions give smooth transitions between specified points. The degree of a Bernstein polynomial is determined by the number of control points to be specified. For a patch defined by 6 points in the u direction and 4 points in the v direction, the Bernstein polynomials are

$$\begin{aligned}
B_1(u) &= (1-u)^5 & \\
B_2(u) &= 5u(1-u)^4 & B_1(v) &= (1-v)^3 \\
B_3(u) &= 10u^2(1-u)^3 & B_2(v) &= 3v(1-v)^2 \\
B_4(u) &= 10u^3(1-u)^2 & B_3(v) &= 3v^2(1-v) \\
B_5(u) &= 5u^4(1-u) & B_4(v) &= v^3 \\
B_6(u) &= u^5 &
\end{aligned} \tag{6.18}$$

By expressing the Berstein polynomial results as vectors $\vec{B(u)}$ and $\vec{B(v)}$, the following equations can be used to find an appropriate (x, y, z) coordinates in space for each (u, v) coordinate pair on the surface. The $\boldsymbol{x_{patch}}$, $\boldsymbol{y_{patch}}$, and $\boldsymbol{z_{patch}}$ matrices contain the control points.

$$\begin{aligned}
x(u,v) &= \vec{B(u)}^T \, \boldsymbol{x_{patch}} \, \vec{B(v)} \\
y(u,v) &= \vec{B(u)}^T \, \boldsymbol{y_{patch}} \, \vec{B(v)} \\
z(u,v) &= \vec{B(u)}^T \, \boldsymbol{z_{patch}} \, \vec{B(v)}
\end{aligned} \tag{6.19}$$

So for a two-position system, two sets of (u, v) parameters are given to define a pantomesh. The dimensions of each pantograph element are determined in the same manner as in the previous section, performed for each position. What remains, however, is the bent-link angles α and β. These angles cannot be determined by the location of the link endpoints of a single position, but can be found by analyzing the element links in both positions and finding their *poles*.

A *pole* for a link is the center of rotation between two positions. Figure 6.2 shows a schematic of two pantograph elements superimposed on top of each other in a single plane. $\boldsymbol{\mathcal{P}}_{A1}$ is set as the origin for both positions, while $\boldsymbol{\mathcal{P}}_{A2}$ for both po-

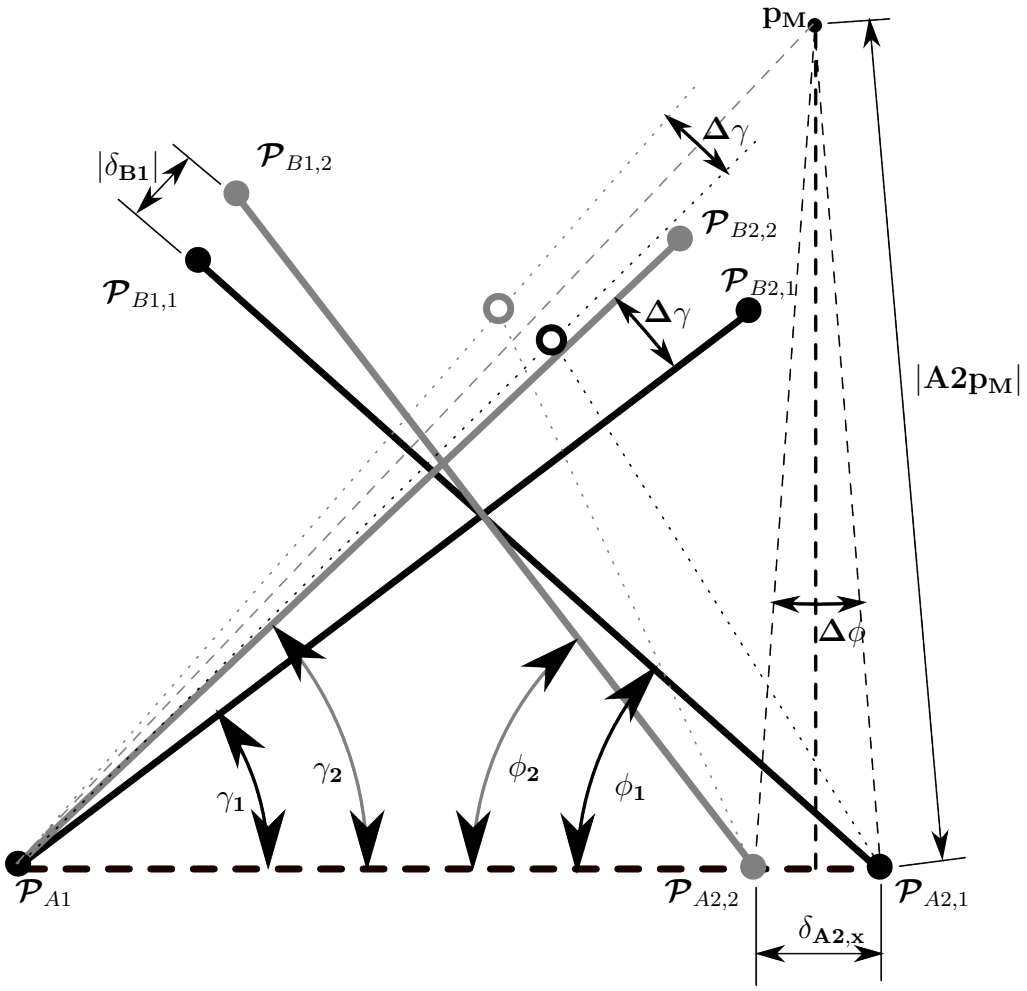


Figure 6.2: Superimposed pantograph elements used to find synthesized pivot of a skew pantograph element from two positions.

sitions ($\boldsymbol{\mathcal{P}}_{A2,1}$ and $\boldsymbol{\mathcal{P}}_{A2,2}$) are fixed onto the x-axis. In order to find the bent-link dimensions, first the pole of link M (p_M) must be found.

The left-hand bent-link pole may be found by comparing both positions of each pantograph element as shown in Figure 6.2.

$$\delta_{A2,x} = j_{x,2} - j_{x,1} \tag{6.20}$$

$$\delta_{B1,x} = j_{x,2} + M_{x,2} - (j_{x,1} + M_{x,1}) \tag{6.21}$$

$$\delta_{B1,y} = M_{y,2} - M_{y,1} \tag{6.22}$$

$$|\delta_{B1}| = \sqrt{\delta_{B1,x}^2 + \delta_{B1,y}^2} \tag{6.23}$$

$$p_{M,x} = \frac{j_{x,1} + j_{x,2}}{2} \tag{6.24}$$

Now the cosine law may be used to solve for the y-coordinate of the pole p_M: the center of rotation of link M between positions 1 and 2. Since $\sin\phi = \cos(90^\circ - \phi)$, the two measurements of the pole's y-coordinate are:

$$|A2p_M|_1 = \frac{-(-2M_1 \sin\phi_1) + \sqrt{(-2M_1 \sin\phi_1)^2 - 4\left(1 - |\delta_{B1}|\right) M_1}}{2\left(1 - |\delta_{B1}|\right)} \tag{6.25}$$

$$|A2p_M|_2 = \frac{-(-2M_1 \sin\phi_2) + \sqrt{(-2M_1 \sin\phi_2)^2 - 4\left(1 - |\delta_{B1}|\right) M_2}}{2\left(1 - |\delta_{B1}|\right)} \tag{6.26}$$

$$p_{M,y,1} = \sqrt{|A2p_M|_1^2 - \left(\frac{\delta_{A2,x}}{2}\right)} \tag{6.27}$$

$$p_{M,y,2} = \sqrt{|A2p_M|_2^2 - \left(\frac{\delta_{A2,x}}{2}\right)} \tag{6.28}$$

Using the coordinates of the pole for the M link, α is found by recognizing that the pivot will be offset half of the rotation angle γ from the line between the origin and p_M, as shown in Eqn. (6.29).

$$\alpha = \tan^{-1}\left(\frac{p_{M,y,1}}{p_{M,x,1}}\right) - \frac{\gamma_2 - \gamma_1}{2} - \gamma_1 \tag{6.29}$$

Furthermore, e may be found by using the cosine law expression for g^2 twice, equating them, and solving for e.

$$g^2 = e^2 + j_{x,1}^2 - 2ej_{x,1}\cos(\gamma_1 + \alpha) \tag{6.30}$$

$$= e^2 + j_{x,2}^2 - 2ej_{x,2}\cos(\gamma_2 + \alpha) \tag{6.31}$$

$$e = \frac{j_{x,2}^2 - j_{x,1}^2}{2(j_{x,2}\cos(\gamma_2 + \alpha) - j_{x,1}\cos(\gamma_1 + \alpha))} \tag{6.32}$$

The remaining link quantities may be found as follows:

$$f = \sqrt{L^2 + e^2 - 2Le\cos\alpha} \tag{6.33}$$

$$g = \sqrt{e^2 + j_{x,1}^2 - 2ej_{x,1}^2\cos(\gamma_1 + \alpha)} \tag{6.34}$$

$$\beta = \sin^{-1}\left(\frac{e}{g}\sin(\gamma_1 + \alpha)\right) - \phi_1; \tag{6.35}$$

$$h = \sqrt{M^2 + g^2 - 2Mg\cos\beta} \tag{6.36}$$

Given these calculated link dimensions, a list of "errors" may be created to repre-

sent undesirable characteristics of the "guessed" pantomesh. The following errors are calculated for each pantograph element to make sure that the link dimensions remain the same between positions:

$$\mathcal{E}_d = d_2 - d_1 \tag{6.37}$$

$$\mathcal{E}_L = L_2 - L_1 \tag{6.38}$$

$$\mathcal{E}_M = M_2 - M_1 \tag{6.39}$$

In order for a skew, normal-pivot bent-link pantomesh to be mobile, each pantopatch must obey the mobility equations from § 3.3.2. Therefore, for all pantopatches,

$$\mathcal{E}_{mobility(first)} = \frac{d(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B1}})}\frac{d(\boldsymbol{j}^2_{\boldsymbol{B1}})}{d(\boldsymbol{k}^2_{\boldsymbol{A2}})} - \frac{d(\boldsymbol{k}^2_{\boldsymbol{B2}})}{d(\boldsymbol{j}^2_{\boldsymbol{B2}})}\frac{d(\boldsymbol{j}^2_{\boldsymbol{B2}})}{d(\boldsymbol{k}^2_{\boldsymbol{A2}})} \tag{6.40}$$

$$\mathcal{E}_{mobility(A2,A1,1)} = e_{A2}h_{A2} - g_{A1}f_{A1} \tag{6.41}$$

$$\mathcal{E}_{mobility(A2,A1,2)} = e^2_{A2} + h^2_{A2} + d^2_{A2} - g^2_{A1} + f^2_{A1} + d^2_{A1} \tag{6.42}$$

$$\mathcal{E}_{mobility(B1,A1,1)} = e_{B1}g_{B1} - h_{A1}f_{A1} \tag{6.43}$$

$$\mathcal{E}_{mobility(B1,A1,2)} = e^2_{B1} + g^2_{B1} + d^2_{B1} - h^2_{A1} + f^2_{A1} + d^2_{A1} \tag{6.44}$$

$$\mathcal{E}_{mobility(B2,A2,1)} = g_{B2}e_{B2} - f_{A2}h_{A2} \tag{6.45}$$

$$\mathcal{E}_{mobility(B2,A2,2)} = g^2_{B2} + e^2_{B2} + d^2_{B2} - f^2_{A2} + h^2_{A2} + d^2_{A2} \tag{6.46}$$

$$\mathcal{E}_{mobility(B2,B1,1)} = e_{B2}h_{B2} - g_{B1}f_{B1} \tag{6.47}$$

$$\mathcal{E}_{mobility(B2,B1,2)} = e^2_{B2} + h^2_{B2} + d^2_{B2} - g^2_{B1} + f^2_{B1} + d^2_{B1} \tag{6.48}$$

Additional constraints are needed to ensure proper formation of the pantograph elements. The first requirement is for $e < L$ and $g < M$ so that each link is properly formed. Secondly, the bend angles of the links should be limited to a reasonable angle, say 30°.

$$\mathcal{E}_{tri,RH} = \max\left((e - L + 1), 0\right) \tag{6.49}$$

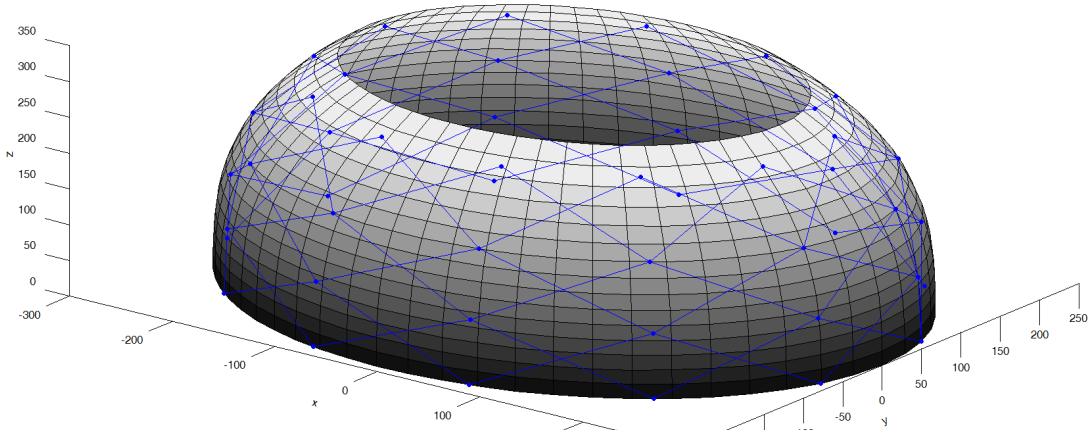
$$\mathcal{E}_{tri,LH} = \max\left((g - M + 1), 0\right) \tag{6.50}$$

$$\mathcal{E}_{alpha,max} = \max\left((\alpha - 30^\circ), 0\right) \tag{6.51}$$

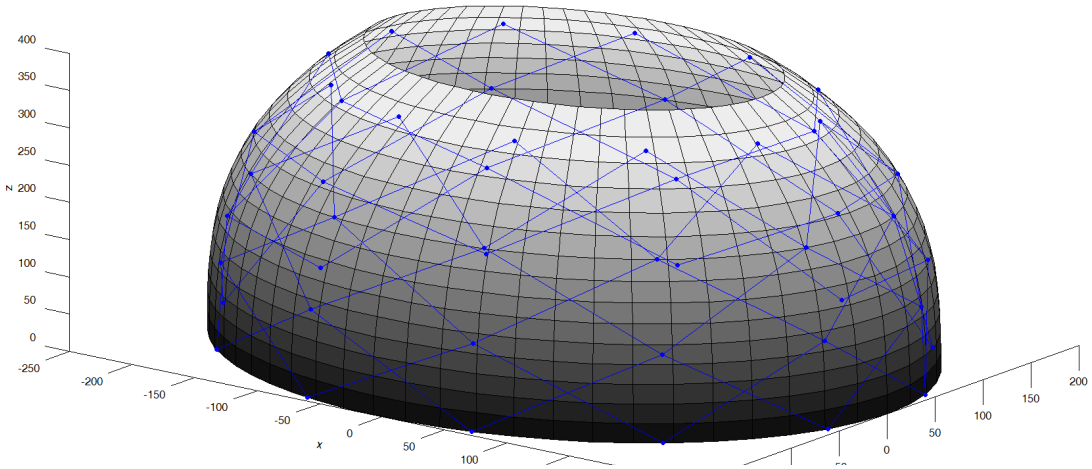
$$\mathcal{E}_{beta,max} = \max\left((\beta - 30^\circ), 0\right) \tag{6.52}$$

The total error of each proposed pantomesh is defined as the root-mean-square of

all of the error values. The goal of the optimization routine is to minimize this error.



(a) Position 1 (wider and shorter)



(b) Position 2 (thinner and taller)

Figure 6.3: Solution for two specified shapes.

Using the procedure outlined at the beginning this chapter, a pantomesh was synthesized to fit two oblong dome shapes as shown in Figure 6.3. The total error of this pantomesh, when weighted for the average link span length, is less than 2%. A MATLAB-style program listing is provided in Appendix B.

6.4 Conclusion

This chapter presents a numerical method for synthesizing pantomeshes. Using well-known optimization techniques, a pantomesh may be designed to meet certain shape requirements. A straight-link pantomesh was synthesized using a

multi-dimensional optimization method to satisfy three point-on-curve constraints for a single position. Using a similar technique, a three-row pantomesh was synthesized to satisfy point-on-surface constraints for two separate shapes.

Chapter 7

Multi-Link Spherical Joints

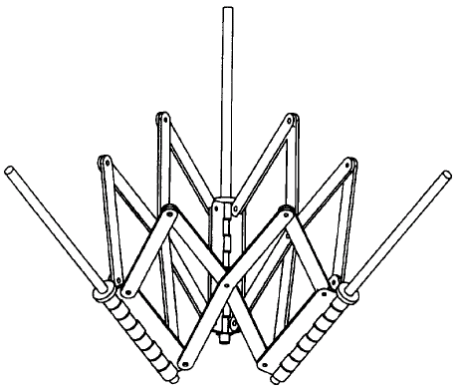
Pantomeshes make extensive use of spherical joints to connect scissor pairs. A *spherical joint* has three rotational degrees of freedom (DOF) around a single spherical center yet allows no translational motion between two links relative to that center, commonly realized by a ball-and-socket pair. By extension, a *multi-link* or *collocated* spherical joint has multiple links connected so that each has three rotational DOF between each link about a single spherical center. Multi-link spherical joints are necessary when more than two scissor pairs are joined at one point. While the ball-and-socket joint is conceptually simple, its manufacture and assembly in large quantities is more difficult, especially as the number of links increases. Practical designs for multi-link spherical joints are needed.

This chapter reviews various approaches to multi-link spherical joints necessary for pantomeshes. First, existing work in collocated spherical joints is discussed. Then a description of how conventional rigid multi-link spherical joints may be incorporated into pantomeshes is presented, followed by a compliant multi-link spherical joint of similar design. Finally, a new crossed-tendon design is described that may provide a more manufacturable option than rigid links while providing more off-axis rigidity than compliant designs.

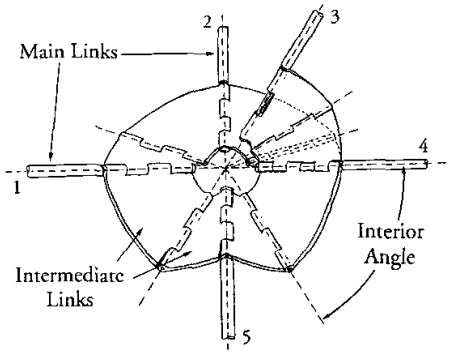
7.1 Background

A significant challenge in fabricating pantomeshes is the need for collocated spherical joints. That is, where four links will intersect at one joint. Simple ball-and-socket joints cannot contain this, so other mechanisms are required to maintain mo-

tion while remaining compact. Two such joints are shown below, in Figure 7.1 [33–35]. These linkages, among others, provide a simple method for collocating spher-



(a) Scissor-link Sphercial Joint [33,34]



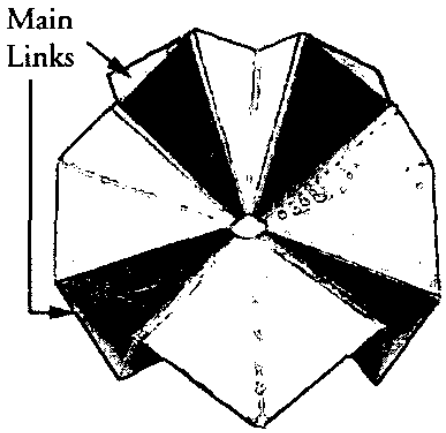
(b) Spherical Mechanism [35]

Figure 7.1: Existing multiple collocated spherical joints.

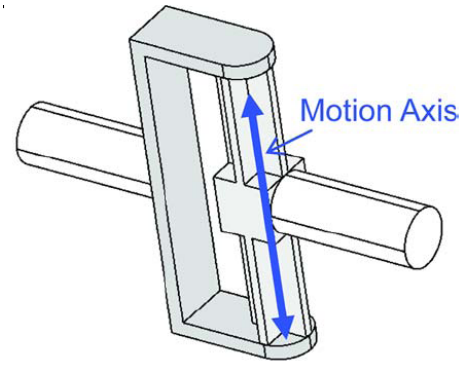
ical joints for multiple links. However, the number of additional links required to make each spherical joint adds to complexity and cost related to manufacturing and assembly. The design shown in Figure 7.1b is perhaps better suited to this application because of the fewer number of links.

Compliant linkages may be a lower-cost solution to the problem of manufacturability. Instead of "rigid" joints, i.e. those that involving close-fitting surface-to-surface contact, compliant joints consist of members that flex. Therefore, rigid spherical and revolute joints shown in Figure 7.1 could be replaced by those shown in Figure 7.2. While these joints restrict the range of motion of the entire linkage, as well as increase the applied force requirements, there is a distinct advantage in manufacturing. By making the entire linkage a single piece, it can be injection molded and require no assembly. This will result in a great cost reduction in the per-unit cost. However, without large volumes, the cost benefit of a compliant device may not be fully realized.

Pantomeshes require specific features of multi-link spherical joints. Typically, either two or four links meet at one point, and a ball-and-socket joint may be used for the two-link joint. A special property of pantomesh spherical joints is that of opposing pantograph link movement: when two links move toward each other, the opposing two links also move toward each other and further away from the first two. Finally, the links should never need to cross each other in typical design scenarios. These facts impose clearance and compliance features of various



(a) Compliant Multi-Link Spherical Joint [35]



(b) Compliant Revolute Joint [36]

Figure 7.2: Existing compliant joints suitable for pantomeshes.

joints that may not apply in the general case of multi-link spherical joints. Those implications will be discussed in each of the following sections.

7.2 Rigid (Non-compliant) Multi-Link Spherical Joints

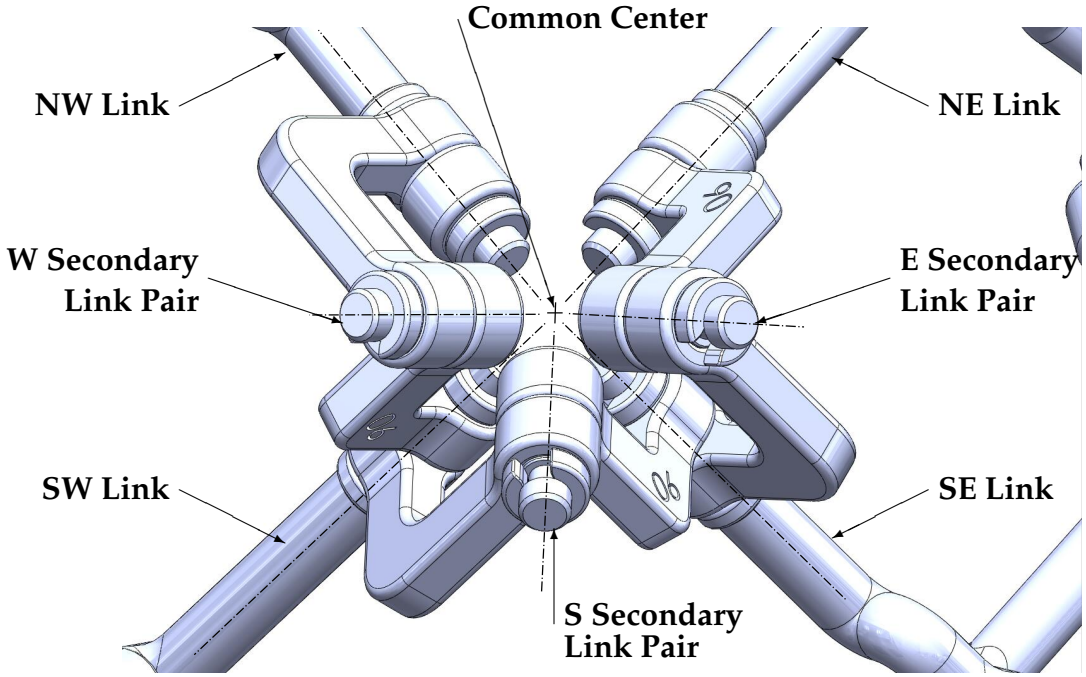


Figure 7.3: Rigid multi-link spherical joint.

The most conventional type of joint is a *rigid* or non-compliant joints. Rigid joints are neither fixed nor immobile; rather it is the joint *components* that are rigid and

close-fitting. For a rigid revolute joint, often a cylindrical pin rotates in a hole to allow a single rotational degree of freedom and no movement in any other degree of freedom. A ball-and-socket joint is a rigid spherical joint for two links. A multi-link spherical joint may be created by using multiple revolute joints that share a common center, as shown in Figure 7.3 and described by Bosscher [35].

Given four *primary* links labeled **SE**, **SW**, **NW**, and **NE**, the Bosscher linkage uses four secondary link pairs, **N**, **E**, **W**, and **S**. The **N** pair is not shown in Figure 7.3 Between any two adjacent primary links there are three revolute joints that share a common center, creating a spherical joint. All joints are directly connected to each other via these secondary link pairs except the **N** position, which is redundant and therefore omitted in Figure 7.4. This redundancy is due to the indirect coupling of the **NW** and **NE** links through the rest of the linkage. Adding the redundant **N** secondary link pair may be helpful to reduce error due to lack of joint stiff, but it may also naturally increase the actuation friction by adding more surface area contact.

The range of motion of the multi-link spherical joint is bounded by two main factors: link collision and joint link extension. The bulky revolute joints comprising the multi-link spherical joint limit its range of motion due to colliding links. This is an unavoidable consequence of intersecting axes; otherwise, the standard solution of offsetting the revolute joints would be appropriate.

The joints that connect neighboring links also have an extension limit based on the angle between link joint axes. Assuming both secondary links between primary links have an angle α between revolute axes, then the maximum angle between primary links is 2α. However, the larger α for the secondary links, the increased chance that they may collide with the opposing secondary link pair when collapsed.

The error possible with this joint design is strictly a function of manufacturing precision of each revolute joint, which may be improved with more expensive fabrication techniques, and therefore has a very low theoretical error.

Once assembled, the complexity of a larger pantomesh is clear (see Figure 7.4). The number of components required is a minimum of six for each multi-link spherical joint, and as many as 13 for figure shown. Manufacturability therefore may be difficult and quite costly. These drawbacks are addressed in the following section

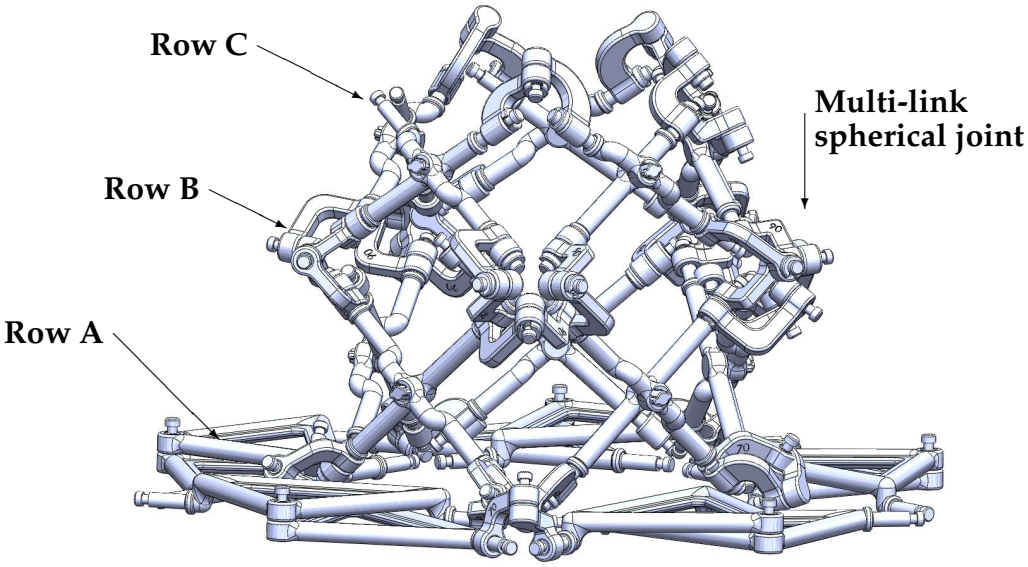


Figure 7.4: Pantomesh with rigid multi-link spherical mechanism.

which replaces the rigid multi-link spherical joint with a compliant one.

7.3 Compliant Multi-Link Spherical Joints

As opposed to rigid-body mechanisms, compliant mechanisms owe either some or all of their mobility to the deflection of flexible members [5]. The primary advantage of using compliant joints for a multi-link spherical joint is the reduction of the total part count, and therefore reduced assembly costs.

In addition to the compliant joints described in the introduction, similar compliant joints have been developed that use *distributed compliance*, rather than the focused flex zones suggested by the design shown in Figure 7.2a. Otherwise, the design goals are similar: high compliance in three rotational degrees of freedom and low compliance in all translational degrees of freedom.

With many multi-link spherical joints in a pantomesh, it is important to minimize the resistance to motion of each compliant joint to reduce the total actuation force of the assembled pantomesh. Also, a large displacement must be available; therefore, distributing the compliance over a long, thin member is better than concentrating the compliance along a line or point. Concentrated compliance also leads to concentrated stresses, thereby limiting the range of motion due to the stress or

strain limit of the material.

Two distributed-compliance designs are presented: 1) a two-sided compliant joint that has a symmetric response to applied forces and 2) a one-sided compliant joint for container-style pantomeshes (see Chapter 8) where a double-side joint could interfere with operation. Both designs were evaluated on the following criteria (see Figure 7.5 for reference):

1. High rotational compliance θ/M. θ is along the same axis as M.
2. Low translational compliance δ/F. δ is along the same direction as F.
3. High range-of-motion θ_{max}, as determined by maximum allowable stress.

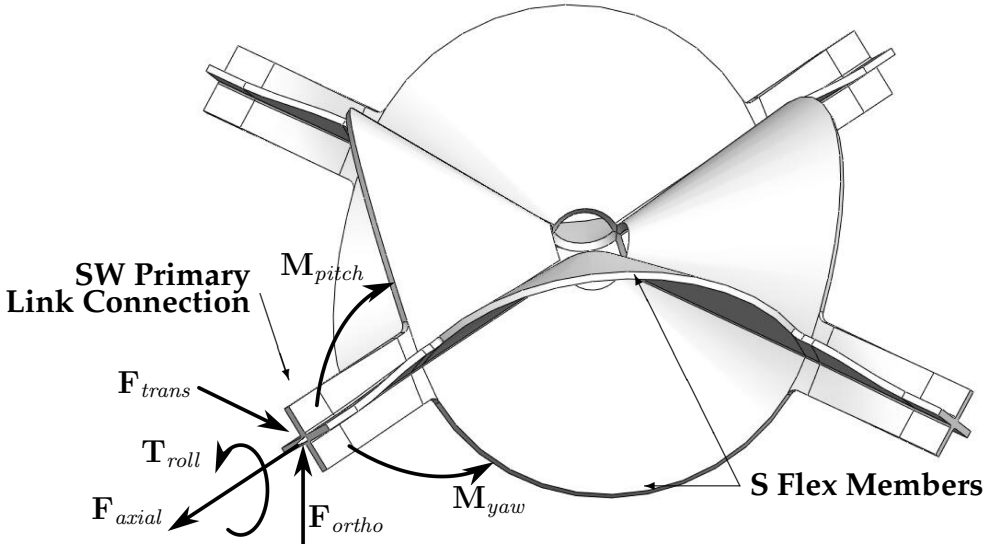


Figure 7.5: Double-sided compliant multi-link spherical joint.

A double-sided compliant multi-link spherical joint is shown in Figure 7.5. The primary links are connected to four "x"-shaped members that allow some axial twist. Those axial twist members are connected to adjacent members with *flex members* that are designed to allow motion in certain directions and resist motion in others. They are also designed to distribute stress along the entire flex member, thereby increasing the joint's range of motion.

A one-sided compliant multi-link spherical joint is shown in Figure 7.6. Instead of two identical flex members as used in the symmetric two-sided model, only one

flex member is used. This approach has an advantage for container-type applications that surround a space such as the breast stabilization system described in Chapter 8. The primary disadvantage of this system is the non-symmetric error and range-of-motion figures for motion studies.

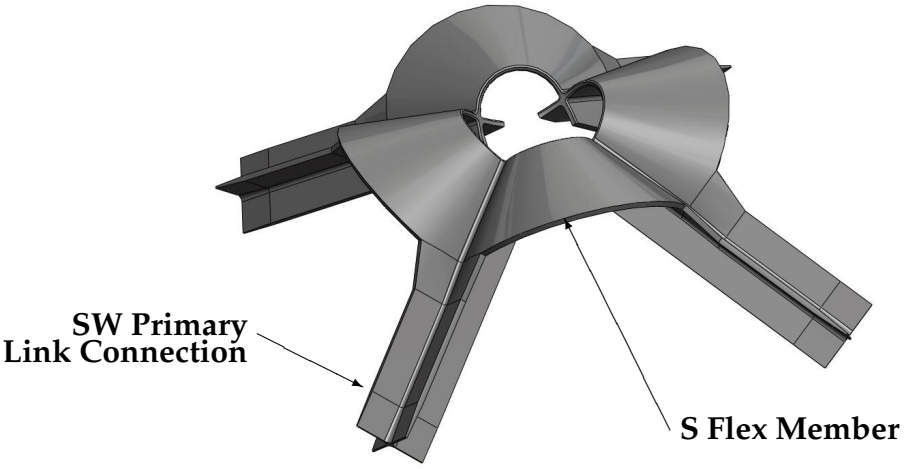


Figure 7.6: One-sided compliant multi-link spherical joint.

Several design parameters may be varied to optimize the joint design, including inner web thickness, outer web thickness, inner web radius, web width, and web arc angle. Both the single-sided and double-sided designs were given similar input parameters.

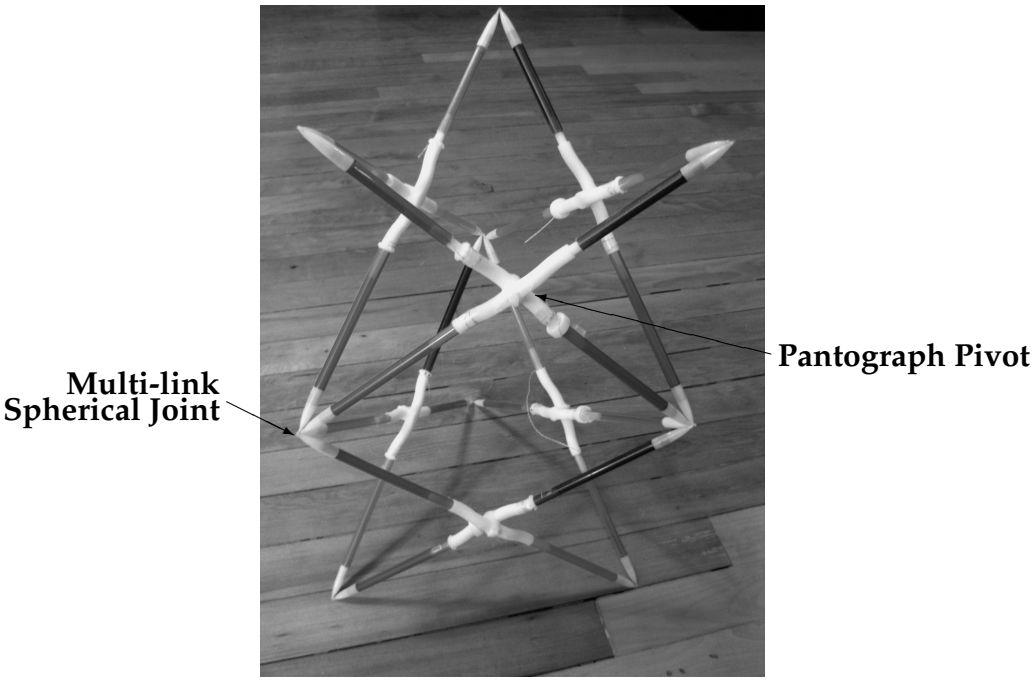
The results of both optimized designs are shown in Table 7.1. A finite element analysis (FEA) was performed using the COSMOSworks package integrated with the SolidWorks (Dassault, Concord, MA) computer-aided design system. Four different loading conditions were measured: an axial force, an transverse force, an orthogonal force, and an axial twist, each applied to the link connection. Four total displacements were measured: the angular displacement of the link as roll (γ), pitch (β), or yaw (α); and ϵ, the absolute error. The distance ϵ is measured as the projected error of a central point, which is actually a phantom node generated in the FEA strictly for this measurement. Furthermore, for each loading the maximum von Mises stress was used, along with the angular displacement, to linearly extrapolate a maximum displacement for each rotational DOF. This assumes the use of rapid prototyping material DuraForm EX (3D Systems, Rock Hill, SC), with an elastic modulus of 1310 MPa and a yield strength of 37 MPa [37].

Measurement	Single-sided	Double-sided	Units
Torsional Compliance $\left(\frac{\alpha}{T_{roll}}\right)$	0.991	0.117	$^{\text{rad}}/_{\text{N}\cdot\text{mm}}$
Orthogonal Compliance $\left(\frac{\beta}{F_{ortho}}\right)$	0.565	0.449	$^{\text{rad}}/_{\text{N}}$
Transverse Compliance $\left(\frac{\gamma}{F_{trans}}\right)$	1.46	0.0619	$^{\text{rad}}/_{\text{N}}$
Axial Stiffness $\left(\frac{F_{axial}}{\epsilon}\right)$	4.47	24.9	$^{\text{N}}/_{\text{mm}}$
Orthogonal Stiffness $\left(\frac{F_{ortho}}{\epsilon}\right)$	1.75	0.295	$^{\text{N}}/_{\text{mm}}$
Transverse Stiffness $\left(\frac{F_{trans}}{\epsilon}\right)$	0.12	30.4	$^{\text{N}}/_{\text{mm}}$
Maximum Yaw (α_{max})	26.2	28.5	deg
Maximum Pitch (β_{max})	18.2	7.09	deg
Maximum Roll (γ_{max})	7.09	3.67	deg
Yaw / Err $\left(\frac{\gamma}{M_{yaw}}\right)\left(\frac{F_{ortho}}{\epsilon}\right)$	7.24×10^{-3}	3.52×10^{-4}	$^{\text{rad}}/_{\text{m}^2}$
Pitch / Err $\left(\frac{\beta}{M_{pitch}}\right)\left(\frac{F_{trans}}{\epsilon}\right)$	1.80×10^{-4}	2.62×10^{-1}	$^{\text{rad}}/_{\text{m}^2}$
Roll / Err $\left(\frac{\alpha}{T_{roll}}\right)\left(\frac{F_{axial}}{\epsilon}\right)$	4.43	2.90	$^{\text{rad}}/_{\text{m}^2}$

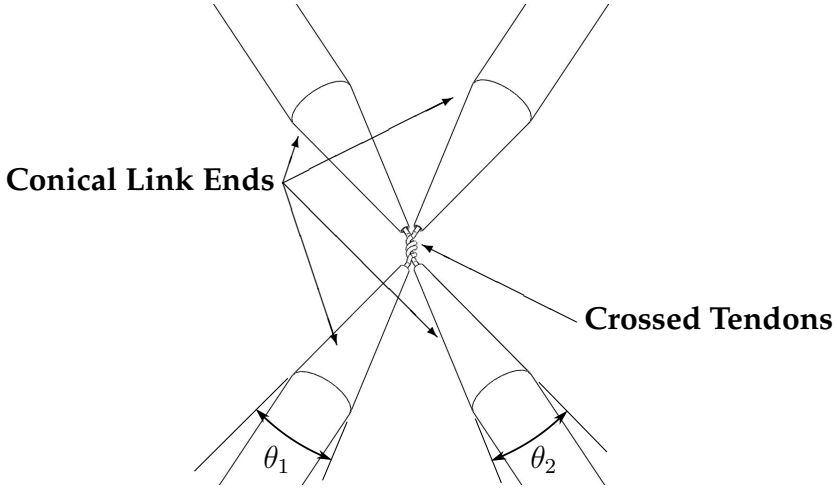
Table 7.1: Relative compliance in several degrees of freedom for chosen designs.

As shown in Table 7.1, the single-sided design, as analyzed, is more flexible and therefore has greater maximum allowable angular displacement. All the numbers in the table are better when they are *larger*, i.e. rotational compliance, displacement stiffness, and range of motion are all desired characteristics. The final three terms relate how much rotational displacement can be achieved with minimal error. These particular figures show where the displacement per error for the single-sided design is three orders of magnitude less in the pitch direction compared to the double-sided design. This means the error is relatively smaller in the double-sided case. Each design may be modified to allow greater flexibility at the cost of greater error.

Care must be taken to avoid component failure due to yield stress. Motion limits should be built into the design to avoid this failure.



(a) Photograph of a Crossed-Tendon Pantomesh Assembly



(b) Drawing of a Crossed-Tendon Spherical Joint

Figure 7.7: Crossed-tendon spherical joint.

7.4 Crossed-Tendon Multi-Link Spherical Joints

A compromise solution for multi-link spherical joints uses crossed tendons and pointed link ends to achieve reasonable performance with minimal cost. In effect, this would be a *hybrid* system, with rigid link members joined with separate, highly-flexible compliant members (see Figure 7.7a).

In theory, each link would taper to an infinitely thin section that is infinitely flexible, which then broadens to form another link or broadens in multiple directions to form multiple link. In practice, this can be accomplished connecting two links with a tendon or string through their centers and pulled tight as shown in Figure 7.7b. The tendon acts as an highly flexible member between two links, allowing three rotational degrees of freedom. The tightly-held tendon and the collision between the two link tips restrict all translational motion, therefore completing the requirements for a spherical joint.

Two of these link pairs may be combined by twisting the tendons, thereby creating a four-link spherical joint (see Figure 7.7b). The range of motion for tendon-joined multi-link spherical joint is limited by the angle of the link taper. The longer the taper, the closer that two adjacent links may approach each other. For two taper angles θ_1 and θ_2, the minimum angle between those corresponding links is $\frac{1}{2}(\theta_1 + \theta_2)$ (see Figure 7.7b). However, the taper length is limited by the distance to the pantograph pivot.

The error of a crossed-tendon multi-link spherical joint is limited to how taught the tendons are tied and how narrow the conical tips become. Tendon slack causes translational error at a maximum of the distance between the link tips. The link tips cannot come to a sharp point because the tendon must be routed through its center. Therefore, the theoretical sharp tips of two joined links may not always coincide even for perfectly tight tendons, therefore introducing further error.

7.5 Summary

Pantomeshes require multi-link spherical joint where four scissor pairs are joined at a single point. Three different approaches were described: a conventional rigid joint design, a compliant design, and a crossed-tendon design. A comparison of those designs is summarized in Table 7.2.

Joint Type	Range of Motion	Durability	Joint Error	Production Cost	Compact
Rigid	+	+	+	-	-
Compliant	-	-	-	+	0
Crossed-Tendon	+	+	0	0	+

Table 7.2: Comparison of multi-link spherical joints.

Crossed-tendon joints provide the best performance for most applications, comparing favorably to the other designs in all performance characteristics except joint error, where the rigid design may be fabricated with high precision. The compliant joint, as designed, is most useful in high-volume applications where cost is of utmost importance.

Chapter 8

Breast Cradle

This chapter describes a breast stabilization device for minimally-invasive breast cancer procedures, especially those under real-time magnetic resonance imaging (MRI) guidance. Breast cancer is a major health care issue in the United States and worldwide. Worldwide, over 1.15 million new cases of breast cancer and 410,000 deaths occur each year [38], while in the U.S. over 212,000 new cases of breast cancer will be diagnosed in women, and over 40,000 breast cancer deaths will occur. That makes breast cancer the second leading cause of cancer death in women in the U.S. [39]. Early diagnosis and treatment of breast cancer is a major factor for efficient patient management; however, a highly efficient approach to diagnose and characterize breast cancer is not yet available [40]. To this end, breast MRI has shown great promise in early detection of breast cancer. Recent evidence has suggested that breast MRI is the best screening modality for those carrying BRCA1 and/or BRCA2, genes know to correlate highly with the incidence of breast cancer, for women 25 years or older [41].

A breast needs to be stabilized during a minimally-invasive procedure such as a core needle biopsy. A biopsy is needed to have a definitive diagnosis of a particular cancer for treatment planning. A particular example of such a need is a robotic breast biopsy method [42] performed with magnetic resonance imaging (MRI) guidance.

This chapter proceeds as follows. Current stabilization methods are detailed in the next section, followed by a section detailing a new breast stabilization device for interventional procedures and guidelines for design. Finally, a dimensioned

example is created using those guidelines.

8.1 Current State of the Art

Nearly all current breast stabilization methods use bilateral compression plates such as those in Figure 8.1. A major drawback of existing breast stabilization systems is the limited orientation of the compression plates. Most of the devices provide breast stabilization by compression along a specific orientation, usually medial-lateral, which may not be optimal since they often do not transverse the shortest path in the tissue [43]. In practice, oblique orientation of compression, as compared to standard medial-lateral or posterior-anterior orientations, and oblique trajectories, as compared to trajectories perpendicular to the compression plane, are better operation strategies for several cases [43–46].

Another major disadvantage of bilateral compression plates is the restriction of blood flow. Dynamic contrast enhanced breast MRI uses the perfusion of contrast through a tumors blood vessels. If flow through these vessels is restricted by extreme compression, the diagnostic capabilities of breast MRI may be severely limited. In cases where only MR imaging is to be performed without interventions, the breast is often only lightly held [47].

The final disadvantage to bilateral compression plates is patient discomfort, which can be intense. Neither MRI nor ultrasound requires the intensity of breast compression like that of x-ray mammography. An improved stabilization apparatus would decrease patient discomfort and anxiety.

8.2 Design Approach

The breast cradle described here provides equal compression around the breast, as opposed to typical bilateral compression plates used in both manual and robotic biopsy systems [43–46]. The cradle allows full 360° access around the breast *after* it has been compressed.

To address these issues, an axisymmetric breast stabilization mechanism, such as the one shown in Figure 8.2, has been developed. This example is actuated by pulling the top portion of the cradle. The device is axisymmetric because of the quasi-axisymmetric shape of the breast. In general, the device must provide consistent and comfortable breast stabilization, with a known applied pressure when

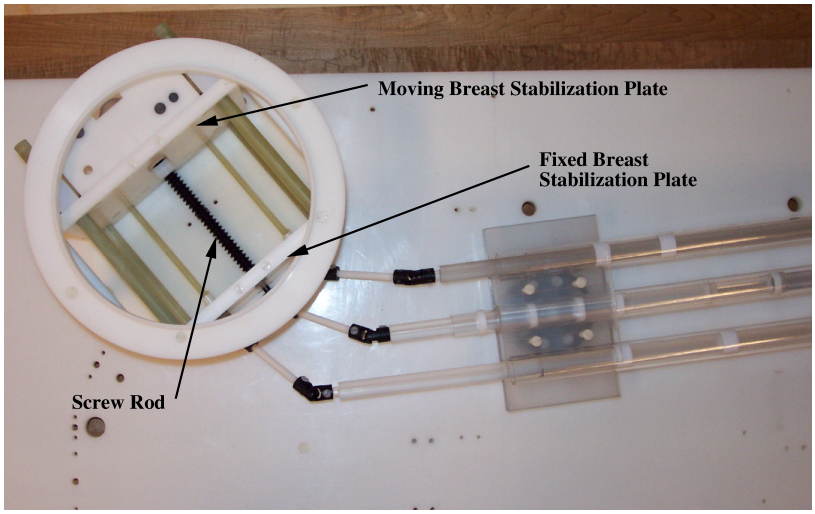
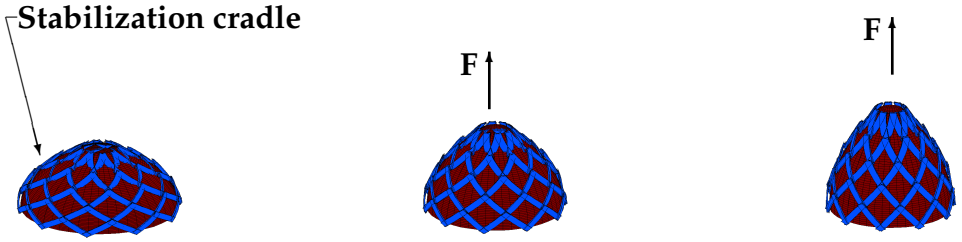


Figure 8.1: An example of bilateral compression plates to provide breast stabilization as presented in B. Larson's masters thesis [42].

actuated. Several sizes of breast cradles may be developed for different anatomies. Each different size may correspond with their own MRI coil, ensuring very close placement.



(a) Initial (relaxed) configuration

(b) Partial actuation

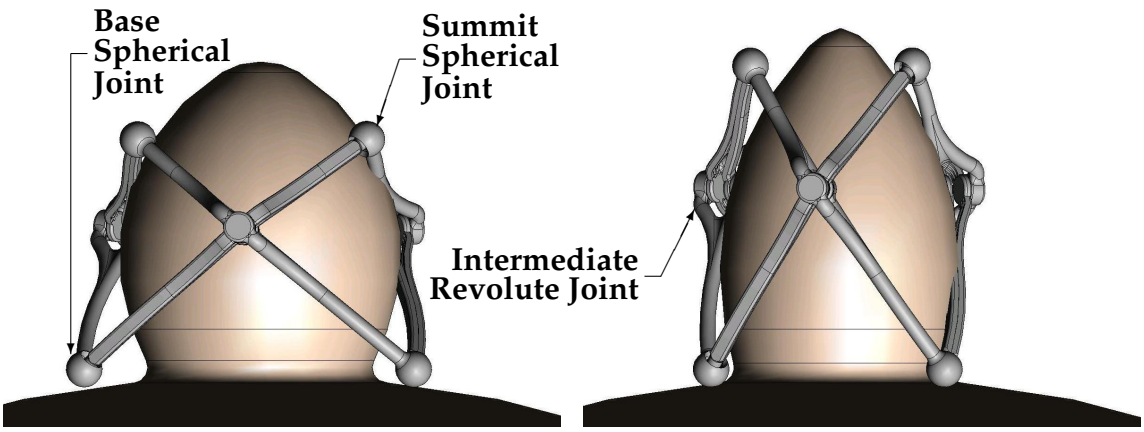
(c) Full actuation

Figure 8.2: Proposed breast stabilization device.

This device will entail the use of a pantomesh closed in a ring, as described in the next sections. These pantomesh rings may be made with solid links and joints or using compliant links and joints.

8.3 Closed Pantomesh Rings

The general form of the device is as shown in Figure 8.3, with several pantograph elements connected in a ring around the breast. Each pantograph element has two links that meet at an intermediate revolute joint. They are joined to neighboring



(a) Initial (relaxed) configuration (b) Final (compressed) configuration

Figure 8.3: Single layer pantograph mesh breast cradle.

elements with spherical joints at the base and summit. The *base* joints are placed on the chest wall, while the *summit* joints are close to the nipple. With radial forces applied to the base joints, the linkage collapses to compress the breast and pulls it away from the chest wall.

The first step in designing such a device is to determine the desired initial and final dimensions. A schematic of a breast profile is shown in Figure 8.4.

The initial shape of the device should be designed such that it easily fits over a free-hanging breast of a prone patient. The final shape of the device should be based on light compression, therefore the final volume should be slightly less than the initial volume. To properly design the dimensions of the pantomesh, the profiles of prone, free-hanging breasts must be determined.

To design the actual link dimensions, the *kinematics*, or motion of the links, must be understood. Assuming the layout shown in Figure 8.3, the radial forces are applied equally and the base spherical joints are constrained to be coplanar; held close to the chest wall. In this case, the motion of the summit spherical joints is given by:

$$r_{summit} = (f/e) r_{base} \tag{8.1}$$

$$z^2_{summit} = L^2 - r^2_{base}\left[1 + (f/e)^2 - 2(f/e)\cos\left(\tfrac{360^\circ}{N}\right)\right] \tag{8.2}$$

Where r is the radial distance from the central axis and z is the distance from the chest wall. The parameter f/e is the ratio of the distance from the intermediate joint

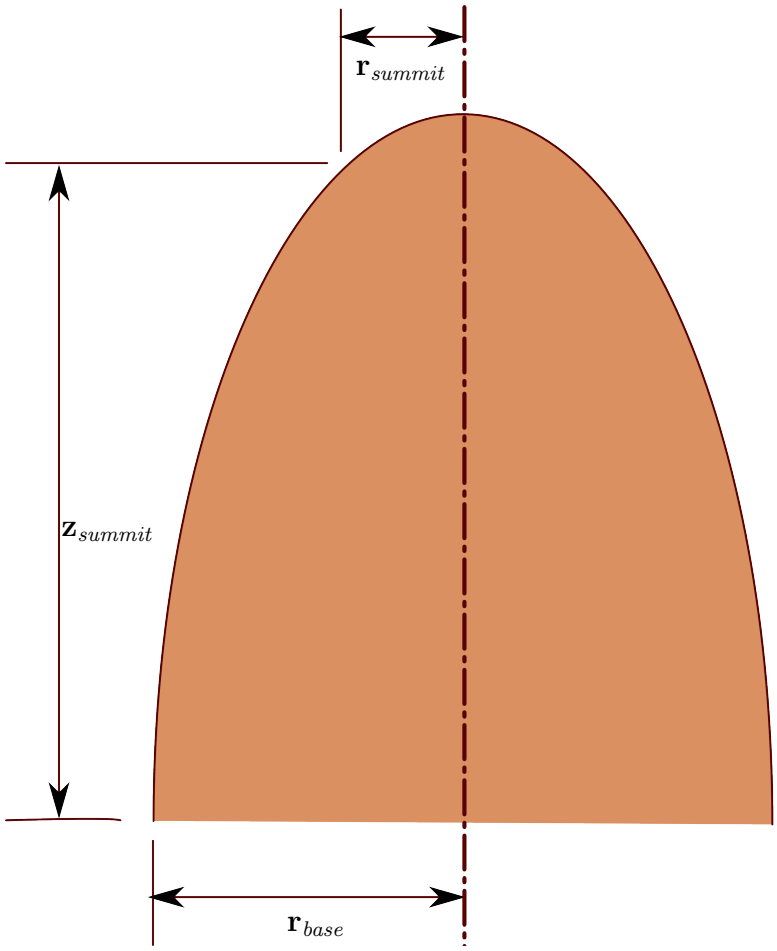


Figure 8.4: Schematic profile representation of a breast.

to the summit joint (f) and the distance from the intermediate joint to the base joint (e) for each link that is of length L. In the case of straight-link elements $e + f = L$. N is the number of scissor pairs.

The link itself may be curved as shown in Figure 8.3 to provide better form-fitting with the breast. To maintain the kinematics of the linkage, the link lengths and collapsing ratio f/e should remain the same, but the location of the intermediate revolute joint may occur anywhere along its axis. The collapsing ratio f/e is measured from where the revolute joint axis crosses the line between the base and summit spherical joints.

8.4 Designed Example

A single cradle was designed to fit a sample breast. Using published breast MRI images take from free-hanging breasts, an image analysis program was used to

"measure" the profile of breast to create a best fit with the cradle. The measured dimensions were:

$$\begin{aligned} r_{base,initial} &= 75\text{mm} \\ z_{summit,initial} &= 65\text{mm} \\ r_{summit,initial} &= 20\text{mm} \end{aligned}$$

Using Eqn. (8.1), we find that

$$f/e = \frac{r_{summit,initial}}{r_{base,initial}} = 0.27 \tag{8.3}$$

Similarly, by rearranging Eqn. (8.2) we can find L:

$$L = \sqrt{z_{summit}^2 + r_{base}^2 \left[1 + (f/e)^2 - 2(f/e)\cos\left(\frac{360^\circ}{N}\right)\right]} = 110\text{mm} \tag{8.4}$$

Assuming a parabolic profile, the theoretical apex of the parabola is taken to be $z_{max,initial} = 82$mm. The volume of a paraboloid of revolution is:

$$V = 2\pi a z_{max}^2 \tag{8.5}$$

For the initial position, $a_{initial} = 27$mm, and therefore $V_{initial} = 5.0 \times 10^5\text{mm}^3$. To provide light compression, a volume reduction of 5% is specified. To pull the breast well away from the chest wall, a theoretical apex of the final parabolic shape is chosen to be $z_{max,final} = 120$mm. Using the volume equation above, we obtain $a_{final} = 5.2$mm. The final desired summit dimensions are measured on this parabola; therefore, assuming that $r_{base,final} = 50$mm:

$$\begin{aligned} r_{summit,final} &= (f/e) r_{base,final} && = 5.3\text{mm} \\ z_{summit,final} &= \sqrt{L^2 - r_{base,final}^2 \left[1 + (f/e)^2 - 2(f/e)\cos\left(\frac{360^\circ}{N}\right)\right]} && = 94\text{mm} \end{aligned}$$

To summarize, the dimensions of each link are $L = 110\text{mm}, e = 87, f = 23\text{mm}$. As mentioned in the previous section, curved links may be used to match the curved shape of the breast as long as the pivot axis intersects the link span at the proper ratio. A schematic of such a curved link is shown in Figure 8.5.

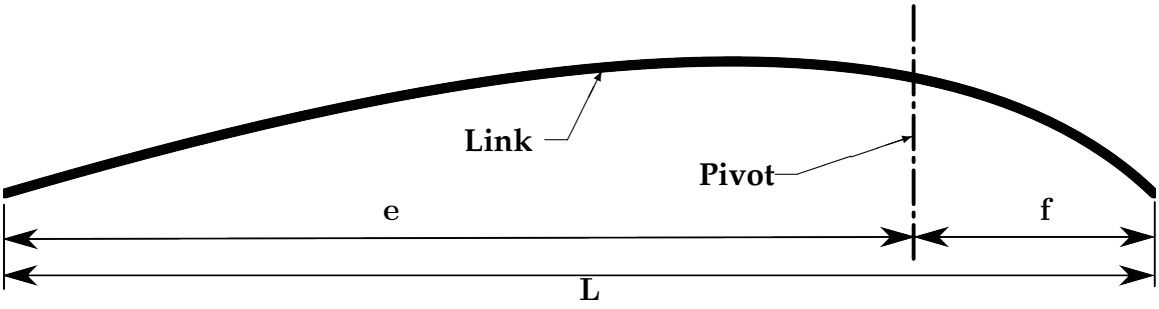


Figure 8.5: Schematic of a curved link with a set pivot ratio.

8.5 Discussion

While the linkage in Figure 8.3 is conceptually simple, there may be better arrangements. For example, multiple rows of scissor pairs may be connected to provide a more complex shape profile. Such multi-row pantomeshes also provide the extra benefit of *enforcing* the shape of the linkage by limiting the mobility to a single *degree-of-freedom*. Single-row pantomesh rings, however, have $(N-2)$ degrees of freedom, which allows three-element pantomeshes have a single DOF, but any more would require more inputs to determine the shape.

8.6 Conclusions

In summary, a novel breast stabilization system has been developed to provide radial compression force to the breast while allowing access for percutaneous interventions. The device uses scissor-pair rings, which collapse on an axis while elongating along that access, thereby pulling the breast away from the chest wall. Equations for designing the dimensional parameters of the device are given so that several sizes can be made to accommodate different breast sizes. Finally, a dimensioned example was created to meet chosen specifications.

Chapter 9

Conclusions

This work provides a complete description of the *pantomesh*, including properties, synthesis, and an application. The definition of a pantomesh, and existing pantomeshes are presented in Chapter 1, along with the six different types of pantograph elements as shown in Table 1.1 and Figure 1.4. While a small sub-section of the planar elements have been used in earlier pantomeshes, an thorough investigation of all six types has not been presented elsewhere.

A detailed examination of the kinematics of pantograph elements is presented in Ch. 2. Special note is taken of the relative motion of the bounding quadrliateral edges, defined by the link endpoints. All relations between these edge lengths is required to determine the mobility of pantopatches in Ch. 3. The mobility of an entire closed pantopatch may be ensured by breaking up an assembled pantomesh into 2x2 pantopatches and determining their mobility. While the mobility requirements for the straight-link pantopatches were quite similar, the bent-link cases increased in complexity as the elements themselves increased in complexity, culminating in a somewhat complicated summary of requirements for skew, angled-pivot bent-link pantopatches. The mobility chapter ended with a brief discussion of a graphical method to determine mobility.

A novel method for determining mobility of a variety of mechanisms, including pantomeshes, is presented in Ch. 4. Mobility is determined by breaking links and joints into the polygonal elements of edges and vertices. The basic form of this formula, $\mathcal{M} = 3V - E - 6$, is valid for a number of situations where the commonly-known Grübler-Kutzbach mobility formula is insufficient in determining the mo-

bility of spatial linkages.

Given these analyses of how pantomeshes operate, Chapter 5 shows how a pantomesh may be constructed graphically using computer-aided design software. This new technique shows a way for a designer to use a solid-modeling package to define some desired geometry while using graphical constraints to construct the remaining geometry. Furthermore, Chapter 6 shows how pantomeshes may be constructed to fit a shape specification using numerical optimization techniques. Two examples of numerical synthesis are presented: one for a single-position, planar straight-link pantomesh, and another for a two-position, skew, bent-link pantomesh.

Further practical considerations are discussed in Ch. 7, where multi-link spherical joints are discussed. Multi-link spherical joints are necessary to join together more than two pantograph elements to form a pantomesh. Several existing joints are shown, as well as novel compliant joints and a crossed-tendon joint.

Finally, a breast cradle is developed and examined in Chapter 8. The breast cradle provides uniform radial compression of the breast to provide stabilization during cancer-related procedures.

Bibliography

[1] J. Q. Brown, "Trolley," Jul 5 1904. U.S. Patent No. 764,224.

[2] C. Scheiner, *Pantographice seu Ars delineandi res quaslibet per parallelogrammum lineare seu cavum, mechanicum, mobile*. Rome: Ludovico Grigani, 1631.

[3] A. Kaveh and A. Davaran, "Analysis of pantograph foldable structures," *Computers & Structures*, vol. 59, no. 1, p. 131, 1996.

[4] G. N. Sandor and A. G. Erdman, *Advanced Mechanism Design: Analysis and Synthesis*. Prentice-Hall, Inc, 1984.

[5] L. L. Howell, *Compliant Mechanisms*. New York: John Wiley and Sons, 2001.

[6] A. B. Kempe, "On conjugate four-piece linkages," *Proceedings of the London Mathematical Society*, vol. s1-9, pp. 133–149, November 1 1877.

[7] C. Hoberman, "Reversibly expandable doubly-curved truss structure," Jul. 24 1990. U.S. Patent No. 4,942,700.

[8] Z. You and S. Pellegrino, "Foldable bar structures," *International Journal of Solids and Structures*, vol. 34, no. 15, pp. 1825–1847, 1997.

[9] K. Wohlhart, "Double-chain mechanisms," in *IUTAM-IASS Symposium on Deployable Structures: Theory and Applications* (S. Pellegrino and S. D. Guest, eds.), (Netherlands), pp. 457–466, Kluwer Academic Publishers, 2000.

[10] J. B. Fenby, "Camp stool," 1881. U.S. Patent No. 244,215.

[11] Z. You, "Deployable structures for masts and reflector antennas," 1994.

[12] G. R. Luckey, "Nesting three dimensional lazy tong structure," 1972. U.S. Patent No. 3,672,104.

[13] J. Patel and G. K. Ananthasuresh, "A kinematic theory for planar hoberman and other novel foldable mechanisms," in *Proceedings of the ASME 2006 International Design Engineering Technical Conferences*, pp. 1–9, ASME, September 10-13, 2006 2006.

[14] C. Hoberman, "Hoberman designs, home page," 2006.

[15] E. P. Pinero, "Materia-estructa-forma," *Hogar y arquitectura*, vol. 40, pp. 24–30, 1962.

[16] F. Escrig, "Expandable space structures," *Space Structures*, vol. 1, pp. 79–91, 1985.

[17] O. F. Maclaren, "Seating structures," 1964. U.S. Patent No. 3,124,387.

[18] T. Hermanson, "Umbrella canopy frame and staff construction," 1980. U.S. Patent No. 4,193,415.

[19] L. K. Selden, "Umbrella," 1859. U.S. Patent No. 23,503.

[20] C. J. Gantes, J. J. Connor, and R. D. Logcher, "Structural analysis and design of deployable structures," *Computers & Structures*, vol. 32, no. 3/4, pp. 661–669, 1989.

[21] K. Atake, "Framework structure," June 9, 1998 1986. U.S. Patent No. 5,761,871.

[22] C. Hoberman, "Radial expansion/retraction truss structures," Jun. 18 1991. U.S. Patent No. 5,024,031.

[23] *CRC Standard Mathematical Tables and Formulae*. Chapman & Hall/CRC, 2002.

[24] M. Grübler, *Getriebelehre: eine Theorie des Zwanglaufes und der ebenen Mechanismen*. Berlin: Springer, 1917.

[25] K. Kutzbach, "Mechanische leitungsverzweigung, ihre gesetze und anwendungen," *Maschinenbau, der Betrieb*, vol. 8, pp. 710–716, 1929.

[26] A. G. Erdman and G. N. Sandor, *Mechanism Design: Analysis and Synthesis*, vol. 1. Prentice-Hall, Inc, 1991.

[27] N. Altshiller-Court, *Modern Pure Solid Geometry*. New York: Macmillan, 1935.

[28] L. N. M. Carnot, *Géométrie de position*. Paris: J.B.M. Duprat, 1803.

[29] L. N. M. Carnot, *Mémoire sur la relation qui existe entre les distances respectives de cinq points quelconques pris dans l'espace: suivi d'un Essai sur la théorie des transversales*. Paris: Courcier, 1806.

[30] C. W. Wampler, B. T. Larson, and A. G. Erdman, "A new mobility formula for spatial mechanisms," in *ASME 2007 International Design Engineering Technical Conference and Computers and Information in Engineering Conference*, ASME, Sept. 4-7 2007.

[31] E. Polak, *Computational methods in optimization: a unified approach*, vol. 77. Academic Press, 1971.

[32] L. Piegl and W. Tiller, *The NURBS book*. Berlin: Springer-Verlag, 1995.

[33] G. J. Hamlin and A. C. Sanderson, "A novel concentric multilink spherical joint with parallel robotics applications," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1267–1272, IEEE, 05/08/1994 - 05/13/1994 1994.

[34] G. J. Hamlin, "Concentric joint mechanism," 1997. U.S. Patent No. 5,657,584.

[35] P. Bosscher and I. Ebert-Uphoff, "A novel mechanism for implementing multiple collocated spherical joints," vol. 1, pp. 336–341, 2003.

[36] B. P. Trease, Y.-M. Moon, and S. Kota, "Design of large-displacement compliant joints," *Journal of Mechanical Design*, vol. 127, pp. 788–798, July 2005 2005.

[37] 3D Systems, "Duraform EX plastic," 23 Apr 2007. Available at http://www.3dsystems.com.

[38] D. M. Parkin, F. Bray, J. Ferlay, and P. Pisani, "Global cancer statistics, 2002," *CA: A Cancer Journal for Clinicians*, vol. 55, pp. 74–108, Mar 2005.

[39] A. Jemal, T. Murray, E. Ward, A. Samuels, R. C. Tiwari, A. Ghafoor, E. J. Feuer, and M. J. Thun, "Cancer statistics, 2005," *CA: A Cancer Journal for Clinicians*, vol. 55, pp. 10–30, Jan 2005.

[40] S. E. Harms, "MRI in breast cancer diagnosis and treatment," *Current problems in diagnostic radiology*, vol. 25, pp. 193–215, 1996.

[41] E. Warner, D. B. Plewes, K. A. Hill, P. A. Causer, J. T. Zubovits, R. A. Jong, M. R. Cutrara, G. DeBoer, M. J. Yaffe, S. J. Messner, W. S. Meschino, C. A. Piron, and S. A. Narod, "Surveillance of BRCA1 and BRCA2 mutation carriers with magnetic resonance imaging, ultrasound, mammography, and clinical breast examination," *JAMA*, vol. 292, pp. 1317–1325, 9/15 2004.

[42] B. T. Larson, N. V. Tsekos, A. G. Erdman, E. Yacoub, P. V. Tsekos, and I. G. Koutlas, "Design of an MRI-compatible stereotactic device for minimally invasive interventions in the breast," *Journal of Biomechanical Engineering*, vol. 126, pp. 458–465, August 2004.

[43] U. Fischera, L. Kopka, and E. Grabbe, "Magnetic resonance guided localization and biopsy of suspicious breast lesions," *Topics in Magnetic Resonance Imaging*, vol. 9, pp. 44–59, 1998.

[44] S. G. Orel, M. D. Schnall, R. W. Newman, C. M. Powell, M. H. Torosian, and E. F. Rosato, "MR imaging-guided localization and biopsy of breast lesions: Initial experience," *Radiology*, vol. 193, pp. 97–102, 1994.

[45] C. K. Kuhl, A. Elevelt, C. C. Leutner, J. Gieseke, E. Pakos, and H. H. Schild, "Interventional breast MR imaging: clinical use of a stereotactic localization and biopsy device," *Radiology*, vol. 204, pp. 667–675, sep 1997.

[46] W. Doler, U. Fischer, I. Metzget, D. Harder, and E. Grabbe, "Stereotaxic add-on device for MR-guided biopsy of breast lesions," *Radiology*, vol. 200, pp. 863–864, 1996.

[47] M. Garwood, July 2003. Interview by B. T. Larson and M. K. Mackenzie; Center for Magnetic Resonance Research, University of Minnesota, Minneapolis.

Appendix A

Nomenclature

Fixed Variables

These variables are fixed quantities such as link length that do not change during actuation of a pantomesh.

$\mathcal{P}$ A point in three-dimensional space, often denoted as the endpoint for a pantograph element.

$\mathfrak{X}$ The location of the pivot point for straight-link elements, otherwise the intersection of link spans in an element.

L The distance between the "lower left" and "upper right" points of a pantograph element.

M The distance between the "lower right" and "upper left" points of a pantograph element.

e The distance between the "lower left" point and the pivot of a pantograph element.

f The distance between the pivot and "upper right" points of a pantograph element.

g The distance between the "lower right" point and the pivot of a pantograph element.

h The distance between the pivot and "upper left" points of a pantograph element.

α The angle between the right-handed diagonal (L) and the right-handed lower pivot distance (e).

β The angle between the left-handed diagonal (M) and the left-handed lower pivot distance (g).

d In skew pantograph elements, the minimum distance between the two link spans.

ν The angle between the right-hand link span and the pivot axis.

ν The angle between the left-hand link span and the pivot axis.

δ The shortest distance between the right-hand link span and the pivot axis.

ϵ The shortest distance between the left-hand link span and the pivot axis.

$\tilde{L}_{A1}$ The distance between the "lower left" and "upper right" points on a quadrilateral project from an angled-pivot element.

$\tilde{M}_{A1}$ The distance between the "lower right" and "upper left" points on a quadrilateral project from an angled-pivot element.

$\tilde{e}_{A1}$ The distance between the "lower left" point and the pivot on a quadrilateral project from an angled-pivot element.

$\tilde{f}_{A1}$ The distance between the pivot and "upper right" point on a quadrilateral project from an angled-pivot element.

$\tilde{g}_{A1}$ The distance between the "lower right" point and the pivot on a quadrilateral project from an angled-pivot element.

$\tilde{h}_{A1}$ The distance between the pivot and "upper left" point on a quadrilateral project from an angled-pivot element.

Actuation Variables

$\boldsymbol{j_{A1}}$ The "bottom" edge of the bounding quadrilateral (or tetrahedron) for pantograph element $A1$.

$\boldsymbol{k_{A1}}$ The "left" edge of the bounding quadrilateral (or tetrahedron) for pantograph element $A1$.

$\boldsymbol{k_{A2}}$ The "right" edge of the bounding quadrilateral (or tetrahedron) for pantograph element and also the "left" edge of the bounding quadrilateral (or tetrahedron) for pantograph element $A2$.

$\boldsymbol{j_{B1}}$ The "top" edge of the bounding quadrilateral (or tetrahedron) for pantograph element $A1$ and also the "bottom" edge of the bounding quadrilateral (or tetrahedron) for pantograph element $B1$.

$\boldsymbol{\gamma}$ The angle between the bottom edge of the bounding quadrilateral (or tetrahedron) and its right-handed diagonal.

$\boldsymbol{\psi}$ The angle between the bottom edge of the bounding quadrilateral (or tetrahedron) and its left-handed diagonal.

$\boldsymbol{\theta_s}$ The angle between the right-handed lower pivot distance (e) and the left-handed lower pivot distance (e); also the "south" of the four angles made by the four link edges that meet at the pivot.

$\boldsymbol{\theta_w}$ The angle between the right-handed lower pivot distance (e) and the left-handed upper pivot distance (h); also the "west" of the four angles made by the four link edges that meet at the pivot.

$\boldsymbol{\theta_e}$ The angle between the left-handed lower pivot distance (e) and the right-handed upper pivot distance (f); also the "east" of the four angles made by the four link edges that meet at the pivot.

$\boldsymbol{\theta_w}$ The angle between the right-handed upper pivot distance (f) and the left-handed upper pivot distance (h); also the "north" of the four angles made by the four link edges that meet at the pivot.

$\boldsymbol{\mathfrak{R}}$ The infinite-length line that defines the rotation axis for the pantograph element.

$\boldsymbol{\mathfrak{X}}$ The intersection of the diagonals created by the bounding quadrilateral of a planar pantograph element.

$\boldsymbol{k^{\star}_{A2/A1}}$ The "right" edge of a flattened skew quadrilateral for pantograph element $A1$.

$\boldsymbol{k^{\star}_{A2/A2}}$ The "left" edge of a flattened skew quadrilateral for pantograph element $A2$. Does not necessarily equal $\boldsymbol{k^{\star}_{A2/A1}}$.

$\boldsymbol{j}^{\star}_{B1/A1}$ The "top" edge of a flattened skew quadrilateral for pantograph element $A1$.

$\boldsymbol{j}^{\star}_{B1/B1}$ The "bottom" edge of a flattened skew quadrilateral for pantograph element $B1$. Does not necessarily equal $\boldsymbol{j}^{\star}_{B1/A1}$.

$\tilde{\boldsymbol{k}}_{A2/A1}$ The "right" edge of a quadrilateral projected from angled-pivot element $A1$.

$\tilde{\boldsymbol{j}}_{B1/A1}$ The "top" edge of a quadrilateral projected from angled-pivot element $A1$.

$\gamma^{\star}$ The angle between the bottom edge of the a flattened skew quadrilateral and its right-handed diagonal.

$\psi^{\star}$ The angle between the bottom edge of the a flattened skew quadrilateral and its left-handed diagonal.

Appendix B

Numerical Analysis Code for Shape Synthesis

B.1 Introduction

Below is the program listing for a pantomesh synthesis routine, using optimization techniques. The program was run using Freemat v.4.0, an open-source MATLAB clone.

B.2 Program Listing

```
1  function [varargout] = MeshSolver(varargin)
2
3      global numrows numcols numpos numiters maxiters ...
           criteria thread
4      numrows = 3;
5      numcols = 12;
6      numpos = 2;
7      num_t_rows = 14;
8      numiters = 1;
9      maxiters = 1000;
10     numresets = 0;
11     totalresets = 0;
12     maxresets = 3;
13     criteria = 1e-6;
```

```
14
15      tic;
16      beep = wavread('ding.wav');
17      done = wavread('tada.wav');
18
19      fignum(1) = 1;
20      fignum(2) = 2;
21      errtxt(1:3)   = {'dist(1) ', 'dist(2) ', 'dist(3) '};
22      errtxt(4:12)  = {'mob(2)  ', 'mob(3)  ', 'mob(4)  ', '...
          mob(5)  ', 'mob(6)  ', 'mob(7)  ', 'mob(8)  ', 'mob...
          (9)  ', 'mob(10) '};
23      errtxt(13:21) = {'mob(2)  ', 'mob(3)  ', 'mob(4)  ', '...
          mob(5)  ', 'mob(6)  ', 'mob(7)  ', 'mob(8)  ', 'mob...
          (9)  ', 'mob(10) '};
24      errtxt(22:27) = {'Ldiff(1)', 'Ldiff(2)', 'Ldiff(3)', '...
          Mdiff(1)', 'Mdiff(2)', 'Mdiff(3)'};
25      errtxt(28:33) = {'l_tri(1)', 'l_tri(2)', 'l_tri(3)', '...
          m_tri(1)', 'm_tri(2)', 'm_tri(3)'};
26      errtxt(34:39) = {'alpha(1)', 'alpha(2)', 'alpha(3)', '...
          beta(1) ', 'beta(2) ', 'beta(3) '};
27
28      [t limits] = Init();
29      printf( '%s', show(reshape(t, numcols, num_t_rows)', 't...
          ', '%12.3g') );
30       % a = threadnew();
31       % threadstart(a, 'plot_pts',0,t,h,);
32            plot_pts(t,fignum);
33      [x, y, z] = get_pts(t, numrows, numcols);
34            printf( '%s', show(x, 'x:', '%10.5g') );
35            printf( '%s', show(y, 'y:', '%10.5g') );
36            printf( '%s', show( sqrt(x.^2 + y.^2), 'r:', '...
                %10.5g') );
37            printf( '%s', show(z, 'z:', '%10.5g') );
38      [L, M, e, f, g, h, alpha, beta, dist, bside, lside] = ...
          get_links(x,y,z);
39            printf( '%s', show(L, 'L:', '%10.5g') );
```

```
40          printf( '%s', show(M, 'M:', '%10.5g') );
41          printf( '%s', show(e, 'e:', '%10.5g') );
42          printf( '%s', show(f, 'f:', '%10.5g') );
43          printf( '%s', show(g, 'g:', '%10.5g') );
44          printf( '%s', show(h, 'h:', '%10.5g') );
45          printf( '%s', show(rad2deg(alpha), 'alpha[deg]:', ...
              '%10.5g') );
46          printf( '%s', show(rad2deg(beta), 'beta[deg]:', '...
              %10.5g') );
47          printf( '%s', show(dist, 'dist:', '%10.5g') );
48          printf( '%s', show(bside, 'bside:', '%10.5g') );
49          printf( '%s', show(lside, 'lside:', '%10.5g') );
50      errorcalc = calc_error(t);
51      str = 'errorcalc:\n';
52      [r c] = size(errorcalc);
53      for (i=1:r)
54          str = [str, errtxt{i}, '\t'];
55          for (j=1:c)
56              str = sprintf([str, '%12.5g', '\t'], errorcalc(i,...
                  j));
57          end %for(j)
58          str = [str, '\n'];
59      end %for(i)
60      printf(str);
61      err(numiters) = xnrm2(errorcalc);
62      printf('Error: %12.9g\n', err(numiters));
63      steepest_prev = -gradient(t, err(numiters));
64      printf( '%s', show(reshape(steepest_prev, num_t_rows, ...
          numcols), 'Steepest Descent:', '%12.5g') );
65      conj_prev = steepest_prev;
66      [conj_alpha errorcalc] = linesearch(t, conj_prev, ...
          limits);
67      printf('conj_alpha: %10.6g\t', conj_alpha);
68      % printf( '%s', show(errorcalc, 'errorcalc', '%15.5g') ...
          );
69      t = t + conj_alpha * conj_prev;
```

```
70    printf( '%s', show(reshape(t, numcols, num_t_rows)', 't...
          ', '%12.5g') );
71    % threadwait(a);
72    % threadfree(a);
73
74    while (numiters < maxiters && err(numiters) >= criteria...
          && numresets < maxresets)
75    numiters = numiters+1;
76    err(numiters) = xnrm2(errorcalc);
77    printf('Iteration: %d \t New Error: %20.14g \t ...
          Previous Error: %20.14g \t', numiters, err(numiters),...
           err(numiters-1));
78    steepest_curr = -gradient(t, err(numiters));
79    printf('%s', show(reshape(steepest_curr, num_t_rows, ...
          numcols), 'Steepest Descent:', '%12.5g') );
80    if ( ((err(numiters-1) - err(numiters)) /err(numiters))...
          > criteria )
81       conj_beta = (steepest_curr' * (steepest_curr - ...
             steepest_prev)) / (steepest_prev' * steepest_prev)...
             ;
82       conj_beta = max(0, conj_beta);    %this should be ...
             the same test as the 'if' statement
83       conj_curr = steepest_curr + conj_beta * conj_prev;
84       numresets = 0;
85    else
86       numresets = numresets+1;
87       totalresets = totalresets+1;
88       printf('RESET no. %d, ', numresets);
89       wavplay(beep);
90       conj_curr = steepest_curr;
91       conj_beta = 0.0;
92    end %if(err>eps_off)
93    [conj_alpha errorcalc] = linesearch(t, conj_curr, ...
          limits);
94    printf('%d total resets; \t conj_beta: %10.6g \t ...
          conj_alpha: %10.6g\n', totalresets, conj_beta, ...
```

```
        conj_alpha);
95      t = t + conj_alpha * conj_curr;
96      printf( '%s', show(reshape(t, numcols, num_t_rows)', 't...
          ', '%12.5g') );
97  %     plot_pts(t,fignum);
98  %    drawnow;
99      %     show(t, x_curve, y_curve, z_curve);
100     [x, y, z] = get_pts(t, numrows, numcols);
101           printf( '%s', show(x, 'x:', '%10.5g') );
102           printf( '%s', show(y, 'y:', '%10.5g') );
103           printf( '%s', show( sqrt(x.^2 + y.^2), 'r:', '...
                %10.5g') );
104           printf( '%s', show(z, 'z:', '%10.5g') );
105     [L, M, e, f, g, h, alpha, beta, dist, bside, lside] = ...
          get_links(x,y,z);
106           printf( '%s', show(L, 'L:', '%10.5g') );
107           printf( '%s', show(M, 'M:', '%10.5g') );
108           printf( '%s', show(e, 'e:', '%10.5g') );
109           printf( '%s', show(f, 'f:', '%10.5g') );
110           printf( '%s', show(g, 'g:', '%10.5g') );
111           printf( '%s', show(h, 'h:', '%10.5g') );
112           printf( '%s', show(rad2deg(alpha), 'alpha[deg]:', ...
                '%10.5g') );
113           printf( '%s', show(rad2deg(beta), 'beta[deg]:', '...
                %10.5g') );
114           printf( '%s', show(dist, 'dist:', '%10.5g') );
115           printf( '%s', show(bside, 'bside:', '%10.5g') );
116           printf( '%s', show(lside, 'lside:', '%10.5g') );
117     str = 'errorcalc:\n';
118     [r c] = size(errorcalc);
119     for (i=1:r)
120        str = [str, errtxt{i}, '\t'];
121        for (j=1:c)
122           str = sprintf([str, '%12.5g', '\t'], errorcalc(i,...
                j));
123        end %for(j)
```

```
124         str = [str, '\n'];
125     end %for(i)
126     printf(str);
127     steepest_prev = steepest_curr;
128     conj_prev = conj_curr;
129 end %while(numiters)
130 final_show(t, err);
131 printf('Total seconds elapsed:\n %6.0f\n', toc );
132 wavplay(done);
133
134 function [x, y, z] = get_pts(t, numrows, numcols)
135           global numpos
136     % this function will contain the constraints and return...
          the coordinates
137     x = zeros(numrows+1, numcols+1, numpos);
138     y = zeros(numrows+1, numcols+1, numpos);
139     z = zeros(numrows+1, numcols+1, numpos);
140     [u, v] = get_uv(t);
141     % find all patch points
142     for (i=1:numrows+1)
143         for (j=1:numcols)
144             for (k=1:2)
145                 [x(i,j,k) y(i,j,k) z(i,j,k)] = get_patch( u(i,...
                      j,k), v(i,j,k), k );
146             end %for(k)
147         end %for(j)
148     end %for(i)
149
150     % now do the wrap-around closing...
151     for (i=1:numrows+1)
152         for (k=1:2)
153             x(i,numcols+1,k) = x(i,1,k);
154             y(i,numcols+1,k) = y(i,1,k);
155             z(i,numcols+1,k) = z(i,1,k);
156         end %for(k)
157     end %for(i)
```

```
158
159 function [r1_A r2_A r1_D r2_D z_D] = get_dims()
160     r1_A = [300 250];
161     r2_A = [240 200];
162     r1_D = [200 150];
163     r2_D = [150 100];
164     z_D =  [300 390];
165     % r1_A = [300 280];
166     % r2_A = [280 220];
167     % r1_D = [150 140];
168     % r2_D = [140 110];
169     % z_D =  [300 380];
170
171 function [x, y, z] = get_patch(u, v, pos)
172     global numrows numcols;
173     [r1_A r2_A r1_D r2_D z_D] = get_dims();
174
175         % The following numbers are a best-fit to a circle...
                based on 20 points
176     x_template = [  1    0.977492         -2.18816  -2.18816...
          0.977492  1]; %mult by major axis radius
177     y_template = [  0    1.678248  2.002597 -2.002597 ...
        -1.678248  0];  %mult by minor axis radius
178     % x_template = [  1 1    -2.2  -2.2   1      1]; %mult ...
        by major axis radius
179     % y_template = [  0 1.75  1.75 -1.75 -1.75  0];  %mult ...
        by minor axis radius
180
181         % printf('u: %8.5g \t v: %8.5g pos: %d\n', u, v, ...
              pos);
182     x_patch(1,:) = r1_A(pos)*x_template;
183     y_patch(1,:) = r2_A(pos)*y_template;
184     z_patch(1,:) = 0*x_template;
185     % x_patch(2,:) = (2*r1_A(pos)+r1_D(pos))/3*x_template;
186     % y_patch(2,:) = (2*r2_A(pos)+r2_D(pos))/3*y_template;
187     x_patch(2,:) = 1.0*r1_A(pos)*x_template;
```

```
188     y_patch(2,:) = 1.0*r2_A(pos)*y_template;
189     z_patch(2,:) = z_patch(1,:) + z_D(pos)/3;
190     % x_patch(3,:) = 1.0*r1_A(pos)*x_template;
191     % y_patch(3,:) = 1.0*r2_A(pos)*y_template;
192           % x_patch(3,:) = (r1_A(pos)+2*r1_D(pos))/3*...
                x_template;
193     % y_patch(3,:) = (r2_A(pos)+2*r2_D(pos))/3*y_template;
194     x_patch(3,:) = (r1_A(pos)+r1_D(pos))/2*x_template;
195     y_patch(3,:) = (r2_A(pos)+r2_D(pos))/2*y_template;
196     z_patch(3,:) = z_patch(1,:) + 2/3*z_D(pos);
197     x_patch(4,:) = r1_D(pos)*x_template;
198     y_patch(4,:) = r2_D(pos)*y_template;
199     z_patch(4,:) = z_patch(1,:) + z_D(pos);
200
201     bernstein_u(1,1) = (1-u)^5;
202     bernstein_u(2,1) = 5*(1-u)^4*u;
203     bernstein_u(3,1) = 10*(1-u)^3*u^2;
204     bernstein_u(4,1) = 10*(1-u)^2*u^3;
205     bernstein_u(5,1) = 5*(1-u)*u^4;
206     bernstein_u(6,1) = u^5;
207
208     bernstein_v(1,1) = (1-v)^3;
209     bernstein_v(2,1) = 3*v*(1-v)^2;
210     bernstein_v(3,1) = 3*v^2*(1-v);
211     bernstein_v(4,1) = v^3;
212
213     x = bernstein_u' * x_patch' * bernstein_v;
214     y = bernstein_u' * y_patch' * bernstein_v;
215     z = bernstein_u' * z_patch' * bernstein_v;
216           r = sqrt(x^2 + y^2);
217
218     % pause
219
220 function [L, M, e, f, g, h, alpha, beta, dist, bside, ...
      lside] = get_links(x,y,z)
221     global numrows numcols numpos;
```

```
222     L = -ones(numrows, numcols+1, numpos);
223     M = -ones(numrows, numcols+1, numpos);
224     e = -ones(numrows, numcols+1);
225     f = -ones(numrows, numcols+1);
226     g = -ones(numrows, numcols+1);
227     h = -ones(numrows, numcols+1);
228     alpha = -ones(numrows, numcols+1);
229     beta = -ones(numrows, numcols+1);
230     dist = -ones(numrows, numcols+1, numpos);
231     bside = -ones(numrows, numcols+1, numpos);
232     lside = -ones(numrows, numcols+1, numpos);
233     for (i=1:numrows)
234         for (j=1:numcols)
235             for (k=1:numpos)
236                 P_A1    = [x(i,j   ,k), y(i,j   ,k), z(i,j   ,k)];
237                 P_A2    = [x(i,j+1,k), y(i,j+1,k), z(i,j+1,k)];
238                 vect_L = [x(i+1,j+1,k) - x(i,j   ,k), y(i+1,j...
                        +1,k) - y(i,j   ,k), z(i+1,j+1,k) - z(i,j   ,k...
                        )];
239                 vect_M = [x(i+1,j   ,k) - x(i,j+1,k), y(i+1,j   ...
                        ,k) - y(i,j+1,k), z(i+1,j   ,k) - z(i,j+1,k)...
                        ];
240                 crtmp = cross(vect_L, vect_M);
241                 normal = crtmp / xnrm2( crtmp );
242                 d1 = dot( normal, P_A1 );
243                 d2 = dot( normal, P_A2 );
244                 dist(i,j,k) = abs(d1-d2);
245                 P_A2_trans = P_A2 + dist(i,j,k)*normal;
246                 j_trans = P_A2_trans - P_A1;
247                 L(i,j,k) = xnrm2( vect_L );
248                 M(i,j,k) = xnrm2( vect_M );
249                 % calculate some terms
250
251                 % now present a 2D version, with P_A1 at ...
                        (0,0,0) and the rest in the x-y plane
```

```
252             cos_angle_L(k) = dot( j_trans, vect_L ) / ( ...
                    xnrm2(j_trans) * xnrm2(vect_L) );
253             sin_angle_L(k) = xnrm2( cross( j_trans, vect_L...
                    ) )/( xnrm2(j_trans) * xnrm2(vect_L) );
254             cos_angle_M(k) = dot( -j_trans, vect_M ) / ( ...
                    xnrm2(j_trans) * xnrm2(vect_M) );
255             sin_angle_M(k) = xnrm2( cross( -j_trans, ...
                    vect_M ) )/( xnrm2(-j_trans) * xnrm2(vect_M)...
                    );
256             gamma(k) = atan2( sin_angle_L(k), cos_angle_L(...
                    k) );
257             phi(k) = atan2( sin_angle_M(k), cos_angle_M(k)...
                    );
258             M_x(k) = -M(i,j,k) * cos_angle_M(k);
259             M_y(k) = M(i,j,k) * sin_angle_M(k);
260             j_x(k) = xnrm2( j_trans );
261             bside(i,j,k) = xnrm2(j_trans);
262             lside(i,j,k) = xnrm2(j_trans + vect_M);
263                         % disp('pause');
264                         % pause
265         end %for(k)
266 %
267 % The pivot point can be found by finding the poles of the...
        L link and the M link
268 % The system can be set up as a four-bar slider mechanism,...
        with L pivoting at P_A1, set to the origin
269 %  The M link slides along the x-axis.  We can find the ...
       pole by graphical construction
270 %
271         delta_B1x = (j_x(2) + M_x(2)) - (j_x(1) + M_x(1))...
                ;                               % distance ...
                between the x-coord of the A1-B1 vectors in the...
                 two positions (taking A1 as origin for both ...
                positions)
272         delta_B1y = (   0    + M_y(2)) - (    0    + M_y(1))...
                ;                               % distance ...
```

```
        between the y-coord of the A1-B1 vectors in the...
         two positions (taking A1 as origin for both ...
        positions)
273     x_Mpp = 0.5 * ( j_x(1) + j_x(2) );                ...
                                                          ...
                        % The x-coordinate of the M-...
        link pole is at the mid-point of travel between...
         P_A2(1) and P_A2(2)
274                     % quadratic formula to solve
275                     % (1-delta_B1/delta_A2)*d_A2_Mpp^2...
                            - 2*M*cos(90-phi)*d_A2_Mpp + M..
                        ^2
276                     %
277                      delta_A2 = j_x(2) - j_x(1);
278                      delta_B1 = sqrt( delta_B1x^2 + ...
                            delta_B1y^2 );
279                      d_A2_Mpp1 = ( -(-2*M(i,j,1)*cos(pi...
                            /4-phi(1))) + sqrt( (-2*M(i,j,1)...
                            *cos(pi/4-phi(1)))^2 - 4*(1-...
                            delta_B1/delta_A2)*M(i,j,1)) ) /...
                             (2*(1-delta_B1/delta_A2) );
280                      d_A2_Mpp2 = ( -(-2*M(i,j,2)*cos(pi...
                            /4-phi(2))) + sqrt( (-2*M(i,j,2)...
                            *cos(pi/4-phi(2)))^2 - 4*(1-...
                            delta_B1/delta_A2)*M(i,j,2)) ) /...
                             (2*(1-delta_B1/delta_A2) );
281                      % now use simple pythagorean
282                      y_Mpp1 = sqrt( d_A2_Mpp1^2 - (...
                            delta_A2/2)^2 );
283                      y_Mpp2 = sqrt( d_A2_Mpp2^2 - (...
                            delta_A2/2)^2 );
284     s = ( j_x(1) + M_x(1) + delta_B1x/2 - x_Mpp) / ...
        delta_B1y;       % This is the slope of the line...
         perpendicular to the travel between P_B1(1) ...
        and P_B1(2)
```

```
285             y_Mpp = M_y(1) + delta_B1y/2 + delta_B1x*s;     ...
                                                        % This ...
                   is the y-coordinate of the M-link pole, drawn ...
                   perpendicular from P_B1_mid. The pivot is on ...
                   the line between the two poles
286             delta_gamma = gamma(2) - gamma(1);              ...
                                                        % ...
                   change in angle of acuation for L
287             delta_phi = phi(2) - phi(1);                    ...
                                                % change in ...
                   angle of actuation for M
288             alpha(i,j) = atan2( y_Mpp, x_Mpp ) - delta_gamma...
                   /2 - gamma(1);  %%%%%%%% should this be +/- ...
                   delta_gamma/2 ????
289                             alpha1 = atan2( y_Mpp1, x_Mpp) - ...
                                   delta_gamma/2 - gamma(1);
290                             alpha2 = atan2( y_Mpp2, x_Mpp) + ...
                                   delta_gamma/2 - gamma(2);
291             tmp1 = tan( delta_phi /2 )/(tan( delta_gamma /2 )...
                    + tan( delta_phi /2 ));
292 %               printf( '%s', show(tmp1, 'tmp1', '%20.4g') );
293 %               printf( '%s', show([x_Mpp y_Mpp], 'Mpp', '%20.4...
       g') );
294             e(i,j) = xnrm2( [x_Mpp; y_Mpp] ) * tmp1 / cos(...
                   delta_gamma/2);
295                             e1 = (j_x(2)^2 - j_x(1)^2)/(2*(j_x...
                                   (2)*cos(gamma(2)+alpha1) - j_x...
                                   (1)*cos(gamma(1)+alpha1) ) );
296                             e2 = (j_x(2)^2 - j_x(1)^2)/(2*(j_x...
                                   (2)*cos(gamma(2)+alpha2) - j_x...
                                   (1)*cos(gamma(1)+alpha2) ) );
297                             f(i,j) = sqrt( L(i,j,1)^2       + ...
                                   e(i,j)^2      - 2*L(i,j,1)*e(i,j...
                                   )*cos(alpha(i,j)));
298                             f1 = sqrt( L(i,j,1)^2   + e1^2  - ...
                                   2*L(i,j,1)*e1*cos(alpha1));
```

```
299                     f2 = sqrt( L(i,j,2)^2    + e2^2  - ...
                           2*L(i,j,2)*e2*cos(alpha2));
300         g(i,j) = sqrt( e(i,j)^2          + j_x(1)^2       - ...
            2*e(i,j)  *j_x(1)*cos(gamma(1)+alpha(i,j)) );
301         g1 = sqrt( e1^2          + j_x(1)^2       - 2*e1  *...
            j_x(1)*cos(gamma(1)+alpha1) );
302         g2 = sqrt( e2^2          + j_x(2)^2       - 2*e2  *...
            j_x(2)*cos(gamma(2)+alpha2) );
303         beta(i,j) = asin( e(i,j)/g(i,j)*sin(gamma(1)+...
            alpha(i,j)) ) - phi(1);
304         beta1 = asin( e1/g1*sin(gamma(1)+alpha1) ) - phi...
            (1);
305         beta2 = asin( e2/g2*sin(gamma(2)+alpha2) ) - phi...
            (2);
306         h(i,j) = sqrt( M(i,j,1)^2        + g(i,j)^2       - ...
            2*M(i,j,1)*g(i,j)*cos(beta(i,j)) );
307         h1 = sqrt( M(i,j,1)^2  + g1^2  - 2*M(i,j,1)*g1*...
            cos(beta1) );
308         h2 = sqrt( M(i,j,2)^2  + g2^2  - 2*M(i,j,2)*g2*...
            cos(beta2) );
309                     % printf('y_Mpp: %6.5g \t y_Mpp1: ...
                          %6.5g \t y_Mpp2: %6.5g \t', ...
                          y_Mpp, y_Mpp1, y_Mpp2);
310                     % printf('alpha: %6.5g \t alpha1: ...
                          %6.5g \t alpha2: %6.5g \t', ...
                          alpha(i,j), alpha1, alpha2);
311                     % printf('e: %6.5g \t e1: %6.5g \t...
                           e2: %6.5g \t', e(i,j), e1, e2);
312                     % printf('f: %6.5g \t f1: %6.5g \t...
                           f2: %6.5g \n', f(i,j), f1, f2);
313                     % printf('beta: %6.5g \t beta1: ...
                          %6.5g \t beta2: %6.5g \t', beta(...
                          i,j), beta1, beta2);
314                     % printf('g: %6.5g \t g1: %6.5g \t...
                           g2: %6.5g \t', g(i,j), g1, g2);
```

```
315                             % printf('h: %6.5g \t h1: %6.5g \t...
                                    h2: %6.5g \n', h(i,j), h1, h2);
316                             % alpha(i,j) = alpha1;
317                             % e(i,j) = e1;
318                             % f(i,j) = f1;
319                             % beta(i,j) = beta1;
320                             % g(i,j) = g1;
321                             % h(i,j) = h1;
322                             % pause
323         end %for(j)
324     end %for(i)
325     % printf('%s', show(L, 'L: ', '%8.5g') );
326     for (i=1:numrows)
327         e(i,numcols+1,1) = e(i,1,1) ;
328         f(i,numcols+1,1) = f(i,1,1) ;
329         g(i,numcols+1,1) = g(i,1,1) ;
330         h(i,numcols+1,1) = h(i,1,1) ;
331         alpha(i,numcols+1,1) = alpha(i,1,1) ;
332         beta(i,numcols+1,1) = beta(i,1,1) ;
333         for (k=1:2)
334             L(i,numcols+1,k) = L(i,1,k);
335             M(i,numcols+1,k) = M(i,1,k);
336             bside(i,numcols+1,k) = bside(i,1,k) ;
337             lside(i,numcols+1,k) = lside(i,1,k) ;
338                     dist(i,numcols+1,k) = dist(i,1,k) ;
339         end %for(k)
340     end %for(i)
341
342     %     disp('dist = ', dist)
343
344 function m = min_pos(v)
345     m = max(v);
346     m = max(m,1e3);
347     for (i=1:length(v))
348         if ( v(i) > 0 )
349             m = min(m, v(i));
```

```
350         end %if
351     end %for(i)
352
353 function m = max_pos(v)
354     m = min(v);
355     m = min(m,-1e3);
356     for (i=1:length(v))
357         if ( v(i) < 0 )
358             m = max(m, v(i));
359         end %if
360     end %for(i)
361
362 function [x errorcalc] = linesearch(t, dir, limits)
363     printf('Beginning linesearch ... \t');
364     global maxiters criteria numcols numrows numiters ...
          num_t_rows;
365     steps = 100;
366     window_width = 10;
367     %golden ratio search
368     golden = (1+sqrt(5))/2;
369     %find the larger error
370     for( i=1:numrows )
371         for( j=1:numcols )
372             theta((i-1)*numcols+j,1) = sum( t( ((i-1)*numcols...
                  +1):((i-1)*numcols+j) ) );
373         end %for(j)
374     end %for(i)
375     dir_theta = dir(1:3*numcols);
376     uv = t(3*numcols+1:5*numcols);
377     dir_uv = dir(3*numcols+1:5*numcols);
378     lim1 = (limits(:,1) - t) ./ dir;
379     lim2 = (limits(:,2) - t) ./ dir;
380     % printf( '%s', show(reshape(t, numcols, 14)', 't', ...
          '%12.3g') );
381     % printf( '%s', show(reshape(dir, numcols, 14)', 'dir',...
           '%12.3g') );
```

```
382     % printf( '%s', show(reshape(lim1, numcols, 14)', 'lim1...
          ', '%12.3g') );
383     % printf( '%s', show(reshape(lim2, numcols, 14)', 'lim2...
          ', '%12.3g') );
384     right1 = min_pos( lim1 );
385     right2 = min_pos( lim2 );
386     right = min(right1, right2);
387     left1 = max_pos( lim1 );
388     left2 = max_pos( lim2 );
389     left = max(left1, left2);
390     printf('left1: %8.3g \t left2: %8.3g \t right1: %8.3g \...
          t right2: %8.3g \n', left1, left2, right1, right2);
391
392 %%
393 %% Now run an equally-spaced interval search to find a ...
      minimum
394 %%
395     % x_step = linspace(left,right,steps);
396     % err_min = xnrm2(calc_error(t + x_step(1)*dir)); %for...
          starters
397     % i_min = 1;
398     % for( i=1:steps );
399         % err_step(i) = xnrm2(calc_error(t + x_step(i)*dir))...
              ;
400         % if( err_step(i) < err_min )
401             % err_min = err_step(i);
402             % i_min = i;
403         % end %if
404             % printf('i: %d \t i_min: %d \t x_step(i_min): ...
                  %12.9g \t err_min: %12.9g\t\t\r', i, i_min, ...
                  x_step(i_min), err_min);
405     % end %for(x_step)
406     % figure(3);
407         % thd_ls = threadnew;
408         % threadstart(thd_ls, 'semilogy', 0, 3, 1:steps, ...
              err_step, 'o-');
```

```
409     %% semilogy(3,1:steps,err_step,'o-');
410
411     % pause
412     % left  = x_step( max(i_min-window_width/2,     1) );
413     % right = x_step( min(i_min+window_width/2,steps) );
414     % printf( '\nLowest at i=%d, start search at window ...
            between %20.10g and %20.10g \n', i_min, left, right )...
            ;
415
416     error_l = calc_error(t + left*dir);
417     error_r = calc_error(t + right*dir);
418     err_l = xnrm2(error_l);
419     err_r = xnrm2(error_r);
420
421     for (iter=1:maxiters)
422     printf('left: %20.17g \t err_l: %20.17g \t right:...
            %20.17g \t err_r: %20.17g\t\t \r', left, err_l, right...
            , err_r);
423         offset = (right - left) / golden;
424         if (err_l > err_r)
425 %             printf('right is lower; choosing new left; \t')...
        ;
426             left = right - offset;
427             %         mid = left +  dist;
428             error_l = calc_error(t + left*dir);
429             err_l = xnrm2(error_l);
430             x = right;
431             errorcalc = error_r;
432         else
433 %             printf('left is lower; choosing new right; \t')...
        ;
434             right = left + offset;
435             %         mid = right - dist;
436             error_r = calc_error(t + right*dir);
437             err_r = xnrm2(error_r);
438             x = left;
```

```
439             errorcalc = error_l;
440         end %if
441         delta_err = abs(err_r - err_l) / ( (err_l + err_r)/2...
                );
442         if (delta_err <= criteria)
443             low_err = xnrm2(errorcalc);
444                             error0 = calc_error(t);
445                             % if ( low_err > err_min )      % ...
                                    compare with minimum found with ...
                                    interval search
446                                     % x = x_step(i_min);
447                                     % printf('\nchose x=%15.5g...
                                            (i_min: %d) because ...
                                            that error, %15.5g, is ...
                                            less than %15.5g!  \t', ...
                                            x, i_min, err_min, xnrm2...
                                            (errorcalc));
448                                     % errorcalc = calc_error(t...
                                            + x_step(i_min)*dir);
449                                     % low_err = err_min;
450                             % end
451                             if ( low_err > xnrm2(error0) ) % ...
                                    compare with where we started, x...
                                    =0
452                                     printf('\nchose x=0 ...
                                            because that error, ...
                                            %15.5g, is less than ...
                                            %15.5g!  \t', xnrm2(...
                                            error0), xnrm2(errorcalc...
                                            ));
453                                     x = 0;
454                                     errorcalc = error0;
455                                     low_err = xnrm2(error0);
456             end %if
457             printf('\nExit linesearch at %d;\t', iter);
458             break
```

```
459         end %if
460     end %for(iter)
461     % threadwait(thd_ls);
462     % filename = sprintf('linesearchfig.png');
463     % print(filename);
464           % threadfree(thd_ls);
465     %    x = max(x, 0);
466 %    x = (right+left)/2;
467
468 function errorcalc = calc_error(t)
469     global numrows numcols;
470     [x, y, z] = get_pts(t, numrows, numcols);
471     [L, M, e, f, g, h alpha, beta, dist, bside, lside] = ...
          get_links(x,y,z);
472     avg_L = xnrm2( L );
473     for (j=1:numcols)
474 %calculate the errors
475             errorcalc(1,j) = dist(1,j,2) - dist(1,j,1);
476             errorcalc(2,j) = dist(2,j,2) - dist(2,j,1);
477             errorcalc(3,j) = dist(3,j,2) - dist(3,j,1);
478         %% Mobility requirement
479             errnum = 4;
480             for (i=1:numrows-1)
481             del_kB2_del_jB1(i) = 1 - (L(i+1,j   ,1)/e(i+1,j   )...
                )*cos(alpha(i+1,j   )) + ((bside(i+1,j   ,1)^2 - ...
                dist(i+1,j   ,1)^2) - e(i+1,j   )^2 - g(i+1,j   )...
                ^2)*(L(i+1,j   ,1)/e(i+1,j   ))*sin(alpha(i+1,j   ...
                ))/(4*(bside(i+1,j   ,1)^2 - dist(i+1,j   ,1)^2)*...
                e(i+1,j   )^2 - ((bside(i+1,j   ,1)^2 - dist(i+1,...
                j   ,1)^2) + e(i+1,j   )^2 - g(i+1,j   )^2)^2)...
                ^0.5;
482             del_kB2_del_jB2(i) = 1 - (M(i+1,j+1,1)/g(i+1,j+1)...
                )*cos( beta(i+1,j+1)) + ((bside(i+1,j+1,1)^2 - ...
                dist(i+1,j+1,1)^2) - g(i+1,j+1)^2 - e(i+1,j+1)...
                ^2)*(M(i+1,j+1,1)/g(i+1,j+1))*sin( beta(i+1,j...
                +1))/(4*(bside(i+1,j+1,1)^2 - dist(i+1,j+1,1)...
```

```
    ^2)*g(i+1,j+1)^2 - ((bside(i+1,j+1,1)^2 - dist(...
    i+1,j+1,1)^2) + g(i+1,j+1)^2 - e(i+1,j+1)^2)^2)...
    ^0.5;
483 del_jB1_del_kA2(i) = 1 - (M(i  ,j  ,1)/g(i  ,j  )...
    )*cos( beta(i  ,j  )) - ((lside(i  ,j  ,1)^2 - ...
    dist(i  ,j  ,1)^2) - g(i  ,j  )^2 - f(i  ,j  )...
    ^2)*(M(i  ,j  ,1)/g(i  ,j  ))*sin( beta(i  ,j  )...
    )/(4*(lside(i  ,j  ,1)^2 - dist(i  ,j  ,1)^2)*g...
    (i  ,j  )^2 - ((lside(i  ,j  ,1)^2 - dist(i  ,j...
      ,1)^2) + g(i  ,j  )^2 - f(i  ,j  )^2)^2)^0.5;
484 del_jB2_del_kA2(i) = 1 - (L(i  ,j+1,1)/e(i  ,j+1)...
    )*cos(alpha(i  ,j+1)) - ((lside(i  ,j+1,1)^2 - ...
    dist(i  ,j+1,1)^2) - e(i  ,j+1)^2 - h(i  ,j+1)...
    ^2)*(L(i  ,j+1,1)/e(i ,j+1))*sin(alpha(i  ,j+1)...
    )/(4*(lside(i  ,j+1,1)^2 - dist(i  ,j+1,1)^2)*e...
    (i  ,j+1)^2 - ((lside(i  ,j+1,1)^2 - dist(i  ,j...
    +1,1)^2) + e(i  ,j+1)^2 - h(i  ,j+1)^2)^2)^0.5;
485 errorcalc(errnum+0,j) = del_kB2_del_jB1(i)*...
    del_jB1_del_kA2(i) - del_kB2_del_jB2(i)*...
    del_jB2_del_kA2(i);
486 % now onto 2nd & 3rd order mobility
487 errorcalc(errnum+1,j) = 1.0/avg_L * (e(i ...
      ,j+1)  * h(i  ,j+1) ...
                          - g(i  ,j  )  *...
      f(i  ,j  ));
488 errorcalc(errnum+2,j) = 1.0/avg_L * (e(i ...
      ,j+1)^2 + h(i  ,j+1)^2 + dist(i  ,j...
     +1,1)^2 - (g(i  ,j  )^2 + f(i  ,j  )^2 ...
     + dist(i  ,j  )^2));
489 errorcalc(errnum+3,j) = 1.0/avg_L * (e(i...
     +1,j  )  * g(i+1,j  ) ...
                          - h(i  ,j  )  *...
      f(i  ,j  ));
490 errorcalc(errnum+4,j) = 1.0/avg_L * (e(i...
     +1,j+1)^2 + g(i+1,j  )^2 + dist(i+1,j  ...
     ,1)^2 - (h(i  ,j  )^2 + f(i  ,j  )^2 + ...
```

```
                    dist(i   ,j   )^2));
491                 errorcalc(errnum+5,j) = 1.0/avg_L * (g(i...
                    +1,j+1)   * e(i+1,j+1) ...
                                        - f(i   ,j+1)   *...
                     h(i   ,j+1));
492                 errorcalc(errnum+6,j) = 1.0/avg_L * (g(i...
                    +1,j+1)^2 + e(i+1,j+1)^2 + dist(i+1,j...
                    +1,1)^2 - (f(i   ,j+1)^2 + h(i   ,j+1)^2 ...
                    + dist(i   ,j+1)^2));
493                 errorcalc(errnum+7,j) = 1.0/avg_L * (e(i...
                    +1,j+1)   * h(i+1,j+1) ...
                                        - g(i+1,j   )   *...
                     f(i+1,j   ));
494                 errorcalc(errnum+8,j) = 1.0/avg_L * (e(i...
                    +1,j+1)^2 + h(i+1,j+1)^2 + dist(i+1,j...
                    +1,1)^2 - (g(i+1,j   )^2 + f(i+1,j   )^2 ...
                    + dist(i+1,j   )^2));
495                 % errorcalc(errnum+9,j) =
496                 errnum = errnum+9;
497         end %for(i)
498
499         errorcalc(errnum+0,j) = (L(1,j,2) - L(1,j,1));
500         errorcalc(errnum+1,j) = (L(2,j,2) - L(2,j,1));
501         errorcalc(errnum+2,j) = (L(3,j,2) - L(3,j,1));
502         errorcalc(errnum+3,j) = (M(1,j,2) - M(1,j,1));
503         errorcalc(errnum+4,j) = (M(2,j,2) - M(2,j,1));
504         errorcalc(errnum+5,j) = (M(3,j,2) - M(3,j,1));
505 % now the triangle problem
506         % errorcalc(errnum+6,j) = 20* log( max( (e(1,j) - L...
              (1,j,1) + 2), 1 ) );
507         % errorcalc(errnum+7,j) = 20* log( max( (e(2,j) - L...
              (2,j,1) + 2), 1 ) );
508         % errorcalc(errnum+8,j) = 20* log( max( (e(3,j) - L...
              (3,j,1) + 2), 1 ) );
509         % errorcalc(errnum+9,j) = 20* log( max( (g(1,j) - M..
              (1,j,1) + 2), 1 ) );
```

```
510         % errorcalc(errnum+10,j) = 20* log( max( (g(2,j) - M..
                (2,j,1) + 2), 1 ) );
511         % errorcalc(errnum+11,j) = 20* log( max( (g(3,j) - M..
                (3,j,1) + 2), 1 ) );
512             errorcalc(errnum+6,j) = max( (e(1,j) - L(1,j,1) ...
                    + 1), 0 );
513             errorcalc(errnum+7,j) = max( (e(2,j) - L(2,j,1) ...
                    + 1), 0 );
514             errorcalc(errnum+8,j) = max( (e(3,j) - L(3,j,1) ...
                    + 1), 0 );
515             errorcalc(errnum+9,j) = max( (g(1,j) - M(1,j,1) ...
                    + 1), 0 );
516             errorcalc(errnum+10,j) = max( (g(2,j) - M(2,j,1)...
                     + 1), 0 );
517             errorcalc(errnum+11,j) = max( (g(3,j) - M(3,j,1)...
                     + 1), 0 );
518 % add coercive alpha & beta constraints
519             alpha_beta_max = deg2rad( 30 );
520             errorcalc(errnum+12,j) = 10 * max( (alpha(1,j) -...
                    alpha_beta_max), 0 );
521             errorcalc(errnum+13,j) = 10 * max( (alpha(2,j) -...
                    alpha_beta_max), 0 );
522             errorcalc(errnum+14,j) = 10 * max( (alpha(3,j) -...
                    alpha_beta_max), 0 );
523             errorcalc(errnum+15,j) = 10 * max( ( beta(1,j) -...
                    alpha_beta_max), 0 );
524             errorcalc(errnum+16,j) = 10 * max( ( beta(2,j) -...
                    alpha_beta_max), 0 );
525             errorcalc(errnum+17,j) = 10 * max( ( beta(3,j) -...
                    alpha_beta_max), 0 );
526         % errorcalc(errnum+12,j) = 1/50 * (exp( 5*abs(alpha...
                (1,j) ) ) - 1);
527         % errorcalc(errnum+13,j) = 1/50 * (exp( 5*abs(alpha...
                (2,j) ) ) - 1);
528         % errorcalc(errnum+14,j) = 1/50 * (exp( 5*abs(alpha...
                (3,j) ) ) - 1);
```

```
529         % errorcalc(errnum+15,j) = 1/50 * (exp( 5*abs(beta...
                (1,j) ) ) - 1);
530         % errorcalc(errnum+16,j) = 1/50 * (exp( 5*abs(beta...
                (2,j) ) ) - 1);
531         % errorcalc(errnum+17,j) = 1/50 * (exp( 5*abs(beta...
                (3,j) ) ) - 1);
532     end %for(j)
533 %
534 %     errorcalc = min(errorcalc, 1e3*ones(size(errorcalc)))...
        ;
535
536  function g = gradient(t, err)
537 %
538 % This function calculates the error gradient of t[] by...
        perturbing each t[i] some
539 % small amount eps_off (by adding and subtracting) and...
        calculating the difference
540 % between those two different errors.
541 %
542     eps_off = 10^(-3+rand(1));          % due to the "...
          bumpy" search surface, having a variable perturbation...
           may vary the search direction a little
543  printf('Beginning gradient calculation with eps_off=%12.3g...
        ... \n', eps_off);
544     % thd1 = threadnew;
545     % thd2 = threadnew;
546     for (i=1:length(t))
547         tmp1 = t;
548         tmp2 = t;
549         tmp1(i) = t(i) + eps_off;
550         tmp2(i) = t(i) - eps_off;
551         err_mod1 = calc_error(tmp1);
552         err_mod2 = calc_error(tmp2);
553             % threadstart(thd1,'calc_error', 1, tmp1);
554             % threadstart(thd2,'calc_error', 1, tmp2);
555             % err_mod1 = threadvalue(thd1);
```

```
556             % err_mod2 = threadvalue(thd2);
557         grad_err1 = xnrm2(err_mod1);
558         grad_err2 = xnrm2(err_mod2);
559         g(i,1) = (grad_err1 - grad_err2) / (2*eps_off);
560             printf('\t grad(%d)\t = %12.3f \t\r', i, g(i,1))...
                ;
561     end %for(i)
562     % threadfree(thd1);
563     % threadfree(thd2);
564
565 function [u, v] = get_uv(t)
566     global numrows numcols;
567     u = -ones(4,numcols,2);
568     v = -ones(4,numcols,2);
569     u(1,:,1) = t(0*numcols+1:1*numcols,1);
570     v(1,:,1) = 0.0 * u(1,:,1); % v_A, position 1, fixed
571     u(2,:,1) = t(1*numcols+1:2*numcols,1);
572     v(2,:,1) = t(2*numcols+1:3*numcols,1);
573     u(3,:,1) = t(3*numcols+1:4*numcols,1);
574     v(3,:,1) = t(4*numcols+1:5*numcols,1);
575     u(4,:,1) = t(5*numcols+1:6*numcols,1);
576     v(4,:,1) = t(6*numcols+1:7*numcols,1);
577     u(1,:,2) = t(7*numcols+1:8*numcols,1);
578     v(1,:,2) = 0.0 * u(1,:,1); % v_A, position 2, fixed
579     u(2,:,2) = t(8*numcols+1:9*numcols,1);
580     v(2,:,2) = t(9*numcols+1:10*numcols,1);
581     u(3,:,2) = t(10*numcols+1:11*numcols,1);
582     v(3,:,2) = t(11*numcols+1:12*numcols,1);
583     u(4,:,2) = t(12*numcols+1:13*numcols,1);
584     v(4,:,2) = t(13*numcols+1:14*numcols,1);
585
586 function [t limits] = Init()
587 %
588 % This function initializes all of the values in t[] to ...
      some number between 0 and 1. Most of the distribution ...
      is evenly spaced.
```

```
589 %
590     global numrows numcols;
591     u_A_init = 1.0/numcols/2.0;
592     u_A_final = 1.0 - u_A_init;
593     u_B_init = 1.0/numcols/2.0;
594     u_B_final = 1.0 - u_B_init;
595     u_C_init = 1.0/numcols/2.0;
596     u_C_final = 1.0 - u_C_init;
597     u_D_init = 1.0/numcols/2.0;
598     u_D_final = 1.0 - u_D_init;
599         % the following numbers are a best-fit of equally-...
              space thetas of a circular Bezier curve
600         u_tmp = [0.032649        0.1057   0.186186        ...
              0.27194  0.361706        0.453702        0.546297...
                       0.638294        0.72806  0.813813        ...
              0.8943   0.967351];
601         t( 0*numcols+1: 1*numcols,1) = u_tmp;
602         t( 1*numcols+1: 2*numcols,1) = u_tmp;
603         t( 3*numcols+1: 4*numcols,1) = u_tmp;
604         t( 5*numcols+1: 6*numcols,1) = u_tmp;
605         t( 7*numcols+1: 8*numcols,1) = u_tmp;
606         t( 8*numcols+1: 9*numcols,1) = u_tmp;
607         t(10*numcols+1:11*numcols,1) = u_tmp;
608         t(12*numcols+1:13*numcols,1) = u_tmp;
609     % t(0*numcols+1:1*numcols,1) = linspace( u_A_init, ...
          u_A_final, numcols ); % u_A, position 1
610     % v_A is fixed at 0.0
611     % t(1*numcols+1:2*numcols,1) = linspace( u_B_init, ...
          u_B_final, numcols ); % u_B, position 1
612     t(2*numcols+1:3*numcols,1) = 0.30;   % v_B, position 1
613     % t(3*numcols+1:4*numcols,1) = linspace( u_C_init, ...
          u_C_final, numcols ); % u_C, position 1
614     t(4*numcols+1:5*numcols,1) = 0.60;   % v_C, position 1
615     % t(5*numcols+1:6*numcols,1) = linspace( u_D_init, ...
          u_D_final, numcols ); % u_D, position 1
616     t(6*numcols+1:7*numcols,1) = 0.90;   % v_D position 1
```

```
617     % t(7*numcols+1:8*numcols,1) = linspace( u_A_init, ...
            u_A_final, numcols ); % u_A, position 2
618           % v_A is fixed at 0.0
619           % t(8*numcols+1:9*numcols,1) = linspace( u_B_init,...
                  u_B_final, numcols ); % u_B, position 2
620     t(9*numcols+1:10*numcols,1) = 0.30; % v_B, position 2
621     % t(10*numcols+1:11*numcols,1) = linspace( u_C_init, ...
            u_C_final, numcols ); % u_C, position 2
622     t(11*numcols+1:12*numcols,1) = 0.60; % v_C, position 2
623     % t(12*numcols+1:13*numcols,1) = linspace( u_D_init, ...
            u_D_final, numcols ); % u_D, position 2
624     t(13*numcols+1:14*numcols,1) = 0.90; % v_D position 2
625     limits(1:14*numcols,1) = 0.0001;
626     limits(1:8*numcols,2) = 0.9999;
627
628
629 function [x y z] = link2surf( P_A1, vect_L, vect_e )
630     triangle(1,1) = P_A1(1);
631     triangle(2,1) = P_A1(2);
632     triangle(3,1) = P_A1(3);
633     triangle(1,2) = P_A1(1) + vect_e(1);
634     triangle(2,2) = P_A1(2) + vect_e(2);
635     triangle(3,2) = P_A1(3) + vect_e(3);
636     triangle(1,3) = P_A1(1) + vect_L(1);
637     triangle(2,3) = P_A1(2) + vect_L(2);
638     triangle(3,3) = P_A1(3) + vect_L(3);
639     triangle(4,:) = 1;
640     x = [triangle(1,1)/triangle(4,1), triangle(1,2)/...
            triangle(4,2); triangle(1,3)/triangle(4,3), triangle...
            (1,3)/triangle(4,3)];
641     y = [triangle(2,1)/triangle(4,1), triangle(2,2)/...
            triangle(4,2); triangle(2,3)/triangle(4,3), triangle...
            (2,3)/triangle(4,3)];
642     z = [triangle(3,1)/triangle(4,1), triangle(3,2)/...
            triangle(4,2); triangle(3,3)/triangle(4,3), triangle...
            (3,3)/triangle(4,3)];
```

```
643
644 function plot_pts(t, fignum)
645     global numrows numcols thread;
646
647     % return; %skips the rest of this...
648
649     disp('Drawing figure...');
650
651     [r1_A r2_A r1_D r2_D z_D] = get_dims();
652     [x y z] = get_pts(t, numrows, numcols);
653
654     x_line = x;
655     x_line(:,:,3:4) = x(:,:,:);
656     x_line(2,:,1:2) = circshift(x(2,:,:), [0 -1]);
657     x_line(1,:,3:4) = circshift(x(1,:,:), [0 -1]);
658     x_line(4,:,1:2) = circshift(x(4,:,:), [0 -1]);
659     x_line(3,:,3:4) = circshift(x(3,:,:), [0 -1]);
660
661     y_line = y;
662     y_line(:,:,3:4) = y(:,:,:);
663     y_line(2,:,1:2) = circshift(y(2,:,:), [0 -1]);
664     y_line(1,:,3:4) = circshift(y(1,:,:), [0 -1]);
665     y_line(4,:,1:2) = circshift(y(4,:,:), [0 -1]);
666     y_line(3,:,3:4) = circshift(y(3,:,:), [0 -1]);
667
668     z_line = z;
669     z_line(:,:,3:4) = z(:,:,:);
670     z_line(2,:,1:2) = circshift(z(2,:,:), [0 -1]);
671     z_line(1,:,3:4) = circshift(z(1,:,:), [0 -1]);
672     z_line(4,:,1:2) = circshift(z(4,:,:), [0 -1]);
673     z_line(3,:,3:4) = circshift(z(3,:,:), [0 -1]);
674
675         xtriL = zeros(2*numcols*numrows,2,2); % init zeros...
                size of 2*numcols*numrows, 2, numpos
676         ytriL = 0.0*xtriL;
677         ztriL = 0.0*xtriL;
```

```
678         xtriM = 0.0*xtriL; % init zeros size of 2*numcols*...
               numrows, 2, numpos
679         ytriM = 0.0*xtriL;
680         ztriM = 0.0*xtriL;
681     [L, M, e, f, g, h, alpha, beta, dist, bside, lside] = ...
           get_links(x,y,z);
682     for (i=1:numrows)
683        for (j=1:numcols)
684           for (k=1:2)
685              P_A1   = [x(i   ,j   ,k), y(i,j   ,k), z(i,j   ,k)...
                    ];
686              P_A2   = [x(i   ,j+1,k), y(i,j+1,k), z(i,j+1,k)...
                    ];
687              vect_L = [x(i+1,j+1,k) - x(i,j   ,k), y(i+1,j...
                    +1,k) - y(i,j   ,k), z(i+1,j+1,k) - z(i,j   ,k...
                    )];
688              vect_M = [x(i+1,j   ,k) - x(i,j+1,k), y(i+1,j   ...
                    ,k) - y(i,j+1,k), z(i+1,j   ,k) - z(i,j+1,k)...
                    ];
689              normal = cross(vect_L, vect_M) / xnrm2( cross(...
                    vect_L, vect_M) );
690              vect_e(1) = L(i,j,k)*e(i,j) ...
691              * (cos(alpha(i,j)) - vect_L(2)/vect_L(1)*...
                    normal(3)*sin(alpha(i,j)) + vect_L(3)/vect_L...
                    (1)*normal(2)*sin(alpha(i,j)))...
692              / (vect_L(1) + vect_L(2)^2/vect_L(1) + vect_L...
                    (3)^2/vect_L(1));
693              vect_e(2) = ( normal(3)*L(i,j,k)*e(i,j)*sin(...
                    alpha(i,j)) + vect_L(2)*vect_e(1)) / vect_L...
                    (1);
694              vect_e(3) = (-normal(2)*L(i,j,k)*e(i,j)*sin(...
                    alpha(i,j)) + vect_L(3)*vect_e(1)) / vect_L...
                    (1);
695
696              vect_g(1) = M(i,j,k)*g(i,j) ...
```

```
697 * (cos(beta(i,j)) - vect_M(2)/vect_M(1)*normal(3)*sin(-...
      beta(i,j)) + vect_M(3)/vect_M(1)*normal(2)*sin(-beta(i,j...
      ))) ...
698 / (vect_M(1) + vect_M(2)^2/vect_M(1) + vect_M(3)^2/vect_M...
      (1));
699                 vect_g(2) = ( normal(3)*M(i,j,k)*g(i,j)*sin(-...
                      beta(i,j)) + vect_M(2)*vect_g(1)) / vect_M...
                      (1);
700                 vect_g(3) = (-normal(2)*M(i,j,k)*g(i,j)*sin(-...
                      beta(i,j)) + vect_M(3)*vect_g(1)) / vect_M...
                      (1);
701                                         % idx = 2*numcols*(i-1)+2*...
                                              j-1;
702                                         idx = 2*numcols*(i-1)+2*j...
                                              -1;
703                                         % [xtriL ytriL ztriL] = ...
                                              link2surf( P_A1 , vect_L...
                                              , vect_e );
704                                         % [xtriM ytriM ztriM] = ...
                                              link2surf( P_A2 , vect_M...
                                              , vect_g );
705                                         [xtriL(idx:idx+1,:,k) ...
                                              ytriL(idx:idx+1,:,k) ...
                                              ztriL(idx:idx+1,:,k)] = ...
                                              link2surf( P_A1 , vect_L...
                                              , vect_e );
706                                         [xtriM(idx:idx+1,:,k) ...
                                              ytriM(idx:idx+1,:,k) ...
                                              ztriM(idx:idx+1,:,k)] = ...
                                              link2surf( P_A2 , vect_M...
                                              , vect_g );
707                                 % figure(fignum(k));
708                                         % surf(xtriL,ytriL,ztriL);
709                                         % surf(xtriM,ytriM,ztriM);
710                                   end %for(k)
711         end %for(k)
```

```
712     end %for(i)
713
714     utics = 5*numcols;
715     vtics = 5*numrows;
716
717     for (k=1:2)
718         figure(fignum(k));
719         clf;
720         colormap(gray);
721         hold on;
722         for (j = 1:numcols)
723             iter = 0;
724             for (u = 0:utics)
725                 for (v = 0:vtics)
726                     [x_surf(u+1,v+1) y_surf(u+1,v+1) z_surf(u...
                            +1,v+1)] = get_patch( u/utics, v/vtics, k...
                             );
727                     iter = iter + 1;
728                 end %for(v)
729             end %for(u)
730             surf(x_surf, y_surf, z_surf)% , 'alphadatamapping...
                    ', 'direct', 'AlphaData', 0.2 %
731         end %for(j)
732
733         color = idiv(256,numcols) * j * ones( size (ztriL...
                (:,:,k)) );
734                  % printf('%s', show([xtriL, ytriL, ztriL, ...
                        xtriM, ytriM, ztriM], 'tri:', '%9.7g'));
735         surf(xtriL(1,:,k),ytriL(1,:,k),ztriL(1,:,k),color);
736         surf(xtriM(1,:,k),ytriM(1,:,k),ztriM(1,:,k),color);
737
738         plot3(x(:,:,k), y(:,:,k), z(:,:,k), 'bo',...
739             x_line(:,:,k), y_line(:,:,k), z_line(:,:,k), 'b-'...
                    ,...
740             x_line(:,:,k+2), y_line(:,:,k+2), z_line(:,:,k+2)...
                    , 'b-');%,...
```

```
741           % x_curve(:,:,k), y_curve(:,:,k), z_curve(:,:,k),...
                  'k.',...
742         xlabel('x');
743         ylabel('y');
744         zlabel('z');
745                   view(3);
746         hold off;
747     end %for(k)
748     drawnow;
749
750 function str = show(var, vartext, format);
751     [r,c,p] = size(var);
752     str = vartext;
753     for (k=1:p)
754         str = [str, '['];
755         for (i=1:r)
756             str = [str, '\n'];
757             for (j=1:c)
758                 str = sprintf([str, format, '\t'], var(i,j,k))...
                      ;
759             end %for(j)
760         end %for(i)
761         str = [str, ' ]'];
762     end %for(k)
763     str = sprintf('%s\n', str);
764
765 function final_show(t, err)
766     global numrows numcols
767      % [x y z] = get_pts(t, numrows, numcols);
768      % [L, M, e, f, g, h, alpha, beta, dist, bside, lside] ...
           = get_links(x,y,z);
769
770 %     printf( '%s', show(reshape(t, numcols, num_t_rows)', ...
    't', '%12.3g') );
771     printf( '%s', show(err, 'running_err:', '%12.5g') );
772 %     printf( '%s', show(L, 'L:', '%12.3g') );
```

```
773 %     printf( '%s', show(M, 'M:', '%12.3g') );
774 %     printf( '%s', show(e, 'e:', '%12.3g') );
775 %     printf( '%s', show(g, 'g:', '%12.3g') );
776 %     printf( '%s', show(rad2deg(alpha), 'alpha [deg]:', ...
        '%12.3g') );
777 %     printf( '%s', show(rad2deg(beta), 'beta [deg]:', ...
        '%12.3g') );
778       plot_pts(t, [1  2]);
```