2011

Modeling and Parameter Study of Bistable Spherical Compliant Mechanisms

Chester Smith
University of South Florida, chester.law.smith@gmail.com

Modeling and Parameter Study of Bistable Spherical Compliant Mechanisms

by

Chester Law Smith

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mechanical Engineering
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Craig Lusk, Ph.D.
Autar Kaw, Ph.D.
Kyle Reed, Ph.D.

Date of Approval:
March 11, 2011

Keywords: Ortho-Planar, Microelectromechanical System, 3D MEMS, Virtual Work, Static Jump

Table of Contents

List of Tables iii

List of Figures iv

Abstract vi

Chapter 1: Introduction 1
1.1 The Bistable Spherical Compliant Mechanism 2

Chapter 2: Outline 4
2.1 Objective 4
2.2 Motivation 4
2.3 Contribution 5
2.4 Research Approach 5

Chapter 3: Background 6
3.1 Microelectromechanical System History 6
3.2 Microelectromechanical System Fabrication 7
3.2.1 Bulk Micromachining 7
3.2.2 Surface Micromachining 8
3.2.3 LIGA (Lithographie, Galvanoformung, Abformung) 8
3.2.4 New Fabrication Methods 9
3.3 3D Microelectromechanical Systems 10
3.4 Ortho-Planar Mechanisms 11
3.5 Spherical Mechanisms 11
3.5.1 Spherical Trigonometry 12
3.6 Compliant Mechanisms 13
3.6.1 Small-Length Flexural Pivots 14
3.7 The Pseudo-Rigid-Body Model 15
3.8 Virtual Work Analysis 16
3.8.1 General Form of a Torsional Spring 16
3.8.2 Virtual Work 17

Chapter 4: Mathematical Model of the Bistable Spherical Compliant Mechanism 18
4.1 Spherical Geometry 18
4.1.1 Spherical Closed Form Equations 19
4.2 The Twisting and Bending Model 19
4.3 Virtual Work 21
4.3.1 First Equation of Virtual Work 21
4.3.2 Second Equation of Virtual Work 24

Chapter 5: The Finite Element Model 26

Chapter 6: Results 29

6.1 The Pure Bending Model 29

6.2 The Variable Parameter: The Length of Compliant Joint 2 30

6.3 The Twisting and Bending Model 30

6.4 Macro Scale Models 32

6.5 Finite Element Analysis: Beam Element Model 34

6.6 Static Jumps 36

6.7 Bistability 37

Chapter 7: Conclusion 38

References 39

Appendices 41

Appendix A: ANSYS Batch Files 42

Appendix B: MATLAB M-Files 61

List of Tables

Table 1 Model Parameters 30

List of Figures

Figure 1. A scanning electron microscope image of a micro bistable spherical compliant mechanism 2

Figure 2. The bistable spherical compliant mechanism design and nomenclature 3

Figure 3. (a) The bistable spherical compliant mechanism configuration after an applied input, (b) A close-up of the resultant twisting in compliant joint 2 due to an applied input 4

Figure 4. Great circles, great arcs, dihedral angles, and a spherical triangle shown on a sphere 12

Figure 5. A small-length flexural pivot 14

Figure 6. A simplified diagram of the bistable spherical compliant mechanism in the form of a partially compliant four-bar mechanism 15

Figure 7. The bistable spherical compliant mechanism pseudo-rigid-body model 16

Figure 8. Spherical diagram and nomenclature for the bistable spherical compliant mechanism 18

Figure 9. Pseudo-Rigid-Body model of link 2: Part (a) front view, Part (b) top view 20

Figure 10. Plot of the nodal configuration and constraints used to obtain finite element analysis results 27

Figure 11. For Length 5, plots of input moment, M_2, and the input link rotation, θ_2, for the pure bending model and finite element model 29

Figure 12. Plot of twisting and bending model input moment, M_2, and the input link rotation θ_2 31

Figure 13. A macro scale model of the bistable spherical compliant mechanism undeflected 33

Figure 14. A macro scale model of the bistable spherical compliant mechanism showing the twist in compliant joint 2 33

Figure 15. Compliant joint 2 in a twisted configuration 34

Figure 16. Plot of finite element model input moment, M_2, and the input link rotation, θ_2 35

Figure 17. Plot of input moment, M_2, and the static jumps that occur during counter-clockwise and clockwise rotation of the input link, θ_2 36

Figure 18. Plot of bistable positions of the bistable spherical compliant mechanism 37

Abstract

The bistable spherical compliant mechanism (BSCM) is a novel device capable of large, repeatable, out-of-plane motion, characteristics that are somewhat difficult to achieve with surface micromachined microelectromechanical systems. An improved pseudo-rigid-body model (PRBM) to predict the behavior of the BSCM is presented. The new model was used to analyze seven different versions of the device, each with a different compliant joint length. The new model, which adds torsion, is compared with a finite element analysis (FEA) beam model. The new model more closely approximates the results yielded by FEA than previous models used to analyze the BSCM. Future work is needed to quantify stress-stiffening interactions between bending and torsion. Both FEA and the current models show that increasing the length of the compliant segment decreases the amount of force required to actuate the device.

Chapter 1

Introduction

Integrated circuit technology has allowed for the fabrication of increasingly complex micromechanisms, the majority of micromechanisms being found in a variety of sensors and optical switches. Although these sensors are used in a wide range of applications, they do not exhibit large mechanical motion. The displacements demonstrated by these micro mechanisms are, in the case of a sensor, most often used to measure a change through piezoresistive, thermal, or capacitive transduction methods (1), or in the case of a micro-mirror, alter the mirror's angle. A device such as the micro-mirror performs the largest displacement, but it is still a relatively small fraction of the size of the device. The development of large mechanical displacement is an area vital to expanding the functionality of micromechanisms. The bistable spherical compliant mechanism (BSCM), shown in Figure 1, exhibits large, repeatable, bistable, out-of-plane motion, when given a planar input, which provides a gateway to enhanced micromechanism functionality. Compliance in MEMS allows designs to be greatly simplified and averts the complications associated with micro-hinges. Bistability allows precise and accurate movement between two stable positions. Although the BSCM concept has previously been reported in a thesis by Choueifati (2), here we provide an alternative analytical model and an analysis of seven different configurations each with a different compliant joint length.

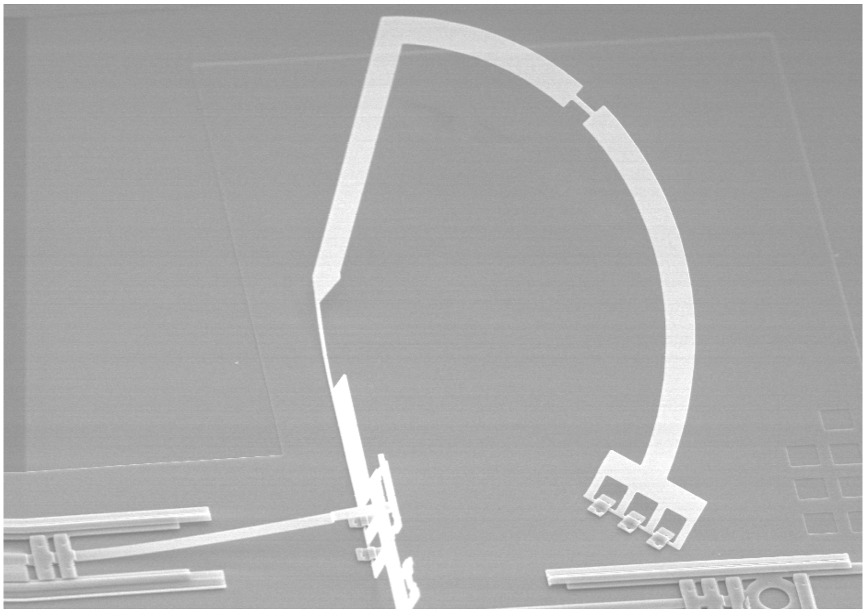


Figure 1. A scanning electron microscope image of a micro bistable spherical compliant mechanism

1.1 The Bistable Spherical Compliant Mechanism

The BSCM was fabricated using the Multi-User MEMS Process (MUMPS), a three-layer surface-micromachining method for polysilicon. Spherical geometry was used in the design of the mechanism. The BSCM has three basic components: two sliders and a compliant spherical four-bar mechanism, with links r_1, r_2, r_3, and r_4 as shown in Figure 2. The ground link is r_1, the input link r_2, r_3 is the coupler link, and r_4, the follower link. Links r_2 and r_4 are joined to the substrate by a staple hinge that allows 180^o rotation. Link r_3 is connected to r_2 and r_4 by compliant joints, nominally small-length flexural pivots (SLFPs) which are discussed in section 3.6.1. The two sliders act as mechanical actuators and are connected to the input link, r_2, by staple hinges (2). An analytical model in the form of a pseudo-rigid-body model (PRBM) is used to analyze the mechanism and compare with results obtained from a finite element analysis (FEA).

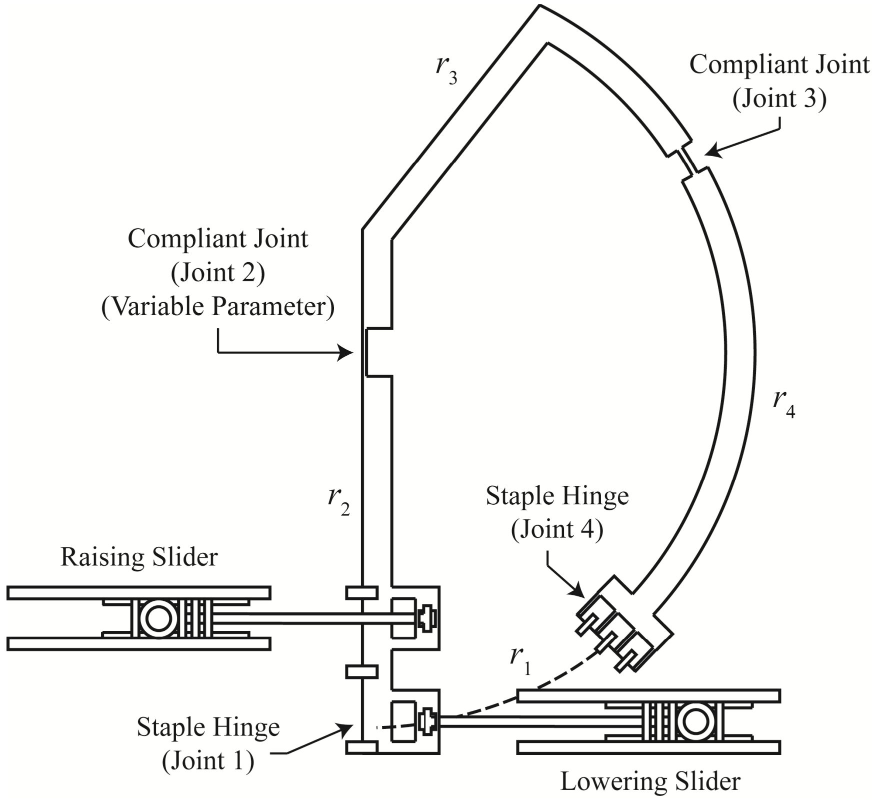


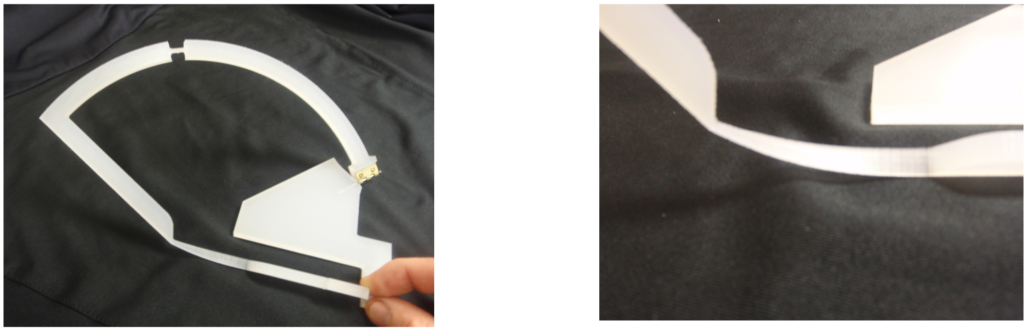
Figure 2. The bistable spherical compliant mechanism design and nomenclature

Chapter 2

Outline

2.1 Objective

The objective of this research is to improve the PRBM of the BSCM by including the torsion in joint 2 to a previous PRBM and comparing it to results obtained through FEA. The primary factor affecting the torsion of the device is assumed to be the length of compliant joint 2, thus seven versions of the device are analyzed, each with a different compliant joint length, as shown in Figure 2. The torsion in compliant joint 2 can be seen in Figure 3.



(a) (b)

Figure 3. (a) The bistable spherical compliant mechanism configuration after an applied input, (b) A close-up of the resultant twisting in compliant joint 2 due to an applied input

2.2 Motivation

The addition of new micromechanisms capable of large out-of-plane motion is crucial to the development of miniaturization technology. The lack of new designs is primarily due to fabrication and assembly limitations on the microscale, and the lack of design processes capable of predicting input-output, or force-deflection relationships, necessary to design micromechanisms to meet predetermined criteria in order to complete specific objectives. Spherical compliant mechanisms alleviate many of the complications associated with microfabrication, thus accurate PRBMs greatly benefit the advancement microsystem design.

2.3 Contribution

An improved PRBM for the BSCM is presented and compared to FEA results for seven different versions of the BSCM. The examination of the different versions of the device allowed for quantitative similarities to be observed indicating that the new PRBM captures the main characteristics of the input-output relationships.

2.4 Research Approach

A review of the various fields of knowledge applicable to the BSCM's fabrication, design, and analysis is presented to provide a background and reinforce the contribution of this work. The mathematical model and theory of the BSCM is then expressed, followed by the approach taken toward a finite element analysis. Finally, results and conclusions are presented.

Chapter 3

Background

The BSCM is designed using a multitude of methods from various fields of knowledge. An overview of the associated fields is presented, specifically: MEMS history, integrated circuit (IC) fabrication methods, 3D MEMS, ortho-planar mechanisms, spherical mechanisms, compliant mechanisms, pseudo-rigid-body models, and the method of virtual work.

3.1 Microelectromechanical System History

The possibilities and difficulties of MEMS were realized during the 1960s, and the first silicon based micromechanical structures were developed as early as 1965 (3). During the 1970s, numerous fabrication advancements were made that spurred the evolution of some of the most prevalent MEMS devices produced today. One of the earliest MEMS to become prevalent in industry was the pressure sensor, created using IC compatible technologies such as doping, thin film materials, and bonding (4). The importance of silicon as a mechanical material and its numerous fabrication methods were outlined in the 1980s (5)(6). A large drive from the automotive industry, based on a need for batch fabrication at low cost and reduced size, helped bolster the development of MEMS. The automotive industry benefits from the use of an array of micro-sensors including, accelerometers, mass/air flow sensors, load/force sensors, torque, position, temperature, light, and oxygen sensors (1).

Other applications of MEMS include structural health monitoring of critical aircraft components, bio-medical applications, micro-actuators, telecommunications, and IT peripherals such as mobile phones, PDA’s, laptops, and game consoles (7)(8)(9). Advancements in MEMS to these areas will continue to advance MEMS technology. More recent and future applications of MEMS, include micro mirrors, microphones, ink jet printing heads, and micro-spray nozzles (8).

3.2 Microelectromechanical System Fabrication

The most common methods for MEMS fabrication are bulk micromachining, surface micromachining, and LIGA (Lithographie Galvanoformung Abformung). Newer methods of MEMS fabrication include deep reactive ion etching (DRIE), extreme ultraviolet lithography (EUVL), charged particle beam lithography (E-Beam Lithography, EBL), ion beam lithography (I-Beam Lithography, IBL), and electrochemical fabrication (EFAB) (1). MEMS materials are limited to silicon, its compounds, glasses, and some metals, but it is gradually extending to more metals, ceramics, and polymers (10). Although the BSCM was fabricated using surface micromachining, each of the aforementioned fabrication methods were researched in order to asses if the design of the BSCM could be improved and to see if any other devices have been produced that are similar to the BSCM.

3.2.1 Bulk Micromachining

Bulk micromachining is a relatively fast and inexpensive method of fabricating MEMS due to its single substrate layer of single crystal material, most commonly silicon.

The silicon substrate is patterned using photolithography. A chemical etchant, such as potassium hydroxide, is used to remove selected areas. After etching is complete, the desired structure remains (1). Bulk micromachining cannot produce a design as complex as the BSCM.

3.2.2 Surface Micromachining

Surface micromachining differs from batch micromachining in that multiple layers are utilized, and etched, on top of an untouched substrate. This technology stems from IC's where dozens of layers is not uncommon. However, micromechanisms generally do not exceed six layers (1).

The separate layers used in surface micromachining are termed structural and sacrificial layers. Structural layers usually consist of a thin film of polysilicon, sacrificial layers, silicon dioxide. These thin films are generally one to two micrometers thick. The polysilicon is deposited using low pressure chemical vapor deposition and the silicon dioxide layers are grown by thermal oxidation. Each consecutive layer is patterned using photolithography and etching. Finally, the sacrificial layers are dissolved using hydrofluoric acid and the mechanism is said to have been released (11). This layered process allows the links of the BSCM to be fabricated free of the substrate, attached only by the staple hinges, so that actuation is possible.

3.2.3 LIGA (Lithographie, Galvanoformung, Abformung)

LIGA is an acronym for the German words lithographie galvanoformung abformung, meaning lithography, electroplating, and molding (12). LIGA uses x-ray radiation produced by a synchrotron to pattern a polymethyl methacrylate substrate into a mold. The mold is then electroplated with nickel, copper, or nickel iron. A chemical rinse is used to dissolve the mold leaving the metal part free and ready for assembly. The metal part may also be used as a mold to form a polymer or ceramic parts by injection molding. Advantages to LIGA are high aspect ratios, on the order of 100:1, and a greater selection of materials. The number of facilities able to perform LIGA is limited due to the necessity of a synchrotron, and the assembly of parts is usually required, which can be difficult (1).

3.2.4 New Fabrication Methods

There is a gradual shift toward using the term microsystem technology, as opposed to MEMS, as miniaturization science begins to expand to include systems that focus on dynamic motion without the need for integrating electrical circuitry. IC fabrication methods have thus become more diverse regarding the use of lithography techniques. IC's priorities require throughput, finer geometries, and batch processing whereas miniaturization science requires modularity, good depth of focus, extending the z-direction, incorporating nontraditional materials, replication, and batch fabrication is not always a prerequisite (1).

Newer fabrication methods such as, Extreme Ultraviolet Lithography (EUVL) (13), Deep X-Ray Lithography (DXRL) (14), Charged Particle Beam Lithography (E-Beam Lithography, EBL) (15), and Ion Beam Lithography (I-Beam Lithography, IBL) (16), succeed in achieving high aspect ratios, some at multiple angles to the substrate, with varying fabrication difficulties ranging from simple and fast methods to complex methods that require controlled work environments. Regardless of the improvement, the outcome is still dominantly planar, where additional benefits of angled designs can be negated by restricted complexity of design. Relatively organic shapes can be produced using a fabrication method termed EFAB (electrochemical fabrication). EFAB has the ability to produce true 3-D structures due to its ability to use a significant number of layers, several dozen, compared to say Sandia's SUMMIT V program which comprises of five layers.

Advances in MEMS fabrication have produced finer line widths, higher features, high aspect ratios, and the introduction of new materials. These new materials, such as SiC, allow MEMS to perform in environments with extreme temperature, vibration, and harsh chemical media. Polymers are becoming more prevalent due to their increased fracture strength, low Young's modulus, high elongation at break, and their compatibility in biological and chemical applications (17). Complex ortho-planar mechanisms continue to elude fabrication due to the limited capabilities of micromachining, microassembly, and layered fabrication methods imposing planar designs of MEMS (10). Surface micromachining is still the optimal choice for fabricating the BSCM.

3.3 3D Microelectromechanical Systems

Recent advancements to MEMS fabrication have focused on improving current planar methods to produce finer line widths and increased design heights, but the complexity of designs remains near the same (16)(17). Organic shapes may be fabricated using a high number of layers through methods such as EFAB, or microstereolithography, and the assembly of parts fabricated with robust materials through LIGA fabrication show promise for the future. Benefits of microassembly include combining parts that use different or incompatible materials or manufacturing processes and the ability to increase the number of layers of a device to form true 3D mechanisms. A review of micro-scale assembly technologies has been carried out by Cohn et al. (1998) and Probst et al. (2009) (9). EFAB, although able to create organic shapes, is still limited to planar design limitations, and microassembly is not economically viable due to time consuming assembly techniques.

3.4 Ortho-Planar Mechanisms

Orthogonally-Planar or ortho-planar mechanisms allow motion in one plane to be transferred in a direction perpendicular to that plane. Ortho-Planar designs are therefore able to take advantage of the planar nature of IC fabrication methods and can produce motion in the z direction, greatly increasing the functionality of MEMS devices.

3.5 Spherical Mechanisms

A spherical mechanism is one in which all axes of rotation of its revolute joints intersect at a single point that defines the origin of the sphere. Circles drawn on the surface of the sphere and in a plane that intersect the center of the sphere are termed great circles. The intersection of two great circles on a sphere defines a dihedral angle. The intersection of three great circles defines the sides of a spherical triangle, and the sides of the spherical triangle are termed great arcs. Each great arc has a corresponding dihedral angle that is opposite to it (18). Great circles, great arcs, dihedral angles and a spherical triangle are shown in Figure 4.

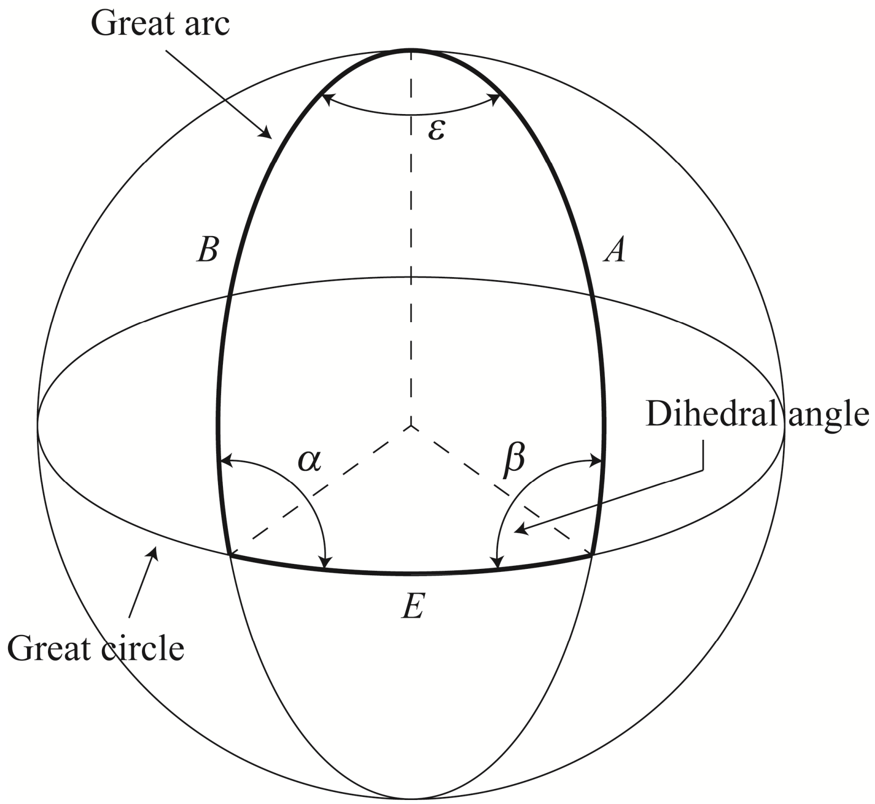


Figure 4. Great circles, great arcs, dihedral angles, and a spherical triangle shown on a sphere

3.5.1 Spherical Trigonometry

Much like planar geometry, laws of sines and cosines exist that are specific to spherical triangles (19). The spherical law of cosines is later used to derive the closed form equations of the BSCM. The spherical law of cosines can be applied to a spherical triangle using either three arcs and one dihedral angle or three dihedral angles and one arc. In the following equations, great arcs are represented by *A*, *B*, and *E*, and dihedral angles are represented by α, β, and ε, as shown in Figure 4.

Three arcs and one dihedral angle.

$$\cos(A) = \cos(B)\cos(E) + \sin(B)\sin(E)\cos(\alpha) \tag{1}$$

Three dihedral angles and one arc:

$$\cos(\alpha) = -\cos(\beta)\cos(\varepsilon) + \sin(\beta)\sin(\varepsilon)\cos(A) \tag{2}$$

3.6 Compliant Mechanisms

A Compliant mechanism transfers, or transforms motion, force, or energy using deflection of flexible members rather than moveable joints (21). The advantage of compliant mechanisms at the micro level are mainly due to deflection as opposed to constraints due to pin joints (10)(21). Compliant mechanisms reduce the total number of parts, reduce friction and wear, require less space and are less complex, have less clearance due to pin joints resulting in higher precision, require no assembly, integrate energy storage elements (springs) with other components, and can be fabricated in a plane, all of which are desirable in MEMS (21)(23). These benefits overcome mechanical problems such as: binding, clearances which are large compared to device sizes, friction, and wear (20). Compliant mechanisms are often monolithic adding to its desirability for batch fabrication of assembly free components, or entire devices, and complex motions are still achievable even with the constraints of micromachining (10).

A disadvantage of compliant MEMS is that performance is highly dependent on material properties and the selection of materials is limited to the few materials available for MEMS fabrication. Material properties do not always behave as predicted due to model sizing being of the same order as material grain size. Even so, compliant mechanisms can be very robust at the micro level and often last longer than do components that use pin joints or other elements that induce wear. Compliant mechanisms bridge the gap between planar fabrication and large out of plane motion by simplifying designs while achieving repeatable out of plane motion. There will continue to be a requirement for compliant mechanisms due to their precision, repeatability, and ability to perform bistability.

3.6.1 Small-Length Flexural Pivots

A small-length flexural pivot is a type of compliant joint that obtains its characteristic motion due to isolated bending in the short segment of the beam, as shown in Figure 5. The wider segment of the beam remains essentially straight due to its relatively high rigidity compared to the low rigidity of the narrower segment. The BSCM's compliant segments allow for the large deflections exhibited by the mechanism. Additional benefits of compliance in microsystems include: simplification of design, reduced part count, reduced wear, and lower production cost (21).

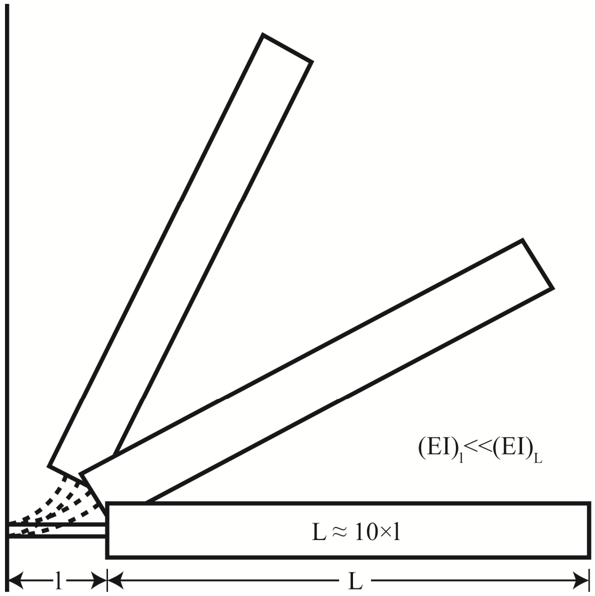


Figure 5. A small-length flexural pivot. Adapted from Howell

The replacement of two rigid-body pin-joints with small-length flexural pivots can be seen in the simplified diagram of the BSCM as shown in Figure 6. Although compliant joints 2 and 3 are fully compliant, the mechanism as a whole is termed 'partially compliant' due the combination of rigid-body joints and compliant joints. The partially compliant four-bar represented in Figure 6 is used as the basis for the pseudo-rigid-body model discussed next.

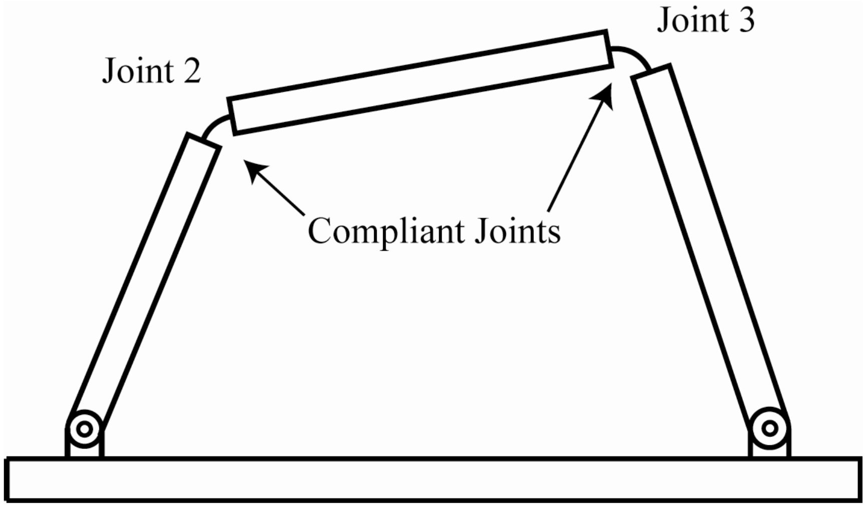


Figure 6. A simplified diagram of the bistable spherical compliant mechanism in the form of a partially compliant four-bar mechanism

3.7 The Pseudo-Rigid-Body Model

Pseudo-Rigid-Body models can be used to replace flexible members with rigid-body components that have equivalent force-deflection characteristics. Rigid-Body mechanics can then be used to analyze the compliant system (21). In the BSCM, small-length flexural pivots are replaced with pin joints and torsional springs, as shown in Figure 7, where k_1, k_2, and k_3 are the small-deflection stiffness constants.

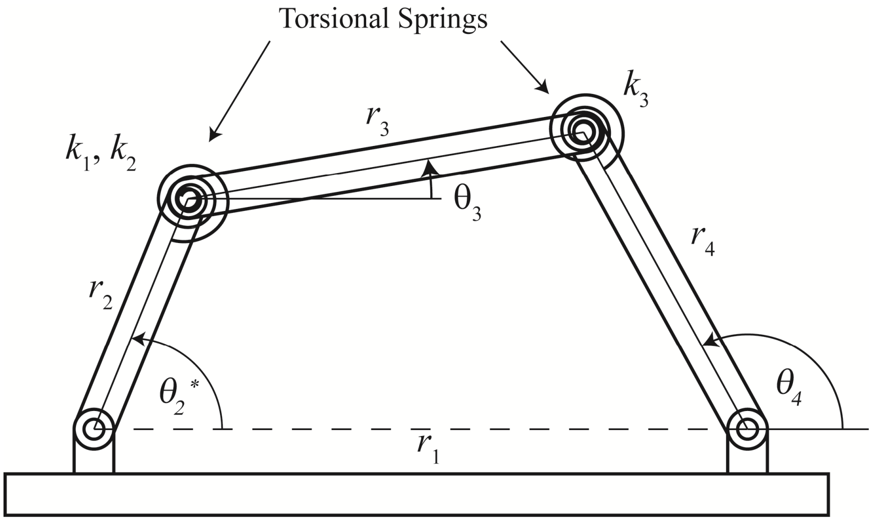


Figure 7. The bistable spherical compliant mechanism pseudo-rigid-body model

The pseudo-rigid-body model allows for the method of virtual work to be applied to the mechanism.

3.8 Virtual Work Analysis

The method of virtual work is used to derive the coupling of rigid links due to the presence of the torsional springs in the pseudo-rigid-body model. The virtual work is found by taking the derivative of the strain energy of the torsional springs and multiplying it by their respective virtual displacements (21).

3.8.1 General Form of a Torsional Spring

The general equation for torsional spring strain energy can be written as

$$V = \int_{\psi_o}^{\psi} m(\psi) d\psi \, , \; \psi = (\theta - \theta_o) , \; m = k(\theta - \theta_o) \tag{3}$$

where *m* is the moment.

After integration, the torsional spring strain energy, *V*, becomes

$$V = \frac{1}{2}k(\theta - \theta_o)^2 \tag{4}$$

3.8.2 Virtual Work

The virtual work is found from the derivative of the torsional spring strain energy with respect to the generalized coordinate to be

$$\delta W = -\frac{dV}{dq}\delta q \tag{5}$$

$$\delta W = -m(\psi)\frac{d\psi}{dq}\delta q \tag{6}$$

The virtual work due to a moment on a rigid link, M_i, and a virtual angular displacement, δq_i, where q_i is the generalized coordinate, is

$$\delta W = \sum M_i \cdot \delta q_i \tag{7}$$

The virtual work due to a moment at a small-length flexural pivot, T_{i}, and a virtual angular displacement, δq_{i}, where q_{i} is the generalized coordinate is

$$\delta W = \sum T_i \cdot \delta q_i \tag{8}$$

The total virtual work of the system is

$$\delta W = \sum M_i \cdot \delta q_i + \sum T_i \cdot \delta q_i \tag{9}$$

Chapter 4

Mathematical Model of the Bistable Spherical Compliant Mechanism

4.1 Spherical Geometry

The spherical geometry nomenclature of the BSCM is shown in Figure 8, where r_1, r_2, r_3, and r_4 represent great arcs and θ_2^*, α, β, γ, ε, and ζ represent the dihedral angles. The arc δ is used to separate the spherical four-bar into two spherical triangles in order to apply spherical trigonometry.

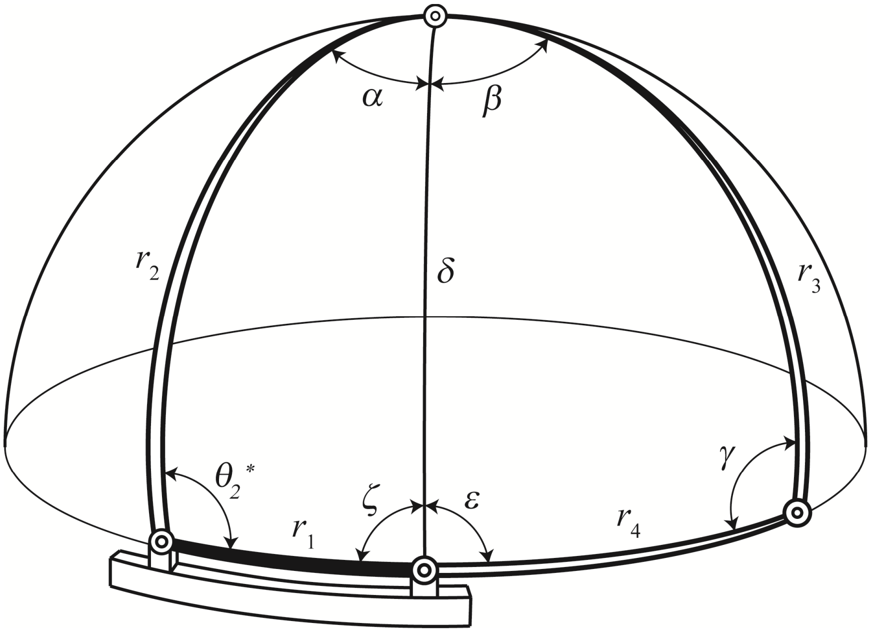


Figure 8. Spherical diagram and nomenclature for the bistable spherical compliant mechanism

4.1.1 Spherical Closed Form Equations

The variables δ, α, β, γ, ε, and ζ, are derived using the background outlined in section 3.5.1. Using the spherical law of cosines, equation (1), and Figure 8 one finds:

$$\delta = \cos^{-1}\left(\cos(r_1)\cos(r_2) + \sin(r_1)\sin(r_2)\cos(\theta_2^*)\right) \tag{10}$$

$$\alpha = \cos^{-1}\left(\frac{\cos(r_1) - \cos(r_2)\cos(\delta)}{\sin(r_2)\sin(\delta)}\right) \tag{11}$$

$$\beta = \cos^{-1}\left(\frac{\cos(r_4) - \cos(r_3)\cos(\delta)}{\sin(r_3)\sin(\delta)}\right) \tag{12}$$

$$\gamma = \cos^{-1}\left(\frac{\cos(\delta) - \cos(r_3)\cos(r_4)}{\sin(r_3)\sin(r_4)}\right) \tag{13}$$

$$\varepsilon = \cos^{-1}\left(\frac{\cos(r_3) - \cos(r_4)\cos(\delta)}{\sin(r_4)\sin(\delta)}\right) \tag{14}$$

$$\zeta = \cos^{-1}\left(\frac{\cos(r_2) - \cos(r_1)\cos(\delta)}{\sin(r_1)\sin(\delta)}\right) \tag{15}$$

These equations allow the calculation of the motion of the BSCM for a given input θ_2^*. The angle θ_2^* is equal to the input rotation θ_2 in the pure bending model and Ω in the twisting and bending model as discussed next.

4.2 The Twisting and Bending Model

To solve for the torsion of compliant joint 2, it is necessary to know two angles to define the angle of twist. In this case, the primary angle is θ_2, and the secondary angle, Ω. The primary angle, θ_2, is the exact value of the input link, r_2, as it is rotated off of the substrate; consequently, it is also the angle at the end of compliant joint 2 that connects to the input link, r_2. The secondary angle, Ω, defines the angle of twist in compliant joint 2. It is the angle at the end of compliant joint 2 that connects to the coupler link, r_3.

As shown in Figure 9, the angle Ω shares the same axis of rotation as θ_2. From a kinematic standpoint (neglecting the torsional spring that couples their motion), the motion of the spherical four-bar depends entirely on the rotation of Ω, and not at all on the rotation of θ_2. The method of virtual work is used to derive the coupling due to the presence of the torsional spring in the PRBM.

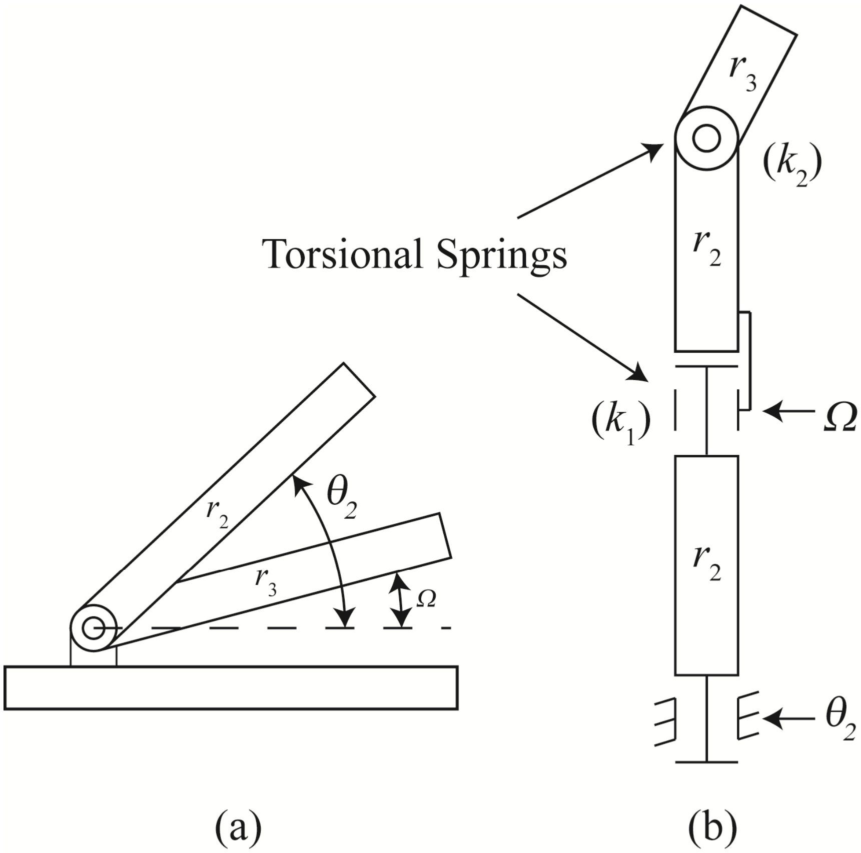


Figure 9. Pseudo-Rigid-Body model of link 2: Part (a) front view, Part (b) top view

A system of two equations was necessary to solve for Ω as a function of θ_2. The first equation of virtual work is derived with θ_2 as the independent variable and therefore concentrates on the input link, r_2. The second equation of virtual work is derived with Ω as the independent variable and includes the twisting and bending associated with compliant joint 2 and the bending of compliant joint 3. The second equation of virtual work captures the deflection of the mechanism that is responsible for the twisting in compliant joint 2.

The method of virtual work was also used to solve for the moment of joint 1 as a function of θ_2. In other words, the moment required to actuate the BSCM for a given input.

4.3 Virtual Work

4.3.1 First Equation of Virtual Work

The first equation of virtual work is used to form a correlation between the generalized co-ordinates θ_2 and Ω. This correlation allows us to solve for $\theta_2(\Omega)$, which is necessary to define the twist. The virtual work with the generalized co-ordinate as θ_2 is

$$\delta W = M_2 \cdot \delta\theta_2 + T_1 \cdot \delta\psi_1 + T_2 \cdot \delta\psi_2 + T_3 \cdot \delta\psi_3 \tag{16}$$

$$0 = \left(M_2 - m_1(\psi_1)\frac{\partial\psi_1}{\partial\theta_2} - m_2(\psi_2)\frac{\partial\psi_2}{\partial\theta_2} - m_3(\psi_3)\frac{\partial\psi_3}{\partial\theta_2} \right)\delta\theta_2 \tag{17}$$

The moments associated with the torsion and bending of compliant joint 2, are m_1, and m_2, respectively, and the moment associated with the bending of compliant joint 3, m_3, are defined as

$$m_1 = -k_1\psi_1,\ m_2 = -k_2\psi_2,\ m_3 = -k_3\psi_3 \tag{18}$$

where k_1, k_2, and k_3, are small-deflection stiffness constants found in Ugural et al. (22). The value $k_{1,}$ is the torsional spring constant, and k_2 and k_3 are the bending spring constants. Each spring constant is dependent on the material and geometric properties of the compliant joints of the BSCM. *J* is the *polar moment of inertia*, *G* is the *shear modulus of elasticity*, *L* is the *length* of the compliant segment, *E* is *Young's modulus, I* is the *second moment of area*, v is *Poisson's ratio*, and β is a constant dependant on the width and height of the rectangular cross section of compliant joint 2 (22). The torsional small-deflection stiffness constant is defined as

$$k_1 = \frac{JG}{L_2} \tag{19}$$

where the product JG is the torsional rigidity (22) and the length L_2 is the length of compliant joint 2.

$$J = \beta_T ab^3 \tag{20}$$

$$J = \beta_T wh^3, \; \beta_T = 0.2125 \tag{21}$$

$$G = \frac{E}{2(1+\nu)}, \; E = 169 \times 10^3 MPa, \; \nu = 0.35 \tag{22}$$

The bending spring constants are defined as

$$k_2 = \frac{EI_2}{L_2} \tag{23}$$

$$k_2 = \frac{E(wh^3)_2}{L_2} \tag{24}$$

$$k_3 = \frac{EI_3}{L_3} \tag{25}$$

$$k_3 = \frac{E(wh^3)_3}{L_3} \tag{26}$$

The angular displacements of the torsion of compliant joint 2, the bending of compliant joint 2, and the bending of compliant joint 3 are defined by ψ_1, ψ_2, and ψ_3 respectively. The angle ψ_1 is the difference between the primary and secondary angles of compliant joint 2, which defines the angle of twist, as seen in Figure 9.

$$\psi_1 = (\theta_2 - \Omega) \tag{27}$$

The angle ψ_2, is the angle between links r_2 and r_3, as defined in Figure 8, and the angle ψ_3 is the angle between links r_3 and r_4, also defined in Figure 8.

$$\psi_2 = (\alpha_o - \alpha) + (\beta_o - \beta), \ \psi_3 = (\gamma_o - \gamma) \tag{28}$$

The partial derivatives of the angular displacements with respect to the generalized co-ordinate are termed "kinematic coefficients" (sensitivities) and are determined in eq. (24 & 25). They are based on the assumption that θ_2 and Ω are kinematically independent but statically coupled due to the torsional spring. In this way, the kinematic coefficient of torsion in compliant joint 2 becomes

$$\frac{\partial \psi_1}{\partial \theta_2} = 1 \tag{29}$$

and the kinematic coefficient of bending in compliant joints 2 and 3 both become 0, as they do not depend on the generalized co-ordinate θ_2.

$$\frac{\partial \psi_2}{\partial \theta_2} = 0, \ \frac{\partial \psi_3}{\partial \theta_2} = 0 \tag{30}$$

Equation (30) simplifies equations (16 & 17) to

$$0 = M_2 + T_1 \tag{31}$$

Equation (31) can now be substituted into the second equation of virtual work to solve for $\theta_2(\Omega)$.

4.3.2 Second Equation of Virtual Work

The second equation of virtual work is based on the generalized co-ordinate Ω and takes into consideration both the torsion and bending of joint 2 and the bending of joint 3. The spring stiffness', k_1, k_2, and k_3, and the angular displacements ψ_1, ψ_2, and ψ_3, are the same as in the first equation of virtual work.

$$\delta W = M_2 \cdot \delta\theta_2 + T_1 \cdot \delta\psi_1 + T_2 \cdot \delta\psi_2 + T_3 \cdot \delta\psi_3 \tag{32}$$

$$0 = \left(M_2 \frac{\partial\theta_2}{\partial\Omega} - m_1(\psi_1)\frac{\partial\psi_1}{\partial\Omega} - m_2(\psi_2)\frac{\partial\psi_2}{\partial\Omega} - m_3(\psi_3)\frac{\partial\psi_3}{\partial\Omega} \right)\delta\Omega \tag{33}$$

The first kinematic coefficient is zero due to θ_2's independence of Ω.

$$\frac{\partial\theta_2}{\partial\Omega} = 0 \tag{34}$$

The torsion and bending kinematic coefficients are based on the partial derivatives of equations (27 & 28) with respect to Ω, where the angles α, β, and γ are found in equations (10-12) and are dependent on δ in equation (9). For simplicity, $c(x)=cos(x)$, $s(x)=sin(x)$, $s^2(x)=sin^2(x)$, etc.

$$\frac{\partial\psi_1}{\partial\Omega} = -1 \tag{35}$$

$$\frac{\partial\psi_2}{\partial\Omega} = \left(\frac{\partial\alpha}{\partial\Omega}\right) + \left(\frac{\partial\beta}{\partial\Omega}\right) \tag{36}$$

$$\frac{\partial\alpha}{\partial\Omega} = -\left(\frac{s(r_1)s(\Omega)}{s(\alpha)s^3(\delta)}\right)\left(c(r_2)s^2(\delta) + c(r_2)c^2(\delta) - c(r_1)c(\delta)\right) \tag{37}$$

$$\frac{\partial\beta}{\partial\Omega} = -\left(\frac{s(r_1)s(r_2)s(\Omega)}{s(r_3)s(\beta)s^3(\delta)}\right)\left(c(r_3)s^2(\delta) + c(r_3)c^2(\delta) - c(r_4)c(\delta)\right) \tag{38}$$

$$\frac{\partial\gamma}{\partial\Omega} = \left(\frac{s(r_1)s(r_2)s(\Omega)}{s(r_3)s(r_4)s(\gamma)}\right) \tag{39}$$

From equation (33) and substituting equations (34-39) we find:

$$0 = -T_1 + T_2 \cdot \frac{\partial \psi_2}{\partial \Omega} + T_3 \cdot \frac{\partial \psi_3}{\partial \Omega} \tag{40}$$

$$0 = k_1 \left(\theta_2 - \Omega \right) + T_{2.} \cdot \frac{\partial \psi_2}{\partial \Omega} + T_3 \cdot \frac{\partial \psi_3}{\partial \Omega} \tag{41}$$

$$\theta_2 = \Omega - \frac{1}{k_1} \left(T_2 \cdot \frac{\partial \psi_2}{\partial \Omega} + T_3 \cdot \frac{\partial \psi_3}{\partial \Omega} \right) \tag{42}$$

From equation (33) and substituting equations (31 & 34-39) we find:

$$0 = -T_1 + T_2 \cdot \frac{\partial \psi_2}{\partial \Omega} + T_3 \cdot \frac{\partial \psi_3}{\partial \Omega} \tag{43}$$

$$0 = M_2 + T_2 \cdot \frac{\partial \psi_2}{\partial \Omega} + T_3 \cdot \frac{\partial \psi_3}{\partial \Omega} \tag{44}$$

$$M_2 = -\left(T_2 \cdot \frac{\partial \psi_2}{\partial \Omega} + T_3 \cdot \frac{\partial \psi_3}{\partial \Omega} \right) \tag{45}$$

The moment required to actuate the device can now be plotted against the input, θ_2.

Chapter 5

The Finite Element Model

The finite element model was constructed in ANSYS using the BEAM4 beam element model, a 3D elastic beam. The large deflection of the BSCM required the use of the non-linear iterative solver. The BEAM4 model allows six degrees of freedom, three in translation, and three in rotation. The beam is constrained to allow rotation about the Y axis at two points: one at the staple hinge of joint 1, and one also on link, r_2, but before compliant joint 2. The staple hinge at joint 4 is also constrained to only allow rotation about the x and y axis in Figure 10. The staple hinges can be viewed in Figure 2 and the constraints can be viewed in Figure 10.

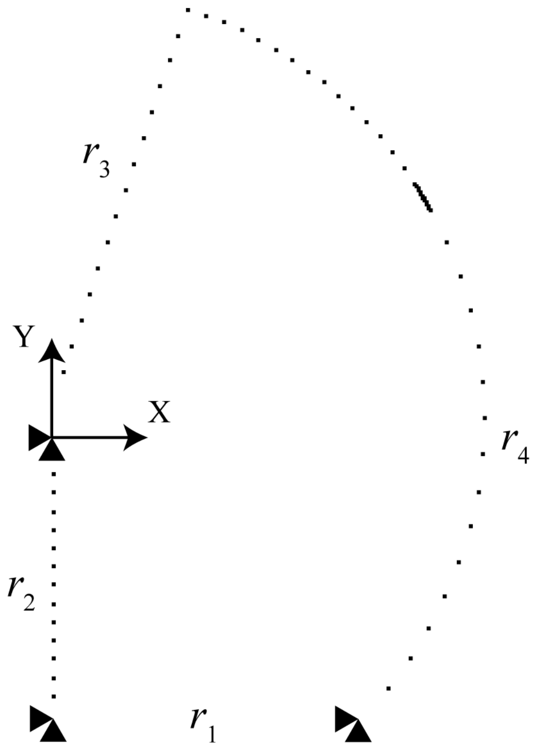


Figure 10. Plot of the nodal configuration and constraints used to obtain finite element analysis results

Obtaining the FEA results took a great deal of time due to singularities encountered while rotating the model from its initial position, as shown in Figure 10, through -180^o. The sign of the rotation of the model is based on the right hand rule applied to Figure 10. Each time a singularity was encountered, the program failed to converge and no data was stored. Multiple singularities were encountered for each model as a result of the BSCM storing potential energy and then releasing that potential energy, causing the mechanism to flip into a new orientation for a single value of input. This phenomenon is examined in chapter 6. Each time a singularity was encountered, the program needed to be switched from displacement to force mode, or vice-versa. A counter was placed within the program in order to gain an understanding of the amount of steps completed before divergence occurred and the program ended. Although the counter provided an approximate location of the divergence, it did not discern the step size required to successfully reach the end of one mode, to continue the program in the alternate mode. If the step sizes were large, the program would end and it would not be known how far to turn back. If the step sizes were too small, the program could be ran successfully, but it would not be known how far the program could go before a singularity was reached. Thus, the results were achieved through trial and error and can be viewed in Figure 16.

Chapter 6

Results

6.1 The Pure Bending Model

Choueifati (2) described the pure bending model in 2007, in which the bending of compliant joints 2 and 3 were analyzed, but not the torsion. The results of Choueifati's model is provided in Figure 11 to compare to new results achieved by the twisting and bending model discussed in the next section.

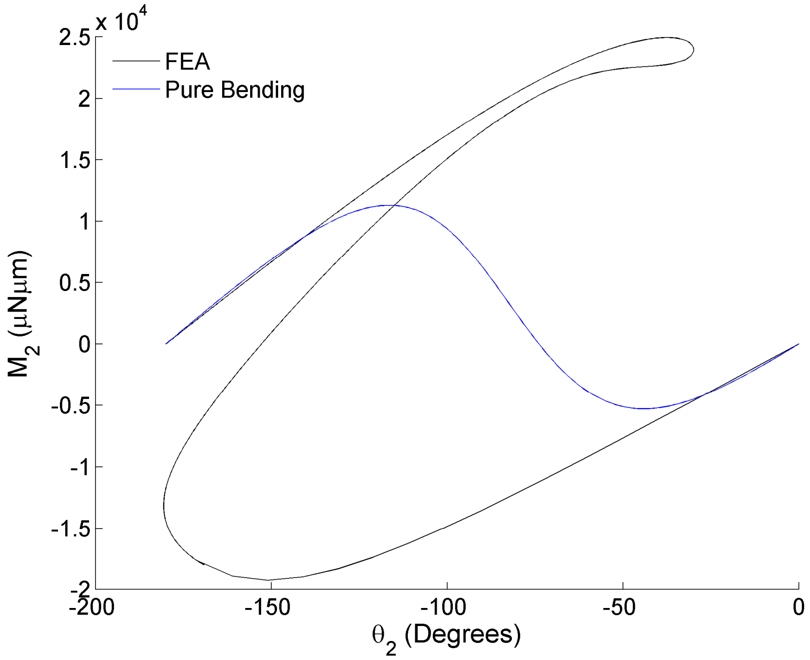


Figure 11. For Length 5, plots of input moment, *M_2*, and the input link rotation, *θ_2*, for the pure bending model and finite element model

The model yields a smooth curve, some initial similarities to the FEA model, and captures the change in direction of the moment. It does not incorporate the twisting in compliant joint 2.

6.2 The Variable Parameter: The Length of Compliant Joint 2

The twisting and bending model, discussed next, differs based on the length of compliant joint 2. Seven different lengths of compliant joint 2 are studied; however, the width and height of compliant joint 2 remains constant. The geometry of the rigid members remains constant, as does the geometry of compliant joint 3, as shown in Table 1.

Table 1. Model Parameters

		Width (μm)	Height (μm)	Length (μm)
Rigid Members		25	2	----------
Joint 3		5	2	22.17
Joint 2	Length 1	3.5	2	30
	Length 2	3.5	2	40
	Length 3	3.5	2	50
	Length 4	3.5	2	60
	Length 5	3.5	2	80
	Length 6	3.5	2	100
	Length 7	3.5	2	120

6.3 The Twisting and Bending Model

The twisting and bending model incorporates the twisting that occurs in compliant joint 2 to the PRBM. The results of the new model are shown in Figure 12. It is immediately apparent that the results of the new model achieve a significant improvement over the results of the previous pure bending model of Figure 11. The twisting and bending model captures the highly non-linear characteristics of the BSCM that are also seen in the FEA analysis as shown in Figure 16. The highly non-linear behavior is a result of the compliance and stiffness of the BSCM, where the stiffness is indicated by the slope of the line. Although the correct stiffness trend is captured, the trend in the maximum moment is only partially correct.

The trend in the moment is easiest described by looking at values above and below the value of zero moment. Below the value of zero moment, in the negative values, the twisting and bending model shows a slight increase in the maximum moment to actuate the device as the length of compliant joint 2 increases. FEA however, shows the opposite trend, a decrease in the maximum moment, as shown in Figure 16. The FEA model shows the correct trend and although the twisting and bending model does not, future work will improve the trend. Above values of zero moment, the positive values, the correct trend does exist in the twisting and bending model. Both the twisting and bending, and FEA, models indicate an increasing maximum moment as the length of compliant joint 2 increases.

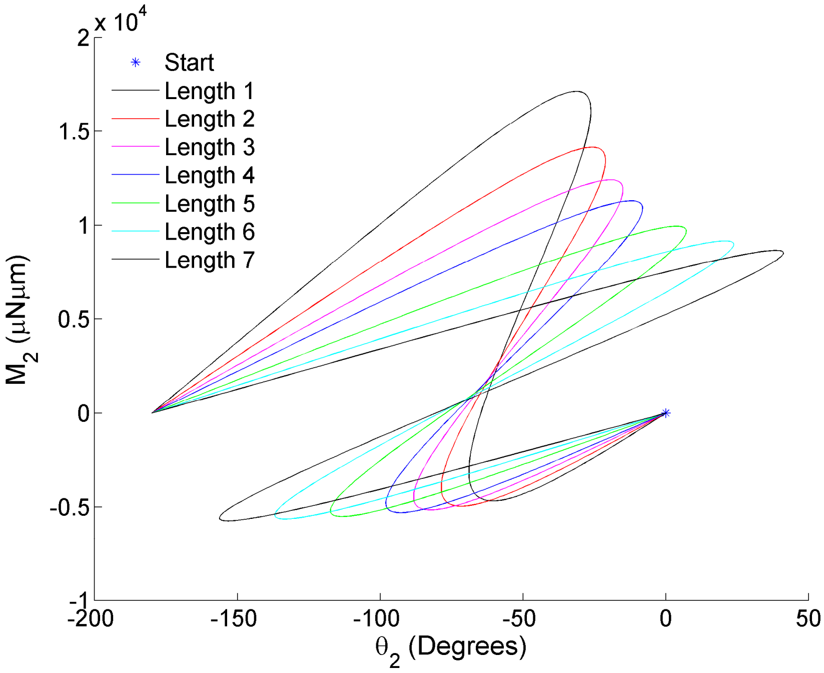


Figure 12. Plot of twisting and bending model input moment, M_2, and the input link rotation θ_2

The highly non-linear behavior of the BSCM, as shown in Figure 12, is in part due to the mechanism having multiple moment values for a single input rotation value.

This is a result of the compliant segments storing potential energy, and at a certain point, releasing it back into the mechanism. The rotation of the input link, r_2, produces a displacement of the follower link, r_4, that is controlled by the rotation of the input link and the potential energy in the system. As the input link, r_2, is rotated, θ_2 increases, as does θ_4, but at a slower rate due to storage of potential energy in the compliant joints. The input link rotation, $\theta_{2,}$ is able to increase to a maximum value termed by Young et al. (24) as 'the load threshold', at which point the stored potential energy is converted to kinetic energy and the mechanism moves spontaneously, or jumps, from one statically stable position to a different one. The tendency for the mechanism to jump is the cause of the multi-valued plots of Figure 12 and Figure 16, and is discussed further in section 6.6.

6.4 Macro Scale Models

It is observed in Figure 12, that as the length of the SLFP of compliant joint 2 increases, the moment required to actuate the device increases. This is not the intuitive relationship one would deduce from adding increased flexibility to the mechanism. The results of the FEA analysis, Figure 16, show that increasing the length of compliant joint 2 decreases the moment required to actuate the device. This is what one would expect due to the reduced torsional spring constant associated with longer beam lengths where low torsion and high rotation would occur before the mechanism achieved large displacements. Macro scale models supported this theory by exhibiting the characteristics portrayed by the FEA results. An example of the macro scale models can be seen in Figure 13, Figure 14, and Figure 15.

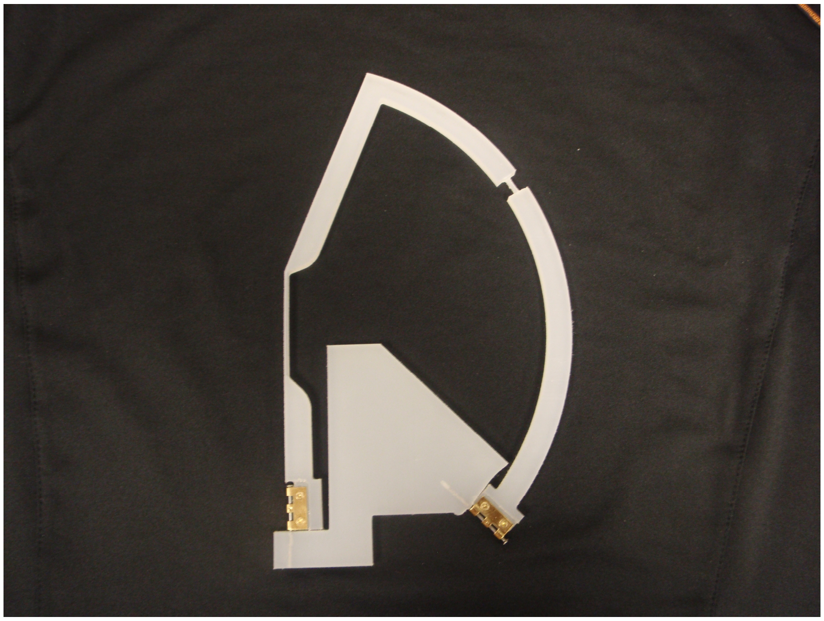


Figure 13. A macro scale model of the bistable spherical compliant mechanism undeflected

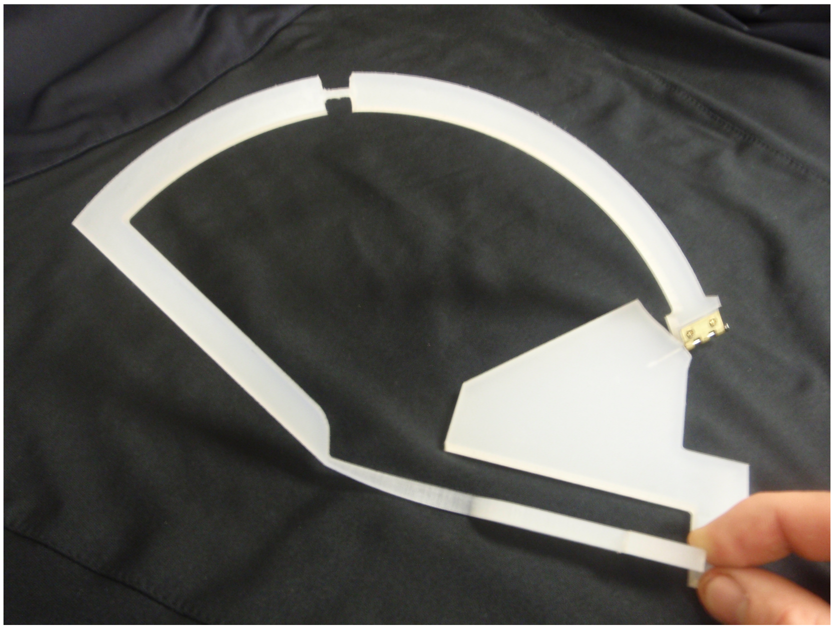


Figure 14. A macro scale model of the bistable spherical compliant mechanism showing the twist in compliant joint 2

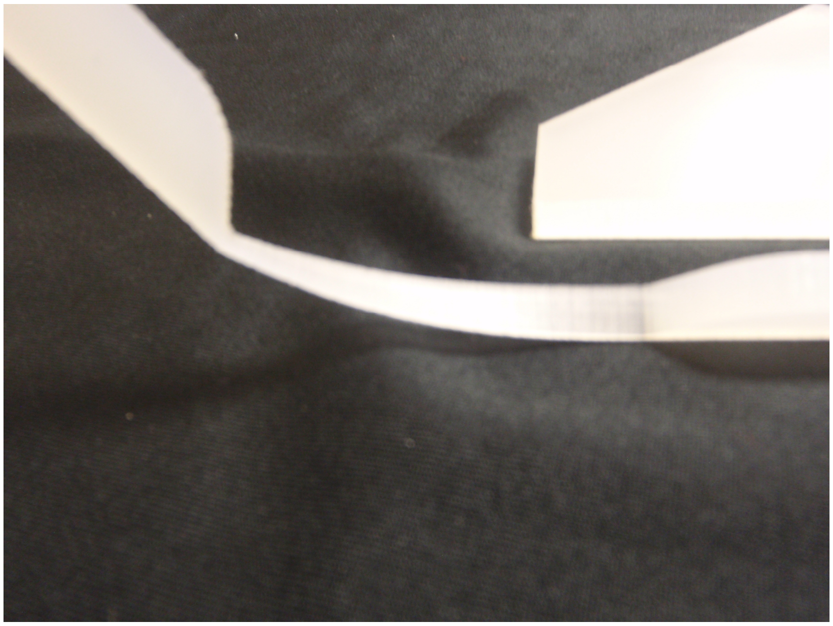


Figure 15. Compliant joint 2 in a twisted configuration

6.5 Finite Element Analysis: Beam Element Model

The finite element analysis results in Figure 16 indicate qualitative similarities compared to the new analytical model. Although qualitative similarities are observed, it is noted that the moment is greater than in the analytical model, suggesting that our current analytical model underestimates the stiffness of the mechanism. Thus, the bending and torsional stiffness' given in equation (19, 24, 26), do not account for stress-stiffening interactions that occur when a bent beam is twisted or when a twisted beam is bent.

Building the FEA model required switching between load control and displacement control as singularities were encountered. As each singularity was reached, the final load step was recorded, in order to begin the program again starting from the last load step but in the new control mode.

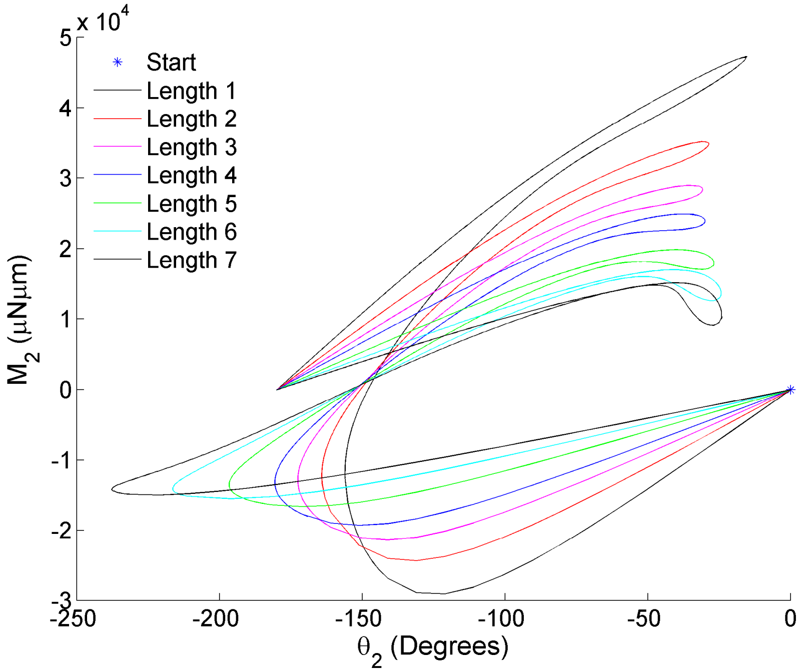


Figure 16. Plot of finite element model input moment, M_2, and the input link rotation, θ_2

6.6 Static Jumps

The plots in Figure 12, and Figure 16, predict some unusual behaviors. As a moment is applied to the input link, $r_{2,}$ rotation occurs until the load threshold of θ_2 is reached, at which point multiple values of M_2 are seen for the same rotation. In order to continue motion, the plots suggest that the moment must be released and then applied in the opposite direction, to allow a decrease in rotation before resuming the desired counter-clockwise rotation. This suggested release of moment and backwards motion is not characteristic of four-bar motion, or compliant mechanisms, and does not exist in physical models. Rather, as a moment is applied to the input link, r_2, rotation occurs to its initial maximum value, and its corresponding initial moment, and then jumps to a second value of moment before continuing to the final point of rotation at -180°. This phenomenon was termed a static jump by Young et al. (24) and is shown graphically in Figure 17.

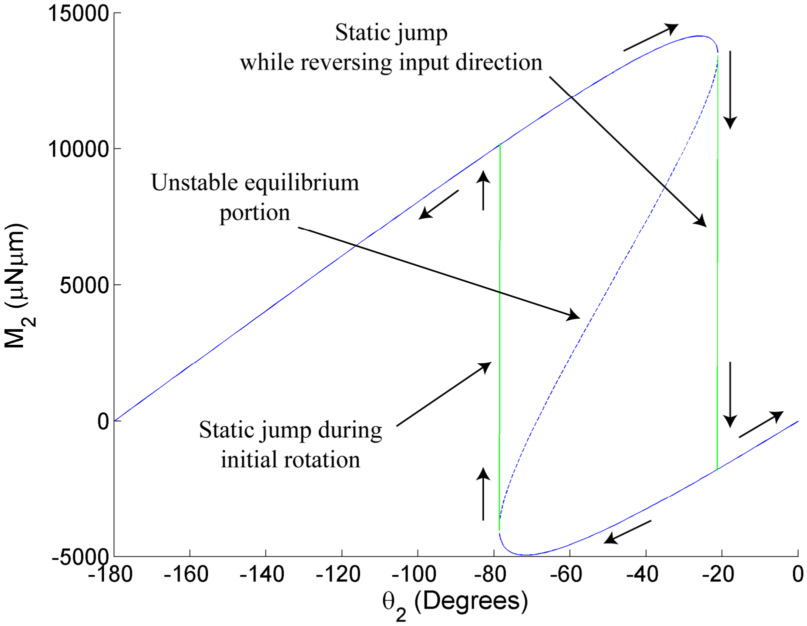


Figure 17. Plot of input moment, M_2, and the static jumps that occur during counter-clockwise and clockwise rotation of the input link, θ_2

6.7 Bistability

The deflection of compliant members incorporates both motion control and energy storage leading to the possibility of accurate and precise bistable mechanisms without the use of joints and springs. Bistability is a highly desired characteristic in micromechanisms because power is not necessary to hold the mechanism in either of its stable positions (25). The BSCM has two stable positions at 0^o and 180^o, as shown in Figure 18.

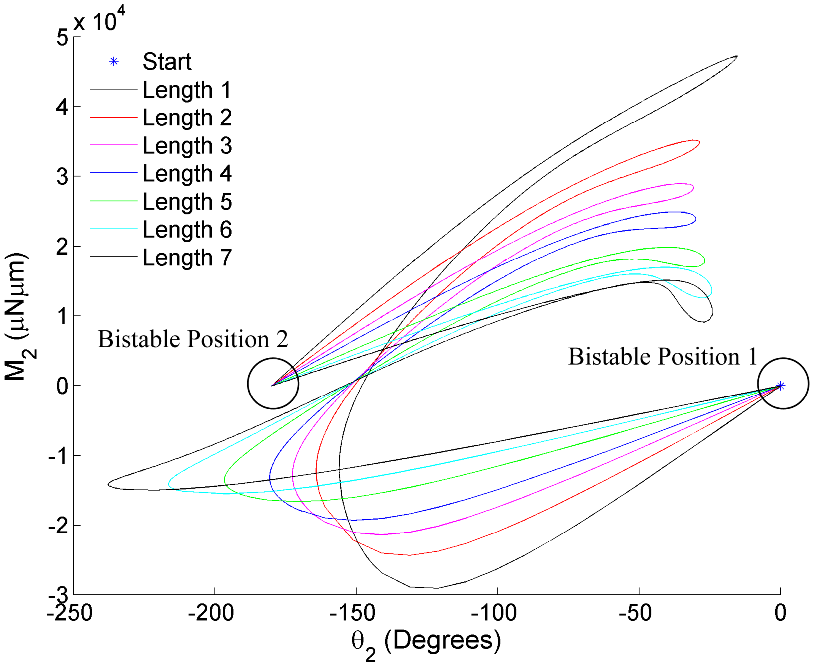


Figure 18. Plot of bistable positions of the bistable spherical compliant mechanism

Chapter 7

Conclusion

The bistable compliant spherical mechanism adds a new mechanism to the very limited field of ortho-planar micromechanisms. The addition of twist to the analytical model shows a marked improvement over the previous model and allowed for the quantification of the dynamic characteristics unique to the bistable spherical compliant mechanism. The analysis of seven different lengths of compliant joint 2 allowed for the primary factor affecting the torsion of compliant joint 2 to be examined. Quantitative similarities are shown indicating that the model captures the main characteristics that define the input-output relationships but the model still underestimates the overall rigidity. Future work will determine the appropriate pseudo-rigid-body model for mechanisms with bending and torsion interactions.

References

(1) Madou, M. (2002), *Fundamentals of microfabrication*. Boca Raton, NY: CRC Press.

(2) Choueifati, J. G. (2007). Design and modeling of a bistable spherical compliant mechanism, M.S. Thesis. University of South Florida

(3) Frazier, B. Warrington, R.O. Friedrich, C. (1995). The miniaturization technologies: past, present, and future. *IEEE Transactions on Industrial Electronics 42*, no. 5, pp 423-430.

(4) Smith, J.H. Eaton, W.P. (1997). Micromachined pressure sensors: review and recent developments. Office of scientific and technical information

(5) Lee, A. P. (1996), Microelectromechanical systems (MEMS) for engineers, Lawrence Livermore National Laboratory, Livermore, CA.

(6) Fujita, H. (1997). A decade of MEMS and its future. Institute of Industrial Science, University of Tokyo.

(7) Polla, D. (1997), MEMS for integrated diagnostic applications, *American Society of Mechanical Engineers*, Tribology Division, vol 7, pp. 19-24.

(8) Urban, A., Laermer, F., (2008), Bosh process-DRIE success story, new applications and products, Proceedings of the Material Research Society, Jan.

(9) Tamadazte, B. Marchand, E. Dembele, S. Li Fort-Piat, N. (2010). CAD model-based Tracking and 3D visual based control for MEMS microassembly, international journal of robotics research, vol. 29, no. 11, pp 1416-34.

(10) Ananthasuresh, G. K. Howell, L. L. Mechanical design of compliant microsystems-a perspective and prospects, 2005, *Journal of Mechanical Design*, vol. 127, no. 4, pp 736-738.

(11) Sniegowski, J.J. (1996). Multi-Level polysilicon surface micromachining technology: applications and issues. international mechanical engineering congress and exhibition, Atlanta, GA, Nov 17-22.

(12) Bonivert, W.D. et al. (1999). LIGA micromachining: infrastructure establishment. Office of scientific and technical information

(13) Anderson, C.N. (2009). Extreme ultraviolet lithography: a few more pieces of the puzzle, Lawrence Berkeley National Laboratory; Oak Ridge, Tenn

(14) Green, G.K. (1977). Synchrotron radiation: its characteristics and applications. Brookhaven National Laboratory ; Oak Ridge, Tenn

(15) Liu, J. Vaclav, V. (2007). Synchrotron facilities and free electron lasers. United States. Dept. of Energy ; Oak Ridge, Tenn

(16) Lindroos, V. Markku, T, Lehto, A. Motooka, T. (2010) *Handbook of Silicon Based MEMS Materials and Technologies*. William Andrew Publishers, Oxford. UK.

(17) Adams, T.M. Layton, R.A. (2010). *Introductory MEMS: fabrication and applications*. Springer. New York.

(18) Henderson, D. W. (2001). *Experiencing geometry in euclidian, spherical, and hyperbolic spaces.* Upper Saddle River, NJ: Prentice Hall.

(19) Spiegel, M.R. Liu, J. (1999). *Schaum's outlines: mathematical handbook of formulas and tables*. McGraw-Hill, Washington, D.C

(20) Kota, S. Ananthasuresh, G.K. Crary, S. Wise, K.D. (1994). Design and fabrication of microelectromechanical systems. *Journal of Mechanical Design* 116, pp 1081-1088.

(21) Howell, L. (2001). *Compliant mechanisms*. New York, NY: John Wiley & Sons.

(22) Ugural, C. A., & Fenster, S. K. (2003). *Advanced strength and applied elasticity*. Hamilton in Casleton, NY: Pearson Education.

(23) Howell, L. Midha, A. (1994). A method for the design of compliant mechanisms with small-length flexural pivots. *Transactions of the ASME* 116, pp 280-290.

(24) Young, S.O., Kota, S. (2009). Synthesis of multistable equilibrium compliant mechanisms using combinations of bistable mechanisms. ASME. *Journal of Mechanical Design.* Feb. Vol. 131.

(25) Opdahl, P.G. Jensen, B.D. Howell, L. (1998). An investigation into compliant bistable mechanisms. *Design Engineering Technical Conferences.* Atlanta: ASME, pp 1-10.

Appendices

Appendix A: ANSYS Batch Files

The following code was originally written and documented in a thesis by Choueifati, and has since been amended as documented in this thesis. Choueifati wrote code for seven different models, but only model 7 was functional. The following ANSYS batch files provide a working code for the seven different models.

A.1 Batch File 1

```
!*************************************
/CONFIG,NRES,1000000
!/CWD,'C:\Documents and Settings\aleon2\Desktop\Work'

!*************************************



!*****************************************
!******* Set Up Model Variables *********
!*****************************************


/title,3D Beam Non-linear Deflection
/PREP7


R=315  ! length in micrometers
PI=acos(-1.)
h1=2
b1=25

b2=5
h2=2

b3=27.6
h3=2


b4=3.5
h4=2
!*********** Area properties **************
A1 = h1*b1
```

```
Iy1= 1/12*b1*h1*h1*h1

Iz1= 1/12*h1*b1*b1*b1
E1= 169E3  ! Young's modulus in MPa, Force will be micro Newtons

!***********************************

A2=  h2*b2

Iz2= 1/12*h2*b2*b2*b2
Iy2= 1/12*b2*h2*h2*h2

E2= 169e3

!***********************************

A3=  h3*b3

Iz3= 1/12*h3*b3*b3*b3
Iy3= 1/12*b3*h3*h3*h3

E3= 169e3
!***********************************

A4=  h4*b4

Iz4= 1/12*h4*b4*b4*b4
Iy4= 1/12*b4*h4*h4*h4

E4= 169e3

!***********************************

!********** Declare an element type: Beam 4 (3D Elastic) *********

ET,1,BEAM4
KEYOPT,1,2,1
KEYOPT,1,6,1


!********** Set Real Constants and Material Properties *********


R,1,A1,Iy1,Iz1,h1,b1, ,          !******Check on the assumptions being made ******
```

```
R,2,A2,Iy2,Iz2,h2,b2, ,

R,3,A3,Iy3,Iz3,h3,b3, ,

R,4,A4,Iy4,Iz4,h4,b4, ,

MPTEMP,1,0
MPDATA,EX,1,,E1
MPDATA,PRXY,1,,0.35              ! Material properties for material 1 and 2

MPTEMP,1,0
MPDATA,EX,2,,E2
MPDATA,PRXY,2,,0.35

MPTEMP,1,0
MPDATA,EX,3,,E3
MPDATA,PRXY,3,,0.35

MPTEMP,1,0
MPDATA,EX,4,,E4
MPDATA,PRXY,4,,0.35

!***************************************



!********** Create Keypoints 1 throug 7: K(Point #, X-Coord, Y-Coord, Z-Coord)
*********

K,1,0,0,0,
K,2,0,-220,0,
K,3,0,-15,0,
K,4,0,15,0,
K,5,R*cos(PI/180*72),R*SIN(PI/180*72),0,
K,6,R*cos(PI/180*33),R*sin(PI/180*33),0,
K,7,R*cos(PI/180*29),R*sin(PI/180*29),0,
K,8,R*cos(PI/180*44),-R*sin(PI/180*44),0,

!********* Create Beam using Lines and Arcs and divide into segments *********


LSTR,     2,     3
LSTR,     4,     5
```

Appendix A (Continued)

```
LSTR,     3,     4           ! Draws lines connecting keypoints 1 through 7

LSTR,     6,     7
LARC,5,6,1,R,                  ! Defines a circular arc
LARC,7,8,1,R,
LESIZE,ALL,,,15                  ! Specifies the divisions and spacing ratio on unmeshed
lines, *****Try making 32 smaller


!*********** MESH ***********

real,3                  ! Use real constant set 3
type,1                  ! Use element type 1
mat,3                   ! use material property set 3
LMESH,1,2               ! mesh lines 1-2

real,2                  ! Use real constant set 2
type,1                  ! Use element type 1
mat,2                   ! use material property set 2
LMESH,4                 ! mesh line 4

real,4                  ! Use real constant set 4
LMESH,3                         ! mesh line 3

real,1                  ! Use real constant set 1
type,1                  ! Use element type 1
mat,1                   ! use material property set 1
LMESH,5,6               ! mesh line 5-6



!******* Get Node Numbers at chosen keypoints *******


ksel,s,kp,,2
nslk,s
*get,nkp2,node,0,num,max          ! Retrieves a value and stores it as a scalar parameter
or part of an array parameter***********
nsel,all
ksel,all

ksel,s,kp,,3
```

```
nslk,s

*get,nkp3,node,0,num,max
nsel,all

ksel,all
ksel,s,kp,,4
nslk,s
*get,nkp4,node,0,num,max
nsel,all
ksel,all

ksel,s,kp,,5
nslk,s
*get,nkp5,node,0,num,max
nsel,all
ksel,all

ksel,s,kp,,6
nslk,s
*get,nkp6,node,0,num,max
nsel,all
ksel,all

ksel,s,kp,,7
nslk,s
*get,nkp7,node,0,num,max
nsel,all
ksel,all


ksel,s,kp,,8
nslk,s
*get,nkp8,node,0,num,max
nsel,all
ksel,all
FINISH

!***********************************************************
!********************** SOLUTION ***************************
!***********************************************************

/SOL
```

Appendix A (Continued)

```
ANTYPE,0                          ! Specifies the analysis type and restart status and "0" means
that it Performs a static analysis. Valid for all degrees of freedom

NLGEOM,1                          ! Includes large-deflection effects in a static or full transient
analysis


!************************************

DK,2, ,0, , , ,UX,UY,UZ,ROTX,ROTZ,  ! Boundary conditions on keypoint 2
DK,3, ,0, , , ,UX,UY,UZ,ROTX,ROTZ,  ! Boundary conditions on keypoint 2

DK,8, ,0, , , ,UX,UY,UZ,ROTZ,
LOCAL,11,CART,0,0,0,-44,0,0,
CSYS,11
DK,8,ROTY,0
CSYS,0

!************************************
*DIM,my1,TABLE,10000

lsnum =0

*DO,step,1,141,10

        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO
stop1 = lsnum


*DO,step,141,156,1
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO
stop1 = lsnum

DKDELE,2,ROTY
```

Appendix A (Continued)

```
*DO,step,-12450,47150,100                    !Maximum Moment Value -2.1065e+004
          mmy1 = step*1
          FK,2,MY,mmy1

       lsnum=lsnum+1
       *set,MY1(lsnum),step*1

          LSWRITE,lsnum
*ENDDO

FKDELE,2,my


*DO,step,47150,47250,10                                !Maximum Moment Value -
2.1065e+004
          mmy1 = step*1
          FK,2,MY,mmy1

       lsnum=lsnum+1
       *set,MY1(lsnum),step*1

          LSWRITE,lsnum
*ENDDO

FKDELE,2,my


*DO,step,16,180,0.5
          theta=-1*step
          DK,2,ROTY,theta*PI/180
          lsnum=lsnum+1
          LSWRITE,lsnum
*ENDDO

DKDELE,2,ROTY

*Do,nn,1,lsnum

LSSOLVE,nn

/output,progress,txt,,append
*VWRITE,nn                ! Writes data to a file in a formatted sequence
%16.8G
```

Appendix A (Continued)

```
/output
*enddo
/STATUS,SOLU
FINISH

!*****************************
!******* GET RESULTS *********
!*****************************
loadSteps=lsnum
/POST1
*DIM,rotY2,TABLE,loadSteps
*DIM,disX3,TABLE,loadSteps
*DIM,disY3,TABLE,loadSteps
*DIM,disZ3,TABLE,loadSteps
*DIM,disX5,TABLE,loadSteps
*DIM,disY5,TABLE,loadSteps
*DIM,disZ5,TABLE,loadSteps

*DIM,my2,TABLE,loadSteps

/output,dofsoln,txt,,append                    ! Writes data to a file in a formatted
sequence
NLIST,ALL, , , ,NODE,NODE,NODE
/output

*Do,nn,1,lsnum
 set,nn
*GET,roty,Node,nkp2,ROT,Y
*SET,rotY2(nn),roty
*GET,mmy2,Node,nkp2,RF,MY
*SET,my2(nn),mmy2

/output,dofsoln,txt,,append                    ! Writes data to a file in a formatted
sequence
PRNSOL,U,COMP
/output

*ENDDO

/output,output_arc%arclength%_asp%aspect%,txt,,

*MSG,INFO,'t','w','R','E','arclength'        ! Writes an output message via the ANSYS
message subroutine
%-8C %-8C %-8C %-8C %-8C
```

Appendix A (Continued)

```
*VWRITE,h2,b2,R,E2,arclength                    ! Writes data to a file in a formatted
sequence
%16.8G %-16.8G %-16.8G %-16.8G %-16.8G

*MSG,INFO,'roty2','my1','my2'
%-8C %-8C %-8c

*VWRITE,rotY2(1),my1(1),my2(1)
%16.8G %-16.8G %-16.8G
/output


FINISH
```

A.2 Batch File 2

```
!********** Create Keypoints 1 throug 7: K(Point #, X-Coord, Y-Coord, Z-Coord)
*********

K,1,0,0,0,
K,2,0,-225,0,
K,3,0,-20,0,
K,4,0,20,0,
K,5,R*cos(PI/180*72),R*SIN(PI/180*72),0,
K,6,R*cos(PI/180*33),R*sin(PI/180*33),0,
K,7,R*cos(PI/180*29),R*sin(PI/180*29),0,
K,8,R*cos(PI/180*44),-R*sin(PI/180*44),0,

*DO,step,1,161,10
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO
stop1 = lsnum

*DO,step,161,164,1
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO
```

Appendix A (Continued)

```
stop1 = lsnum

DKDELE,2,ROTY

*DO,step,-13300,35000,100
        mmy1 = step*1
        FK,2,MY,mmy1

     lsnum=lsnum+1
     *set,MY1(lsnum),step*1

        LSWRITE,lsnum
*ENDDO


FKDELE,2,my

*DO,step,35000,35190,10
        mmy1 = step*1
        FK,2,MY,mmy1

     lsnum=lsnum+1
     *set,MY1(lsnum),step*1


        LSWRITE,lsnum
*ENDDO

FKDELE,2,my

*DO,step,31,180,1
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO

DKDELE,2,ROTY
```

Appendix A (Continued)

A.3 Batch File 3

```
!********** Create Keypoints 1 throug 7: K(Point #, X-Coord, Y-Coord, Z-Coord)
*********

K,1,0,0,0,
K,2,0,-225,0,
K,3,0,-25,0,
K,4,0,25,0,
K,5,R*cos(PI/180*72),R*SIN(PI/180*72),0,
K,6,R*cos(PI/180*33),R*sin(PI/180*33),0,
K,7,R*cos(PI/180*29),R*sin(PI/180*29),0,
K,8,R*cos(PI/180*44),-R*sin(PI/180*44),0,

*DO,step,1,161,10
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum


*ENDDO
stop1 = lsnum

DKDELE,2,ROTY


*DO,step,161,172,1
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO

stop1 = lsnum

DKDELE,2,ROTY


*DO,step,-14000,29000,100          !Maximum Moment Value -2.1065e+004
        mmy1 = step*1
        FK,2,MY,mmy1
     lsnum=lsnum+1
     *set,MY1(lsnum),step*1
```

```
	LSWRITE,lsnum
*ENDDO

FKDELE,2,my

*DO,step,35,180,1
	theta=-1*step
	DK,2,ROTY,theta*PI/180
	lsnum=lsnum+1
	LSWRITE,lsnum
*ENDDO

DKDELE,2,ROTY
```

A.4 Batch File 4

```
!********** Create Keypoints 1 throug 7: K(Point #, X-Coord, Y-Coord, Z-Coord)
*********

K,1,0,0,0,


K,2,0,-225,0,
K,3,0,-30,0,
K,4,0,30,0,
K,5,R*cos(PI/180*72),R*SIN(PI/180*72),0,
K,6,R*cos(PI/180*33),R*sin(PI/180*33),0,
K,7,R*cos(PI/180*29),R*sin(PI/180*29),0,
K,8,R*cos(PI/180*44),-R*sin(PI/180*44),0,

*DO,step,1,171,10
	theta=-1*step
	DK,2,ROTY,theta*PI/180
	lsnum=lsnum+1
	LSWRITE,lsnum
*ENDDO
stop1 = lsnum


DKDELE,2,ROTY

*DO,step,-18000,24900,100				!Maximum Moment Value -
2.1065e+004
	mmy1 = step*1
```

Appendix A (Continued)

```
          FK,2,MY,mmy1

          lsnum=lsnum+1
      *set,MY1(lsnum),step*1

          LSWRITE,lsnum
*ENDDO

FKDELE,2,my

*DO,step,34,180,0.5
          theta=-1*step
          DK,2,ROTY,theta*PI/180
          lsnum=lsnum+1
          LSWRITE,lsnum
*ENDDO
```

A.5 Batch File 5

```
!********** Create Keypoints 1 throug 7: K(Point #, X-Coord, Y-Coord, Z-Coord) *********

K,1,0,0,0,
K,2,0,-225,0,
K,3,0,-40,0,
K,4,0,40,0,
K,5,R*cos(PI/180*72),R*SIN(PI/180*72),0,
K,6,R*cos(PI/180*33),R*sin(PI/180*33),0,
K,7,R*cos(PI/180*29),R*sin(PI/180*29),0,
K,8,R*cos(PI/180*44),-R*sin(PI/180*44),0,

*DO,step,1,181,10
          theta=-1*step
          DK,2,ROTY,theta*PI/180
          lsnum=lsnum+1
          LSWRITE,lsnum
*ENDDO
stop1 = lsnum


DKDELE,2,ROTY

*DO,step,181,196,1
```

Appendix A (Continued)

```
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO

stop1 = lsnum


DKDELE,2,ROTY

*DO,step,-14500,18100,100                       !Maximum Moment Value -
2.1065e+004
        mmy1 = step*1
        FK,2,MY,mmy1

        lsnum=lsnum+1
        *set,MY1(lsnum),step*1

        LSWRITE,lsnum
*ENDDO

FKDELE,2,my



*DO,step,18100,18140,10                         !Maximum Moment Value -
2.1065e+004
        mmy1 = step*1
        FK,2,MY,mmy1

        lsnum=lsnum+1
        *set,MY1(lsnum),step*1

        LSWRITE,lsnum
*ENDDO

FKDELE,2,my


*DO,step,53,27,-1
        theta=-1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
```

```
            LSWRITE,lsnum
*ENDDO

DKDELE,2,ROTY



*DO,step,17500,19875,25          !Maximum Moment Value -2.1065e+004
            mmy1 = step*1
            FK,2,MY,mmy1
        lsnum=lsnum+1
        *set,MY1(lsnum),step*1

            LSWRITE,lsnum
*ENDDO

FKDELE,2,my


*DO,step,40,180,1
            theta=-1*step
            DK,2,ROTY,theta*PI/180
            lsnum=lsnum+1
            LSWRITE,lsnum
*ENDDO
```

A.6 Batch File 6

```
!********** Create Keypoints 1 throug 7: K(Point #, X-Coord, Y-Coord, Z-Coord)
*********

K,1,0,0,0,
K,2,0,-225,0,
K,3,0,-50,0,
K,4,0,50,0,
K,5,R*cos(PI/180*72),R*SIN(PI/180*72),0,
K,6,R*cos(PI/180*33),R*sin(PI/180*33),0,
K,7,R*cos(PI/180*29),R*sin(PI/180*29),0,
K,8,R*cos(PI/180*44),-R*sin(PI/180*44),0,

*DO,step,1,215,15
            theta=-1*step
            DK,2,ROTY,theta*PI/180
```

Appendix A (Continued)

```
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO
stop1 = lsnum


DKDELE,2,ROTY

*DO,step,-15100,15900,100                          !Maximum Moment Value -
2.1065e+004
        mmy1 = step*1
        FK,2,MY,mmy1

     lsnum=lsnum+1
     *set,MY1(lsnum),step*1

        LSWRITE,lsnum
*ENDDO

FKDELE,2,my

*DO,step,-56,-25,1
        theta=1*step
        DK,2,ROTY,theta*PI/180
        lsnum=lsnum+1
        LSWRITE,lsnum
*ENDDO

DKDELE,2,ROTY

*DO,step,12925,16975,25                            !Maximum Moment Value -
2.1065e+004
        mmy1 = step*1
        FK,2,MY,mmy1

     lsnum=lsnum+1
     *set,MY1(lsnum),step*1

        LSWRITE,lsnum
*ENDDO

FKDELE,2,my

*DO,step,-39,-180,-1
```

```
            theta=1*step
            DK,2,ROTY,theta*PI/180
            lsnum=lsnum+1
            LSWRITE,lsnum
*ENDDO
```

A.7 Batch File 7

```
!********** Create Keypoints 1 throug 7: K(Point #, X-Coord, Y-Coord, Z-Coord)
*********

K,1,0,0,0,
K,2,0,-225,0,
K,3,0,-60,0,
K,4,0,60,0,
K,5,R*cos(PI/180*72),R*SIN(PI/180*72),0,
K,6,R*cos(PI/180*33),R*sin(PI/180*33),0,
K,7,R*cos(PI/180*29),R*sin(PI/180*29),0,
K,8,R*cos(PI/180*44),-R*sin(PI/180*44),0,

*DO,step,1,231,10
            theta=-1*step
            DK,2,ROTY,theta*PI/180
            lsnum=lsnum+1
            LSWRITE,lsnum
*ENDDO
stop1 = lsnum

!DKDELE,2,ROTY


*DO,step,231,237,1
            theta=-1*step
            DK,2,ROTY,theta*PI/180
            lsnum=lsnum+1
            LSWRITE,lsnum
*ENDDO
stop1 = lsnum

DKDELE,2,ROTY
```

```
*DO,step,-14400,14600,100                           !Maximum Moment Value -
2.1065e+004
         mmy1 = step*1
         FK,2,MY,mmy1

      lsnum=lsnum+1
      *set,MY1(lsnum),step*1


         LSWRITE,lsnum
*ENDDO

FKDELE,2,my


*DO,step,14600,14790,10                             !Maximum Moment Value -
2.1065e+004
         mmy1 = step*1
         FK,2,MY,mmy1

      lsnum=lsnum+1
      *set,MY1(lsnum),step*1

         LSWRITE,lsnum
*ENDDO

FKDELE,2,my


*DO,step,-48,-24,1
         theta=1*step

         DK,2,ROTY,theta*PI/180
         lsnum=lsnum+1
         LSWRITE,lsnum
*ENDDO

DKDELE,2,ROTY

*DO,step,10000,15150,25                             !Maximum Moment Value -
2.1065e+004
         mmy1 = step*1
         FK,2,MY,mmy1
```

```
        lsnum=lsnum+1
        *set,MY1(lsnum),step*1

            LSWRITE,lsnum
*ENDDO

FKDELE,2,my


*DO,step,39,180,1
            theta=-1*step
            DK,2,ROTY,theta*PI/180
            lsnum=lsnum+1
            LSWRITE,lsnum
*ENDDO
```

Appendix B: MATLAB M-Files

B.1 FEA and Twisting and Bending Plots

This MATLAB file selects the correct data from the previous FEA results and plots it against the twisting and bending model results also produced by this M-file.

```
clc

input_range=[pi/2:pi/288:3*pi/2];
%----------------------------------------------------------------------

%spherical trig wrt theta2 as input

r1=46*pi/180;                           %spherical trig angle
r1=46degrees
r2=pi/2;                                %spherical trig angle
r2=90degrees
r3=pi/2;                                %spherical trig angle
r3=90degrees
r4=75*pi/180;                           %spherical trig angle
r4=75degrees

omega = input_range; %Input theta2
flag1 = sign(sin(omega));
delta_wrt2 = acos(cos(r1)*cos(r2)+sin(r1)*sin(r2)*cos(omega));
alpha_wrt2 = flag1.*(acos((cos(r1)-
cos(r2)*cos(delta_wrt2))./(sin(r2)*sin(delta_wrt2))));
beta_wrt2 = acos((cos(r4)-
cos(r3)*cos(delta_wrt2))./(sin(r3)*sin(delta_wrt2)));
gamma_wrt2 = acos((cos(delta_wrt2)-
cos(r3)*cos(r4))./(sin(r3)*sin(r4)));
epsilon_wrt2 = acos((cos(r3)-
cos(r4)*cos(delta_wrt2))./(sin(r3)*sin(delta_wrt2)));
zeta_wrt2 = flag1.*(acos((cos(r2)-
cos(r1)*cos(delta_wrt2))./(sin(r1)*sin(delta_wrt2))));
theta4_wrt2 = pi-epsilon_wrt2-zeta_wrt2;
%----------------------------------------------------------------------
E = 169*10^3;                  %youngs modulus 169 GPa
a = 3.5;                 %width of compliant joint
b = 2;                    %height of compliant joint

omega_match_ansys = -(omega-pi/2);

%moment of bending joint3
L3 = 22.17;                %length of joint 3 in micrometers
Ix = a*b^3/12;            %second moment of area, compliant mechanisms
pg.399
K3 = -E*Ix/L3;             %bending coefficient
```

```
delta_psi3 = (gamma_wrt2-gamma_wrt2(1)); %need to make sure this gets
smaller
d_Phi3_d_omega=(sin(r1).*sin(r2).*sin(omega))./(sin(r3).*sin(r4).*sin(g
amma_wrt2));
%U_bending_J3 = 0.5*K3*(delta_psi3).^2;
MJ3 = K3.*delta_psi3.*d_Phi3_d_omega;
%----------------------------------------------------------------------
---

%1square Analytical
L=20;

%coefficient of twist K1
nu = 0.35;               %poisson's ratio
beta = 0.2125;           %pg 255 elasticity
J = beta*a*b^3;          %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));        %shear modulus
K1 = -J*G./L;              %coefficient of twist

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_1 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_1=(MJ2+MJ3);

M2_1(145)=(M2_1(144)+M2_1(146))/2;
theta2_1(145)=(theta2_1(144)+theta2_1(146))/2;

% 1 Square Ansys
filename = ['ansys_output_1square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
```

Appendix B (Continued)

```
roty2_1 = [0; A(:,1)];                      % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                       % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                       % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                    % assigns a zero to values of
zero and 1 to values other than zero
my_1 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%-----------------------------------------------------------------------
---

%2square Analytical
L=30;

%coefficient of twist K1
nu = 0.35;                 %poisson's ratio
beta = 0.2125;             %pg 255 elasticity
J = beta*a*b^3;            %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));          %shear modulus
K1 = -J*G./L;                %coefficient of twist

%moment of bending joint2.
Ix = a*b^3/12;           %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;            %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_2 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_2=(MJ2+MJ3);

M2_2(145)=(M2_2(144)+M2_2(146))/2;
theta2_2(145)=(theta2_2(144)+theta2_2(146))/2;

% 2 Square Ansys
filename = ['ansys_output_2square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);          % opens the file
ABT = fread(fid1);                         % reads the file into variable
ABT
fclose(fid1);                              % closes the data file
```

```
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_2 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_2 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%-----------------------------------------------------------------------
---

%3square Analytical
L=40;

%coefficient of twist K1
nu = 0.35;                %poisson's ratio
beta = 0.2125;            %pg 255 elasticity
J = beta*a*b^3;           %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));         %shear modulus
K1 = -J*G./L;               %coefficient of twist

%moment of bending joint2.
Ix = a*b^3/12;          %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;            %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);;
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_3 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_3=(MJ2+MJ3);

M2_3(145)=(M2_3(144)+M2_3(146))/2;
theta2_3(145)=(theta2_3(144)+theta2_3(146))/2;

% 3 Square Ansys
filename = ['ansys_output_3square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
```

Appendix B (Continued)

```
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_3 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_3 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_3(156)=0; % deletes incorrect data in 3square program to remove
spike
%-----------------------------------------------------------------------
---

%4square Analytical
L=50;

%coefficient of twist K1
nu = 0.35;               %poisson's ratio
beta = 0.2125;           %pg 255 elasticity
J = beta*a*b^3;          %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));        %shear modulus
K1 = -J*G./L;              %coefficient of twist

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_4 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_4=(MJ2+MJ3);
```

```
M2_4(145)=(M2_4(144)+M2_4(146))/2;
theta2_4(145)=(theta2_4(144)+theta2_4(146))/2;

% 4 Square Ansys
filename = ['ansys_output_4square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_4 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_4 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_4(171)=0; % deletes incorrect data in 4square program to remove
spike
%----------------------------------------------------------------------
---

%5square Analytical
L=60;

%coefficient of twist K1
nu = 0.35;               %poisson's ratio
beta = 0.2125;           %pg 255 elasticity
J = beta*a*b^3;          %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));        %shear modulus
K1 = -J*G./L;              %coefficient of twist

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
```

```
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_5 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_5=(MJ2+MJ3);

M2_5(145)=(M2_5(144)+M2_5(146))/2;
theta2_5(145)=(theta2_5(144)+theta2_5(146))/2;

% 5 Square Ansys
filename = ['ansys_output_5square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_5 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_5 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_5(200)=0; % deletes incorrect data in 5square program to remove
spike
%----------------------------------------------------------------------
---

%6square Analytical
L=80;

%coefficient of twist K1
nu = 0.35;               %poisson's ratio
beta = 0.2125;           %pg 255 elasticity
J = beta*a*b^3;          %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));        %shear modulus
K1 = -J*G./L;              %coefficient of twist

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
```

```
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_6 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_6=(MJ2+MJ3);

M2_6(145)=(M2_6(144)+M2_6(146))/2;
theta2_6(145)=(theta2_6(144)+theta2_6(146))/2;

% 6 Square Ansys
filename = ['ansys_output_6square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_6 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_6 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_6(182)=0; % deletes incorrect data in 6square program to remove
spike
%-----------------------------------------------------------------------
---

%7square Analytical
L=100;

%coefficient of twist K1
nu = 0.35;                 %poisson's ratio
beta = 0.2125;             %pg 255 elasticity
J = beta*a*b^3;            %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));          %shear modulus
K1 = -J*G./L;                %coefficient of twist
```

Appendix B (Continued)

```
%moment of bending joint2.
Ix = a*b^3/12;          %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;           %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_7 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_7=(MJ2+MJ3);

M2_7(145)=(M2_7(144)+M2_7(146))/2;
theta2_7(145)=(theta2_7(144)+theta2_7(146))/2;

% 7 Square Ansys
filename = ['ansys_output_7square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_7 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2    = [0; A(:,2)];                   % pulls my2 from file and adds
a zero to beginning
my3    = [0; A(:,3)];                   % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_7 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_7(168)=0; % deletes incorrect data in 7square program to remove
spike
%----------------------------------------------------------------------
---

%8square Analytical
L=120;

%coefficient of twist K1
```

```
nu = 0.35;               %poisson's ratio
beta = 0.2125;           %pg 255 elasticity
J = beta*a*b^3;          %J is polar moment of inertia, JG is the
torsional rigidity, pg. 191 compliant mechanisms
G = E/(2*(1+nu));        %shear modulus
K1 = -J*G./L;              %coefficient of twist

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

theta2_8 = omega_match_ansys-1/K1.*(MJ2+MJ3);
M2_8=(MJ2+MJ3);

M2_8(145)=(M2_8(144)+M2_8(146))/2;
theta2_8(145)=(theta2_8(144)+theta2_8(146))/2;

% 8 Square Ansys
filename = ['ansys_output_8square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_8 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_8 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_8(177)=0; % deletes incorrect data in 8square program to remove
spike
%-----------------------------------------------------------------------
---
```

```
%individual plots of ansys vs analytical

figure(1)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_1*180/pi,my_1,'k') plot(theta2_1*180/pi,M2_1,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_combined_1')

figure(2)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_2*180/pi,my_2,'k') plot(theta2_2*180/pi,M2_2,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
print(gcf,'-dtiff','-r300', 'Graph_MomVsRot_combined_2')

figure(3)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_3*180/pi,my_3,'k') plot(theta2_3*180/pi,M2_3,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
print(gcf,'-dtiff','-r300', 'Graph_MomVsRot_combined_3')

figure(4)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_4*180/pi,my_4,'k') plot(theta2_4*180/pi,M2_4,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
```

Appendix B (Continued)

```
print(gcf,'-dtiff','-r300', 'Graph_MomVsRot_combined_4')

figure(5)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_5*180/pi,my_5,'k') plot(theta2_5*180/pi,M2_5,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
print(gcf,'-dtiff','-r300', 'Graph_MomVsRot_combined_5')

figure(6)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_6*180/pi,my_6,'k') plot(theta2_6*180/pi,M2_6,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
print(gcf,'-dtiff','-r300', 'Graph_MomVsRot_combined_6')

figure(7)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_7*180/pi,my_7,'k') plot(theta2_7*180/pi,M2_7,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
print(gcf,'-dtiff','-r300', 'Graph_MomVsRot_combined_7')

figure(8)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
h=[plot(roty2_8*180/pi,my_8,'k') plot(theta2_8*180/pi,M2_8,'b')];
p=legend(h,'ANSYS','Analytical');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',14)
print(gcf,'-dtiff','-r300', 'Graph_MomVsRot_combined_8')
```

```
% %--------------------------------------------------------------------
-----

%all combined plots

%combined plot for models 1-8
figure(9)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
thinline=plot(roty2_2*180/pi,my_2,'k');
thickline=plot(theta2_2*180/pi,M2_2,'k','linewidth',3);
plot(theta2_1*180/pi,M2_1,'k','linewidth',3)
plot(theta2_2*180/pi,M2_2,'r','linewidth',3)
plot(theta2_3*180/pi,M2_3,'m','linewidth',3)
plot(theta2_4*180/pi,M2_4,'b','linewidth',3)
plot(theta2_5*180/pi,M2_5,'g','linewidth',3)
plot(theta2_6*180/pi,M2_6,'c','linewidth',3)
plot(theta2_7*180/pi,M2_7,'k','linewidth',3)
plot(theta2_8*180/pi,M2_8,'r','linewidth',3)
h=[plot(theta2_1(1)*180/pi,M2_1(1),'*')
thinline
thickline
plot(roty2_1*180/pi,my_1,'k')
plot(roty2_2*180/pi,my_2,'r')
plot(roty2_3*180/pi,my_3,'m')
plot(roty2_4*180/pi,my_4,'b')
plot(roty2_5*180/pi,my_5,'g')
plot(roty2_6*180/pi,my_6,'c')
plot(roty2_7*180/pi,my_7,'k')
plot(roty2_8*180/pi,my_8,'r')];
p=legend(h,'Start','Ansys','Analytical','Model 1','Model 2','Model
3','Model 4','Model 5','Model 6','Model 7','Model 8',2);
set(p,'box', 'off')
set(p,'location','northeast')
set(p,'fontsize',10)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_Allcombined_2-8')

%combined plot for models 2-8
figure(10)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('Moment (\muN\mum)','fontsize',17)
thinline=plot(roty2_2*180/pi,my_2,'k');
thickline=plot(theta2_2*180/pi,M2_2,'k','LineWidth',3);
plot(theta2_2*180/pi,M2_2,'k','LineWidth',3)
plot(theta2_3*180/pi,M2_3,'r','LineWidth',3)
plot(theta2_4*180/pi,M2_4,'m','LineWidth',3)
plot(theta2_5*180/pi,M2_5,'b','LineWidth',3)
plot(theta2_6*180/pi,M2_6,'g','LineWidth',3)
```

Appendix B (Continued)

```
plot(theta2_7*180/pi,M2_7,'c','LineWidth',3)
plot(theta2_8*180/pi,M2_8,'k','LineWidth',3)
h=[plot(theta2_1(1)*180/pi,M2_1(1),'*')
thinline
thickline
plot(roty2_2*180/pi,my_2,'k')
plot(roty2_3*180/pi,my_3,'r')
plot(roty2_4*180/pi,my_4,'m')
plot(roty2_5*180/pi,my_5,'b')
plot(roty2_6*180/pi,my_6,'g')
plot(roty2_7*180/pi,my_7,'c')
plot(roty2_8*180/pi,my_8,'k')];
p=legend(h,'Start','Ansys','Analytical','Model 2','Model 3','Model
4','Model 5','Model 6','Model 7','Model 8',2);
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',10)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_Allcombined_2-8')
```

B.2 FEA and Pure Bending Plots

The following plots the pure bending model against FEA results.

```
clc

input_range=[pi/2:pi/288:3*pi/2];
%----------------------------------------------------------------------
---

%spherical trig wrt theta2 as input

r1=46*pi/180;                           %spherical trig angle
r1=46degrees
r2=pi/2;                                %spherical trig angle
r2=90degrees
r3=pi/2;                                %spherical trig angle
r3=90degrees
r4=75*pi/180;                           %spherical trig angle
r4=75degrees

omega = input_range; %Input theta2
flag1 = sign(sin(omega));
delta_wrt2 = acos(cos(r1)*cos(r2)+sin(r1)*sin(r2)*cos(omega));
alpha_wrt2 = flag1.*(acos((cos(r1)-
cos(r2)*cos(delta_wrt2))./(sin(r2)*sin(delta_wrt2))));
beta_wrt2 = acos((cos(r4)-
cos(r3)*cos(delta_wrt2))./(sin(r3)*sin(delta_wrt2)));
gamma_wrt2 = acos((cos(delta_wrt2)-
cos(r3)*cos(r4))./(sin(r3)*sin(r4)));
epsilon_wrt2 = acos((cos(r3)-
cos(r4)*cos(delta_wrt2))./(sin(r3)*sin(delta_wrt2)));
```

```
zeta_wrt2 = flag1.*(acos((cos(r2)-
cos(r1)*cos(delta_wrt2))./(sin(r1)*sin(delta_wrt2))));
theta4_wrt2 = pi-epsilon_wrt2-zeta_wrt2;
%----------------------------------------------------------------------
---

E = 169*10^3;                  %youngs modulus 169 GPa
a = 3.5;                 %width of compliant joint
b = 2;                    %height of compliant joint

omega_match_ansys = -(omega-pi/2);

%moment of bending joint3
L3 = 22.17;                %length of joint 3 in micrometers
Ix = a*b^3/12;            %second moment of area, compliant mechanisms
pg.399
K3 = -E*Ix/L3;             %bending coefficient
delta_psi3 = (gamma_wrt2-gamma_wrt2(1)); %need to make sure this gets
smaller
d_Phi3_d_omega=(sin(r1).*sin(r2).*sin(omega))./(sin(r3).*sin(r4).*sin(g
amma_wrt2));
%U_bending_J3 = 0.5*K3*(delta_psi3).^2;
MJ3 = K3.*delta_psi3.*d_Phi3_d_omega;
%----------------------------------------------------------------------
---

%1square Analytical
L=20;

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

M2_1=(MJ2+MJ3);
M2_1(145)=(M2_1(144)+M2_1(146))/2;

% 1 Square Ansys
filename = ['ansys_output_1square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
```

```
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_1 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_1 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%-----------------------------------------------------------------------
---

%2square Analytical
L=30;

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

M2_2=(MJ2+MJ3);
M2_2(145)=(M2_2(144)+M2_2(146))/2;

% 2 Square Ansys
filename = ['ansys_output_2square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_2 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
```

```
my2   = [0; A(:,2)];                       % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                       % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                    % assigns a zero to values of
zero and 1 to values other than zero
my_2 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

%3square Analytical
L=40;

%moment of bending joint2.
Ix = a*b^3/12;          %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;           %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);;
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

M2_3=(MJ2+MJ3);
M2_3(145)=(M2_3(144)+M2_3(146))/2;

% 3 Square Ansys
filename = ['ansys_output_3square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                         % reads the file into variable
ABT
fclose(fid1);                              % closes the data file
GBT = native2unicode(ABT)';                % changes data from machine
code to text
s_iB = findstr('my2',GBT);                 % finds end of header
A=str2num(GBT(s_iB+4:end));                % turns the data into a
numerical matrix
roty2_3 = [0; A(:,1)];                       % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                       % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                       % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                    % assigns a zero to values of
zero and 1 to values other than zero
my_3 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
```

Appendix B (Continued)

```
my_3(156)=0; % deletes incorrect data in 3square program to remove
spike
%----------------------------------------------------------------------
---

%4square Analytical
L=50;

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

M2_4=(MJ2+MJ3);
M2_4(145)=(M2_4(144)+M2_4(146))/2;

% 4 Square Ansys
filename = ['ansys_output_4square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_4 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_4 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_4(171)=0; % deletes incorrect data in 4square program to remove
spike
%----------------------------------------------------------------------
---
```

Appendix B (Continued)

```
%5square Analytical
L=60;

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;


M2_5=(MJ2+MJ3);
M2_5(145)=(M2_5(144)+M2_5(146))/2;


% 5 Square Ansys
filename = ['ansys_output_5square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_5 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_5 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_5(200)=0; % deletes incorrect data in 5square program to remove
spike
%-----------------------------------------------------------------------
---

%6square Analytical
L=80;

%moment of bending joint2.
```

Appendix B (Continued)

```
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

M2_6=(MJ2+MJ3);
M2_6(145)=(M2_6(144)+M2_6(146))/2;

% 6 Square Ansys
filename = ['ansys_output_6square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_6 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_6 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_6(182)=0; % deletes incorrect data in 6square program to remove
spike
%-----------------------------------------------------------------------
---

%7square Analytical
L=100;

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
```

```
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

M2_7=(MJ2+MJ3);
M2_7(145)=(M2_7(144)+M2_7(146))/2;

% 7 Square Ansys
filename = ['ansys_output_7square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_7 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_7 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_7(168)=0; % deletes incorrect data in 7square program to remove
spike
%-----------------------------------------------------------------------
---

%8square Analytical
L=120;

%moment of bending joint2.
Ix = a*b^3/12;         %second moment of area, compliant mechanisms pg.
399
K2 = -E*Ix./L;          %bending coefficient
delta_phi2 = (alpha_wrt2(1)-alpha_wrt2)+(beta_wrt2(1)-beta_wrt2);
d_alpha_d_omega=((sin(r1).*sin(omega))./(sin(alpha_wrt2).*sin(delta_wrt
2).^3)).*(cos(r1).*cos(delta_wrt2)-cos(r2).*cos(delta_wrt2).^2-
cos(r2).*sin(delta_wrt2).^2);
d_beta_d_omega=((sin(r1).*sin(r2).*sin(omega))./(sin(beta_wrt2).*sin(r3
).*sin(delta_wrt2).^3)).*(cos(r4).*cos(delta_wrt2)-
cos(r3).*sin(delta_wrt2).^2-cos(r3).*cos(delta_wrt2).^2);
```

Appendix B (Continued)

```
d_phi2_d_omega=d_alpha_d_omega+d_beta_d_omega;
%U_bending_J3 = 0.5*K2*(delta_psi2).^2;
MJ2 = K2*delta_phi2.*d_phi2_d_omega;

M2_8=(MJ2+MJ3);
M2_8(145)=(M2_8(144)+M2_8(146))/2;

% 8 Square Ansys
filename = ['ansys_output_8square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_8 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_8 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
my_8(177)=0; % deletes incorrect data in 8square program to remove
spike
%----------------------------------------------------------------------
---

%individual plots of ansys vs analytical

figure(1)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_1*180/pi,my_1,'k')
plot(omega_match_ansys*180/pi,M2_1,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_1')

figure(2)
clf
hold on
```

Appendix B (Continued)

```
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_2*180/pi,my_2,'k')
plot(omega_match_ansys*180/pi,M2_2,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_2')

figure(3)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_3*180/pi,my_3,'k')
plot(omega_match_ansys*180/pi,M2_3,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_3')

figure(4)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_4*180/pi,my_4,'k')
plot(omega_match_ansys*180/pi,M2_4,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_4')

figure(5)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_5*180/pi,my_5,'k')
plot(omega_match_ansys*180/pi,M2_5,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_5')

figure(6)
```

Appendix B (Continued)

```
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_6*180/pi,my_6,'k')
plot(omega_match_ansys*180/pi,M2_6,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_6')


figure(7)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_7*180/pi,my_7,'k')
plot(omega_match_ansys*180/pi,M2_7,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_7')

figure(8)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_8*180/pi,my_8,'k')
plot(omega_match_ansys*180/pi,M2_8,'b')];
p=legend(h,'FEA','Pure Bending');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_8')
%-----------------------------------------------------------------------
---

%pure bending combined plots

figure(9)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(omega_match_ansys(1)*180/pi,M2_1(1),'*')
```

Appendix B (Continued)

```
plot(omega_match_ansys*180/pi,M2_1,'k')
plot(omega_match_ansys*180/pi,M2_2,'r')
plot(omega_match_ansys*180/pi,M2_3,'m')
plot(omega_match_ansys*180/pi,M2_4,'b')
plot(omega_match_ansys*180/pi,M2_5,'g')
plot(omega_match_ansys*180/pi,M2_6,'c')
plot(omega_match_ansys*180/pi,M2_7,'k')
plot(omega_match_ansys*180/pi,M2_8,'r')];
p=legend(h,'Start','Model 1','Model 2','Model 3','Model 4','Model
5','Model 6','Model 7','Model 8');
set(p,'box', 'off')
set(p,'location','northeast')
set(p,'fontsize',11)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_PureBending_All')
```

B.3 Combined FEA Plots

The following program plots the FEA results for each model.

```
clc

% 1 Square
filename = ['ansys_output_1square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_1 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_1 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

% 2 Square
filename = ['ansys_output_2square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
```

Appendix B (Continued)

```
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_2 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_2 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

% 3 Square
filename = ['ansys_output_3square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_3 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_3 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

% 4 Square
filename = ['ansys_output_4square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
```

Appendix B (Continued)

```
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_4 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_4 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

% 5 Square
filename = ['ansys_output_5square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_5 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_5 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

% 6 Square
filename = ['ansys_output_6square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_6 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
```

Appendix B (Continued)

```
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_6 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

% 7 Square
filename = ['ansys_output_7square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_7 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_7 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%----------------------------------------------------------------------
---

% 8 Square
filename = ['ansys_output_8square.txt'];
string1 = ['I:\Thesis\Chester\Thesis Data\Ansys\Combined Data\'];
fid1 = fopen([string1,filename]);       % opens the file
ABT = fread(fid1);                      % reads the file into variable
ABT
fclose(fid1);                           % closes the data file
GBT = native2unicode(ABT)';             % changes data from machine
code to text
s_iB = findstr('my2',GBT);              % finds end of header
A=str2num(GBT(s_iB+4:end));             % turns the data into a
numerical matrix
roty2_8 = [0; A(:,1)];                    % pulls roty2 from file and
adds a zero to beginning
my2   = [0; A(:,2)];                    % pulls my2 from file and adds
a zero to beginning
my3   = [0; A(:,3)];                    % pulls my3 from file and adds
a zero to beginning
```

```
flag2 = abs(sign(my2));                 % assigns a zero to values of
zero and 1 to values other than zero
my_8 = (flag2.*my2+(1-flag2).*my3); %combines my2 and my3 by using
previous 1's and zeros to form one column,' my
%-----------------------------------------------------------------------
---

my_3(156)=0; % deletes incorrect data in 3square program to remove
spike
my_4(171)=0; % deletes incorrect data in 4square program to remove
spike
my_5(200)=0; % deletes incorrect data in 5square program to remove
spike
my_6(182)=0; % deletes incorrect data in 6square program to remove
spike
my_7(168)=0; % deletes incorrect data in 7square program to remove
spike
my_8(177)=0; % deletes incorrect data in 8square program to remove
spike

%for the sake of the paper, 2square has been re-labeled model 1.
figure(1)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_1(1),my_1(1),'b*')
plot(roty2_2*180/pi,my_2,'k')
plot(roty2_3*180/pi,my_3,'r')
plot(roty2_4*180/pi,my_4,'m')
plot(roty2_5*180/pi,my_5,'b')
plot(roty2_6*180/pi,my_6,'g')
plot(roty2_7*180/pi,my_7,'c')
plot(roty2_8*180/pi,my_8,'k')];
p=legend(h,'Start','model 1','model 2','model 3','model 4','model
5','model 6','model 7');
set(p,'box', 'off')
set(p,'location','northwest')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_Ansys_Combined_2-8')
%-----------------------------------------------------------------------
---

figure(2)
clf
hold on
set(gca,'fontsize',15)
xlabel('\theta_2 (Degrees)','fontsize',17)
ylabel('M_2 (\muN\mum)','fontsize',17)
h=[plot(roty2_1(1),my_1(1),'b*')
plot(roty2_1*180/pi,my_1,'k')
plot(roty2_2*180/pi,my_2,'r')
plot(roty2_3*180/pi,my_3,'m')
```

Appendix B (Continued)

```
plot(roty2_4*180/pi,my_4,'b')
plot(roty2_5*180/pi,my_5,'g')
plot(roty2_6*180/pi,my_6,'c')
plot(roty2_7*180/pi,my_7,'k')
plot(roty2_8*180/pi,my_8,'r')];
p=legend(h,'Start','model 1','model 2','model 3','model 4','model
5','model 6','model 7','model 8');
set(p,'box', 'off')
set(p,'location','northeast')
set(p,'fontsize',15)
print(gcf, '-dtiff','-r300', 'Graph_MomVsRot_Ansys_Combined_1-8')
%----------------------------------------------------------------------
---
```