2013-04-23

Fully Compliant Mechanisms for Bearing Subtraction in Robotics and Space Applications

Ezekiel G. Merriam
Brigham Young University - Provo

Fully Compliant Mechanisms for Bearing Subtraction

in Robotics and Space Applications

Ezekiel G. Merriam

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Larry L. Howell, Chair
Spencer P. Magleby
Mark B. Colton

Department of Mechanical Engineering

Brigham Young University

April 2013

ABSTRACT

Fully Compliant Mechanisms for Bearing Subtraction
in Robotics and Space Applications

Ezekiel G. Merriam
Department of Mechanical Engineering, BYU
Master of Science

Robotics and space applications represent areas where compliant mechanisms can continue to make a significant impact by reducing costs and weight while improving performance. Because of the nature of these applications, a common need is for bearing replacement mechanisms, or mechanisms that perform the function of a bearing without the complexity and failure modes associated with bearings. Static balancing is a design strategy that attempts to reduce the actuation effort of a mechanism, and has been applied to compliant mechanisms in some applications. Monolithic construction, especially by means of 3D printing technology, is a strategy whereby the mechanism links and joints are built as a single "chunk" of material. This eliminates assembly and failure modes associated with wear and friction in traditional joints. In this work we examine these design strategies in the context of robotics and space applications. Matlab and Ansys batch files can be found in Appendix A.

A fully compliant zero-torque, statically balanced mechanism is described that can undergo greater than 100° of motion. Because compliant mechanisms achieve their motion from the deflection of their constituent members, there is some strain energy associated with actuated positions. By introducing an appropriate preload, strain energy can be held constant. This can reduce or nearly eliminate the input force required from the actuating device. This paper describes the statically balanced rotary joint concept, and demonstrates its optimization, testing, and implementation for a haptic pantograph mechanism. The statically balanced properties of the constituent joints result in a mechanism with two balanced degrees of freedom. Matlab and Ansys batch files can be found in Appendix B.

The conception, modeling, and development of a fully compliant two-degree-of-freedom pointing mechanism for application in spacecraft thruster, antenna, or solar array systems is described. The design objectives and the advantages of a compliant solution are briefly discussed. A single design concept is selected for final development from a field of generated concepts. Analytical and numerical models are accompanied by prototype testing and measurements in several iterations. A final design is described in detail, a fully compliant prototype is fabricated in titanium, and its performance is measured.

Keywords: compliant mechanisms, static balancing, monolithic, space mechanisms, pointing mechanism, robotics

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation through grants 0800606 and 1240417. Also, I would like to thank my advisor for his leadership and professional example. Thanks go also to my committee for their willingness to lend support when needed. Thanks also to Ken Forster and Eric Wilcox for their assistance building numerous prototypes. The support of Jonathan Jones and Marshall Space Flight Center for EBM fabrication of the test flexures and final titanium mechanism is gratefully acknowledged. Many thanks to Jason Lund for help with ANSYS. Thanks to my lab-mates for their insight and valuable discussion. Finally, thanks to my wife Kayla, for making it all worth it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

FEA	Finite element analysis
SLFP	Small length flexural pivot
CAFP	Cross axis flexural pivot
CH	cartwheel hinge
STF	Split tube flexure
θ	Defined in Figure 2.1 and re-defined in Figures 2.4 and 3.4
g	Gravitational constant and angular length of coupler as shown in Figure 3.4
k	A spring stiffness
F	An applied force
l	A link or flexible member length
L	A length defined in Figure 2.4
H	Thickness of semi-rigid member, defined in Figure 2.4
t	Width of SLFP (see Figure 2.4) or thickness (see Equation 3.1
h	Thickness of SLFP, see Figure 2.4
T	Material thickness, see Figure 2.4
l_{slfp}	Length of SLFP, see Figure 2.4
S_y	Yield strength
E	Elastic modulus
σ	Stress
S_θ	A CAFP coefficient, $S_\theta = 0.965$
γ	Rotation angle defined in Figure 3.4
η	Rotation angle defined in Figure 3.4
ϕ	Rotation angle defined in Figure 3.4
ψ	Rotation angle defined in Figure 3.4
$[R_x(\beta)]$	Rotation matrix about x by angle β
$[R_y(\beta)]$	Rotation matrix about y by angle β
$[R_z(\beta)]$	Rotation matrix about z by angle β
T_ϕ	Actuation torque about ϕ
T_ψ	Actuation torque about ψ
q_i	A generalized coordinate for calculating virtual work
$\vec{N}$	A vector normal to the surface of the pointer platform
$\vec{AB}$	A vector along the surface of the pointer platform
$\vec{AC}$	A vector along the surface of the pointer platform

CHAPTER 1. INTRODUCTION

1.1 Problem Statement

The purpose of this research is to create preliminary strategies for bearing subtraction by incorporating compliant mechanisms in robotics and space applications. The validity of these strategies is demonstrated by the design, fabrication, and testing of statically balanced mechanisms, and monolithic mechanisms.

1.2 Background on Surrogate Bearings

Robots are widely used in industry to automate repetitive or dangerous tasks that would otherwise require human workers. Such robots are quite sophisticated, but because of their programming and construction they are often dangerous if they come in contact with humans [2, 3, 4]. Adapting robots for use in activities that require human interaction, such as haptic interfaces or physical therapy, requires that these robots be safe. Additionally, they must satisfy other customer needs including quiet actuators, long battery life, and appropriate size and weight, while movement accuracy may be relaxed [5, 3]. Of the many components that could be improved, bearings are inherently parasitic, and as Michell observed:

> “To the machine designer all bearings are of course only necessary evils, contributing nothing to the product or function of the machine; and any virtues they can have are only of the negative order. Their merits consist in absorbing as little power as possible, wearing out as slowly as possible, occupying as little space as possible, and costing as little as possible.” [6]

Because bearings are a mature technology, bearing design methods are already well understood and unlikely to change. To improve some aspects of performance a paradigm shift may be necessary, and compliant mechanisms may be able to supply that shift. Compliant mechanisms in general offer reduced cost, reduced wear, reduced friction, reduced weight, and reduced maintenance [7, 8]. However, one major disadvantage of traditional compliant mechanisms is that significant actuation energy is needed because of the strain energy associated with the displaced mechanism. Stapel et al. suggest using static balancing in the form of a pre-load to reduce the needed actuation force in a medical grasper [9]. In some applications these principles can be applied to reduce the need for bearings in order to meet specific design objectives, resulting in inexpensive robotic joints with zero friction and near-zero actuation force. Going a step further, mechanisms with the proper force deflection behavior could be designed to compensate for the weight of a robotic appendage, again reducing needed actuation force [3] and ”subtracting” the bearing from the assembly while reducing actuator size. In this work, a compliant mechanism that can be substituted for a traditional joint is referred to as a surrogate bearing.

Many areas in robotics stand to benefit from surrogate bearings. Studies show that therapy featuring increased intensity and duration can speed recovery and avoid harmful side effects of stroke, and that robots are well suited to providing more intense therapy for longer periods [10]. Making such therapy robots inexpensive enough to be purchased for in-home use could greatly facilitate their distribution and impact. Exoskeletons for performance augmentation are meant to reduce worker injury and increase productivity in the workplace and on the battlefield [4] and could also benefit from lighter structures and smaller actuators made possible by surrogate bearings. Haptic interfaces, where mechanism transparency, low inertia, and static balancing are desirable, have already seen some development in this area [11, 12], but could still benefit from a reduction of mechanism size.

Compliant surrogate bearings are already available for some applications, but still have serious drawbacks. For example, C-Flex bearings have a range of motion of less than ±15° of rotation and are not statically balanced [13]. The butterfly pivot has applications in aerospace and provides very precise motion but again has very limited rotation [8]. Living hinges easily undergo large deflections, but transmitted power is generally low and the materials choices are limited. Balanced four-bar mechanisms are used in desk lamps and other applications, but these are generally not fully compliant. A special case of the four-bar, the cross axis flexural pivot, has been explored for large deflection applications and static balancing, but this joint is more complicated than other options and might be simplified to reduce costs.

By developing fully compliant statically balanced mechanisms that undergo rotational motion (referred to as zero-torque mechanisms), we can take a step towards surrogate bearing development.

1.3 Background on Monolithic, Additive-Manufacturing Produced Mechanisms

Another area of compliant mechanisms with potential for improvement is monolithic construction of multi-axis systems. Monolithic construction of planar systems is well understood, while multi-axis systems are generally an assembly of parts. Monolithic construction has a number of benefits, the exploitation of which could advance mechanism design. By exploring monolithic construction of multi-axis systems it is possible to reduce part count and assembly cost while increasing performance.

One general type of multi-axis mechanism that could have wide application is the pointing mechanism. On a spacecraft, the capability of pointing a thruster could eliminate the need for multiple thruster arrays, reducing part count and potential failure points. The ability to accurately point a communications antenna could decrease the power required to send data. Similarly, pointing a solar array would give the ability to orient it for optimal capturing of solar radiation, increasing efficiency of energy capture.

However, in the harsh environment of space bearings are prone to failure from cold welding and lubrication out-gassing. For this reason, moving parts are avoided in space applications whenever possible. In order for a pointer mechanism to be useful, it must be designed without using bearings or sliding contact to achieve its motion.

Chapter 3 of this work discusses research into a method of designing a pointing mechanism for use in spacecraft that achieves the desired motion without traditional bearings. Compliant mechanisms achieve their mobility through deflection of flexible members [7, 14]. This avoids sliding contact, eliminating friction and wear along with the need for lubrication. Additionally, compliant mechanisms lend themselves to monolithic construction, which greatly enhances a mechanism's robustness to thermal deformation [15]. The recently developed manufacturing process of Electron Beam Melting that has enabled the production of monolithic parts with complicated geometries will be explored.

1.4 Research Objectives

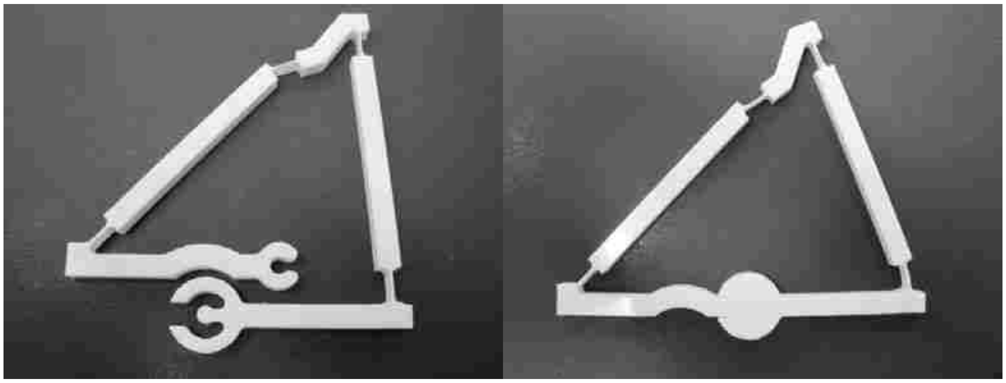
It has been demonstrated that removing bearings from traditional mechanisms and replacing them with compliant elements is an avenue for significant improvements in robot and spacecraft cost, safety, and performance. Therefore, the objective of this research is to provide designers alternatives to traditional bearing technology by:

- Creating, analyzing, and evaluating topologies for zero-torque, statically balanced compliant mechanisms suitable for a variety of robotics applications
- Creating and demonstrating a monolithic two-degree-of-freedom pointing mechanism

1.5 Research Approach

Novel concepts, and current mechanisms were evaluated for potential to become statically balanced. The field of ideas were narrowed to the topology that showed the widest potential application. This was modeled using the Pseudo Rigid Body Model and the principle of virtual work [7], and finite element analysis. In this way, general force deflection curves for a given topology were obtained. If unknowns are properly selected, this yields an analytical, though non-linear, solution to the problem. The machining facilities at BYU were used to construct prototypes for testing and validation of the analytical and finite element models. Figure 1.1 below shows a compliant four-bar designed and prototyped using these concepts. The surrogate bearing behavior was compared to that needed for a haptic pantograph to demonstrate the viability of the solution.

A fully compliant 2-DOF pointing mechanism was designed to provide pointing without traditional bearings. Marshall Space Flight Center produced prototypes using electron beam melting with titanium. These prototypes were tested and compared to FEA models to validate the model and prove the usefulness of the design.



(a) (b)

Figure 1.1: One concept for a statically balanced four bar. With one link grounded and the robot arm or joint mounted to a second link, this becomes a zero friction rotational joint. The left picture shows the mechanism as manufactured. On the right is the assembled mechanism.

CHAPTER 2. OPTIMIZED KINEMATIC CHAIN WITH SMALL LENGTH FLEXURAL PIVOTS

This work describes a fully compliant statically balanced mechanism that can undergo greater than 100° of motion. Because compliant mechanisms achieve their motion from the deflection of their constituent members, there is some strain energy associated with actuated positions. By introducing an appropriate preload, strain energy can be held constant. This can reduce or nearly eliminate the input force required from the actuating device. This work describes the statically balanced concept and demonstrates its optimization, testing, and implementation for a haptic pantograph mechanism. The statically balanced properties of the constituent mechanisms result in an assembly with two balanced degrees of freedom.[1]

2.1 Introduction

The objective of this work is to describe a fully compliant statically balanced mechanism. Such a mechanism could influence applications where small actuator size is critical, such as consumer robotics and space applications. Also, it could benefit from the advantages of compliant mechanisms: reduced cost, wear, friction, mass, and maintenance [7, 8].

A potential topology was identified for development into a statically balanced, fully compliant mechanism. This mechanism was modeled using progressively sophisticated techniques until we achieved the necessary degree of accuracy. Once an accurate model was obtained, optimization was performed to determine the geometry that would yield the desired behavior. The mechanism was prototyped and its behavior was found to closely follow the predicted behavior. Finally, the joint design was integrated into a simplified two-degree-of freedom haptic controller and tested. This application was selected to serve as a first

[1] This chapter has been accepted for publication in IDETC 2013 with Mark Colton, Spencer Magleby, and Larry Howell contributing as co-authors.

step in evaluating statically balanced compliant joints in haptics, an area of human-robot interaction in which low friction and reduced mass are critical for realistic force feedback.

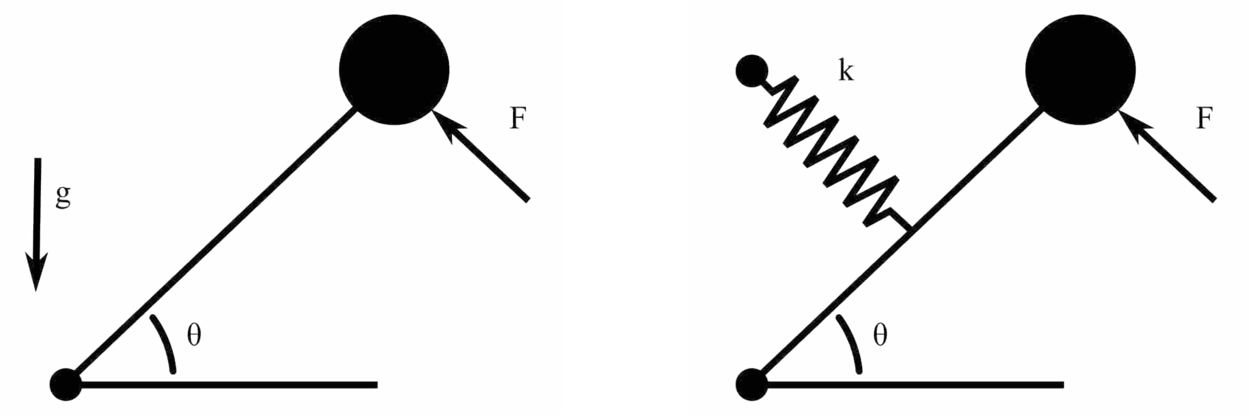
2.2 Background

Compliant mechanisms derive their motion from the deflection of their constituent members. Often this includes deflections large enough that linear beam theory becomes inadequate. Other methods, such as elliptic integrals, the pseudo rigid body model (PRBM) [7, 1], or finite element analysis (FEA), must be used to model their behavior. However, the strain energy associated with the actuated mechanism means that the force required to maintain position is non-zero [16]. For many applications this results in increased actuator size compared to traditional mechanisms, which leads to higher mass and cost. Static balancing offers the potential of exploiting the benefits of compliant mechanisms but mitigating the effects of strain energy.

Herder described the concept of using springs to compensate for undesired changes in strain energy [17]. Much work has been done in reducing input force in surgical instruments and prosthetics [9, 3]. These designs incorporate a preload and some finite potential energy. Stored energy is released from the preloading members as the device is actuated. This energy release aids in the actuation of the mechanism. Because the net change in energy is small, the input force is reduced, and in some cases can be nearly eliminated [18].

Consider the gravity balanced mechanism in Figure 2.1. For the unbalanced system, a non-zero input force is needed to maintain position, except when $\theta = \pm 90^\circ$; the system's potential energy is a function of θ. In the balanced system, this need not be the case. It can be shown that with a zero-free-length spring and the proper selection of the spring constant, k, $F = 0$ for all θ; system energy is constant [19, 20, 21]. Thus, the introduction of a preloaded spring has decoupled mechanism position from mechanism potential energy, reducing the input force.

A compliant mechanism with motion similar to a four-bar mechanism was selected as a fundamental component of the statically balanced mechanism. Researchers have used buckled beams to achieve rectilinear balanced mechanisms [16, 18, 22, 23]. Previous work by Dede et al. has shown that a compliant four-bar can be balanced with gravity compen-



(a) Unbalanced system where a non-zero value of F is required to maintain the position.

(b) A balanced system where the position can be maintained with $F = 0$.

Figure 2.1: An illustration of static balancing.

sation [24]. Additionally, many methods for balancing require the addition of springs or auxiliary bodies [21].

2.3 Mechanism Model

Because not all devices operate in a vertical plane, and space mechanisms operate in low gravity environments, gravity compensation is not always practical. Therefore, our mechanism will not use gravity compensation but will instead be balanced by mechanism strain energy in its own compliant joints [23], without adding external springs or auxiliary bodies.

To make the four-bar mechanism compliant, the revolute joints were replaced with small-length flexural pivots (SLFPs). Figure 2.2 shows the progression from a rigid-link four-bar mechanism to a compliant pseudo-four-bar mechanism modeled using the PRBM approximations [7]. Using the principle of virtual work together with the PRBM [7], a computer model was developed to analyze the torque deflection curve. To apply a preload to the mechanism, the mechanism was made from two pieces that were subsequently rigidly joined. As Figure 2.3 shows, the pieces cannot be assembled into a closed loop without deflection. This deflection provides the preload. Figure 2.3 also indicates the characteristic

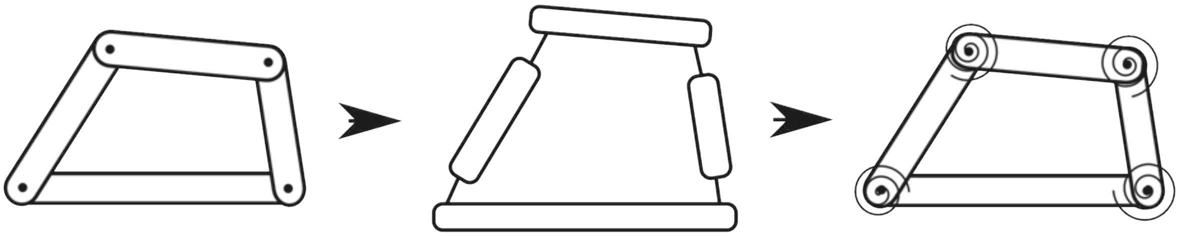


Figure 2.2: Replace the pin joints in a four-bar with small length flexural pivots. The pivots are then modeled as torsion springs [1].

pivot, about which rotation of the input link is measured, the ground link, and the input link.

The PRBM showed that a pseudo-four-bar mechanism has the potential for good statically balanced behavior. However, the PRBM assumes the segments between the SLFPs are rigid; it does not capture compliance in the links. While the PRBM and FEA results might be brought into agreement by using stiffer links (possibly by increasing H_1, H_2, or H_3, see Figure 2.4), the increase in mass would be undesirable. The mechanism could be more accurately modeled as an under-actuated twelve-bar mechanism, but it was decided to use FEA instead of approximate analytical methods. The geometric parameters of the finite element model are indicated in Figure 2.4.

A commercially available FEA program (ANSYS) with non-linear and parametric capabilities was used for the analysis. The BEAM23 element was selected for its non-linear capabilities and relative simplicity. A mesh refinement study was conducted to determine an appropriate mesh size. Three load steps were used to control the solution. On the first load step, the mechanism was assembled, applying the preload. The second load step applied a rotational displacement to the first extreme position. Lastly, the rotational displacement was changed to the second extreme position. Extremes of motion were selected to give the required range of motion while avoiding excessive stresses. Adaptive substeps were used to record data at intermediate positions and increase solver speed. The beam model is shown in Figure 2.5.

The objective of our design is a flat torque curve with a magnitude of zero and large rotational displacement. Therefore, design fitness was calculated as the sum of the squares

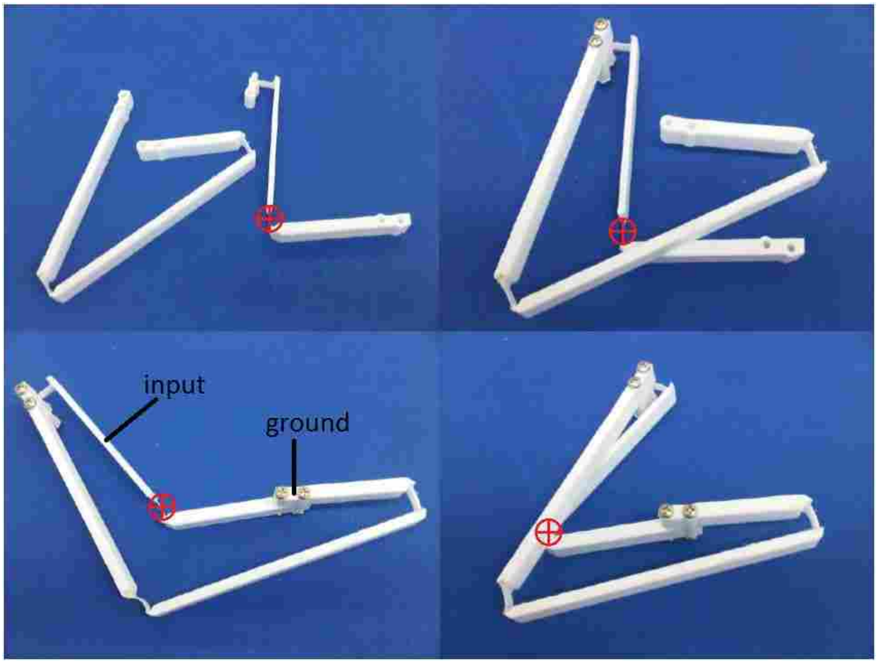


Figure 2.3: Upper left shows the mechanism as manufactured. Upper right shows the partially assembled mechanism before the preload is applied. The lower pictures show the fully assembled mechanism at two deflected positions. Note that the deflected positions are maintained without use of fasteners or restraints, indicating balanced behavior in these positions.

of the error between the desired torque-deflection curve and the torque-deflection curve predicted by the FEA model. A large penalty was added to the fitness if at any point the stress in a member exceeded the yield stress of the material. The geometry minimum dimensions were constrained by the capabilities of the available manufacturing process (computer numeric controlled (CNC) machining). Maximum dimensions were chosen to be on the order of 15 cm to keep the mechanism at an easily manageable size. Transmission angle was not considered a vital parameter for performance. The design fitness would be minimized. A genetic algorithm was selected because the multiplicity of local minima in the design space rendered gradient methods impractical. Matlab was used to write a custom genetic algorithm with the parameters listed in Table 2.1. Elitism was used to ensure that the best designs persisted from one generation to the next. To increase diversity, the same design was not allowed to be both parents.

The model parameters were passed between the FEA and optimization programs. An FEA simulation was run for each design generated by the genetic algorithm, and the

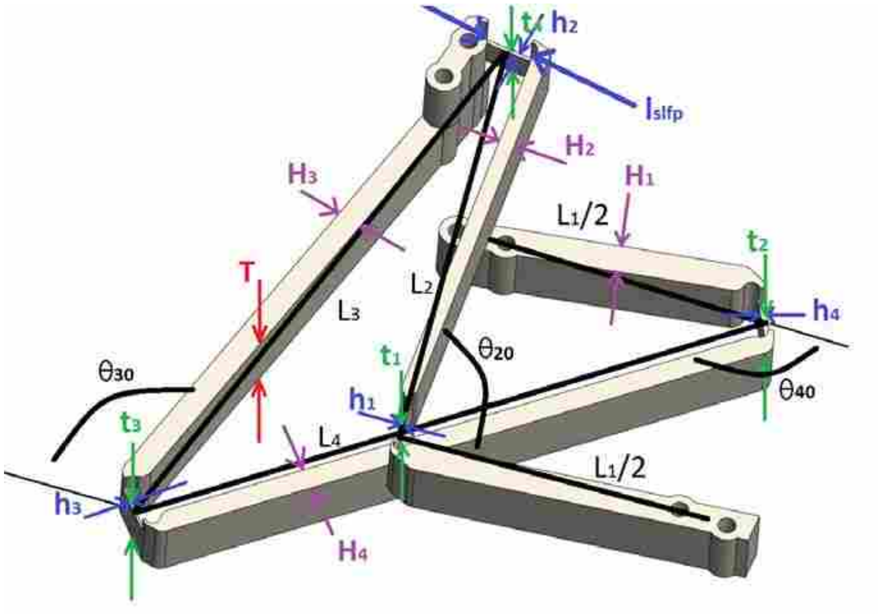


Figure 2.4: Geometric parameters of FE model.

FEA results were used to calculate design fitness. This required the FEA and optimization programs to run simultaneously. In our implementation the genetic algorithm waited for the FEA software to solve all the designs in a generation before continuing on to the next generation. The FEA could be parallelized to speed algorithm completion by analyzing multiple designs simultaneously.

Table 2.1: Parameters for the genetic algorithm.

Parameter	Value
Population Size	16
Generations	100
Crossover Probability	50%
Mutation Probability	4%
Tournament Size	3
Number of Variables	22

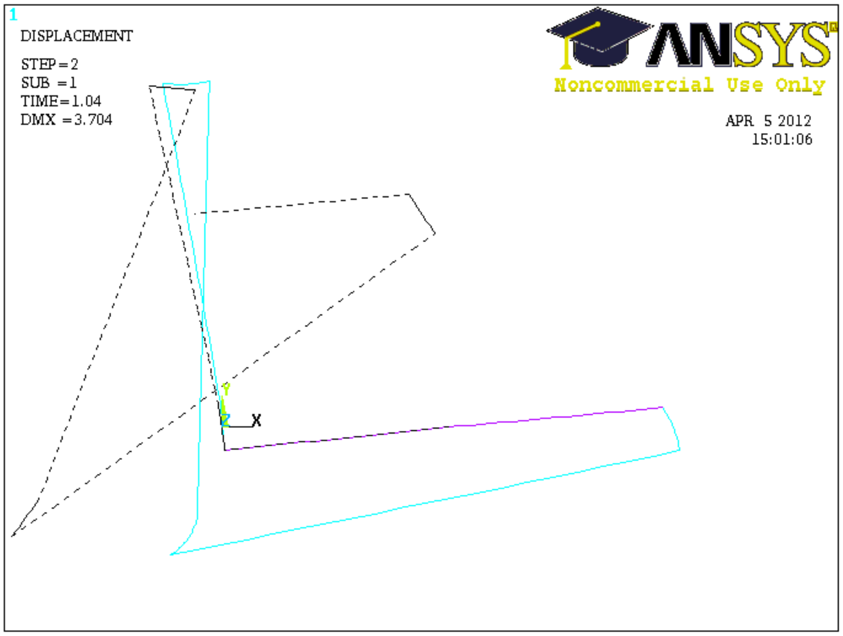


Figure 2.5: The ANSYS beam model used in the genetic optimization algorithm.

2.4 Prototypes

The results from the genetic algorithm were used to build and test a prototype to verify the design. Tabs were added to the prototype design to allow incorporation into a pantograph. The pantograph was assembled and tested, with results discussed below.

2.4.1 Mechanism Design

Table 2.2 summarizes the design parameters from the best design of the final generation. The algorithm terminated early (after about 20 generations) because the average fitness stopped changing, but the final design showed enough promise that efforts to increase diversity were deemed unnecessary. Increasing population size, decreasing tournament size, or increasing the mutation rate could increase diversity.

2.4.2 Prototype Fabrication

First, a single mechanism was built to validate the geometric design parameters selected by the genetic algorithm. We built our prototypes out of polypropylene for its low cost and high $\frac{S_y}{E}$ ratio. All prototypes were modeled in the commercial solid modeling package SolidWorks and exported as parasolids (.x_t files). Tool paths were generated using GibbsCAM and the parts were made on a CNC mill. Figure 2.3 shows the prototype in various stages of assembly.

2.4.3 Testing

After assembly the mechanism was tested using an Omega digital force gage and visual angle measurement. The force gage was attached perpendicular to the input link 5 cm from the characteristic pivot and force was measured at different points across the mechanism's range of motion. The displacement angle is the angle between the input link and ground. Torque was calculated by multiplying the measured force by the distance from the characteristic pivot to the point where the force gage was attached. Figure 2.6 shows

Table 2.2: Geometric design parameters. See Figure 2.4 for explanation.

Parameter	Value
L_1	12.90 cm
L_2	9.87 cm
L_3	13.23 cm
L_4	13.68 cm
h_1	0.450 mm
h_2	0.759 mm
h_3	0.467 mm
h_4	1.186 mm
t_1	0.49 cm
t_2	0.59 cm
t_3	0.88 cm
t_4	0.74 cm
θ_{20}	1.707 rad
θ_{30}	1.902 rad
θ_{40}	2.486 rad
H_1	0.64 cm
H_2	0.32 cm
H_3	0.52 cm
H_4	0.47 cm
l_{slfp}	1.27 cm
T	1.27 cm

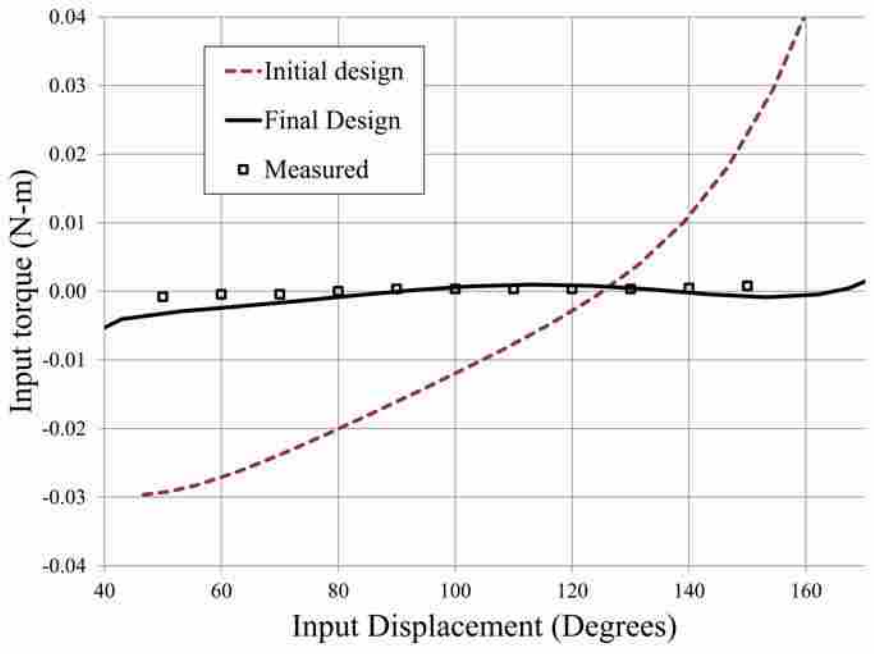


Figure 2.6: The predicted and measured torque displacement of the pseudo-four-bar balanced mechanism compared with an unoptimized initial design.

the measured torque-displacement curve, with the predicted torque of the final design and a randomly generated design from the first generation of the genetic algorithm for comparison. The initial design has only one stable point and lacks a constant torque region. The final design has three stable positions and a large region of near-zero input torque. The measured data show good agreement with the predicted input torque.

It was determined that an increase in off-axis stiffness was needed before implementation into an application. To this end we fabricated a second mechanism and stacked the two together. Figure 2.7 shows the assembly. The resulting symmetry yielded a significant increase in off-axis stiffness.

Two configurations are possible when implementing the rotary joint into a mechanism. Figure 2.7 depicts these configurations. Rigidly joining links to the members highlighted in green results in a mechanism with its characteristic pivot at A. Correspondingly, rigidly joining links to the members highlighted in red results in a mechanism with a characteristic pivot located at B. For the pantograph presented below, pivot A is used. Tabs and bearing mounts were added to the appropriate links to facilitate assembly of the pantograph.

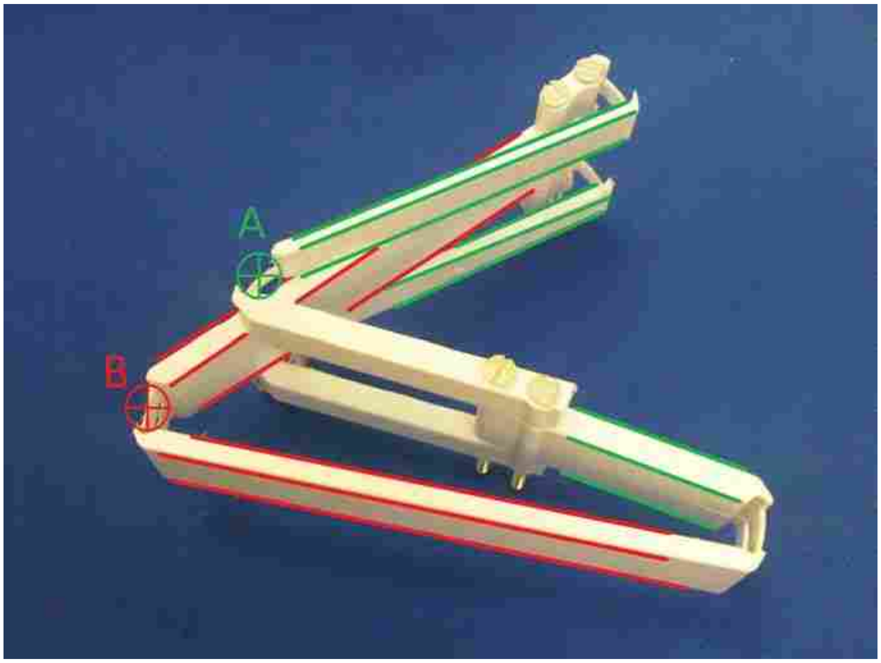


Figure 2.7: Stacking two mechanisms together resulted in a significant increase in off-axis stiffness. Rigidly joining links to the green-highlighted members results in a characteristic pivot at A, while rigidly joining links to the red-highlighted members results in a characteristic pivot at B.

2.5 Implementation in a 2 DOF Pantograph

Haptic interfaces are force-feedback robotic devices that give users a sense of touch while interacting with virtual or distant environments. Pantograph mechanisms have been implemented in haptic interfaces because of their large workspace and relative uniformity of response over the workspace [25, 26, 12], and have been selected as the basis for some commercial haptic interfaces [27]. The compliant haptic pantograph that was developed as a test platform for this work (Figure 2.8) is similar in structure to other haptic pantographs. The user typically grips a knob at A and moves it through the device's planar workspace. Motors at B and C exert torques that are converted to feedback forces at the user's hand through the device's links, which are typically connected by ball bearings at D and E. In the device shown in Figure 2.8 the links and bearings have been replaced by integrated statically balanced compliant joints. The motors and knob are also absent and have been replaced by low-friction ball bearings with hardware for attachment to the testing apparatus. The potential advantage of integrating compliant joints into haptic interfaces is a significant

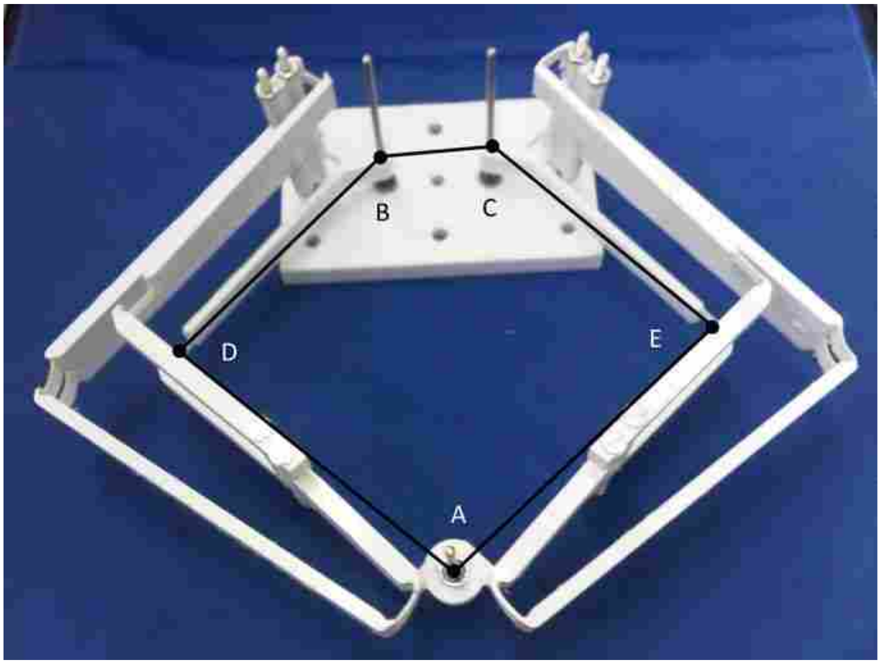


Figure 2.8: The partially compliant pantograph assembled. The dark lines show the conventional five-bar that has been replaced by two statically balanced mechanisms.

reduction in friction and mass, two factors that typically detract from the realism of the forces felt by the user. The use of statically balanced joints is significant because it eliminates the return-to-center force that would otherwise be felt by the user during haptic interactions.

Figure 2.9 shows the mechanism mounted in a custom force-displacement testing apparatus. A linear actuator takes the mechanism through its range of motion while linear potentiometers measure precise displacement. The actuator is coupled with the y-axis potentiometer, which rest on a carriage that is mounted to an air bearing to allow free movement along the x-axis. X-axis displacement is measured by a second potentiometer. The force measurement is obtained from a Kistler model 9212 quartz dynamic load cell mounted to the end of the actuator.

The resulting data was filtered with a running average to reduce noise and is shown in Figure 2.10. Each set of points represents an extension or retraction of the actuator. As can be seen, input force is low along both axes, with a nearly constant force for as much as 150 mm of displacement. The irregularities at the extremes of motion show the limits of the constant force behavior.

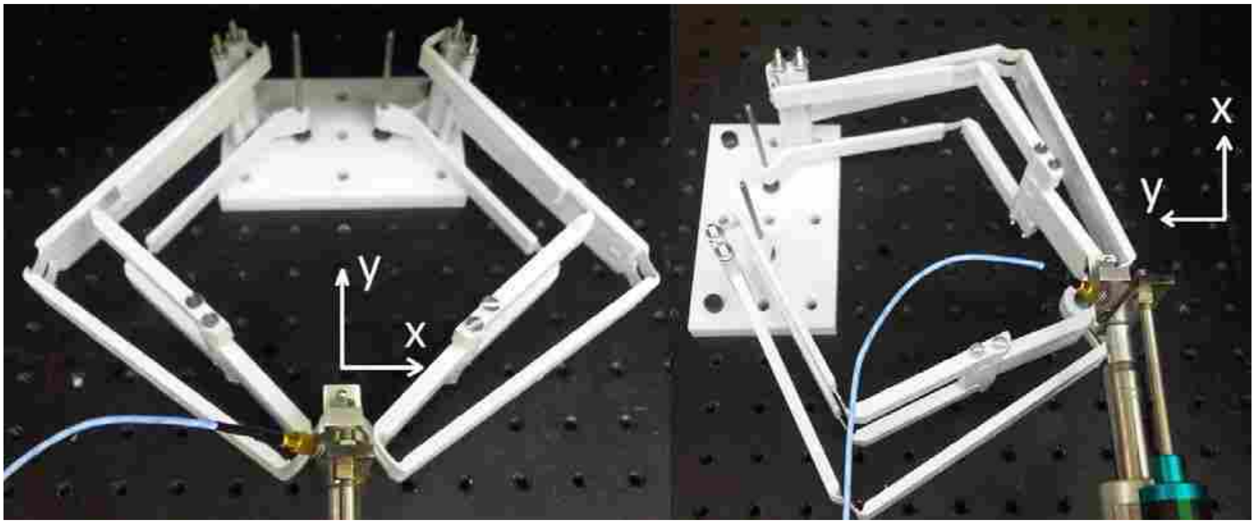


Figure 2.9: The partially compliant pantograph final design mounted in the testing apparatus. Orthogonal mounting positions allow testing along two axes.

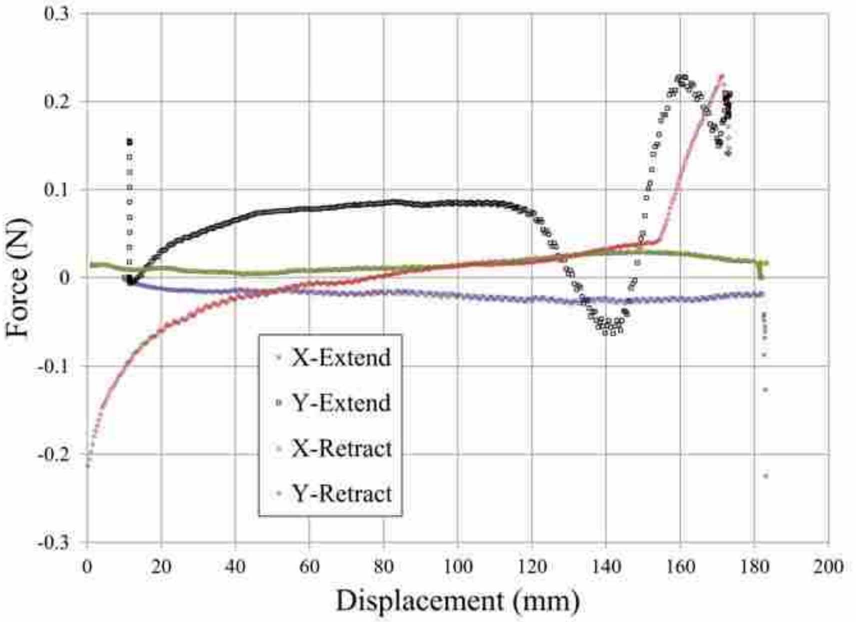


Figure 2.10: Force displacement characteristics of the completed pantograph.

2.6 Conclusion

We have described the development of a zero-torque, large displacement compliant mechanism. With the aid of a custom genetic algorithm and finite element modeling we found a configuration that resulted in the desired behavior. A prototype demonstrated the viability of the balancing concept and exhibited large displacement with low torque. This balanced design was incorporated in a five-bar pantograph. The results demonstrate the viability of optimized fully compliant rotary joints in statically balanced mechanisms.

CHAPTER 3. MONOLITHIC TWO-DEGREE-OF-FREEDOM POINTING MECHANISM

This work describes the conception, modeling, and development of a fully compliant, two-degree-of-freedom pointing mechanism for application in spacecraft thruster, antenna, or solar array systems. The design objectives and the advantages of a compliant solution are briefly discussed. A design concept is selected for final development from a field of generated concepts. Detailed design decisions to meet project objectives are described. Analytical and numerical models are developed and subsequently validated by prototype testing and measurements in several iterations. A final design is described in detail and its performance is measured.[1]

3.1 Introduction

Current space mechanisms have arrived at a state of highly optimized design [28]. They are subject to many harsh environmental conditions not often encountered on earth, including thermal gradients, vibrational loading, and operation in a vacuum [15]. As designers seek to reduce space mechanism size and cost, new mechanism types and topologies must be considered. One general type of mechanism that could have wide applications in spacecraft design is the pointing mechanism. The capability of pointing a thruster could eliminate the need for multiple thruster arrays, reducing part count and potential failure points. The ability to accurately point a communications antenna could decrease the power required to send data. Similarly, pointing a solar array would give the ability to orient it for optimal capturing of solar radiation, increasing efficiency of energy capture.

The goal of this work is to develop a versatile pointing mechanism for use in spacecraft that is light weight and eliminates friction.

[1] This chapter has been accepted for review in the Journal of Mechanical Sciences with Jonathan Jones, Spencer Magleby, and Larry Howell contributing as co-authors.

3.2 Background

Various types of pointing mechanisms have been evaluated for use in space. The classic gimbals, sometimes known as a Cardan suspension, can provide good accuracy but proper arrangement of actuators is a non-trivial problem. Single-pivot pointing mechanisms using Hooke or ball-and-socket joints have several feasible configurations, but their actuation is complex. Azimuth-over-elevation pointing mechanisms are essentially two tilting mechanisms mounted in series with orthogonal axes of rotation. These tend to be heavier than other pointing mechanisms and must have slip rings or cables with a service loop for signal transfer. [15]

More recent developments have addressed the issue of positioning and orienting an object with devices such as the Agile Eye [29, 30] and the Canfield joint [31]. The Canfield joint has a large workspace and parallel architecture that allows wires and plumbing to be routed without risk of pinching or shearing, and served as early inspiration for this work.

This new pointer design is approached by applying the advantages of compliant mechanisms. Compliant mechanisms achieve their mobility through deflection of flexible members [7, 14]. This avoids sliding contact, eliminating friction and wear along with the need for lubrication. Reducing the number of traditional joints can reduce or eliminate backlash, which is a contributing factor to the use of compliant mechanisms in precision instrumentation [32]. Additionally, compliant mechanisms lend themselves to monolithic construction, which greatly enhances a mechanism's robustness to thermal deformation [15]. Finally, compliant mechanisms can often be designed with significant weight savings when compared to traditional mechanisms. [7]

3.3 Design Objectives

Design objectives for the proposed pointer are given in Table 3.1. It is intended that the rotational range specification be increased for future design iterations: fifteen degrees was chosen as a baseline. Repeatability must be high to eliminate the need for active position sensing. Center shift must be low to reduce reaction torques when the thruster load is applied, and to increase pointing accuracy. The thruster load was chosen for a standard thruster size, and the life specification is for earth-orbit satellites.

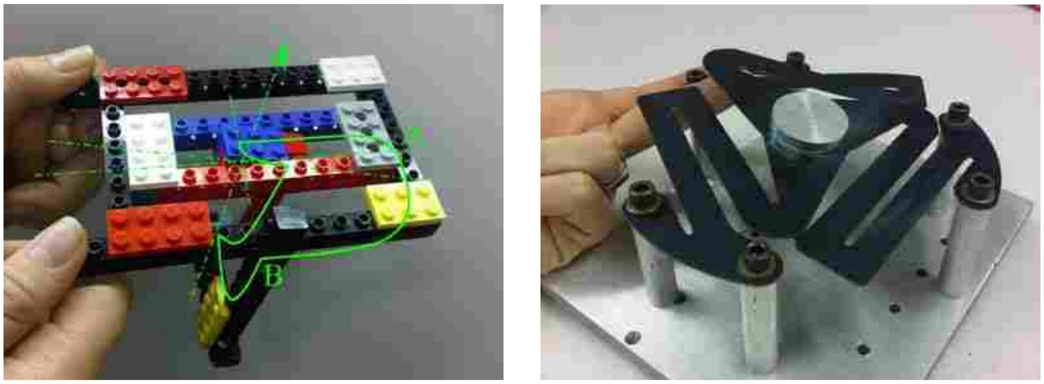
Table 3.1: Design objectives of space pointer

Description	Target Value	Achieved Value
Rotational range	±15°	±15°
Repeatability	High	High
Center Shift	<0.5 mm	0.28 mm
Thruster load	445 N	445 N
Mass	<200 g	93.2 g
First Natural Frequency	>50 Hz	55 Hz
Life (cycles)	>1,000	100,000
Allows space for electronics/fuel lines	Yes	Yes

3.4 Concept Development

Concept generation was performed with various techniques, including brainstorming, biomimetics [33], patent searches, FACT [34], and evaluating analogous designs in use elsewhere, including in origami. Over fifty distinct concepts relating to mechanism topology and performance were generated and considered. Of these, two of the top concepts are illustrated in Figure 3.1. Both give the required motion.

The selected concept (see Figure 3.1a) is a spherical five-bar mechanism. It requires only two actuators and can passively support the applied thruster load. It is superior to a Cardan suspension in this application because with the addition of the driver dyad both actuators can be attached to ground, reducing rotational inertia. With proper attention to geometry, it can have the aforementioned desirable qualities of the Canfield joint, e.g., plumbing and wiring are not at risk of damage from pinching or shearing.



(a) Spherical 5-bar with central pointing platform. Rotations at A and B give two DOF.

(b) Ortho-planar spring with central pointing platform. The platform can be tilted at each corner for three DOF (two rotations and extension).

Figure 3.1: Top concepts for compliant pointer.

3.5 Rigid Body Replacement Synthesis

Rigid body replacement synthesis [14, 35] was used to convert the kinematic concept of a five-bar spherical mechanism to a compliant mechanism. To select a compliant pivot for use in the mechanism, four pivot types were evaluated for suitability: the cross-axis flexural pivot (CAFP) [36], the split-tube flexure (STF) [37], the cartwheel hinge (CH) [38], and the small-length flexural pivot (SLFP) [1]. Metrics for comparison were stiffness, material volume, envelope, and center shift. The results are shown in Table 3.2. For comparison, the flexure geometry was chosen for minimum mass and equal stress (486 MPa) at a specified deflection (15°). Ultimately, the CAFP was selected for most rotary joints because of its low stiffness, low mass, and small envelope. The split-tube flexure was selected to replace one rotary joint because of its high off-axis stiffness and the ease with which it could be integrated into the mechanism structure. Figure 3.2 shows graphically the progression from rigid kinematic mechanism to the fully compliant mechanism.

In this application, flexure design is limited by space, manufacturing, stability, and stress. To increase the range of motion and decrease actuation torque, a thin, narrow, long flexure is desired. Electron Beam Melting (EBM, see Section 3.7.1) material data [39, 40, 41] provided a reasonable assumed ultimate tensile stress (1,034 MPa) and standard deviation (62 MPa). A conservative S-N fatigue curve [42] was generated for EBM produced titanium parts, shown in Figure 3.3. When a load (such as firing a thruster) is applied to the pointer, stress will increase and the number of cycles to failure will decrease.

The maximum stress occurring in the CAFP can be approximated as

$$\sigma = \frac{S_\theta \theta E t}{2l\cos(45^\circ)} \tag{3.1}$$

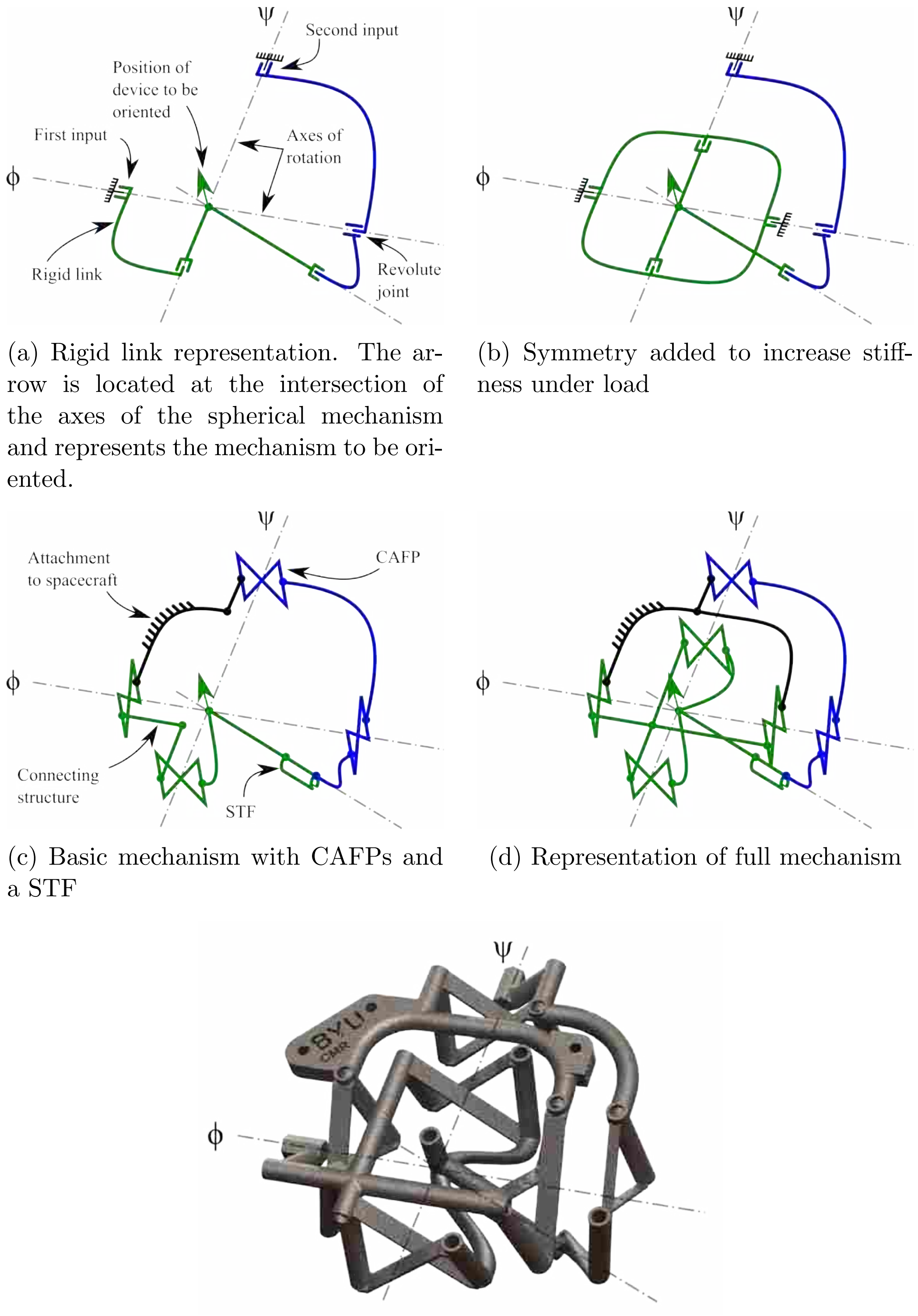
where σ is stress, $S_\theta = 0.965$ is a CAFP coefficient, θ is rotation, E is elastic modulus, t is thickness, and l is segment length [36]. Using $\theta = 0.261$ rad, $E = 111$ GPa, $t = 1$ mm, and $l = 4.06$ cm, the stress was estimated to be 494 MPa. The S-N curve in Figure 3.3 approximates a fatigue life of 100,000 cycles, which is well above the design objective of 1000 cycles. However, this estimate is for low load applications such as positioning communications or solar-power systems.

3.6 Mechanism Models

Before a comprehensive model of the mechanism could be developed, it was desirable to establish accurate models of the component flexures. The flexures were modeled using commercial Finite Element Analysis (FEA) software (ANSYS). The CAFP was represented with BEAM4 elements, and this model was compared to the pin joint model presented by Jensen et al. [36]. The STF was modeled with BEAM188 elements and defined as an arbitrary cross section. Section properties were determined from Pilkey [43] and elementary beam theory. The STF FEA model was verified using the analytical solution by Goldfarb et al. [37]. Table 3.3 contains comparisons of the analytical and FEA models of both the

Table 3.2: Comparison of flexures for equal stress at 15° deflection.

Metric	STF	CH	CAFP	SLFP
Torsional Stiffness (N-m/rad)	20.38	16.22	3.26	3.03
Material Volume (cm^3)	3.09	6.27	0.52	0.52
Envelope (cm^3)	17.79	294.60	5.24	0.52
Center Shift (mm)	-	0.20	0.33	-



(a) Rigid link representation. The arrow is located at the intersection of the axes of the spherical mechanism and represents the mechanism to be oriented.

(b) Symmetry added to increase stiffness under load

(c) Basic mechanism with CAFPs and a STF

(d) Representation of full mechanism

(e) Rendering of CAD model

Figure 3.2: Kinematics

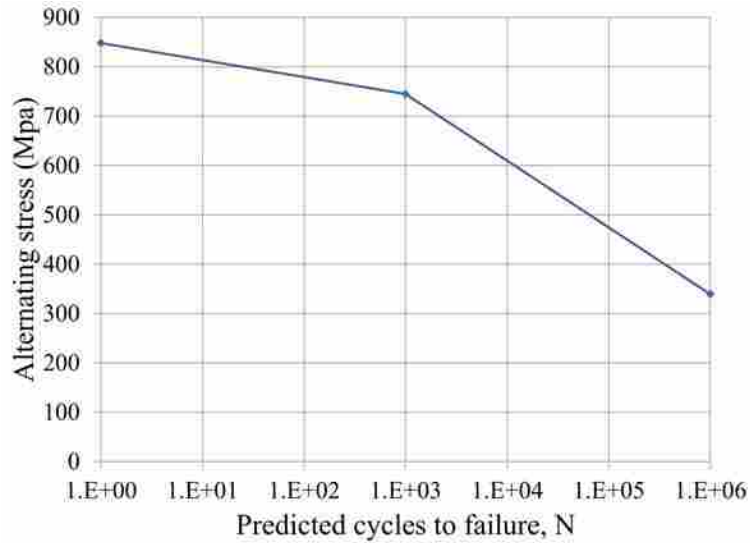


Figure 3.3: Estimated S-N curve for EBM produced Ti6Al4V

CAFP and STF. Agreement between the FEA and analytical solutions gave confidence that the models used have a sufficient degree of accuracy.

The kinematics were derived using 3D rotation matrices, given in Equation 3.2, where variables are depicted in Figure 3.4 and $g = \frac{\pi}{4}$. Multiplying out the matrices gives nine equations, three of which (see Equation (3.3)) can be solved for unknowns θ, γ, and η. γ and η were found using function fmincon in Matlab, while θ could be found directly once γ and η were known. Combining the kinematics, analytical models of the flexures, PRBM [1], and the principle of virtual work [7], an analytical model of the mechanism was developed that is capable of predicting input torques (see Equation (3.4)).

Table 3.3: Comparison of FEA and analytical predictions of stiffness for CAFP and STF. All stiffnesses are in N-m/rad.

	FEA	Analytical	% Difference
CAFP	3.05	3.26	−6.94%
STF	21.34	20.38	4.5%

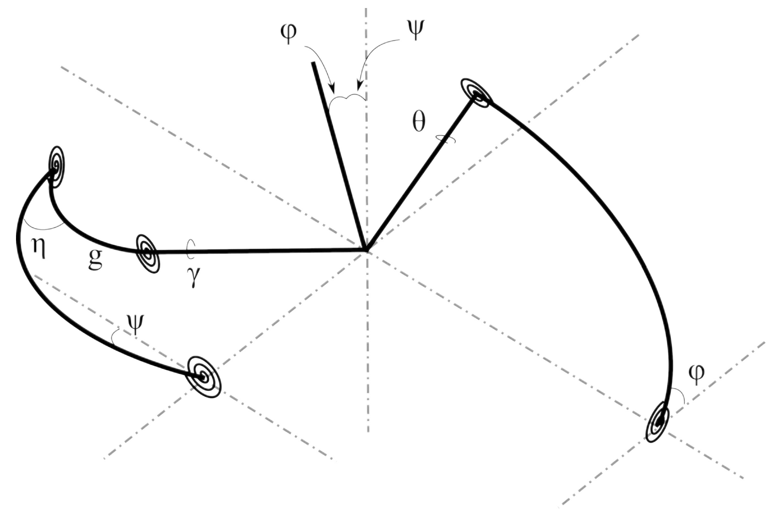


Figure 3.4: Angle nomenclature used to develop Equation (3.2).

$$
\begin{aligned}
&[R_x(90^\circ)][R_y(\theta)][R_z(90^\circ)][R_y(\phi)] = \\
&[R_x(-90^\circ)][R_y(-g)][R_x(\gamma)][R_y(-g)][R_x(\eta)][R_z(-90^\circ)][R_x(\psi)]
\end{aligned}
\tag{3.2}
$$

$$
\begin{aligned}
c(\phi) &= c(\gamma)c(\eta) - \frac{s(\gamma)s(\eta)}{\sqrt{2}} \\
0 &= \frac{s(\gamma)s(\psi)}{\sqrt{2}} - (c(\gamma)s(\eta) + \frac{s(\gamma)s(\eta)}{\sqrt{2}})s(\psi) \\
c(\theta) &= (\frac{1}{2} + \frac{c(\gamma)}{2})c(\psi) - (-\frac{s(\gamma)s(\eta)}{\sqrt{2}} + (-\frac{1}{2} + \frac{c(\gamma)}{2})c(\eta))s(\psi)
\end{aligned}
\tag{3.3}
$$

$$
\begin{bmatrix} \sum_{i=0}^{n} k_i \tilde{\theta}_i \frac{d\theta_i}{dq_1} \\ \sum_{i=0}^{n} k_i \tilde{\theta}_i \frac{d\theta_i}{dq_2} \end{bmatrix} = \begin{bmatrix} \frac{d\phi}{dq_1} & \frac{d\psi}{dq_1} \\ \frac{d\phi}{dq_2} & \frac{d\psi}{dq_2} \end{bmatrix} \begin{bmatrix} T_\phi \\ T_\psi \end{bmatrix}
\tag{3.4}
$$

The FEA flexure models were combined and expanded to represent the entire mechanism. A mesh refinement study was implemented to find an element size for a mesh-independent solution. The resulting FEA model is shown in Figure 3.5. Figure 3.6 shows FEA and analytical predictions of required torque for ten combinations of input angles. The

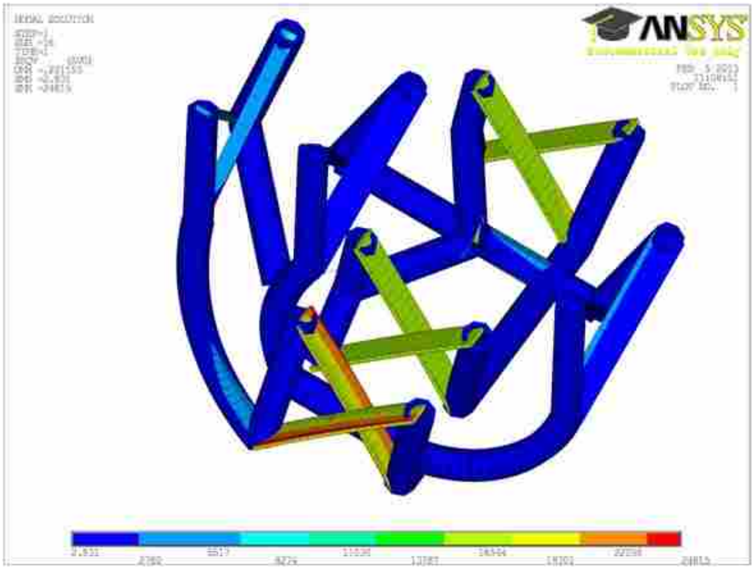


Figure 3.5: FEA representation of the mechanism at a deflected position. Contours are Von Mises stress in lbs/in^2.

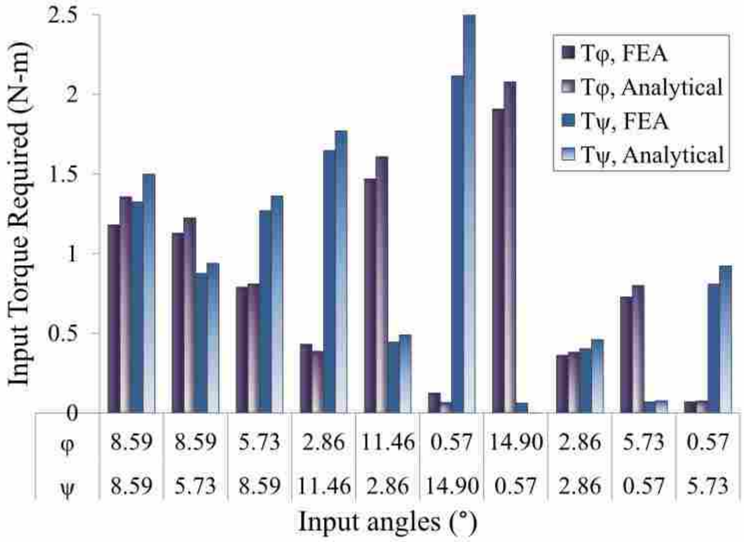


Figure 3.6: Comparison of FEA and analytical models at ten sets of command angles. T_ϕ and T_ψ are torque required for given rotation about the ϕ and ψ axes, respectively.

models agree closely on T_ψ, but T_ϕ predictions show a discrepancy roughly proportional to the magnitude of the commanded angle. This discrepancy is due to parasitic motion in the FEA model; the reduced stiffness of that model results in a lower torque prediction.

3.6.1 Connecting Structure

Three section types were evaluated for use in the structure that connects the pivots: solid square, hollow round, and square channel. The section dimensions were determined by using a constant cross sectional area (.403 cm^2) and a wall thickness (where applicable) of 1.27 mm. This approach allowed connecting structures of equal mass to be evaluated based on their relative stiffness only. To compare each structure's stiffness, an FEA modal analysis was conducted for each section type. The results are listed in Table 3.4, and indicate that for equal mass the round tube structure is stiffer than either of the competing section types. It was found that increasing the stiffness of the connecting structure significantly improved the mechanism's precision by reducing center shift (see Table 3.5). Although increasing structural stiffness increases stress in the flexures, the benefits of improved precision dictate that the connecting structure be as rigid as possible while still maintaining low mass. Therefore, round, hollow tubes were selected for the structure.

3.7 Prototypes

A CAD model was built parametrically in the commercial package Solidworks to allow geometry to be altered if necessary. Mounting points were added to the ground link, and hex stubs incorporated along the ϕ- and ψ-axes to enable actuation during testing. During the testing of early prototypes, torque watches and torque wrenches were used to apply and measure actuation torques, and protractors provided angle measurement.

Table 3.4: Modal analysis of connecting structure sections. First and second frequencies correspond to the desired degrees of freedom. Frequencies in Hz.

Natural Frequency	Solid Square	Hollow Circle	Square Channel
1st	15.5	15.6	14.8
2nd	19.6	19.7	19.3
3rd	55.2	73.8	50.3
4th	86.4	126.1	68.0
5th	108.5	135.8	84.2

First prototypes were built from FlexLinks [44](custom compliant mechanism prototyping elements) and LEGO® bricks (see Figures 3.7a and 3.7b). The prototype in Figure 3.7b was used to further validate the FEA and analytical models, and to better visualize the mechanism behavior. The FEA model correctly predicted the torque required to deflect the prototype to several positions.

Before full-scale prototyping in metal, several titanium flexures were fabricated (see Figures 3.7c and 3.7d) to examine their suitability for EBM and test their behavior. The torque-displacement behavior of these flexures compared with FEA and analytical predictions is shown in Table 3.6. The differences in stiffnesses for different flexures are attributed to variations in manufactured flexure thickness. The differences between measured and predicted values is also due to high surface roughness, causing the effective thickness of the pivot to be less than the measured thickness. Applying a correction factor of 0.83 to the flexure thickness in the FEA model results in a good agreement with measured stiffness.

As a final check of the CAD model before creating a titanium mechanism using EBM, the mechanism was produced on an FDM printer in ABS plastic. This ABS prototype is shown in Figure 3.7e. After minor cosmetic changes the design was produced in Ti6Al4V using EBM at Marshall Space Flight Center. This final prototype is shown in figure 3.7f.

3.7.1 Manufacturing Considerations

To enable the manufacture of the complex geometry in Titanium, Electron Beam Machining (EBM) was selected as the manufacturing process for the final device. EBM is a variant of 3D printing or rapid prototyping, where the part is built from successive layers

Table 3.5: Static analysis of connecting structures. All values are averages over eight representative sets of command angles. Rotational error is a root sum square difference from zero and command angles.

	Solid Square	Hollow Circle	Square Channel
Center Shift (mm)	0.259	0.256	0.363
Rotational Error (rad)	0.019	0.017	0.024
Max Von Mises Stress (MPa)	453	475	422

of metal powder that are melted together one after the other. This additive manufacturing process allows geometries to be created that would be difficault to fabricate using traditional methods. EBM allows parts to be designed with reduced manufacturing constraints [45]. Adherence to certain design guidelines can greatly improve part quality, and were incorporated in the space pointer to facilitate manufacturing. For example, overhanging geometry should be avoided, especially when features are thin [46]. In cases where overhangs are unavoidable, support geometry should be included that could be removed during post-processing. Additionally, the wall thickness should be at least 1.0 mm, and the maximum cross section of one build layer is 100 cm^2.

3.8 Testing

To confirm that the final prototype behaves as desired, a testing regime was implemented using techniques more refined than the simplified methods used for initial prototypes.

3.8.1 Test Method

Custom fixtures were built to attach the mechanism to an optical breadboard. Worm-and-wheel gear sets were used to apply pure rotational displacements in series with Omega torque transducers. Transducer output was measured with a National Instruments USB carrier coupled with a LabView DAQ. Transducer calibration was accomplished by hanging masses from a lever arm of known length to apply a known, constant torque.

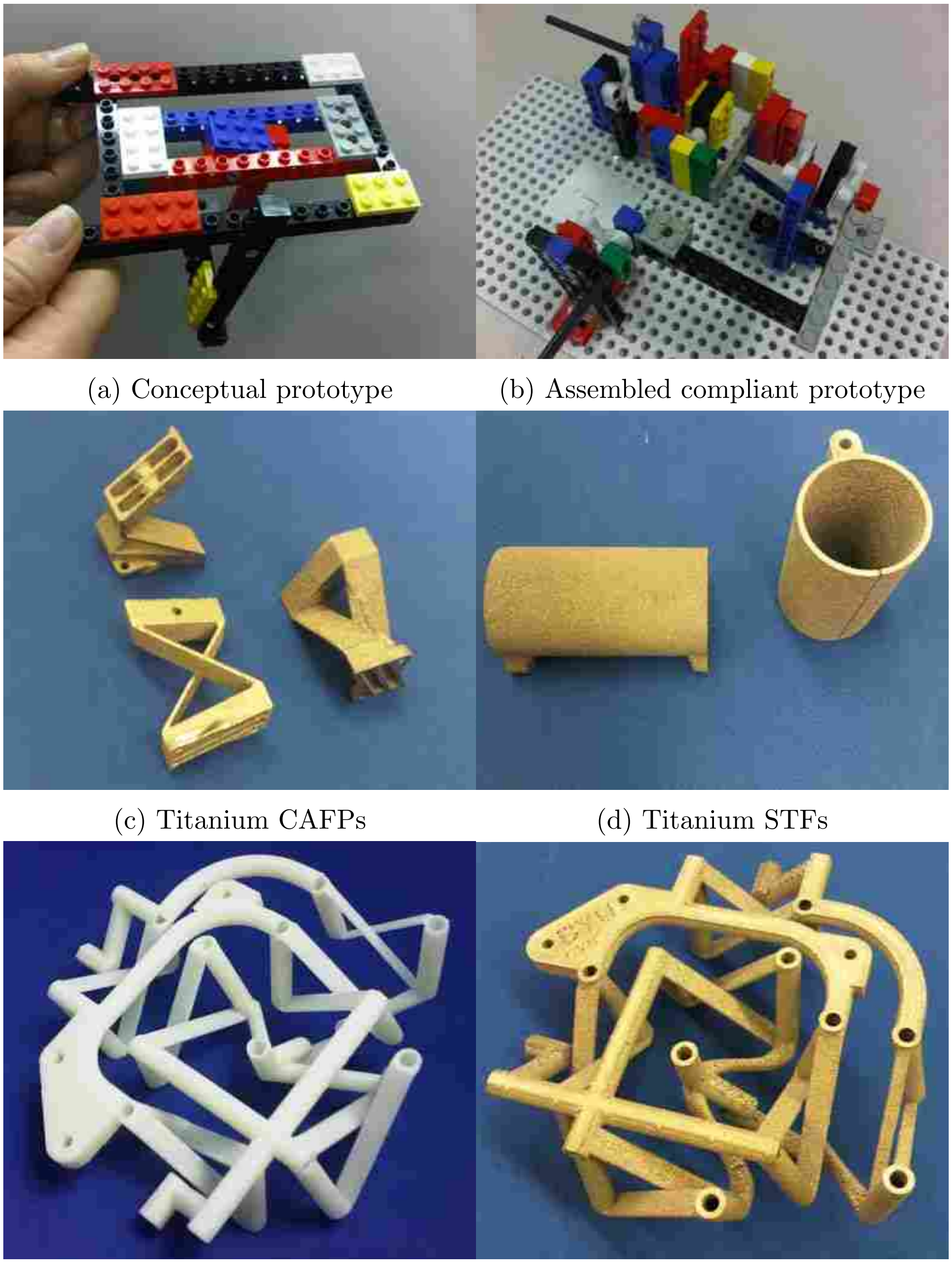
Table 3.6: Predicted and measured stiffnesses of two flexure types. A, B, C, α, and β designate average values of stiffness for each prototype flexure.

(a) CAFPs

	FEA	A	B	C
Stiffness (N-m/rad)	6.0	5.6	3.8	4.1

(b) STFs

	FEA	α	β
Stiffness (N-m/rad)	23.6	10.4	12.9



(a) Conceptual prototype (b) Assembled compliant prototype

(c) Titanium CAFPs (d) Titanium STFs

(e) Monolithic ABS prototype (f) Titanium prototype

Figure 3.7: Prototypes

Stereoscopic imaging was used for position measurement because of its ability to rapidly determine locations in three dimensions. Images were captured using Dino-Lite AM3011 Digital Microscopes. The test set-up is shown in Figure 3.8. Processing and position finding was done using scripts written in Matlab. Platform orientation was found by

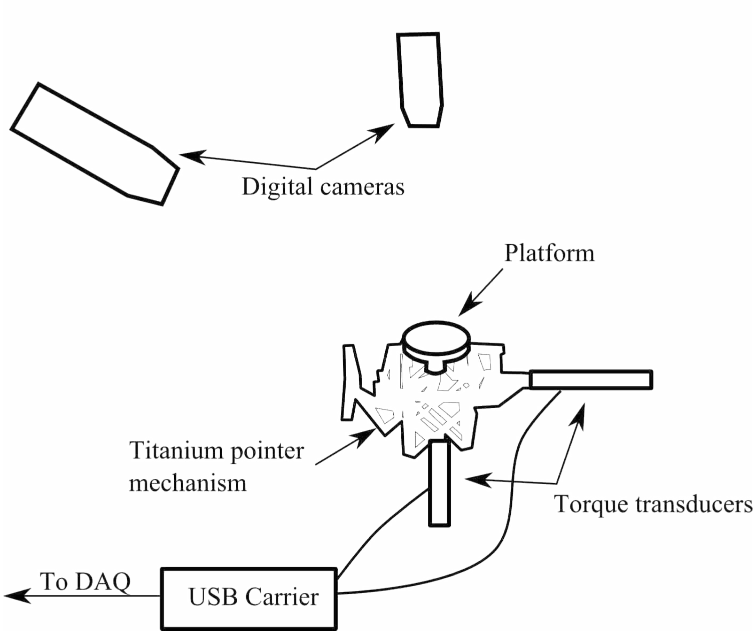


Figure 3.8: Test set-up for measuring applied torque and resulting displacement.

identifying the location of points A, B, and C (see Figure 3.9) and using $\vec{N} = \vec{AB} \times \vec{AC}$. The output angles were found as $\phi = \arctan(\frac{N_x}{N_z})$ and $\psi = \arctan(\frac{-N_y}{N_z})$.

Center shift was found using points B, C, D, and E from Figure 3.9. These points were chosen because they are equidistant from the center of rotation.The point locations were found for all 45 data sets. Gradient based optimization (the Matlab function fmincon) was used to determine the sphere radius, r, that best fit these points to its surface. Using this calculated r and the undeflected positions of B, C, D, and E, a center of rotation (X_0, Y_0, and Z_0) was also found. For each data set a relative center of rotation was found (x_0, y_0, and z_0). The distance from this displaced center to the non-displaced center was then calculated and recorded as center shift.

Repeatability was tested by selecting four sets of command torques T_ϕ and T_ψ and actuating the mechanism to each set four times. The rotations were measured and compared.

3.8.2 Results

To validate the FEA model, torque and displacement were measured at a variety of command angles. The flexure thickness correction factor (0.83) described previously was

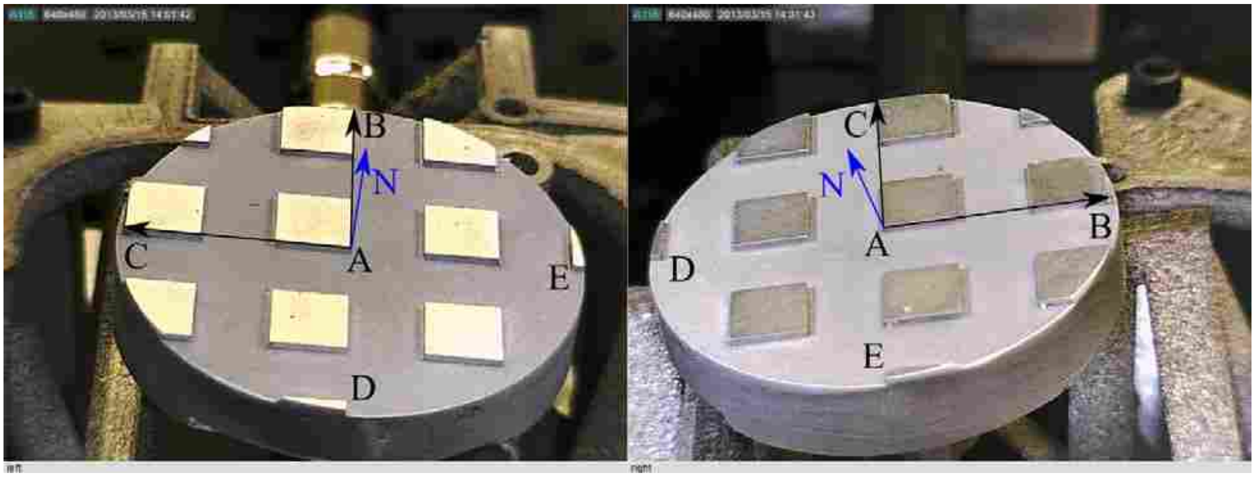


Figure 3.9: Points A, B, C, D, and E with vector $\vec{N}$ used for measurement of ϕ and ψ.

used to correct for changes between the designed and as fabricated flexure thicknesses. Good agreement between measured behavior and the FEA model was achieved, as shown in Figures 3.10, 3.11, and 3.12. Table 3.1 tabulates the final values of the design metrics. Most values meet or exceed the design goals. Note that rotational accuracy will depend on the method of actuation and associated control system, and was not quantified here. Center shift was used as an alternate metric of performance in lieu of rotational accuracy. This is the amount the mechanism center moves from the undeflected position. Center shift data is tabulated in Table 3.7, which shows that center shift is larger than anticipated but still quite small. It was found that the repeatability of the mechanism is high. Figure 3.13 shows that differences in measured position are less than the measurement error for four sets of command torques.

Table 3.7: Center shift data for analytical model, numerical model, and measured behavior. Analytical maximum center shift is a root sum squared total for a single hinge. Numerical and measured data were collected from 45 distinct sets of applied torques. All data in millimeters.

Metric	Analytical	Numerical	Measured
Mean	-	0.18	0.28
Standard Deviation	-	0.10	0.12
Max	0.57	0.46	0.63
Min	-	0.02	0.08

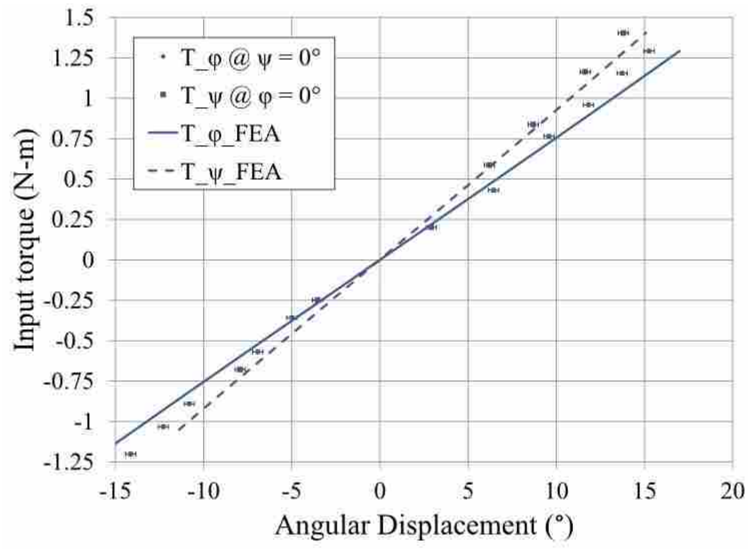


Figure 3.10: Comparison of measured and predicted T_ϕ and T_ψ at $\psi = 0$ and $\phi = 0$, respectively for the titanium prototype.

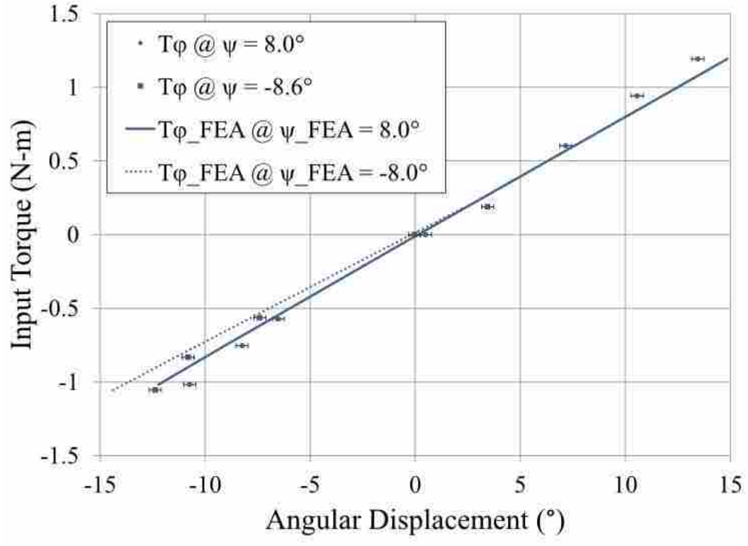


Figure 3.11: Comparison of measured and predicted input torques (T_ϕ) at two values of ψ for the titanium prototype.

3.9 Conclusion

We have described the modeling and development of a fully compliant spherical pointing mechanism suitable for use in space applications. Monolithic construction avoids many design challenges of the space environment. Adequate precision was obtained while providing a platform capable of supporting thruster loads up to 450 N. Every design objective in

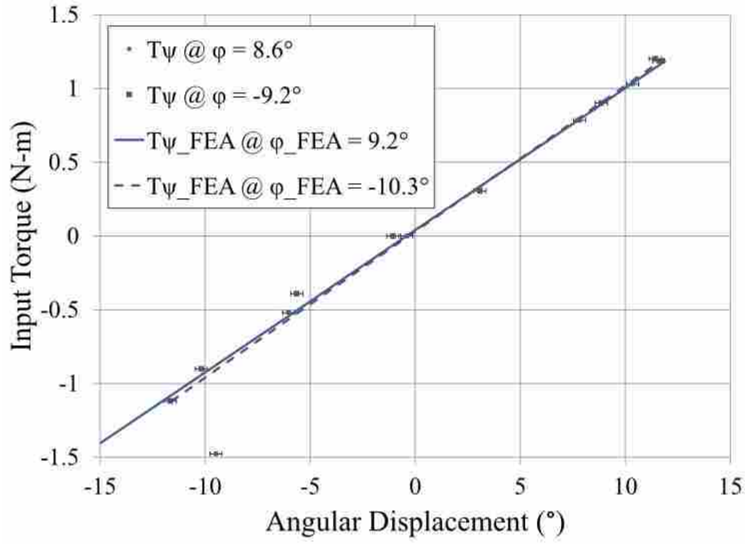


Figure 3.12: Comparison of measured and predicted input torques (T_ψ) at two values of ϕ for the titanium prototype.

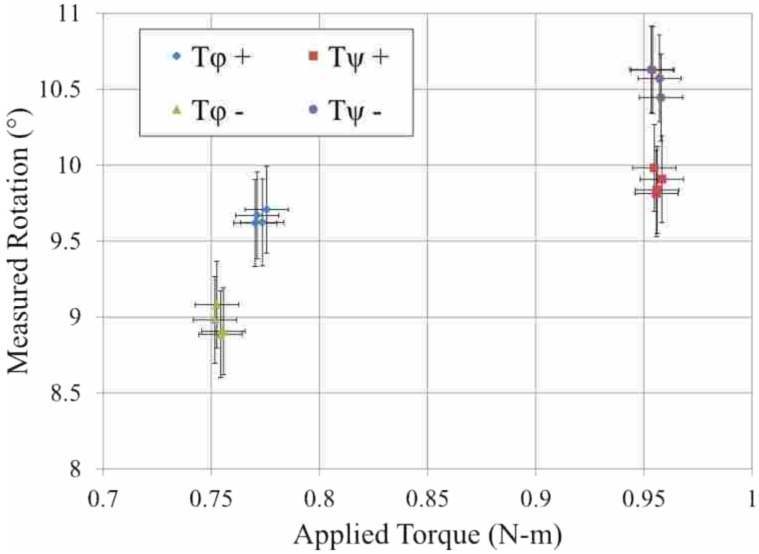


Figure 3.13: Measured rotation for applied torques T_ϕ and T_ψ at $\psi = 0$ and $\phi = 0$, respectively. All data are absolute values with actual sign indicated by the legend. Data is for titanium prototype.

Table 3.1 has been met or exceeded. This mechanism could have applications in thruster platforms or solar array or antenna mountings. The advantages of doing so include reduction in part count, increased robustness, and increased performance.

The performance of this pointer mechanism shows that combining compliant mechanisms with the manufacturing capabilities of 3D printing has the potential to revolutionize the way space mechanisms are designed. This work shows that monolithic mechanisms can provide high performance in multi-axis systems while eliminating friction, wear, backlash, and lubrication issues. Future work will include efforts to utilize static balancing as a strategy for reducing actuation effort, and optimizing for more desirable frequency response while tailoring the design for specific applications.

3.10 Acknowledgments

The authors acknowledge Eric Wilcox for his support in building testing apparatus, and Scott Thompson, PhD for his help with the stereoscopic imaging. This work was supported in part by the National Science Foundation through grants 0800606 and 1240417. The support of Marshall Space Flight Center for EBM fabrication of the test flexures and final titanium mechanism is acknowledged.

CHAPTER 4. CONCLUSIONS AND RECOMMENDATIONS

4.1 Summary

Taken as a whole, the research described herein deals with the advancement of compliant mechanism design, with a particular focus on design strategies and mechanisms that replace traditional bearings in mechanical systems. Current factors that limit the proliferation of compliant mechanisms include high actuation efforts and complex geometries. In this thesis I have addressed these issues and presented solutions.

Chapter 2 describes the development of a zero-torque, large displacement compliant mechanism. With the aid of a custom genetic algorithm and finite element modeling I found a configuration that resulted in the desired behavior. A prototype demonstrated the viability of the balancing concept and exhibited large displacement with low torque. This balanced design was incorporated in a five-bar pantograph. The results demonstrate the viability of optimized fully compliant rotary joints in statically balanced mechanisms.

Chapter 3 describes the modeling and development of a fully compliant spherical pointing mechanism suitable for use in space applications. Monolithic construction avoids many design challenges of the space environment. Adequate precision was obtained while providing a platform capable of supporting thruster loads up to 450 N. This mechanism could have applications in thruster platforms or solar array or antenna mountings. The advantages of doing so include reduction in part count, increased robustness, and increased performance.

4.2 Conclusions

It has been shown that near-zero-torque mechanisms are possible and can be used to replace bearings in a haptic pantograph. This application is general enough that mechanisms developed using similar methods could be used in other applications to replace traditional joints.

Additionally, it has been demonstrated that multi-axis systems can be accurately modeled using FEA and numerical methods. The 2 DOF pointer mechanism showed that constructing complicated monolithic geometries using EBM is practical, and can result in mechanisms that exhibit good performance characteristics without relying on traditional bearings for their motion.

Taken together, these developments expand the capabilities of compliant mechanism design, allowing designers to eliminate bearings in new applications i robotics and space.

4.3 Recommendations

Combining static balancing with monolithic construction has the potential to allow designers to reap the benefits of both design strategies. The difficulty comes in applying a pre-load to a monolithic mechanism. Developing simple strategies for accomplishing this without increasing mechanism complexity should be the focus of future research.

Also, much of the work presented in Chapter 3 was done with the intended application being in space technology. Because of the complex launch environment, the vibration characteristics of mechanisms become very important. Compliant mechanisms generally have a low first natural frequency that corresponds to necessary degrees of freedom. This can result in poor frequency response characteristics. Static balancing will exacerbate this shortcoming by reducing mechanism stiffness even further. Designers would benefit from the development of techniques for improving frequency response of compliant mechanisms. Such developments would allow designers to incorporate compliant mechanisms into new applications.

REFERENCES

[1] Howell, L., and Midha, A., 1994. "A method for the design of compliant mechanisms with small-length flexural pivots." *Journal of Mechanical Design,* **116**(1), March, pp. 280–290.

[2] Gaskill, S., and Went, S., 1996. "Safety issues in modern applications of robots." *Reliability Engineering & System Safety,* **53**(3), pp. 301–307.

[3] Tuijthof, G. J., and Herder, J. L., 2000. "Design, actuation and control of an anthropomorphic robot arm." *Mechanism and Machine Theory,* **35**(7), pp. 945 – 962.

[4] Siciliano, B., and Khatib, O., eds., 2008. *Springer Handbook of Robotics.* Springer.

[5] Shibata, T., 2004. "An overview of human interactive robots for psychological enrichment." *Proceedings of the IEEE,* **92**(11), Nov., pp. 1749 – 1758.

[6] Michell, A., 1929. "Progress of Fluid-Film Lubrication." *Trans. ASME,* **51**(2), pp. 153 – 164.

[7] Howell, L., 2001. *Compliant Mechanisms.* John Wiley & Sons, Inc.

[8] Henein, S., and Spanoudakis, P., 2003. "Flexural Pivot for Aerospace Mechanisms." *10th European Space Mechanisms and Tribology Symposium*, September.

[9] Stapel, A., and Herder, J. L., 2004. "Feasibility Study of a Fully Compliant Statically Balanced Laparoscopic Grasper." *ASME Conference Proceedings,* **2004**(46954), pp. 635–643.

[10] Mihelj, M., Nef, T., and Riener, R., 2007. "ARMin II - 7 DoF Rehabilitation Robot: Mechanics and Kinematics." pp. 4120 –4125.

[11] Leishman, L. C., Ricks, D., and Colton, M. B., 2010. "Design and Evaluation of Statically Balanced Compliant Mechanisms for Haptic Interfaces." *Proc. ASME Dynamic Systems and Control Conference*, September.

[12] Gillespie, R., Shin, T., Huang, F., and Trease, B., 2008. "Automated Characterization and Compensation for a Compliant Mechanism Haptic Device." *Mechatronics, IEEE/ASME Transactions on,* **13**(1), Feb., pp. 136 –146.

[13] C-Flex Bearing Co., Inc., 2012. http://www.c-flex.com Accessed 2012-9-17.

[14] Howell, L. L., Magleby, S. P., and Olsen, B. M., 2013. *Handbook of Compliant Mechanisms.* Wiley.

[15] Fusaro, R., 1999. “NASA space mechanisms handbook and reference guide.” *NASA/TP-1999-206988.*

[16] Kim, C., and Ebenstein, D., 2012. “Curve Decomposition for Large Deflection Analysis of Fixed-Guided Beams With Application to Statically Balanced Compliant Mechanisms.” *Journal of Mechanisms and Robotics,* **4**(4).

[17] Herder, J. L., 2001. *Energy-free Systems. Theory, conception, and design of statically balanced spring mechanisms.*

[18] Pluimers, P., Tolou, N., Jensen, B. D., Howell, L. L., and Herder, J., 2012. “A Compliant On/Off Connection Mechanism for Preloading Statically Balanced Compliant Mechanisms, DETC2012-71509.” In *Proceedings of the ASME International Design Engineering Technical Conferences*, ASME.

[19] Schenk, M., Guest, S., and Herder, J., 2007. “Zero stiffness tensegrity structures.” *International Journal of Solids and Structures,* **44**(20), pp. 6569 – 6583.

[20] Deepak, S. R., and Ananthasuresh, G., 2012. “Static balancing of a four-bar linkage and its cognates.” *Mechanism and Machine Theory,* **48**(0), pp. 62 – 80.

[21] Deepak, S. R., and Ananthasuresh, G. K., 2012. “Perfect Static Balance of Linkages by Addition of Springs But Not Auxiliary Bodies.” *Journal of Mechanisms and Robotics,* **4**(2), p. 021014.

[22] Sonmez, U., and Tutum, C. C., 2008. “A Compliant Bistable Mechanism Design Incorporating Elastica Buckling Beam Theory and Pseudo-Rigid-Body Model.” *Journal of Mechanical Design,* **130**(4), p. 042304.

[23] Tolou, N., 2012. *Statically Balanced Compliant Mechanisms for MEMS and Precision Engineering.*

[24] Dede, E., and Trease, B., 2004. “Statically-balanced Compliant Four-bar Mechanism for Gravity Compensation.” *2004 ASME Student Mechanism Design Competition.*

[25] Ramstein, C., and Hayward, V., 1994. “The pantograph: a large workspace haptic device for multimodal human computer interaction.” In *Conference Companion on Human Factors in Computing Systems*, CHI ’94, ACM, pp. 57–58.

[26] Campion, G., Wang, Q., and Hayward, V., 2005. “The Pantograph Mk-II: a haptic instrument.” In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, IEEE, pp. 193–198.

[27] Quanser 5-DOF Haptic Wand http://www.quanser.com Accessed 2012-11-10.

[28] Fowler, R., Howell, L., and Magleby, S. “Compliant space mechanisms: a new frontier for compliant mechanisms.” *Mechanical Sciences,* **2**.

[29] Gosselin, C., and Hamel, J., 1994. “The agile eye: A high-performance three-degree-of-freedom camera-orienting device.” In *1994 IEEE International Conference on Robotics and Automation*, IEEE, pp. 781–786.

[30] Gosselin, C., et al., 1999. Two degree-of-freedom spherical orienting device, Oct US Patent 5,966,991.

[31] Canfield, S., and Reinholtz, C., 1998. “Development of the carpal robotic wrist.” In *Experimental Robotics V*, A. Casals and A. DeAlmeida, eds., Vol. 232 of *Lecture notes in control and information sciences*, Springer-Verlag, London Ltd, pp. 423–434.

[32] Motsinger, R., 1964. “Flexural devices in measurement systems.” *Measurement Engineering,* **1**, pp. 383–435.

[33] Vakili, V., and Shu, L., 2001. “Towards biomimetic concept generation.” In *Proceedings of the ASME Design Engineering Technical Conference*, Vol. 4, Citeseer, pp. 327–335.

[34] Hopkins, J., and Culpepper, M., 2010. “Synthesis of multi-degree of freedom, parallel flexure system concepts via Freedom and Constraint Topology (FACT)–Part I: Principles.” *Precision Engi-neering,* **34**(2), pp. 259–270.

[35] Olsen, B. M., Issac, Y., Howell, L. L., and Magleby, S. P., 2010. “Utilizing a Classification Scheme to Facilitate Rigid-Body Replacement for Compliant Mechanism Design.” ASME.

[36] Jensen, B., and Howell, L., 2002. “The modeling of cross-axis flexural pivots.” *Mechanism and Machine theory,* **37**(5), May, pp. 461–476.

[37] Goldfarb, M., and Speich, J. E., 1999. “A well-behaved revolute flexure joint for compliant mechanism design.” *Journal of Mechanical Design,* **121**(3), pp. 424–429.

[38] Pei, X., Yu, J., Zong, G., Bi, S., and Su, H., 2009. “The modeling of cartwheel flexural hinges.” *Mechanism and Machine Theory,* **44**(10), pp. 1900–1909.

[39] Robert, Stirling, and Ring, 2008. EOS titanium Ti64 for EOSINT M 270 systems (titanium version) www.eos.info, Dec Accessed 2012-11-10.

[40] Morris Technologies, Inc. Titanium Ti64 www.morristech.com Accessed 2012-11-10.

[41] Svensson, M., 2009. Material properties of EBM-manufactured Ti6Al4V & Ti6Al4V ELI under RAW and HIP conditions Arcam AB.

[42] Norton, R. L., 2006. *Machine Design An Integrated Approach, 3E.* Pearson Prentice Hall.

[43] Pilkey, W. D., 2002. *Analysis and design of elastic beams.* John Wiley &Sons, Inc.

[44] Brigham Young University Compliant Mechanisms Research Group FlexLinks http://compliantmechanisms.byu.edu/downloads/flexlinks Accessed 2012-11-1.

[45] Arcam AB Additive manufacturing (AM) http://www.arcam.com/technology/additive-manufacturing.aspx Accessed 2012-11-10.

[46] Cansizoglu, O., Harrysson, O. L. A., West, II, Harvey A, Cormier, D. R., and Mahale, T., 2008. “Applications of structural optimization in direct metal fabrication.” *Rapid Prototyping Journal,* **14**(2), pp. 114–122.

APPENDIX A. MATLAB AND ANSYS CODE FOR GENETIC ALGORITHM

To employ FEA in conjunction with a genetic algorithm, we wrote a series of Matlab and Ansys batch files that interact with each other to pass needed data back and forth. Figure A.1 depicts this flow of information graphically. The files are included below.

A.1 Genetic Algorithm

This file is the genetic algorithm used for the optimization of the pseudo-four-bar mechanism. Length units are in inches. This code must run at the same time as the Ansys batch file.

```
%clear
crossprob = .5; %probability of crossover
mutaprob = .04; %probability of mutation
generations = 100; %includes original parents
population = 16; %designs in each generation (must be even number)
tourn = 3; %tournament size
numpars = 22; %number of parameters
% param1 is out of plane thickness of rigid segments (can be 1/4", 3/8",
% 1/2"
% param2 is in plane thickness of rigid segments
% param3 is length of SLFPs
% param4 is material: (1 polyprop, 2 HDPE, 3 Nylon, 4 Delrin)
% params5-8 are link lengths
% params9-12 SLFP thicknesses out of plane
% params13-14 are t20, t30, and t40
% params16-19 are SLFP thicknesses in plane

%initialize first generation

s1 = 1; %spread of link1
s2 = 1; %spread of link2
s3 = 1; %spread of link3
s4 = 1; %spread of link4
ll1 = 4.8; %lower limit of link1
ll2 = 3.3; %lower limit of link2
ll3 = 4.8; %lower limit of link3
```

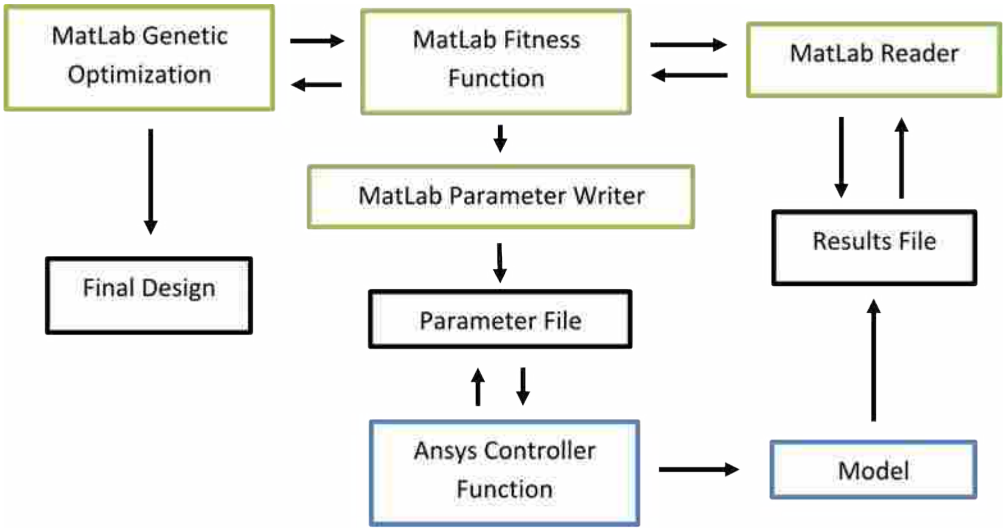


Figure A.1: Flowchart depicting optimization algorithm.

```
ll4 = 4.9; %lower limit of link4
st20 = .7; %Spread t20
st30 = .7; %spread t30
st40 = .7; %spread t40
llt20 = 1.4; %lower limit of t20
llt30 = 1.4; %lower limit of t30
llt40 = 2.3; %lower limit of t40
cont = 0;
if cont == 1
    resume = 10; %generation you want to start from
    genstart = gen(:,:,resume);
    %fitstart = fit(:,resume,:);
    fitstart = fit(1,:);
    gen = zeros(numpars,population,generations);
    gen(:,:,1) = genstart;
    fit = zeros(generations, population);
    fit(1,:);
    %fit = zeros(3,generations,population);
    %fit(:,1,:) = fitstart;
else
    itnum = 1; %identifier used to keep track of results files
    gen = zeros(numpars,population,generations);
    gen(1,:,1) = ceil(rand(1,population)*3); %out of plane thickness
    gen(2,:,1) = rand(1,population)*.25+.25; %In plane thickness of rigid segments
    gen(4,:,1) = ceil(rand(1,population)*4); %possible material properties
    gen(5,:,1) = rand(1,population)*s1+ll1; %link lengths
    gen(6,:,1) = rand(1,population)*s2+ll2;
    gen(7,:,1) = rand(1,population)*s3+ll3;
    gen(8,:,1) = rand(1,population)*s4+ll4;
    thk = rand(4,population);
    for R = 1:population
        if gen(1,R,1) == 1 %out of plane thickness = .25
            m = .125;
            b = .125;
        elseif gen(1,R,1) == 2 %.375
            m = 3/16;
            b = 3/8;
        else
            m = .25;
            b = .25;
        end
            gen(9,R,1) = thk(1,R)*m+b; %SLFP thickness out of plane
            gen(10,R,1) = thk(2,R)*m+b;
            gen(11,R,1) = thk(3,R)*m+b;
            gen(12,R,1) = thk(4,R)*m+b;
    end

    gen(13,:,1) = rand(1,population)*st20+llt20; %initial angles
    gen(14,:,1) = rand(1,population)*st30+llt30;
    gen(15,:,1) = rand(1,population)*st40+llt40;

    gen(3,:,1) = rand(1,population)*.3+.2; %SLFP length
    gen(16,:,1) = rand(1,population)*.01+.015; %SLFP thickness in plane
    gen(17,:,1) = rand(1,population)*.04+.015;
    gen(18,:,1) = rand(1,population)*.01+.015;
    gen(19,:,1) = rand(1,population)*.04+.015;
    gen(20,:,1) = rand(1,population)*.25+.05; %in-plane thickness of link2
    gen(21,:,1) = rand(1,population)*.25+.05; %in-plane thickness of link3
    gen(22,:,1) = rand(1,population)*.25+.05; %in-plane thickness of link4
    %fit = zeros(3,generations,population);
    fit = zeros(generations, population);

    for R = 1:population
        chrom = gen(:,R,1);
        [fit(1,R), itnum] = fitness(itnum,chrom);
        fit(1,R)
    %[fit(2:3,1,R), itnum] = fitness(itnum, chrom);
    %fit(2:3,1,R)
    end
    %A = zeros(3,population);
    %A(1,:) = fit(2,1,
    %fit(1,1,:) = -maximin(fit(2:3,1,:),2);
```

```
end
for G = 2:generations
G;    % determining parents
mothers = ceil(rand(tourn,(population/2))*population); %randomly select which
%designs to enter in tournament
fathers = ceil(rand(tourn,(population/2))*population);
moms = zeros(1,(population/2)); %vector to hold information of which wins tournament
dads = zeros(1,(population/2));
X1 = zeros(tourn,(population/2)); %vector to hold participants in tournament
X2 = zeros(tourn,(population/2));
for S = 1:tourn
for R = 1:(population/2)

X1(S,R) = fit(G-1,mothers(S,R)); %fit(1,G-1,mothers(S,R)); %build vector that
%contains the fitness of each parent
X2(S,R) = fit(G-1,fathers(S,R)); %fit(1,G-1,fathers(S,R));

[~,Y] = min(X1(:,R));
[~,Z] = min(X2(:,R));
moms(R) = mothers(Y,R);
dads(R) = fathers(Z,R);
while moms(R) == dads(R) %ensure that the mother design is distinct from the
%father design
    mothers2 = ceil(rand(tourn,1)*population);
    fathers2 = ceil(rand(tourn,1)*population);
    X1(S,R) = fit(G-1,mothers2(S,1)); %fit(1,G-1,mothers2(S,1));
    X2(S,R) = fit(G-1,fathers2(S,1)); %fit(1,G-1,fathers2(S,1));
    [~,Y] = min(X1(:,R));
    [~,Z] = min(X2(:,R));
    moms(R) = mothers2(Y,1);
    dads(R) = fathers2(Z,1);
end
end
end

 %perform crossover
crossover = rand(numpars,population/2); %use these random numbers to determine
% crossover
for P = 1:population/2
    for C = 1:numpars;
       if C == 4 || C == 1 %discrete variables listed here to preserve integer
%values during crossover
             if crossover(C,P) < crossprob
             gen(C,P,G) = gen(C,dads(P),G-1);
             gen(C,P+population/2,G) = gen(C,moms(P),G-1);
             else
                 gen(C,P,G) = gen(C,moms(P),G-1);
                 gen(C,P+population/2,G) = gen(C,dads(P),G-1);
             end
       else
            if crossover(C,P) < crossprob
            gen(C,P,G) = crossover(C,P)*gen(C,moms(P),G-1)+(1-crossover(C,P))*...
gen(C,dads(P),G-1);
            gen(C,P+population/2,G) = (1-crossover(C,P))*gen(C,moms(P),G-1)+...
crossover(C,P)*gen(C,dads(P),G-1);
            else
                gen(C,P,G) = gen(C,moms(P),G-1);
                gen(C,P+population/2,G) = gen(C,dads(P),G-1);
            end

       end
   end
end

%Do mutations. Must be expanded for every added parameter

mutations = rand(numpars,population,2);
for P = 1:population
     if mutations(1,P,1)<mutaprob
         gen(1,P,G) = ceil(mutations(1,P,2)*3);
    end
    if mutations(2,P,1)<mutaprob
        gen(2,P,G) = mutations(2,P,2)*.05+.01;
    end
    if mutations(3,P,1)<mutaprob
        gen(3,P,G) = mutations(3,P,2)*.3+.2;
    end
    if mutations(4,P,1)<mutaprob
       gen(4,P,G) = ceil(mutations(4,P,2)*4);
    end
    if mutations(5,P,1)<mutaprob
        gen(5,P,G) = mutations(5,P,2)*s1+ll1; %link lengths
    end
    if mutations(6,P,1)<mutaprob
        gen(6,P,G) = mutations(6,P,2)*s2+ll2;
    end
    if mutations(7,P,1)<mutaprob
        gen(7,P,G) = mutations(7,P,2)*s3+ll3;
    end
    if mutations(8,P,1)<mutaprob
        gen(8,P,G) = mutations(8,P,2)*s4+ll4;
    end
    if mutations(9,P,1)<mutaprob
        if gen(1,P,G) == 1 %out of plane thickness = .25
        gen(9,P,G) = mutations(9,P,2)/8+1/8; %SLFP thickness out of plane
        elseif gen(1,P,1) == 2 %.375
        gen(9,P,G) = mutations(9,P,2)*3/16+3/16; %SLFP thickness out of plane
        else
        gen(9,P,G) = mutations(9,P,2)*.25+.25; %SLFP thickness out of plane
        end
    end
    if mutations(10,P,1)<mutaprob
        if gen(1,P,G) == 1 %out of plane thickness = .25
        gen(10,P,G) = mutations(10,P,2)/8+1/8; %SLFP thickness out of plane
        elseif gen(1,P,1) == 2 %.375
        gen(10,P,G) = mutations(10,P,2)*3/16+3/16; %SLFP thickness out of plane
        else
        gen(10,P,G) = mutations(10,P,2)*.25+.25; %SLFP thickness out of plane
        end
    end
    if mutations(11,P,1)<mutaprob
        if gen(1,P,G) == 1 %out of plane thickness = .25
        gen(11,P,G) = mutations(11,P,2)/8+1/8; %SLFP thickness out of plane
        elseif gen(1,P,1) == 2 %.375
        gen(11,P,G) = mutations(11,P,2)*3/16+3/16; %SLFP thickness out of plane
        else
        gen(11,P,G) = mutations(11,P,2)*.25+.25; %SLFP thickness out of plane
        end
    end
    if mutations(12,P,1)<mutaprob
        if gen(1,P,G) == 1 %out of plane thickness = .25
        gen(12,P,G) = mutations(12,P,2)/8+1/8; %SLFP thickness out of plane
        elseif gen(1,P,1) == 2 %.375
        gen(12,P,G) = mutations(12,P,2)*3/16+3/16; %SLFP thickness out of plane
        else
        gen(12,P,G) = mutations(12,P,2)*.25+.25; %SLFP thickness out of plane
        end
    end
    if mutations(13,P,1)<mutaprob
        gen(13,P,G) = mutations(13,P,2)*st20+llt20; %initial angles
    end
    if mutations(14,P,1)<mutaprob
        gen(14,P,G) = mutations(14,P,2)*st30+llt30;
    end
    if mutations(15,P,1)<mutaprob
```

```
        gen(15,P,G) = mutations(15,P,2)*st40+llt40;
    end
    if mutations(16,P,1)<mutaprob
        gen(16,P,G) = mutations(16,P,2)*.01+.015; %SLFP thickness in plane
    end
    if mutations(17,P,1)<mutaprob
        gen(17,P,G) = mutations(17,P,2)*.04+.015;
    end
    if mutations(18,P,1)<mutaprob
        gen(18,P,G) = mutations(18,P,2)*.01+.015;
    end
    if mutations(19,P,1)<mutaprob
        gen(19,P,G) = mutations(19,P,2)*.04+.015;
    end
    if mutations(20,P,1)<mutaprob
        gen(20,P,G) = mutations(20,P,2)*.25+.05;
    end
    if mutations(21,P,1)<mutaprob
        gen(21,P,G) = mutations(21,P,2)*.25+.05;
    end
    if mutations(22,P,1)<mutaprob
        gen(22,P,G) = mutations(22,P,2)*.25+.05;
    end
end

%find fitness of new generation
for R = 1:population
    chrom = gen(:,R,G);
    [fit(G,R), itnum] = fitness(itnum, chrom);
    %[fit(2:3,G,R), itnum] = fitness(itnum, chrom);
    %fit(2:3,G,R)
    fit(G,R)
end
%fit(1,G,:) = -maximin(fit(2:3,G,:),2);
%use elitism
Q = zeros(2*population,numpars+1);
for n = 1:population
    Q(n,:) = [fit(G-1,n),gen(:,n,G-1)'];
    Q(n+population,:) = [fit(G,n),gen(:,n,G)'];
    %Q(n,:) = [fit(1,G-1,n),gen(:,n,G-1)'];
    %Q(n+population,:) = [fit(1,G,n),gen(:,n,G)'];
end
P = sortrows(Q,1);
for n = 1:population
    fit(G,n) = P(n,1);
    for i = 1:numpars
        gen(i,n,G) = P(n,i+1);
    end
end

end %end of generations loop
genwrite(gen,fit);
```

A.2 Fitness Function

This Matlab file contains the function ”fitness” called by the genetic algorithm. It returns a fitness value for each design produced by the algorithm.

```
function [fits, itnum] = fitness(itnum, chrom)

paramswriter(itnum,chrom); %write a parameter file with parameter values given
%in chrom
results = ansysreader(itnum); %read the corresponding output file

err = results(1);
spr = results(7);
pen = 0; %penalty function for high stresses
if chrom(4) == 1%assign material properties(PP, steel, Al, Ti)
    if results(3) > 4500
        pen = pen+5;
    end
    if results(4) > 4500
        pen = pen+5;
    end
    if results(5) > 4500
        pen = pen+5;
    end
    if results(6) > 4500
        pen = pen+5;
    end
elseif chrom(4) == 2
     if results(3) > 3800
        pen = pen+5;
    end
    if results(4) > 3800
        pen = pen+5;
    end
    if results(5) > 3800
        pen = pen+5;
    end
    if results(6) > 3800
        pen = pen+5;
    end
elseif chrom(4) == 3
     if results(3) > 8000
        pen = pen+5;
    end
    if results(4) > 8000
        pen = pen+5;
    end
    if results(5) > 8000
        pen = pen+5;
    end
    if results(6) > 8000
        pen = pen+5;
    end
else
     if results(3) > 9500
        pen = pen+5;
    end
    if results(4) > 9500
        pen = pen+5;
    end
    if results(5) > 9500
        pen = pen+5;
    end
    if results(6) > 9500
        pen = pen+5;
    end
end

fits = err+3*spr+pen;

itnum = itnum+1; %calculate fitness from the results
```

A.3 Parameter Writer

This function writes a text file readable by Ansys that contains model parameters for a single design.

```
function paramswriter(itnum, chrom)
params = fopen('parameters.txt','w');

% write variables
```

```
if chrom(1) == 1 %out of plane thickness of 'rigid' segments
    thk5 = .25;
elseif chrom(1) == 2
    thk5 = .375;
else
    thk5 = .50;
end

h5 = chrom(2); %in plane thickness of 'rigid' segments
slfp = chrom(3); %length of SLFPs
if chrom(4) == 1 %assign material properties(PP, HDPE, Nylon, Acetal (Delrin))
    mod = 200000;
    nu = .36;
elseif chrom(4) == 2
    mod = 155000;
    nu = .36;
elseif chrom(4) == 3
    mod = 350000;
    nu = .34;
else
    mod = 450000;
    nu = .34;
end
link1 = chrom(5); %link lengths
link2 = chrom(6);
link3 = chrom(7);
link4 = chrom(8);

thk1 = chrom(9); %SLFP thicknesses (out of plane)
thk2 = chrom(10);
thk3 = chrom(11);
thk4 = chrom(12);

t20 = chrom(13); %initial angles
t30 = chrom(14);
t40 = chrom(15);

h1 = chrom(16); %in plane thickenss of flexures
h2 = chrom(17);
h3 = chrom(18);
h4 = chrom(19);
h6 = chrom(20);
h7 = chrom(21);
h8 = chrom(22);
```

```
%set parameter values
range = 60*pi/180;

fprintf(params, ’*set,itnum_,%i\r\n’,itnum);
fprintf(params, ’*set,thk5,%f\r\n’,thk5); %change to thk5
fprintf(params, ’*set,h5,%f\r\n’,h5); %change to h5
fprintf(params, ’*set,slfp,%f\r\n’,slfp); %change to slfp
fprintf(params, ’*set,mod,%f\r\n’,mod);
fprintf(params, ’*set,nu,%f\r\n’,nu);
fprintf(params, ’*set,link1, %f\r\n’,link1);
fprintf(params, ’*set,link2, %f\r\n’,link2);
fprintf(params, ’*set,link3, %f\r\n’,link3);
fprintf(params, ’*set,link4, %f\r\n’,link4);
fprintf(params, ’*set,thk1, %f\r\n’,thk1);
fprintf(params, ’*set,thk2, %f\r\n’,thk2);
fprintf(params, ’*set,thk3, %f\r\n’,thk3);
fprintf(params, ’*set,thk4, %f\r\n’,thk4);
fprintf(params, ’*set,t20, %f\r\n’,t20);
fprintf(params, ’*set,t3, %f\r\n’,t30);
fprintf(params, ’*set,t4, %f\r\n’,t40);
fprintf(params, ’*set,h1, %f\r\n’,h1);
fprintf(params, ’*set,h2, %f\r\n’,h2);
fprintf(params, ’*set,h3, %f\r\n’,h3);
fprintf(params, ’*set,h4, %f\r\n’,h4);
fprintf(params, ’*set,h6, %f\r\n’,h6);
fprintf(params, ’*set,h7, %f\r\n’,h7);
fprintf(params, ’*set,h8, %f\r\n’,h8);
fprintf(params, ’*set,range, %f\r\n’,range);

fclose(params);
end
```

A.4 Ansys Reader

This file parses an Ansys output file and returns values to the fitness function so that design fitness can be calculated.

```
% ansys file reader
% adapted from C. Mattson
% Dec 23 2009
% Used to read in Ansys files into matlab.

function results=ansysreader(~)
sc = 34;         % THESE ARE THE SPACES TO THE LEFT OF THE TEXT TO SKIP
%BEFORE READING
```

```
slfirst = 5;    % These are the lines to SKIP before first reading these in.
%sl = 4;        % These are the lines to SKIP between Ansys output.
clo = 6;       % number of parameters to read
results = zeros(clo,1);
%results(2) = itnum-1;
new = 0;
while new == 0
    new = exist('results.txt','file');
    pause(.1);
end
pause(.1);
% while results(2) ˜= itnum
fid = fopen('results.txt','r');

% nlines = 6;   % End on this line

clnum = 0;

for i = 1:slfirst
        fgetl(fid);
        clnum = clnum+1;
end

    curline = fgetl(fid); %read in a line
    resultstr1 = curline(sc:sc+14); %store a piece of the line as a string
    results(1) = str2double(resultstr1); %convert that string to a double.
%integrated error of moment

    fgetl(fid);
    %curline = fgetl(fid);
    %resultstr2 = curline(sc:sc+16);
    %results(2) = str2double(resultstr2); %itnum_

    curline = fgetl(fid);

    resultstr3 = curline(sc:sc+14);
    results(3) = str2double(resultstr3); %max stress slfp1

    curline = fgetl(fid);

    resultstr4 = curline(sc:sc+14);
    results(4) = str2double(resultstr4); %max stress slfp2

    curline = fgetl(fid);
    resultstr5 = curline(sc:sc+14);
    results(5) = str2double(resultstr5); %max stress slfp3

    curline = fgetl(fid);

    resultstr6 = curline(sc:sc+14);
    results(6) = str2double(resultstr6); %max stress slfp4

    curline = fgetl(fid);

    resultstr7 = curline(sc:sc+14);
    results(7) = abs(str2double(resultstr7)); %spread of input torque

fclose(fid);
delete(’results.txt’);
pause(.2);
end
```

A.5 Generation Writer

This file contains the genwrite function that is called at the end of the genetic algorithm. It records the designs arrived at by the genetic algorithm.

```
function genwrite(gen, fit)
genfile = fopen(’generations.txt’,’w’);
fitfile = fopen(’fitness.txt’,’w’);
graphfile = fopen(’lastgen.txt’,’w’);
fprintf(fitfile, ’Gen\tFitness\r\n’);
[˜, pop, gens] = size(gen);
for G = 1:gens

    fprintf(fitfile, ’%i\t%f\r\n’,G,mean(fit(G,:)));%1,G,:)));
    fprintf(genfile, ’Gen\tDes\tOoPThk\tIPThk\tslfpLen\tMat\tr1\tr2\tr3\tr4\...
tb1\tb2\tb3\tb4\tt20\tt30\tt40\th1\th2\th3\th4\tl2thk\tl3thk\tl4thk\tFit\r\n’);
    for P = 1:pop
        fprintf(genfile, ’%i\t%i\t%i\t%f\t%f\t%i\t%f\t%f\t%f\t%f\t%f\...
t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\...
t\r\n’,G,P,gen(:,P,G),fit(G,P)); %1,G,P));
    end
    fprintf(genfile,’Average Fitness: %f\r\n’,mean(fit(G,:))); %1,G,:)));
end

while gen(1,1,gens) == 0
    gens = gens-1;
end
```

```
for P = 1:pop
    fprintf(graphfile, '%i\t%i\t%f\t%f\t%i\t%f\t%f\t%f\t%f\t%f\t%f\...
t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t\r\...
n',P,gen(:,P,gens),fit(G,P)); %1,gens,P));
end
fclose(genfile);
fclose(fitfile);
fclose(graphfile);
end
```

A.6 Ansys Controller File

This batch file must be running when you start running the genetic algorithm. It waits for a new parameter file, executes the file, and then calls the actual mechanism model.

```
fini
/clear
go = 1
itnumold = 0

*dowhile,go

itnumfin = 99999
/INPUT,'parameters','txt','/auto/fse/emerriam/optimization/GeneticProject/',1,0
*if,itnumold,EQ,itnum_,and,itnum_,NE,itnumfin,then
pause = 1
*dowhile,pause
/wait,.1
/INPUT,'parameters','txt','/auto/fse/emerriam/optimization/GeneticProject/',1,0
*if,itnumold,NE,itnum_,then
pause = 0
*endif
*enddo
*endif

itnumold = itnum_

*if,itnum_,eq,itnumfin,then
*go, stop
*endif

/INPUT,'mech','txt','/auto/fse/emerriam/optimization/GeneticProject/',1,0

*enddo
```

A.7 Mechanism model

This file contains the mechanism model. It writes an output file when the model finishes solving.

```
!initial (unstressed angles) of SLFPs
t30 = -t3
t40 = -t4

pi = acos(-1)

ndiv = 20

/prep7

k,1,0,0,0 !pin joint 1
k,2,link2*cos(t20),link2*sin(t20),0 !pin joint 2
k,3,link2*cos(t20)+link3*cos(t30),link2*sin(t20)+link3*sin(t30),0 !pin joint 3
k,4,link2*cos(t20)+link3*cos(t30)+link4*cos(t40+pi),link2*sin(t20)+link3*sin(t30)
+link4*sin(t40+pi),0  !pin joint 4
k,5,link1/2,0,0 !ground coupling 1
k,6,link2*cos(t20)+link3*cos(t30)+link4*cos(t40+pi)-link1/2,link2*sin(t20)+link3*
sin(t30)+link4*sin(t40+pi),0 !ground coupling 2
k,7,slfp/2*cos(t20),slfp/2*sin(t20),0 !SLFP 1
k,8,-slfp/2*cos(t20),-slfp/2*sin(t20),0
!SLFP 2
k,9,link2*cos(t20)+slfp/2*cos((t20+t30)/2),link2*sin(t20)+slfp/2*sin((t20+t30)/2),0
k,10,link2*cos(t20)-slfp/2*cos((t20+t30)/2),link2*sin(t20)-slfp/2*sin((t20+t30)/2),0
k,11,link2*cos(t20)+link3*cos(t30)-slfp/2*cos((t30+t40)/2),link2*sin(t20)+link3*
sin(t30)+slfp/2*sin((t30+t40)/2),0 !SLFP 3
k,12,link2*cos(t20)+link3*cos(t30)+slfp/2*cos((t30+t40)/2),link2*sin(t20)+link3*
sin(t30)-slfp/2*sin((t30+t40)/2),0
k,13,link2*cos(t20)+link3*cos(t30)+link4*cos(t40+pi)+slfp/2*cos(t40+1.5),link2*
sin(t20)+link3*sin(t30)+link4*sin(t40+pi)+slfp/2*sin(t40+1.5),0 !SLFP 4
k,14,link2*cos(t20)+link3*cos(t30)+link4*cos(t40+pi)-slfp/2*cos(t40+1.5),link2*
sin(t20)+link3*sin(t30)+link4*sin(t40+pi)-slfp/2*sin(t40+1.5),0

lstr,7,8   !line 1, SLFP 1
lstr,10,9  !line 2, SLFP 2
lstr,12,11 !line 3, SLFP 3
lstr,14,13 !line 4, SLFP 4
lstr,7,9  !line 5 (7,10)
lstr,10,12  !line 6 (9,12)
lstr,11,13 !line 7
lstr,5,8   !line 8
lstr,14,6  !line 9
```

```
et,1,beam23
keyopt,1,6,0
nlgeom,1

mptemp,1,0  !material properties for flexible segments
mpdata,ex,1,,mod
mpdata,prxy,1,,nu
!tb,biso,1,1,2
!tbtemp,0,1
!tbdata,,sy,tm

!material properties for ground
mpdata,ex,2,,mod*1000
mpdata,prxy,2,,nu

R,1,thk1*h1,thk1*h1**3/12,h1, !real constants for SLFP 1
R,2,thk2*h2,thk2*h2**3/12,h2, !real constants for SLFP 2
R,3,thk3*h3,thk3*h3**3/12,h3, !real constants for SLFP 3
R,4,thk4*h4,thk4*h4**3/12,h4, !real constants for SLFP 4
R,5,thk5*h5,thk5*h5**3/12,h5, !real constants for link1(ground)
R,6,thk5*h6,thk5*h6**3/12,h6, !real constants for link2
R,7,thk5*h7,thk5*h7**3/12,h7, !real constants for link3
R,8,thk5*h8,thk5*h8**3/12,h8, !real constants for link4

lsel,s,line,,1 !select SLFP 1
latt,1,1,
lsel,all
lsel,s,line,,2 !select SLFP 2
latt,1,2,
lsel,all
lsel,s,line,,3 !select SLFP 3
latt,1,3,
lsel,all
lsel,s,line,,4 !select SLFP 4
latt,1,4,
lsel,all
lsel,s,line,,5 !link 2
latt,1,6
lsel,all
lsel,s,line,,6 !link 3
latt,1,7
lsel,all
lsel,s,line,,7 !link 4
latt,1,8
```

```
lsel,all
lsel,s,line,,8,9 !link 1 (ground)
latt,2,5
lsel,all

esize,,ndiv
lmesh,all !mesh all lines


ksel,s,kp,,7
nslk,s
*get,mom,node,,num,max
ksel,all

ksel,s,kp,,7,7
nslk,s
*get,slfp1,node,,num,max
esln,s,0,all
*get,slfpe1,elem,,num,min
ksel,all

ksel,s,kp,,9,9
nslk,s
*get,slfp2,node,,num,max
esln,s,0,all
*get,slfpe2,elem,,num,min
ksel,all

ksel,s,kp,,11,11
nslk,s
*get,slfp3,node,,num,max
esln,s,0,all
*get,slfpe3,elem,,num,min
ksel,all

ksel,s,kp,,13,13
nslk,s
*get,slfp4,node,,num,max
esln,s,0,all
*get,slfpe4,elem,,num,min
ksel,all
nsel,all
esel,all

fini
```

```
/sol

dk,5,all,0
dk,6,rotz,0
dk,6,ux,link1/2-(link2*cos(t20)+link3*cos(t30)+link4*cos(t40+pi)-link1/2)
dk,6,uy,-(link2*sin(t20)+link3*sin(t30)+link4*sin(t40+pi))
nsubst,25,55,15,
outres,erase
outres,all,none
lswrite,1


dk,7,rotz,-range! !displacement on keypoint
nsubst,25,75,15,
outres,erase
outres,all,1
lswrite,2

dk,7,rotz,range*.2 !displacement on keypoint
nsubst,25,75,15
outres,erase
outres,all,1
lswrite,3

lssolve,1,3,1
fini

/post26
numvar, 15
esol,2,slfpe1,slfp1,nmis,1,s1max
esol,3,slfpe2,slfp2,nmis,1,s2max
esol,4,slfpe3,slfp3,nmis,1,s3max
esol,5,slfpe4,slfp4,nmis,1,s4max
rforce,6,mom,M,z,input
nsol,7,mom,rot,Z,disp
abs,13,7,,,absdsp
abs,11,6,,,absl v,  !find absolute value of reaction moments
!integrates the absolute value of the input moment with respect to
!displacement and calls the result error
int1,12,11,1,,error,,

!store the maximum values of stress at each SLFP and the max error
*get,s1max_,vari,2,extrem,vmax
*get,s2max_,vari,3,extrem,vmax
*get,s3max_,vari,4,extrem,vmax
```

```
*get,s4max_,vari,5,extrem,vmax
*get,error_,vari,12,extrem,vmax
*get,tmin,vari,6,extrem,vmin
*get,tmax,vari,6,extrem,vmax
*set,tspread_,tmax-tmin
!*get,failure_,active,0,solu,ncmss
!/delete,'results','txt','/auto/fse/emerriam/optimization/GeneticProject/',
/OUTPUT,'results','txt','/auto/fse/emerriam/optimization/GeneticProject/',
*status,prm_
/output
/output,'results%itnum_%','txt','/auto/fse/emerriam/optimization/GeneticProject/',
prvar,input,disp,s1max,s2max,s3max,s4max
/output
extrem,2,5,1
fini

/prep7
lclear,all
ldele,all,,,1

fini
```

APPENDIX B. ANSYS BATCH FILES AND MATLAB DLT SCRIPTS

This appendix contains Ansys batch files for analyzing the pointer mechanism, and Matlab files for use with stereoscopic vision measurements.

B.1 Basic Model

This file builds the pointer mechanism using lines and beam elements, applies a rotational displacement or a moment to the two input nodes, and solves the model for rotation at the platform and required torque.

```
!this is a model of a spherical 5-bar,
! or a cardan suspension with a driver dyad.
/cwd, 'C:\ANSYS'
finish
/clear
/prep7

!Units in, lbf, psi
/PNUM,KP,1   !turn line and keypoint numbering on
/PNUM,LINE,1
*SET,pi,acos(-1) !pi
*set,len,1.6
*set,cor,len*sin(pi/4)/2 !x and y coordinates

arcg = pi/4
arcf = pi/2
gap = .05 !gap between flexures for manufacturing

mod = 1.61e7 !modulus of titanium
!mod = 30e6 !modulus of inconel or steel
!mod = 200000 !polypropylene
!mod1 = 425000 !nylon
mod1 = mod !everything is titanium
nu = .34
ndiv = 20
esz1 = len/ndiv !element size for flexible members
esz2 = len/ndiv !element size for rigid members
sy = 120000
```

```
tm = 1000

!input angles
psiang = -0.10
fiang = 0.10

load = 00.0

!real constants for flexible beams
wid1 = .25  !width of pivots (.25)
thk1 = .04 !thickness
area1 = thk1*wid1
Izz1 = wid1*thk1**3/12
Iyy1 = thk1*wid1**3/12
Ixx1 = Izz1+Iyy1
tkz1 = thk1
tky1 = wid1
theta1 = pi/4

!real constants for z-axis pivots
r,1,area1,Iyy1,Izz1,tkz1,tky1,,,Ixx1,,,,,

!real constants for x-axis pivots
r,3,area1,Izz1,Iyy1,tky1,tkz1,,,Ixx1,,,,,

!real constants for f-g pivot flexures
wid3 = wid1/2
thk3 = thk1
area3 = wid3*thk3
Iyy3 = wid3*thk3**3/12
Izz3 = thk3*wid3**3/12
Ixx3 = Izz3+Iyy3
tkz3 = thk3
tky3 = wid3

!real constants for off-axis pivots
r,4,area3,Izz3,Iyy3,tkz3,tky3,,,Ixx3,,,,,

wid1 = wid1+gap

!real constants for rigid beams
wid2 = .25
thk2 = .25 !thickness
area2 = thk2*wid2
Izz2 = wid2*thk2**3/12
```

```
Iyy2 = thk2*wid2**3/12
Ixx2 = Izz2+Iyy2
tkz2 = thk2
tky2 = wid2
theta2 = 0

et,2,beam4
r,2,area2,Izz2,Iyy2,tkz2,tky2,theta2,,Ixx2,,,,,

!section properties for split tube flexure
ri = .5 !.085 !inner radius of tube .5
t = .04 !wall thickness .04
ro = ri+t !outer radius
g = .0088*2*6 !1/2 gap, in radians (~1.5 degrees)

!section properties of split tube flexure
area = (ro**2-ri**2)*(pi-g)
a = (ro**2-ri**2)*(ro+ri)/4
Iyy = a/4*(2*sin(2*g)+4*pi-4*g)
Izz = a/4*(4*pi-4*g-2*sin(2*g))
b = (ro+ri)**2*(ro**2-ri**2)/8
Iyz = b/2*(sin(3*pi/2-g)**2-sin(-pi/2+g)**2)
Iw = 0  !warping constant
J = t**3*2*(pi-g)*(ro+ri)/6
CGy = 0 !y coordinate of centroid
CGz = 0 !z coordinate of centroid
SHy = 0 !y coordinate of shear center
SHz = 2*ri*((pi-g)*cos(g)+sin(g))/((pi-g)+sin(g)*cos(g))

platwid = 1. !half-width of platform 1.0

et,1,beam4

et,3,beam188
sectype,5,beam,asec,splittub
secdata,area,Iyy,Iyz,Izz,Iw,J,CGy,CGz,SHy,SHz

et,4,beam188
!define section for rigid supports. uncomment any beam section you wish to
!use and comment all the others.
!sectype,6,beam,T,rigidbeam !for T-beams (shows high stress)
!secdata,.25,.25,.04,.04 !flange width, overall depth, flange thk, stem thk

!sectype,6,beam,rect,rigidbeam !(solid square beam)
!secdata,.25,.25
```

```
!sectype,6,beam,chan,rigidbeam !channel beam
!flange len, flange len, depth, flange thk, flange thk, web thk
!secdata,.3,.3,.3,.05,.05,.05

sectype,6,beam,ctube,rigidbeam !hollow tube beam. seems to work very well
secdat,.085,.125 !Ri,Ro

keyopt,4,1,1 !do not assume warping is zero

mptemp
mptemp,1,0
mpdata,ex,1,,mod
mpdata,prxy,1,,nu

!material used in connecting sections
mptemp
mptemp,1,0
mpdata,ex,2,,mod
mpdata,prxy,2,,nu

mptemp
mptemp,1,0
mpdata,ex,3,,mod1
mpdata,prxy,3,,nu

nlgeom,1 !use non-linear solving

!define keypoints for platform joints
k,1,-cor,-cor,platwid+wid1
k,2,cor,cor,platwid+wid1
k,3,-cor,cor,platwid
k,4,cor,-cor,platwid

lstr,1,2 !line 1 define lines
lstr,3,4 !line 2

k,5,-cor,-cor,-platwid-wid1
k,6,cor,cor,-platwid-wid1
k,7,-cor,cor,-platwid
k,8,cor,-cor,-platwid

lstr,5,6 !line 3 define lines
lstr,7,8 !line 4
```

```
!define keypoints for ground joints
k,9,platwid,-cor,-cor !fix in all DOF
k,10,platwid,cor,cor
k,11,platwid+wid1,-cor, cor
k,12,platwid+wid1,cor,-cor

lstr,9,10 !line 5 define lines
lstr,11,12 !line 6

k,13,-platwid-wid1,-cor,-cor
k,14,-platwid-wid1,cor,cor
k,15,-platwid,-cor,cor
k,16,-platwid,cor,-cor
k,17,0,0,0
k,18,cor,cor,-cor
k,19,cor,cor,platwid*1.67 !central ground

!keypoints for driver dyad
k,20,-cor,-cor,platwid*2+wid1
k,21,cor,cor,platwid*2+wid1
k,22,-cor,cor,platwid*2
k,23,cor,-cor,platwid*2

k,24,(-platwid*2-(wid3+gap))*sin(arcf)-cor*cos(arcf),-cor,(platwid*2+(wid3+gap))*
cos(arcf)-cor*sin(arcf)
k,25,(-platwid*2-(wid3+gap))*sin(arcf)+cor*cos(arcf),cor,(platwid*2+(wid3+gap))*
cos(arcf)+cor*sin(arcf)
k,26,-platwid*2*sin(arcf)+cor*cos(arcf),-cor,platwid*2*cos(arcf)+cor*sin(arcf)
k,27,-platwid*2*sin(arcf)-cor*cos(arcf),cor,platwid*2*cos(arcf)-cor*sin(arcf)

!keypoint that defines link g
k,28,-platwid*2*sin(arcf)*cos(arcg),-platwid*2*sin(arcg),platwid*2*cos(arcf)*
cos(arcg)

k,29,-cor,cor,platwid*1.67

!define better platform
k,30,-cor,0,-platwid
k,31,-cor,0,platwid
k,32,0,-cor,0 !define a node not part of any line to use for orienting elements
 !node for orienting flexible members in f-g pivot
!k,33,platwid*2*sin(arcf),100000000,platwid*2*cos(arcf)

lstr,13,14 !line 7 define lines
```

```
lstr,15,16 !line 8

!connect grounds
lstr,9,12 !line 9
lstr,14,15 !line 10
larc,10,19,18,platwid !lstr,11,19 !line 11

!connect platform to ground
lstr,2,4 !line 12
lstr,7,30 !line 13
lstr,10,11 !line 14
lstr,13,16 !line 15
lstr,12,18 !line 16
lstr,16,18 !line 17

!connect sides of platform
lstr,1,31 !line 18
lstr,6,8 !line 19
lstr,31,17 !line 20
lstr,17,30 !line 21
lstr,2,18 !line 22
lstr,6,18 !line 23
larc,14,29,18,platwid !lstr,15,19 !line 24

!lines for driver dyad flexible members
lstr,20,21 !line 25
lstr,22,23 !line 26
lstr,24,25 !line 27
lstr,26,27 !line 28
lstr,28,17 !split tube flexure, line 29

!lines for driver dyad rigid members
lstr,19,21 !line 30
lstr,21,23 !line 31
lstr,20,22 !line 32
larc,22,25,18,3*platwid !line 33
lstr,25,26 !line 34
lstr,24,27 !line 35
lstr,24,28 !line 36
lstr,19,29 !line 37


lstr,5,30 !line 38
lstr,3,31 !line 39
```

```
/VIEW,1,1,2,3
/ANG,1
/REP,FAST
/REPLOT
LPLOT

esize,esz1,0

lsel,s,line,,1,4 !selects lines for flexible members that are z-axis pivots
lsel,a,line,,25,26
latt,1,3,1 !mat, real, type
lmesh,all
lsel,all
esize,esz1,0

lsel,s,line,,5,8 !selects lines for flexible members that are x-axis pivots
latt,1,1,1
lmesh,all
lsel,all

lsel,s,line,,27,28 !selects lines for flexible members at f-g pivot
latt,1,4,1
lmesh,all
lsel,all

lsel,s,line,,29,29 !selects line for split tube flexure
!secnum,5
!mat,1
!et,3
latt,3,,3,,,,5
lmesh,all
lsel,all

esize,esz2,0

lsel,s,line,,16,17 !select base members
lsel,a,line,,22,24
lsel,a,line,,11,11
lsel,a,line,,37,37
lsel,a,line,,30,30
lsel,a,line,,33,33
!the following line defines element orientation.
!MAT, REAL, TYPE, ESYS, KB, KE, SECNUM. KB, KE determine element orientation
latt,2,,3,,32,32,6
lmesh,all
```

```
lsel,all
lsel,s,line,,34,35 !selects lines for upright rigid members
lsel,a,line,,9,10
lsel,a,line,,12,15
lsel,a,line,,31,32
lsel,a,line,,18,19
lsel,a,line,,38,39

lsel,a,line,,20,21  !select other rigid segments
lsel,a,line,,36,36
!MAT, REAL, TYPE, ESYS, KB, KE, SECNUM. KB, KE determine element orientation
latt,1,,3,,,,6

lmesh,all
lsel,all

/ESHAPE,1
/EFACET,1
/RATIO,1,1,1
/CFORMAT,32,0
/REPLOT

ksel, all

!define a node where we can get reactions
ksel,s,kp,,9
nslk,s
*get,base,node,,num,max
nsel,all
ksel, all

!define a node to get output
ksel,s,kp,,17
nslk,s
*get,end,node,0,num,max
nsel,all
ksel, all

!Get input moment PSI
ksel,s,kp,,20
nslk,s
*get,PSI,node,0,num,max
nsel,all
ksel, all
```

```
!Get input moment FI
ksel,s,kp,,9
nslk,s
*get,FI,node,0,num,max
nsel,all
ksel, all

ksel,s,kp,,23
nslk,s
*get,fpiv,node,0,num,max !the f pivot (where we measure psi)
nsel,all
ksel, all

!define node to apply loads
ksel,s,kp,,17
nslk,s
*get,inpt,node,0,num,max

nsel, all
ksel, all

ksel,s,kp,,17
nslk,s
*get,cntr,node,0,num,max
nsel, all
ksel, all


!define loads and displacements
!ground points, simply supported

dk,11,ux,0,,,uy,uz,roty !the base node
dk,15,uy,0 !make the structure simply supported
dk,23,uy,0 !the f pivot

*do,j,1,31,1 !warping is restrained
dk,j,warp,0
*enddo

!Use this for applying a rotation
dk,20,rotz,psiang
dk,9,rotx,fiang

!Use this for applying a moment
```

```
!fk,20,mz,Rpsi !PSI input angle
!fk,9,mx,Rfi !FI input angle

allsel, all
nsubstp,20,35,15
lswrite,1

!can write additional substeps to analyze more than one loading condition in a
!single run.
!If you do this remember to change lssolve below to lssolve,first load step, last
! load step,increment

finish

/SOL
lssolve,1,1,1

FINISH
/POST1
!*

PLDISP,0

AVPRIN,0, ,

/SHOW,WIN32C
SET,FIRST
/PLOPTS,INFO,3
/CONTOUR,ALL,18
/PNUM,MAT,1
/NUMBER,1
/REPLOT,RESIZE
PLNSOL,S,EQV
/SHOW,WIN32
/REPLOT,RESIZE

finish
/post26
! Define variables
rforce,2,PSI,M,z,Mpsi !the input torque PSI
rforce,3,FI,M,x,Mfi !the input torque THETA
NSOL,4,end,U,Z,uzo

!end effector rotations
NSOL,5,end,rot,x,rxo
```

```
!NSOL,6,end,rot,y,ryo
NSOL,7,end,rot,z,rzo
*get,rxo,node,end,rot,x
*get,ryo,node,end,rot,y
*get,rzo,node,end,rot,z
*get,Rpsi,node,PSI,rot,z
*get,Rfi,node,FI,rot,x

!displacements of platform center
nsol,4,cntr,u,y, cntry
nsol,8,cntr,u,x, cntrx
nsol,6,cntr,u,z, cntrz

finish
/post26

PRVAR,Mpsi,Mfi,rxo,rzo,cntrx,cntry,cntrz
```

B.2 DLT Calibration

DLT stands for direct linear transform, and is the method we use to calculate positions in 3D space of points found on two images. The first step to DLT is building some calibration matrices. This file does that.

```
%takes a set of images and generates L and R matrices from 27 calibration
%points. Adapted from code by Scott Thomson. 6 March 2013

clear all

% Get calibration points
% Read in one image
figure(1)
num_pts = 29;
for i = 1:3;
%generate a file name
name = sprintf('calib_left_%d.png',i);
A=imread(name);
% Display image
image(A); axis image;
% Get pixel (image) coordinates of size calibration points (A through G)
% from one image.  ginput allows you to click on points in the image and
% automatically save the pixel coordinates as uL and vL.
[ul(:,i) vl(:,i)]=ginput(num_pts);

clear A
```

```
end

%build giant vectors containing calibration points
uL = [ul(:,1); ul(:,2); ul(:,3)];
vL = [vl(:,1); vl(:,2); vl(:,3)];

% % Do the same as above, but for other image.
 for i = 1:3;
 name = sprintf('calib_right_%d.png',i);
 A=imread(name);
 image(A); axis image;
 [ur(:,i) vr(:,i)]=ginput(num_pts);
 clear A
 end

uR = [ur(:,1); ur(:,2); ur(:,3)];
vR = [vr(:,1); vr(:,2); vr(:,3)];

%Physical coordinates of the points. (change these to fit your actual problem
x = [19.05 12.7  12.7  12.7  12.7  12.7  6.35   6.35   6.35   6.35   6.35   0   ...
    0   0   0   0   0   0   ...
    -6.35  -6.35  -6.35  -6.35  -6.35  -12.7  -12.7  -12.7  -12.7  -12.7  -19.05];
y = [0  12.7  6.35   0   -6.35  -12.7  12.7  6.35   0   -6.35  -12.7    19.05  ...
    12.7  6.35   0   -6.35  -12.7 -19.05 ...
    12.7  6.35   0   -6.35  -12.7  12.7  6.35   0   -6.35  -12.7 0];
x = [x, x, x];
y = [y, y, y];
z=[-5*ones(1,num_pts), zeros(1,num_pts), 5*ones(1,num_pts)];


% Construct A matrix for left and right views, u component
N=length(x);
for i=1:2:N*2
    k=(i+1)/2;
    AL(i,:)=[x(k) y(k) z(k) 1 0 0 0 0 -uL(k)*x(k) -uL(k)*y(k) -uL(k)*z(k)];
    AR(i,:)=[x(k) y(k) z(k) 1 0 0 0 0 -uR(k)*x(k) -uR(k)*y(k) -uR(k)*z(k)];
    BL(i,1)=[uL(k)];
    BR(i,1)=[uR(k)];
end
% Construct A matrix for left and right views, v component
for i=2:2:N*2
    k=i/2;
    AL(i,:)=[0 0 0 0 x(k) y(k) z(k) 1 -vL(k)*x(k) -vL(k)*y(k) -vL(k)*z(k)];
    AR(i,:)=[0 0 0 0 x(k) y(k) z(k) 1 -vR(k)*x(k) -vR(k)*y(k) -vR(k)*z(k)];
    BL(i,1)=[vL(k)];
    BR(i,1)=[vR(k)];
end
% Calculate L and R calibration matrices
L=(inv(AL'*AL))*(AL'*BL)
R=(inv(AR'*AR))*(AR'*BR)

% This next section compares the calculated point locations using L and R
% with the known point locations.
for i=1:N
    xL=uL(i);
    yL=vL(i);
    xR=uR(i);
    yR=vR(i);
    A=[L(1)-L(9)*xL L(2)-L(10)*xL L(3)-L(11)*xL;...
        L(5)-L(9)*yL L(6)-L(10)*yL L(7)-L(11)*yL;...
        R(1)-R(9)*xR R(2)-R(10)*xR R(3)-R(11)*xR;...
        R(5)-R(9)*yR R(6)-R(10)*yR R(7)-R(11)*yR];
    B=[xL-L(4);yL-L(8);xR-R(4);yR-R(8)];
    X(:,i)=inv(A'*A)*A'*B;
end
XKnown=[x;y;z];
for i=1:size(X,2)
    err(i) = sqrt((XKnown(1,i)-X(1,i))^2 + (XKnown(2,i) - X(2,i))^2 + ...
        (XKnown(3,i) - X(3,i))^2);
end

% Plot the known point locations with the point locations calculated using
% the DLT method.  The red askerisks are the calculated positions, and the
% black dots are the known positions.
figure(2)
plot3(X(1,:),X(2,:),X(3,:),'r*');
hold on
plot3(XKnown(1,:),XKnown(2,:),XKnown(3,:),'k.');
hold off
xlabel(['X']); ylabel(['Y']); zlabel(['Z']);
grid on

error = mean(err)/0.5*100

fid = fopen('calib.txt','w');
fprintf(fid,'%6.6f\n',L);
fprintf(fid,'%6.6f\n',R);
fclose(fid);
```

B.3 Points Finder

After you have a calibration file, you can run this file to find the locations of your reference points. This file also calculates and outputs the orientation of the pointer platform.

```
%This file loads L and R matrices from the file calib.txt and uses them to
%find point locations in loaded photos. Adapted from code by Scott Thomson.
%6 March 2013
clc
clear
base = 2; %1 for yes or anything else for no
%Load L and R from file
fid = fopen('calib.txt','r');
CAL = fscanf(fid,'%f');
fclose(fid);
N = length(CAL);
L = zeros(N/2,1);
R = zeros(N/2,1);
%split the vector CAL into L and R
for i = 1:(N/2)
L(i) = CAL(i);
R(i) = CAL(i+N/2);
end
angles = zeros(16,3);
N = 3; %the number of points to find
for data_set=1:16
angles(data_set,1)=data_set;

% name = sprintf('L_data_%d.png',data_set);
name = sprintf('L%d.png',data_set);
A=imread(name);
% A=imread('L_base.png');
% Display image
image(A); axis image;
% Choose click three points to find position
[uL vL]=ginput(N);
clear A
% Do the same as above, but for other image.
% name = sprintf('R_data_%d.png',data_set);
name = sprintf('R%d.png',data_set);
A=imread(name);
% A=imread('R_base.png');
image(A); axis image;
[uR vR]=ginput(N);

% Calculate coordinates of point using L and R calibration matrices
```

```
for i=1:N
    xL=uL(i);
    yL=vL(i);
    xR=uR(i);
    yR=vR(i);
    A=[L(1)-L(9)*xL L(2)-L(10)*xL L(3)-L(11)*xL;...
        L(5)-L(9)*yL L(6)-L(10)*yL L(7)-L(11)*yL;...
        R(1)-R(9)*xR R(2)-R(10)*xR R(3)-R(11)*xR;...
        R(5)-R(9)*yR R(6)-R(10)*yR R(7)-R(11)*yR];
    B=[xL-L(4);yL-L(8);xR-R(4);yR-R(8)];
    X(:,i)=inv(A'*A)*A'*B;
end
% Print x, y, z coordinates of point to matlab command window
%If this data is from the unrotated state, write a file containing the
%vector normal. If this file is some other state, get the rotation of
%the original position
if base == 1
    %to get a vector normal, find vectors from the origin to p01 and p02, then
    %take the cross product of the two vectors.
    vec_norm = cross(X(:,2)-X(:,1), X(:,3)-X(:,1));
    %normalize to unit vector
    vec_norm = vec_norm./norm(vec_norm,2);
    phi = atan2(vec_norm(1),vec_norm(3))
    psi = atan2(-vec_norm(2),vec_norm(3))
    ang0 = [phi, psi]; %initial phi and psi
    fid = fopen('base.txt','w+');
    fprintf(fid,'%6.6f\n',ang0);
    fclose(fid);
else
    fid = fopen('base.txt','r');
    ang_0 = fscanf(fid,'%f')';
    fclose(fid);
    vec_norm = cross(X(:,2)-X(:,1), X(:,3)-X(:,1));
    %normalize to unit vector
    vec_norm = vec_norm./norm(vec_norm,2)
    %find the rotation angles from vec_norm_0 to vec_norm
    angles(data_set,2) = atan2(vec_norm(1),vec_norm(3))-ang_0(1);
    angles(data_set,3) = atan2(-vec_norm(2),vec_norm(3))-ang_0(2);
end
end
angles
X
```