

2011

Constraint-based synthesis of shape-morphing compliant structures in virtual reality

Denis Dorozhkin
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Dorozhkin, Denis, "Constraint-based synthesis of shape-morphing compliant structures in virtual reality" (2011). *Graduate Theses and Dissertations*. 12105.

<https://lib.dr.iastate.edu/etd/12105>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**Constraint-based synthesis of shape-morphing compliant structures
in virtual reality**

by

Denis Vitalievich Dorozhkin

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Program of Study Committee:
Judy M. Vance, Major Professor
Chris Harding
Adin Mann
Thomas J. Rudolphi
Gloria K. Starns

Iowa State University

Ames, Iowa

2011

Copyright © Denis Vitalievich Dorozhkin, 2011. All rights reserved

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	v
ACKNOWLEDGEMENTS	vii
CHAPTER 1. INTRODUCTION	1
1.1 Overview	1
1.2 Scope	2
1.3 Impact and motivation	4
CHAPTER 2. BACKGROUND	6
2.1 Virtual Reality	6
2.1.1 VR design environments	6
2.1.2 Applications of VR in mechanism design	7
2.2 Synthesis of compliant mechanisms	10
2.2.1 Pseudo-Rigid Body Model	11
2.2.2 Topological Synthesis	11
2.2.3 Constraint-Based Design Method	14
2.2.4 CBDM tools	17
CHAPTER 3. METHODOLOGY	19
3.1 Method overview	19
3.2 Initial anchor selection	23
3.2.1 CBDM anchor selection	23
3.2.2 Kinematic feasibility of the initial anchor placement methods	25
3.3 Optimization problem	26
3.4 Shape segmentation and kinematic analysis	28
3.4.1 Basic kinematic analysis	29
3.4.2 Solution filtering	33
3.4.3 Optimization details	37
3.5 Flexure analysis	39
3.5.1 Linear elastic deformation analysis	39
3.5.2 FEA solver setup	40
3.5.3 Optimization details	43
CHAPTER 4. IMPLEMENTATION	45
4.1 VR design environment	45
4.2 Design problem solution sequence	47
4.2.1 Problem definition	47
4.2.2 Kinematic analysis	49

4.2.3 Finite element analysis	49
4.3 Functionality details	50
4.3.1 Haptically-assisted menu system	50
4.3.2 Source/target curve specification	52
4.3.3 Force feedback	54
4.3.4 Optimization functionality	58
CHAPTER 5. TEST CASES	62
5.1 Sample design problems	62
5.1.1 Simple curve	62
5.1.2 Concave-convex-concave curve	64
5.1.3 Compliant lumbar support	65
5.2 Detailed analysis of the compliant lumbar support	68
CHAPTER 6. CONCLUSIONS AND FUTURE WORK	72
BIBLIOGRAPHY	74
APPENDIX A. HDAPI FUNCTIONALITY FLOWCHART	80
APPENDIX B. INITIAL ANCHOR PLACEMENT STUDY RESULTS	81
APPENDIX C. COMPLIANT LUMBAR SUPPORT SOLUTIONS	88

LIST OF TABLES

Table 2.1. Compliant mechanism design methods	10
Table 3.1. Comparison of large and small deflection results	40
Table 4.1. PHANTOM Omni specifications	55
Table 4.2. Application state synchronization variables	58
Table 4.3. Comparison of optimization algorithm termination times	60
Table 5.1. Simple convex curve control points	62
Table 5.2. Concave-convex-concave curve control points	64
Table 5.3. Compliant lumbar support control points	66
Table 5.4. Solutions to the compliant lumbar support design problem	69

LIST OF FIGURES

Figure 2.1. Isis and VRSpatial	8
Figure 2.2. Flexible element and the pseudo-rigid body analog	11
Figure 2.3. Typical topological synthesis procedure	12
Figure 2.4. Typical disconnected structure results	13
Figure 2.5. TS-generated compliant lumbar support	14
Figure 2.6. 2D constraint cases	15
Figure 2.7. Constraint ambiguity	16
Figure 2.8. CoMeT interface	17
Figure 2.9. CoMeT design flow	18
Figure 3.1. Segmentation of the source and target curves	19
Figure 3.2. Initial curve and constraint members	20
Figure 3.3. Deformation of the rigid four-bar chain	20
Figure 3.4. Deformation of the flexible model	21
Figure 3.5. Design sequence overview	22
Figure 3.6. CBDM-based constraint regions	24
Figure 3.7. CBDM-based initial anchor placement	25
Figure 3.8. Single cell in the segmented deformable structure	29
Figure 3.9. Position vector loop for a fourbar linkage	31
Figure 3.10. The $0^\circ \leq \mu < 180^\circ$ branch of a fourbar cell	35
Figure 3.11. Defect in the fourbar cell chain	36
Figure 3.12. LSE computation for a single segmented element	38
Figure 3.13. 3D beam element	41
Figure 4.1. Design framework in different VR setups	45
Figure 4.2. Compliant structure in its initial and deformed states	46
Figure 4.3. Main design environment	47
Figure 4.4. Source and target curve specification	48
Figure 4.5. Chain of the fourbar linkages in the deflected configuration	49
Figure 4.6. Flexed configuration of the shape-morphing structure	50

Figure 4.7. PHANTOM Omni	55
Figure 5.1. Simple convex curve target profile	62
Figure 5.2. Simple convex curve problem solution	63
Figure 5.3. Concave-convex-concave curve target profile	64
Figure 5.4. Concave-convex-concave curve problem solution	65
Figure 5.5. Compliant lumbar support	66
Figure 5.6. Compliant lumbar support target profile	67
Figure 5.7. Compliant lumbar support solution	67
Figure 5.8. LSE values	69
Figure 5.9. Computation time	70
Figure 5.10. Actuation force	70

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my academic advisor, Dr. Judy Vance, for her continuous encouragement and guidance throughout my academic pursuits. The many opportunities she has given me during my stay at Iowa State University are greatly appreciated.

I would also like to sincerely thank Dr. Chris Harding, Dr. Adin Mann, Dr. Thomas Rudolphi, and Dr. Gloria Starns for being able to take on duties of my POS committee members.

Many thanks go out to all the VRAC students and personnel, especially those I got to work closely with, for the helpful advice and friendship they have given me.

Last, but by no means least, I wish to thank my parents, Vitaly and Elena Dorozhkin, my sister, Masha Dorozhkina, and Dan and Marilyn Kammiller and their family. Without their unconditional love and support this dissertation would not have been possible.

CHAPTER 1. INTRODUCTION

1.1 Overview

The purpose of this research is to establish a novel approach to the design of compliant shape-morphing structures using constraint-based design methods and virtual reality (VR). Compliant mechanisms, as opposed to rigid link mechanisms, achieve motion guidance via the compliance and deformation of the mechanism's members. They are currently being explored as structural components to produce shape changes in products such as aircraft wing and antenna reflectors. The goal is to design a single-piece flexible structure capable of morphing a given curve or profile into a target curve or profile while utilizing the minimum number of actuators..

The successful design of compliant mechanisms requires an understanding of solid mechanics (deformation, stress, strain, etc.) and mechanism kinematics (properties of motion). As a result, only a fairly narrow, experienced group of engineers are successful in designing these mechanisms. The two primary methods prevalent in the design community at this time are the pseudo-rigid body method (PRBM) and topological synthesis. Unfortunately each of these methods has its own limitations. The research presented here takes a different approach by examining the use of the constraint-based design method (CBDM) to solve shape-morphing problems.

The concept of CBDM has generally been confined to the Precision Engineering community and is based on the fundamental premise that all motions of a rigid body are determined by the position and orientation of the constraints (constraint topology) which are placed upon the body. Constraint-based compliant mechanism design theory is a

powerful and rational design process where a desired motion path is first described, then decomposed into combinations of lines, arcs, and rotations which can be achieved through combining a series of compliant mechanism components and sub-components. Any mechanism motion path may then be defined by the proper combination of constraints. In order to apply the CBDM concepts to the design and analysis of shape-morphing compliant structures we propose a tiered design method that relies on kinematics, finite element analysis, and optimization.

The proposed approach consists of two major steps: kinematic modeling and flexible body deformation synthesis. First, the initial and target shape are defined. By segmenting the flexible element that comprises the active shape surface at multiple points in both the initial and the target configurations and treating the resulting individual segments as rigid bodies that undergo a planar or general spatial displacement we are able to apply traditional kinematic theory to rapidly generate sets of potential solutions. Once a feasible design space is identified, the final design is determined via an FEA-augmented optimization sequence. Coupled with an immersive VR interface and a force-feedback input device this approach provides the ability to quickly specify and evaluate multiple potential design problems in order to arrive at the desired solution.

1.2 Scope

The concept of CBDM has generally been confined to the Precision Engineering community and learned via apprenticeship. This method has been developed to design compliant mechanisms, which form the foundation of many precision instruments,

compliant manipulators and consumer products. Although this method has been published in the literature [1, 2] these publications and their application to compliant mechanism design are not well known outside the Precision Engineering community. In addition, proficiency in using constraint-based methods for designing compliant mechanisms requires (1) commitment to a steep learning curve (hence the reason for apprenticeship) and (2) “hands-on” experience to understand the stiffness characteristics of alternate designs.

In the research presented here, a generalized constraint-based concept design process and the supporting optimization engineering decision making tools required for concept selection have been created. These components have been integrated with VR so as to provide an experience which reduces the need for apprentice-based learning. This is particularly important in fields of application in which it is difficult to obtain hands-on experience/intuition. For instance, micro-scale and nano-scale compliant mechanisms are often difficult to design due to the difficulty in (1) obtaining a “feel” for how these devices operate and (2) visualizing how these devices function.

A decade of research into using VR as an engineering design tool has resulted in an understanding of the characteristics of VR that can be used to improve engineering design. Stereo viewing, position tracking and haptic force feedback provide a computer interface that allows participants to move and interact with digital objects as if they were real three-dimensional objects. The interface is particularly useful when designing objects that require three-dimensional specification of the design objective or three-dimensional evaluation of the shape or motion of the resultant design. In the work presented here, an

immersive VR environment has been created to provide the 3D working space required to view, assemble components and interact with CBDM concepts. The interface is augmented with the constraint-based theory and simulation tools discussed in the preceding paragraph.

1.3 Impact and motivation

Development of robust methods for designing shape-morphing structures is the focus of several current research projects, both in the academic and the military communities. Geometric shapes of the individual system components, such as aircraft wings and antenna reflectors, directly affect the performance of the corresponding mechanical systems [3]. Of particular interest is the utilization of compliant mechanisms to achieve the desired adaptive shape change characteristics. Compliant mechanisms, as opposed to the traditional rigid link mechanisms, achieve motion guidance via the compliance and deformation of the mechanism's members. The goal is to design a single-piece flexible structure capable of morphing a given curve or profile into a target curve or profile while utilizing the minimum number of actuators (ideally, just one) [4].

The combination of CBDM methods and VR provide a working/learning/design space that supports the design of compliant mechanisms. This design environment provides designers with (1) a new perspective on how to perform synthesis and analysis of compliant mechanisms, (2) a generalized, well-disseminated design theory of mechanism design, (3) a means to rapidly master design for compliance/compliant mechanisms in fields which are difficult to build competence via hands-on experience,

and (4) a fully immersive, collaborative, interactive design environment. This has the potential to bring the field of compliant mechanism design to a broader audience which will be capable of better understanding how/why compliant mechanisms work, how to synthesize them, how to characterize them with general design metrics and how to best fabricate/integrate them into practical applications.

CHAPTER 2. BACKGROUND

2.1 Virtual Reality

VR refers to computer-generated three-dimensional (3-D) environments, which can be interactively experienced and manipulated by the participants [5]. Stuart [6] defines a virtual environment system as a human-computer interface capable of providing “interactive immersive multi-sensory 3-D synthetic environments.” In such systems the user’s motions are tracked with position sensors and used to update the visual and auditory displays in real-time. This creates the illusion for the participants of being inside of the environment [6]. In addition to providing the ability to explore a design problem in three-dimensional space, VR environments often allow users to manipulate the objects in the environment in an intuitive way using a variety of instrumented gloves and wands.

2.1.1 VR design environments

The scientific and engineering communities have embraced VR as a valuable tool because it offers a unique way to investigate data. Benefits of the VR systems are especially evident in the area of engineering product development, where these systems are used throughout the whole range of the product development cycle: from modeling and evaluation of the first prototypes, to providing training opportunities for end-product users ([7], [8], [9]).

The potential of using VR technology as an interface to design of mechanical systems has been extensively investigated at Iowa State University. Application areas include assembly methods prototyping, factory simulation, shape optimization, and finite element analysis as well as mechanism design [10, 11].

2.1.2 Applications of VR in mechanism design

There are multiple benefits to using VR even in the design of conventional non-compliant mechanisms. The design of planar mechanisms is limited to two-dimensional space, so the traditional human-computer interfaces (HCI) of a monitor, a keyboard and a mouse are well suited for the task of the design problem parameter definition. However, operation of spatial mechanisms is associated with general 3-D space, and usage of a traditional HCI, even well designed, imposes artificial constraints on the ability of the mechanism designers to correctly and efficiently specify the design problem and investigate the spatial mechanism synthesis results.

VR provides a truly three-dimensional alternative to the traditional computer interface. Replacing the mouse and the monitor with a position tracked stereo visual display and a position tracked input device, VR allows the users to interact with the design problem by moving around and performing actions in 3-D space. The potential of using VR technology in the design of spatial mechanisms was first recognized in 1995 by Vance and Osborn [12], when the SphereVR program was created for analysis and synthesis of spherical 4R linkages. It required users to place coordinate frames on graphical representation of a sphere in the VR environment. The Newton-Raphson

iterative approach was used to solve the non-linear equations, which resulted from Suh and Radcliff's displacement matrix mechanism synthesis method.

Investigation of VR as a medium for spherical mechanism synthesis continued in 1996 with the creation of VEMECS (Virtual Reality for MEchanism Synthesis) [13]. VEMECS relied on Sphinx algorithms for its mechanism analysis and synthesis functionality and essentially became a VR interface to the Sphinx program. Following evaluation of the effectiveness of a VR interface compared to the traditional HCI methods [14], in 1999 Furlong et al. developed Isis, as the next generation spherical mechanism design tool [15]. Isis introduced the 'design in context' approach to the design problem definition, where digital models of the design part and of the work environment could be imported into the application and manipulated by the users instead of the conventional abstract coordinate frames (Figure 2.1). A real world design task was investigated and the resulting mechanism built by the designers.

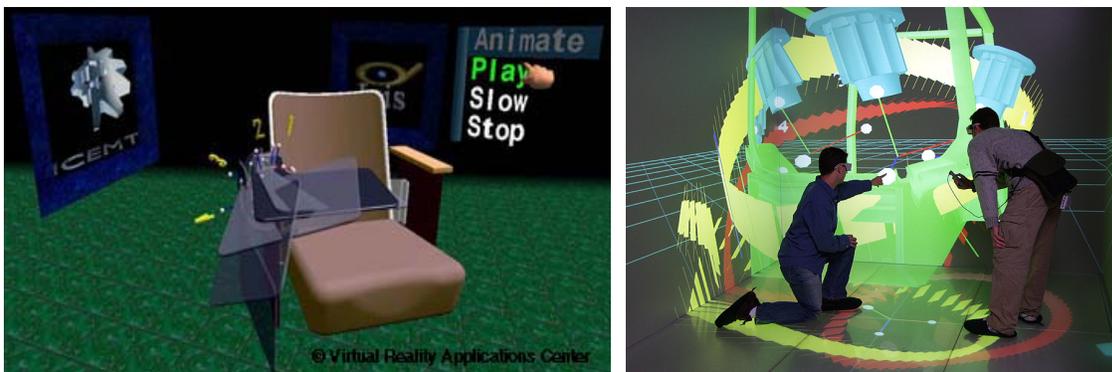


Figure 2.1. Isis and VRSpatial [15]

In 2001 the spectrum of VR-based mechanism design applications was expanded to include analysis and synthesis of spatial 4C mechanisms, with the creation of the

VRNETS program by Kihonge and Vance [16]. Computation routines from SPADES, a PC-based program for design of spatial 4C mechanisms for spatial rigid-body guidance tasks [17], were used to provide the mechanism synthesis functionality of VRNETS. The program allowed users to investigate the design parameters associated with spatial 4C mechanisms, such as the input design positions and the congruence planes, in a 3-D environment. Additionally, it provided the option of networking several instances of the application in order to facilitate a collaborative design process. Operation and functionality of VRNETS has been explored by several mechanism designers. They discovered that while the program proved to be an effective tool in the synthesis and analysis of spatial 4C mechanisms, improvements and modifications to the program's structure and functionality were needed in order to take full advantage of the VR design environment. The suggested changes were focused on improving the user interface, expanding the design problem specification functionality, providing higher degree of flexibility while working with the application, and improving solution evaluation methods. In 2002, the development of the VRSpatial application [18] relied on the experience gained from operation of the VRNETS program, while offering its users an assortment of new and improved features. The range of VR systems capable of running the mechanism design program has been extended to include practically all of the modern VR hardware and software configurations. Methods of specifying the initial design problem have been improved and multiple options were made available for investigating the generated solution space, providing for more effective design. Furthermore, the level of interactivity within the application has been increased through the implementation of a speech recognition interface.

2.2 Synthesis of compliant mechanisms

Compliant mechanism design is performed via one of the three methods outlined in Table 2.1:

Table 2.1. Compliant mechanism design methods

Pseudo-rigid body (lumped compliance)	Continuum topology (distributed compliance)	CBDM (modular kinematic)
<ul style="list-style-type: none"> -Combination of rigid and compliant elements -Compliant joints connect rigid elements to form kinematic chains 	<ul style="list-style-type: none"> -Combination of elements with distributed compliance -Continuum-topology generation based on envelope and inputs-outputs 	<ul style="list-style-type: none"> -Motion driven by constraint topology of mechanism -Concepts generated by combining modular flexures which provide desired constraint/freedom

The Pseudo-rigid Body Model [19] and Topological Synthesis method [3] have been widely used in the kinematics and mechanism communities dating back to as early as the 1980s. The foundations of the constraint-based method were laid out by Maxwell [20] in the 1880s during his quest to design compliant instruments and elastic mechanisms to support his physics research experiments. The method has been developed and continues to be advanced to meet modern challenges via research at several MIT Precision Engineering Labs. The method is attractive because it is based upon motion visualization and is therefore well-suited to conceptual development [2]. Well-known shape and optimization methods may be used to refine concepts after the initial concept generation phase.

2.2.1 Pseudo-Rigid Body Model

The pseudo-rigid body approach models the deflection of flexible members using rigid-body components that have equivalent force-deflection characteristics (Figure 2.2). [19, 21]. The rigid analog of the compliant structure is then analyzed using traditional mechanism design methods and the principle of virtual work to ascertain its kinematic and elastomechanic properties. The primary aim of PRBM is to model rather than synthesize and so it is not ideally suited to generate many different concepts. Pseudo-rigid body modeling (PRBM) is utilized as an alternative to rigorous large-deflection analysis methods in order to provide a more efficient method to arrive at and improve these initial designs.

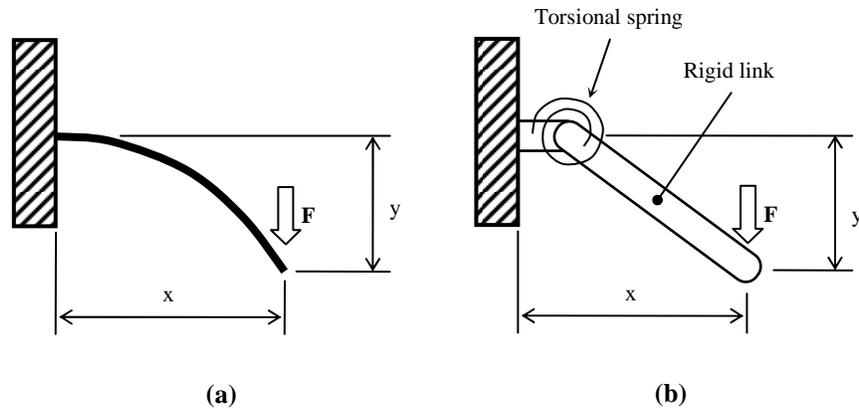


Figure 2.2. Flexible element (a) and the pseudo-rigid body analog (b)

2.2.2 Topological Synthesis

Topological synthesis is a concept synthesis method that is based upon computer algorithms that begin with a starting shape for a compliant mechanism and then determine how to add/subtract material in order to create concepts that satisfy

performance specifications [22, 23]. This approach is highly effective for the rapid synthesis of unique, non-precision compliant mechanism concepts in applications such as robotics, MEMS and aeronautics/airfoils. Unfortunately, topology synthesis cannot be easily used to solve most precision flexure design problems.

The synthesis of shape morphing compliant mechanism is different from the typical single output design problems. This is due to the multiple output points along the morphing boundary. Lu and Kota have developed a genetic algorithm (GA)-based synthesis approach, incorporating a binary ground structure parameterization, to systematically design shape morphing compliant mechanisms [24]. Figure 2.3 represents a typical procedure using this approach.

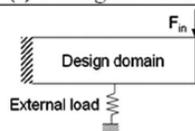
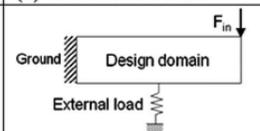
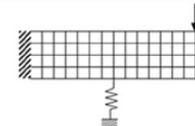
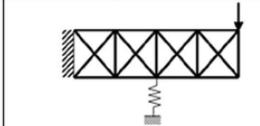
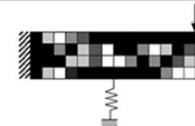
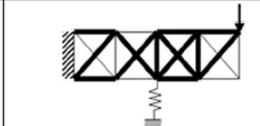
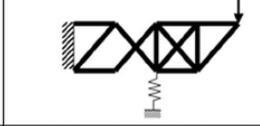
Typical Synthesis Procedure	(a) Homogenization	(b) Ground Structure
Step 1: Problem Specification - Define design domain - Apply boundary conditions		
Step 2: Design Domain Parameterization - Discretize design domain - Define design variables		
Step 3: Topology Optimization - Define objective function - Choose and implement optimization method		
Step 4: Final Design Interpretation - Filter out elements with values under certain threshold - Interpret final topology		

Figure 2.3. Typical topological synthesis procedure (by permission of ASME) [24]

This approach, however, does not always result in a valid solution for every problem. Because of the topology optimization, the result is highly dependent on the

initial mesh configuration and the method sometimes produces disconnected structures (Figure 2.4).

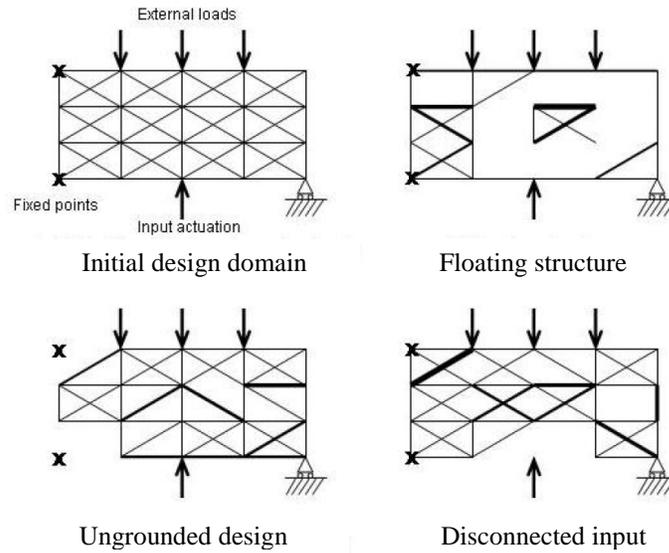


Figure 2.4. Typical disconnected structure results [25]

In subsequent work topological synthesis has been augmented with ‘load path representation, which is used to overcome the issues encountered using the binary ground structure parameterization [3]. At the foundation of the load path approach lays a design domain parameterization method that utilizes the load path of a structure. The topological connection of the method generates three types of paths: from input to outputs, from input to fixed points, and from fixed points to output points. However, the attainable topology connectivity is limited by direct connection between the points. A set of grid points are used as the intermediate ‘connection ports’ to allow additional connections between paths and to increase the variety of available topologies [3].

Utilizing the intermediate grid points the GA is capable to efficiently detect the invalid designs and exclude them from the solution space with design variable data

structure, using the path information. The load path approach offers several advantages over previous methods, such as (a) eliminating the need of an initial ground structure, (b) ensuring structural connectivity, and (c) yielding solutions that generate desired shape change efficiently. However, the designers have little control over the resulting solutions, often ending up with overly-complex topologies (Figure 2.5)

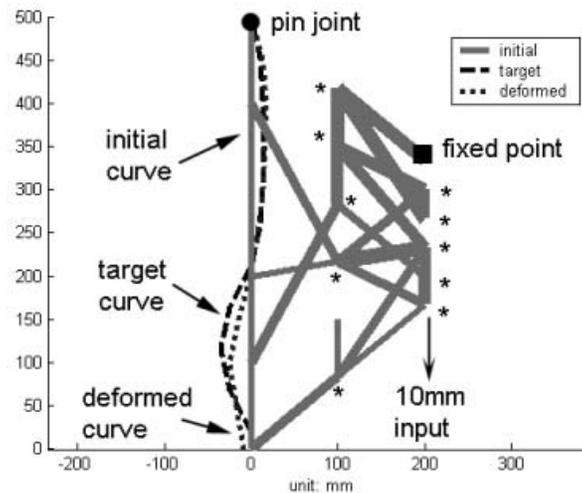


Figure 2.5. TS-generated compliant lumbar support [25]

2.2.3 Constraint-Based Design Method

The fundamental premise of the constraint-based method is that all motions of a rigid body are determined by the position and orientation of the constraint elements which are placed upon the body. An ideal constraint is defined as a member that has zero compliance in one direction and compliance in two directions. Any mechanism motion path can be defined by the proper combination of constraints and non-constraints. An unconstrained 3D rigid object has 6 degrees of freedom (DOF). Proper application of non-redundant constraint elements eliminates a DOF in a one-to-one fashion. Figure 2.6 depicts a circular object constrained by two constraint members. In one configuration, the

allowable motion of the circular object is pure translation. The other configuration results in single axis rotation.

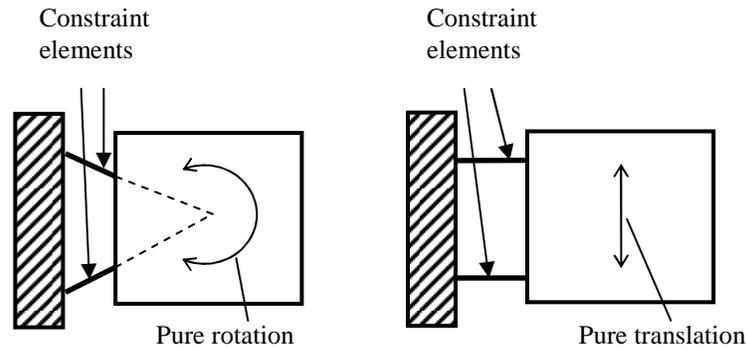


Figure 2.6. 2D constraint cases a) single axis rotation b) pure translation

Maxwell applied the concepts of constraint member to compliant mechanisms, Hooke's Law of elasticity [26], beam flexure theory [27], and Maxwell's own principle of reciprocity. Post Maxwell, physicists and precision engineers used his method in combination with instant centers (2D screws) to visualize and generate individual mechanisms and modular mechanisms. Through the work of Blanding [1] and Hale [2] the early theory of constraint-based method was codified and published.

The six steps in the design method are as follows:

1. Determine design requirements: motion path, stiffness, load capacity, etc.
2. Perform motion path decomposition: arcs, lines, rotation points, sub-paths.
3. Define constraint topology definition: high and low stiffness directions.

4. Generate concepts: mechanisms that satisfy constraint topology and decomposed motion paths.
5. Perform simulation and concept selection: operational range, stiffness characteristics, manufacturability, etc.
6. Perform size and shape optimization: stiffness, load capacity, efficiency, etc.

One of the design difficulties associated with CBDM is the ambiguity of the constraint topology. Consider the combinations of constraints in Figure 2.7. The design objective was to apply constraints to a rigid body to restrict its motion to one degree of freedom pure translation. Two design solutions are illustrated. Each solution is distinctly different yet produces the same motion: a single translational degree of freedom. This problem becomes especially prominent in general topology cases, where the constraints are no longer orthogonal.

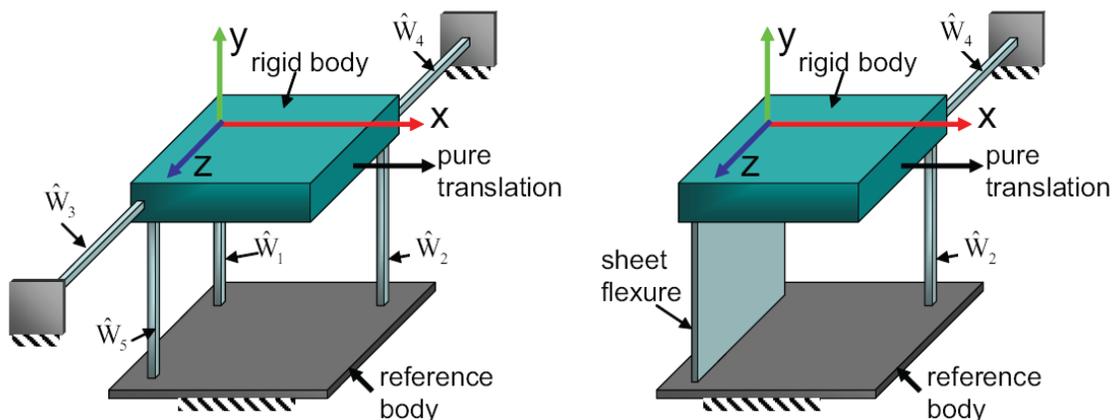


Figure 2.7. Constraint ambiguity [28]

2.2.4 CBDM tools

One of the recent developments in the constraint-based method is a computer-based synthesis tool, which enables a designer to quickly sketch concept designs on a computer and perform rapid simulation and optimization, named CoMeT [29]. It can be used to perform synthesis of 2D and 3D compliant mechanisms via a 2D computer screen (Figure 2.8).

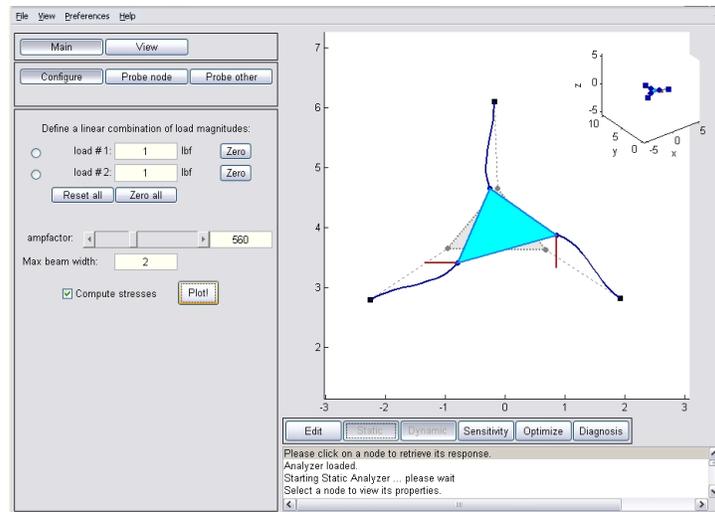


Figure 2.8. CoMeT interface

CoMeT connects the Graphical-User Interface (GUI) with MATLAB computation routines. The results of the mechanism analysis, such as display of the flexed mechanism shape and numerical data which quantify displacement, stress, stiffness and screw axis location/orientation, are provided to the user [30]. This data is presented in numerical form (e.g. as in FEA) and in matrix form (provides direct actuation/motion equations). The CoMeT analysis procedure is shown in Figure 2.9.

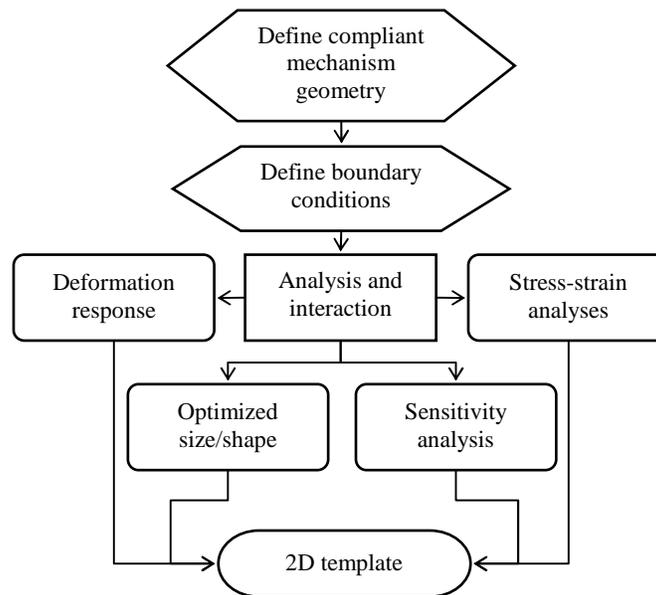


Figure 2.9. CoMeT design flow [31]

The CoMeT program can be used to synthesize compliant mechanisms that move in 2D and 3D. The design of 2D compliant mechanisms is easily accommodated with the traditional human-computer interfaces (HCI) of a monitor, a keyboard and a mouse, or, in case of a Tablet PC, with a touch screen and a stylus. CoMeT relies on linear elastic deformation analysis that, while less accurate than a rigorous large deformation FE analysis, is sufficient to rapidly narrow down a list of possible design concepts.

CoMeT has been created to aid in the design of motion controlling mechanisms. These types of mechanisms are designed to move an external object in a desired motion. This design problem is different than the problem of shape morphing where the design problem is to achieve a desired shape. The rest of this thesis outlines a design methodology to synthesize compliant mechanisms to achieve shape morphing of a given structure.

CHAPTER 3. METHODOLOGY

The research presented here expands the scope of CBDM to the design of shape-morphing structures. The goal is to identify the number and topology of the constraints that will produce the desired shape. The method consists of two distinct steps: modeling the entire desired shape by a series of rigid four-bar linkages to identify candidate constraint anchor point regions, then refining the structure by analyzing the deformable members to identify the best location of the constraint anchor points. The suitability of fit of the final design shape is determined by a least squares error between the target shape and the achieved shape.

3.1 Method overview

The method begins by dividing the source (initial) shape into a number of discrete segments. The endpoints of these segments are also located and identified on the target shape curve (Fig. 3.1).

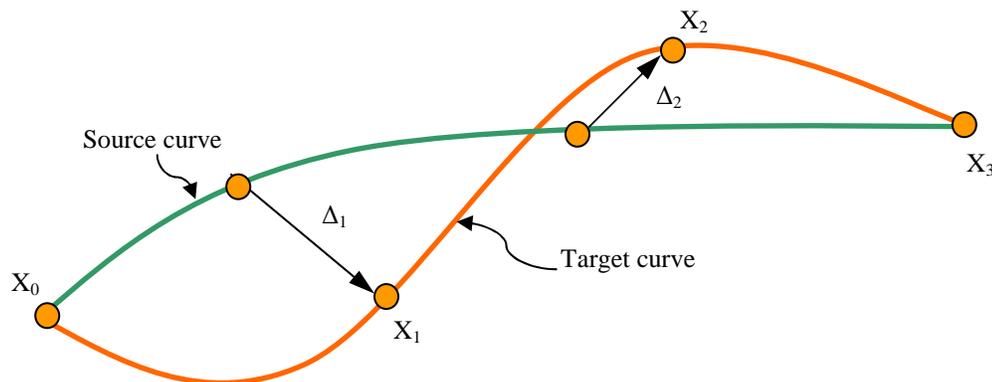


Figure 3.1. Segmentation of the source and target curves

A constraint member is created for each segment point (X_0 , X_1 , X_2 , and X_3). One end of a constraint member is attached to the segment point and the other end of the constraint member is attached to the ground. The ground attachment is called the anchor point. The design goal is to locate the anchor points such that when the mechanism is actuated by a force, the surface bends into the target shape.

The combination of constraint members and curve segments is modeled as a series of four-bar mechanisms to fit the source curve (Fig 3.2). Traditional planar kinematics is used to determine the configuration of each four-bar in the chain for a given input angle θ_0 (Fig. 3.3):

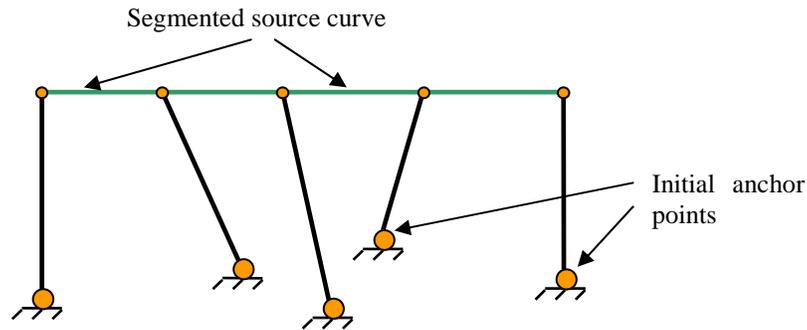


Figure 3.2. Initial curve and constraint members

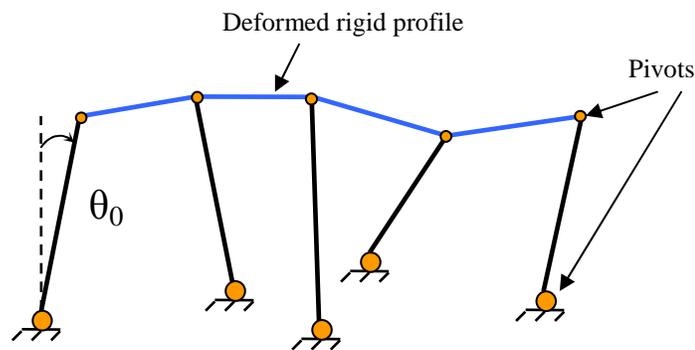


Figure 3.3. Deformation of the rigid four-bar chain

The next step is to optimize the structure to obtain the locations for each anchor point. The objective function follows the method proposed by Kota and Lu [4], which

minimizes the difference between the target and the achieved profiles of the active surface based on the Least-Square Error method. The results of the optimization are a set of potential locations of the anchor points based on the rigid four-bar linkage analysis.

In the next step, the rigid body approximation is replaced with a flexible body model (Figure 3.4). The initial locations of the anchor points and the segment points are retained from the kinematic optimization. A small region around each initial anchor point location is specified as the possible feasible region for the final optimized location of each anchor point. The shape is optimized by varying the location of the anchor points and the input actuation force. The objective function is to minimize the least squares error (LSE) between the target profile and the solution profile.

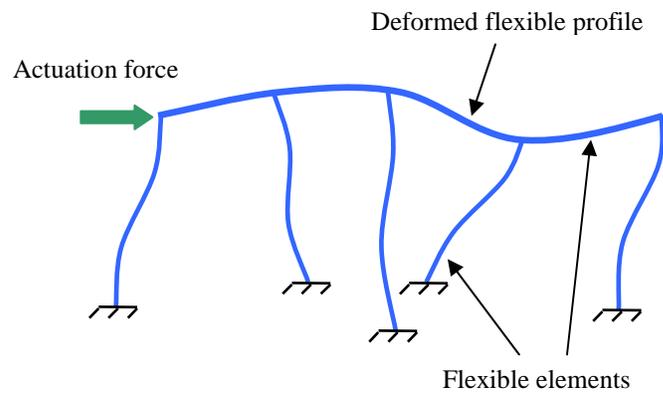


Figure 3.4. Deformation of the flexible model

Figure 3.5 summarizes the design sequence. The steps follow:

- a) Given the values of the anchor points and the input angle (initial or intermediate), and the location of the segmented vertices on the source curve determine

theoretical response of the structure to variations of the input angle using rigid body kinematics methods.

- b) Vary the location of the anchor points within the available anchor region, R_C and change the value of the input angle within the specified bounds, while computing the cumulative difference (LSE) between the attained surface point locations and the desired locations of those points on the segmented target curve.
- c) Stop once the lowest value of LSE is found define a small area around each anchor point location as the feasible domain for the initial anchor locations.
- d) Keep the constraint configurations from the kinematic model, and model the structure as composed of flexible members.
- e) Optimize to find the location of the anchor points by minimizing the LSE between the desired shape and the computed shape.
- f) Examine the final solution.

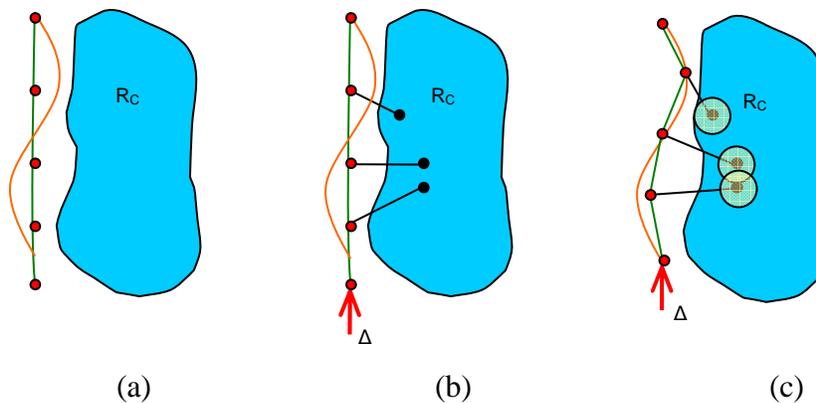


Figure 3.5. Design sequence overview: a) through c) is based on rigid body kinematics;
d) through f) is based on flexible body modeling

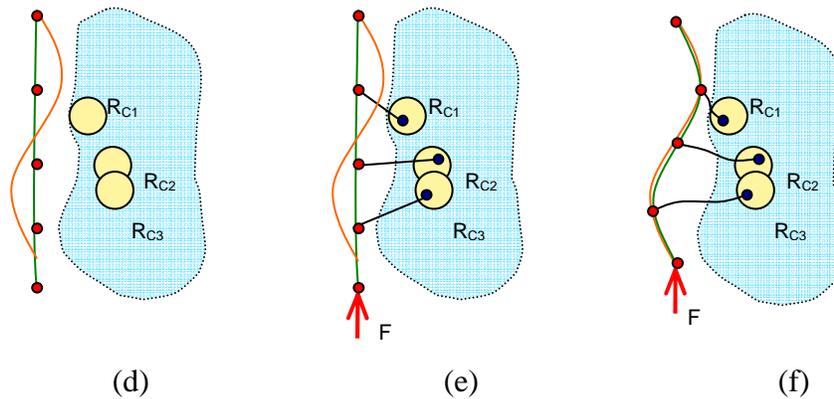


Figure 3.5. (continued)

3.2 Initial anchor selection

Two methods were considered for selecting the initial anchor positions: random and CBDM. Random method populates the available anchor region with the necessary number of anchor points using a random-number generator to assign their x- and y-coordinates. The constraint-based design method defines the possible constraint regions where anchors can be placed.

3.2.1 CBDM anchor selection

CBDM limits the possible constraint regions to just those regions that are feasible with the application of a constraint member. In general, for the displacement of a single point, the anchor of the constraint member would lie on the perpendicular bisector between the two positions of the end of the constraint member as it moves between the source and the target curve. Figure 3.6 shows both the entire available anchor region and the CBDM constraint regions.

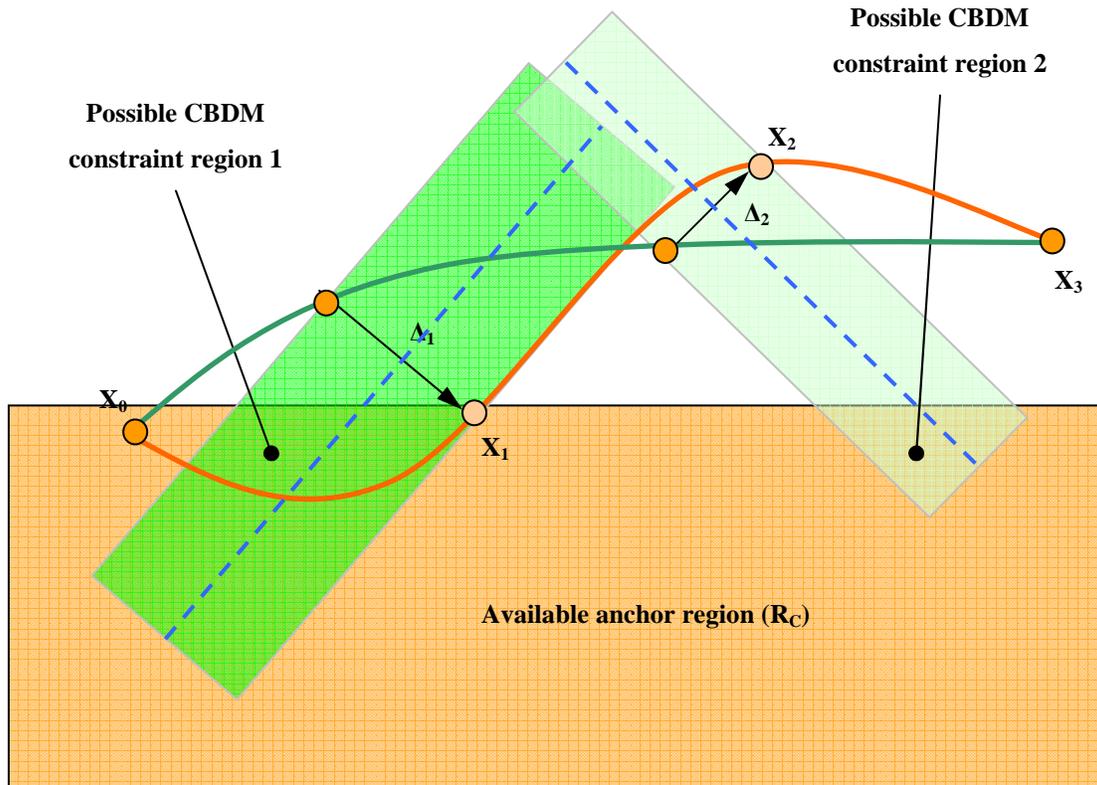


Figure 3.6. CBDM-based constraint regions

The union of the possible constraint regions and the available anchor region result in valid solution regions for anchor positions. The initial anchor placements are then chosen along the perpendicular bisectors that connect points on the segmented source curve to the corresponding points on the target curve. The actual position of each anchor on the perpendicular bisector is determined by the maximum anchor length, specified by the user, as well as the bounds on the available anchor region. Figure 3.7 depicts some of the possible anchor placement scenarios:

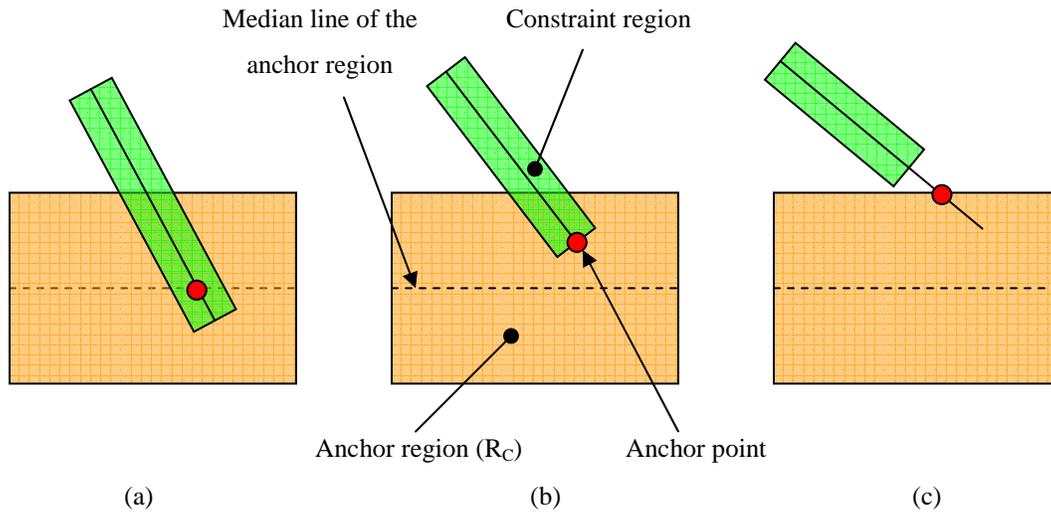


Figure 3.7. CBDM-based initial anchor placement

The following section investigates the viability of the constraint-based method versus the random method for initial anchor placement.

3.2.2 Kinematic feasibility of the initial anchor placement methods

The kinematic analysis of the segmented rigid-body four-bar approximation to the compliant structure is the first step of the design method outlined in this work. Therefore, it was deemed necessary to investigate the behavior of the kinematic solver corresponding to the two available options for generating the initial guesses for the anchor point locations: random and constraint-based. Three design problems (described in detail in Section 5.1) were considered for this study. The global anchor region was populated with 10 sets of randomized anchor positions, followed by a constraint-based anchor position set. The optimized kinematic response of the rigid structure was calculated for different number of anchors, and the least-square error values along with

the algorithm termination times were recorded. Appendix B contains detailed data on the simulation runs.

Based on the results of the study it was concluded that the randomized generation of the initial values for the anchor positions often causes the kinematic solver not to converge on a valid solution, especially for configurations with a large number of anchors. An incorrect kinematic configuration of the segmented rigid-body four-bar approximation to the compliant structure will in turn result in an incorrect FEA-based solution to the flexure response. Therefore, the design method outlined in this thesis utilizes only the constraint-based method of placing the initial anchor points.

3.3 Optimization problem

The goal is to minimize the cumulative difference between the target curve and the achievable curve. Since both shapes are segmented during the synthesis, the objective function results in minimization of the LSE [24] for each segmented endpoint:

$$Diff = \frac{1}{n} \sum_{i=1}^n \sqrt{(X^D_i - X^T_i)^2 + (Y^D_i - Y^T_i)^2}, \quad (1)$$

where (X^T, Y^T) and (X^D, Y^D) are the points on the target curve and the actual curve respectively, and n is the total number of points. The target (and the source) curves are specified by the user, and are used in the constraint determination process. The actual curve, achievable with the designed topology, is computed either during the initial stages of the implemented design sequence via kinematic analysis and later with the aid of a

basic Finite Element Analysis (FEA) code, capable of linear elastic analysis of isotropic structures containing beam and rigid plate elements [30].

The acceptable solutions are constrained to lie within the available anchor region R_C , as outlined in Equation 2.

$$(X^c, Y^c)_i \in R_C, \quad i = 1, m \quad (2)$$

where $(X^c, Y^c)_i$ is an anchor point for a constraint, m is the number of constraints, and R_C is the region of the workspace available for constraint positioning. The analysis proceeds with the following additional assumptions: elastic deformations only, small displacements of the individual constraints, and predefined limited direction and magnitude of actuation input(s).

The coordinates of the endpoints of each constraint element anchor point within the available anchor region serve as the design variables. The base input angle, θ_0 , for the segmented rigid-body analog structure, or the input actuation force, F , in case of the FEA-based analysis step, are also design variables since they determine the input displacement and the resulting shape of the structure. Practically, θ_0 is constrained to the range of $\pi/4$ to $3\pi/4$. The operational envelope of the compliant structure actuation force (direction, magnitude, application node) depends on the material properties associated with the structure and is directly tied to the geometrical profile of individual compliant elements. Furthermore, the number and the size of each segmented element can be adjusted to control the total deflection of the curve.

Finally, the optimization problem can be stated as follows:

$$\min \quad F(\mathbf{X}^C, \mathbf{Y}^C, \theta_0, F) = \frac{1}{n} \sum_{i=1}^n \sqrt{(X^D_i - X^T_i)^2 + (Y^D_i - Y^T_i)^2}$$

where

$$\mathbf{X}^D = G_X(m, \mathbf{X}^C, \mathbf{Y}^C, \theta_0, F),$$

$$\mathbf{Y}^D = G_Y(m, \mathbf{X}^C, \mathbf{Y}^C, \theta_0, F);$$

$$\text{S.T.} \quad (X^C, Y^C)_i \in R_C, \quad i = 1, m$$

$$F_{\text{initial}} \leq F \leq F_{\text{final}}$$

$$\theta_{\text{initial}} \leq \theta_0 \leq \theta_{\text{final}}$$

3.4 Shape segmentation and kinematic analysis

Figure 3.8 depicts a single ‘cell’ of the segmented compliant structure, which spans two neighboring anchor points and the corresponding two points on the deformable profile. These cells are connected in series to provide the ability to determine the locations of all points in the structure.

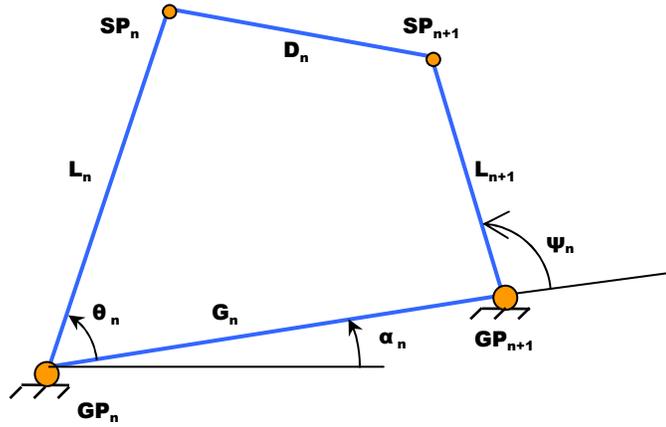


Figure 3.8. Single cell in the segmented deformable structure

Here, SP_n and SP_{n+1} are the two neighboring points on the segmented source surface, and GP_n and GP_{n+1} are the anchor points. Given the four points, it is relatively easy to find the individual link lengths in this 4-bar mechanism, and, ultimately, the expression for the output angle ψ_n angle which relates it to the input θ_n angle, with the aid of traditional planar kinematics analysis [32]. The ψ_n angle can then be used to determine the θ_{n+1} angle, which can then be used to determine the configuration of the next 4-bar ‘cell’ in the structure. This is repeated for all cells within the model. This modeling approach results in the ability to know exactly how the segmented structure, defined by the collection of anchor and surface points, will deform with the given input angle θ_0 .

3.4.1 Basic kinematic analysis

The initial segmentation of the compliant shape-morphing profile and the subsequent solution steps (a), (b), and (c), introduced in Section 3.1, rely only on rigid body modeling. A rigid body is defined by a set of points on an object that always retain

constant distance between any two of them. The discipline of kinematics is concerned with investigation of the geometric aspects of motions of a rigid body (or several connected rigid bodies) without consideration of the forces causing the motions [33]. A mechanism can be defined as a collection of rigid bodies connected together with joints that constrain their relative motion [34]. The connections are designated as kinematic pairs, and every rigid body involved in the construction of a kinematic pair is designated as a link. A sequence of links connected by kinematic pairs forms a kinematic chain, which can be either open or closed. In order for a kinematic chain to be classified as closed, every link in the chain must be connected to at least two other links, with one of the links in the chain being fixed. Furthermore, a simple kinematic chain is defined as a kinematic chain composed exclusively of binary links, that is, links that connect exactly two other links [33]. A mechanism comprising links that move in planes parallel to the base plane and joints with axes that are strictly perpendicular to the base plane, is designated as a planar mechanism [34]. The investigation of the segmented compliant structure's kinematic response presented in this thesis relies on the synthesis and analysis of planar single-loop closed kinematic chains, or mechanisms.

Each cell is represented as a planar four-bar linkage. Vector loop equations are written to analyze the motions of each fourbar cell [35]. Figure 3.9 depicts the vector loop representation of a fourbar, similar to the one in the Figure 3.8, where the links are drawn as position vectors that form a loop.

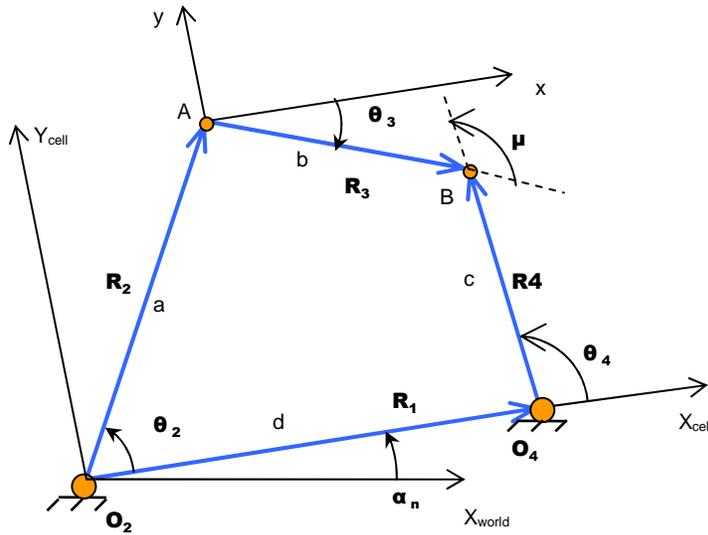


Figure 3.9. Position vector loop for a fourbar linkage

Here R_1 , R_2 , R_3 , and R_4 are vectors and a , b , c , and d are link lengths. The corresponding vector loop equation is as follows:

$$\mathbf{R}_2 + \mathbf{R}_3 - \mathbf{R}_4 - \mathbf{R}_1 = 0 \quad (3)$$

Substituting the complex number notation for each vector we arrive at the following expression:

$$a e^{j\theta_2} + b e^{j\theta_3} - c e^{j\theta_4} - d e^{j\theta_1} = 0 \quad (4)$$

Considering θ_2 as the independent variable and substituting Euler equivalents for the complex numbers results in two trigonometric equations, with θ_3 and θ_4 as the variables. Following substitutions of trigonometric identities, we can arrive at the following expression for θ_4 (it is not necessary to utilize θ_3 in the scope of the design problem at hand):

$$\theta_{4,2} = 2 \arctan \left(\frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \right), \quad (5)$$

where

$$A = \cos \theta_2 - K_1 - K_2 \cos \theta_2 + K_3$$

$$B = -2 \sin \theta_2$$

$$C = K_1 - (K_2 + 1) \cos \theta_2 + K_3$$

and

$$K_1 = \frac{d}{a} \quad K_2 = \frac{d}{c} \quad K_3 = \frac{a^2 - b^2 + c^2 + d^2}{2ac}$$

Note that there are two solutions for Equation 6. These two solutions can be of three types: *real and equal*, *real and unequal*, and *complex conjugate*. The solution is normally expected to be of the *real and unequal type*, which results in two distinct values of θ_4 for any given value of θ_2 . These are classified as the crossed and open configurations, or as the two circuits of the analyzed linkage [35]. Chase and Mirth define a circuit of a linkage as “all possible orientations of the links, which can be realized without disconnecting any of the joints” [36]. If more than one assembly is required in order to guide a mechanism through the specified design positions, the mechanism suffers from a circuit defect. On the same note, a branch is a distinct configuration of the mechanism associated with a given position of the input link [37]. If more than one branch is associated with the prescribed design positions, the mechanism suffers from branch defect. In this case, it is possible that, while passing through a set of positions, the

mechanism may experience a change in branch and enter a singular configuration. This will sometime cause the mechanism to fail because the input link is no longer capable of driving the output link [38]. Also of concern is the possible *complex conjugate* solution, which means that the specified link lengths are not capable of forming a closed fourbar for the chosen value of the input angle θ_2 .

Methods to address these potential problematic scenarios in the context of this research are described in the subsequent section. Note that the θ_2 and θ_4 angles are expressed with respect to the ground link (\mathbf{R}_1) of the fourbar linkage, which is assumed to be coincident with the X-axis of the corresponding coordinate system (Fig. 3.9). Since this is most likely not the case in the general configuration of the segmented structure, with each cell having its own unique orientation in space, θ_2 and θ_4 values that are passed to each subsequent cell for motion analysis are augmented with the α_n angle in order to express them with respect to the global coordinate system .

3.4.2 Solution filtering

The initial configuration of each fourbar cell (link lengths, pivot locations) as well as the overall segmented structure are guaranteed to exist, i.e., the specified link lengths will always be capable of forming a closed fourbar. However, once the anchor pivot locations and the input angle θ_0 are modified, either during the optimization sequence or through direct user input (as outlined in the design framework functionality section), it is expected that one or several of the fourbars in the chain will either undergo a branch change, or will be physically impossible to assemble.

Branch defects pose a significant problem, not only because input link can become no longer capable of driving the output link in the singular configuration of the mechanism, but also due to the fact that each discretized fourbar cell will eventually serve as the basis for the corresponding compliant flexure cell. In the context of the shape-morphing compliant structure design, a branch change will almost certainly result in the catastrophic failure of the corresponding segment of the shape-morphing profile. Due to the random nature of the anchor pivot positioning and the input angle values during the optimization sequence, either of the two values for θ_4 (assuming they are real and unequal) can correspond to a branch change with respect to the original fourbar cell configuration.

To filter out the configuration where θ_4 values result in a branch change, the transition angle value is examined. The transmission angle μ is defined as the angle between the coupler link, represented in Figure 3.9 by vector \mathbf{R}_3 and the output link, represented in Figure 3.9 by vector \mathbf{R}_4 [35]. It is usually assigned the absolute value of the acute angle of the pair of angles at the intersection of the two links. The definition is modified slightly here, and μ is specified as the positive angle between vectors \mathbf{R}_3 and \mathbf{R}_4 (Figure 3.9). The transition angle μ_{def} associated with the segmented fourbar cell in its default (initial) position is computed, compared to the possible two ranges of its values $0^\circ \leq \mu < 180^\circ$ and $180^\circ \leq \mu < 360^\circ$, and its range association is preserved for future comparison. The two transmission angles μ are calculated each time a new pair of the θ_4 values is obtained, either during user interaction or during the optimization sequence. For each value of the μ angle its range association is compared to the default transmission

angle μ_{def} . If they belong to different ranges – the corresponding θ_4 value is discarded, since it will result in a branch change. Figure 3.10 depicts one of the initial configurations of a fourbar cell (a), the corresponding acceptable variations of the transmission angle μ (b, c), and an invalid value for μ that would result in a branch change and is therefore discarded (d).

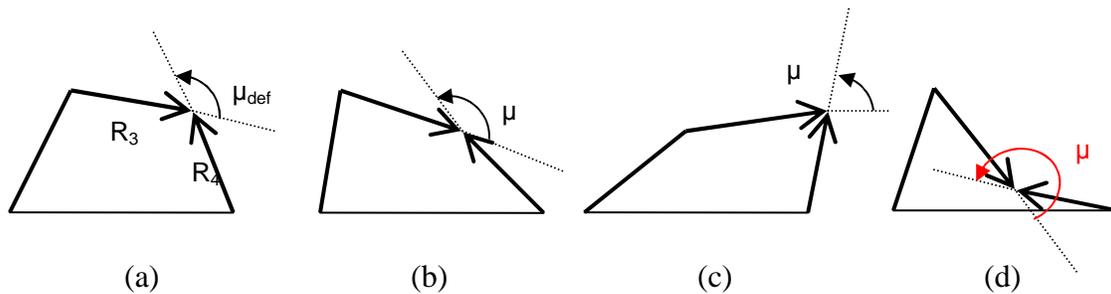


Figure 3.10. The $0^\circ \leq \mu < 180^\circ$ branch of a fourbar cell

The other problem arises when the two θ_4 values obtained from Equation 5 are *complex conjugates*, in which case the fourbar cannot be physically assembled. If such a situation is encountered during the optimization sequence the corresponding variables (anchors positions, input angle) are simply discarded. During the interactive kinematic analysis, where the user gradually varies the input angle, the application continuously keeps track of the valid segmented cell configurations. If the user attempts to specify an unattainable segmented profile - one of the previous valid configurations is retained for the current visualization state of the structure.

The solution filtering methods work well when analyzing each fourbar cell individually. They also hold true as the design framework sequentially processes the segmented fourbar chain representation of the shape-morphing structure. However, a

problem arises if a branch change defect (for example) is encountered a few cells down the chain of the fourbar mechanisms. In that case the preceding cells have already been updated to their respective new deformed states, yet any cells after the problematic chain entry will be rejected as unacceptable, resulting in the discrepancy in the overall deformation behavior of the segmented surface

In case of the fourbar chain in Figure 3.11, fourbar #2 has attained its toggle position (stationary configuration), thus restraining its future motion, as well as the motion of the next fourbar cell in the chain (#3). However, as far as the sequential kinematic solver is concerned, fourbar #1 can continue its motion, since in this case its motion parameters are determined before the rest of the chain.

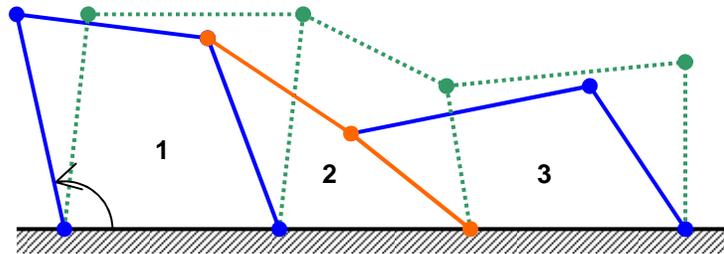


Figure 3.11. Defect in the fourbar cell chain

Due to the iterative nature of the kinematic motion analysis for the entire chain of the fourbar cells the only suitable method to avoid such scenarios is to pre-process the entire fourbar chain with the given input parameters without a permanent geometrical update of the individual cells, instead analyzing each cell for a potential problem. If such a problem is detected for any of the cells regardless of their total number, the entire potential deformation configuration candidate is discarded and the next set of input

parameters is processed. If no problems are detected – a complete analysis sequence is performed on the fourbar chain along with the updates of respective geometric data.

3.4.3 Optimization details

The general standard optimization problem (SOP) statement, outlined in Section 3.3, is modified in order to reflect the specific combination of the input design variables associated with the segmented rigid-body fourbar cell representation of the shape-morphing profile:

$$\min \quad F(\mathbf{X}^C, \mathbf{Y}^C, \theta_0) = \frac{1}{n} \sum_{j=1}^n \sqrt{(X^D_j - X^T_j)^2 + (Y^D_j - Y^T_j)^2} \quad (6)$$

$$\mathbf{X}^D = G_X(m, \mathbf{X}^C, \mathbf{Y}^C, \theta_0),$$

$$\mathbf{Y}^D = G_Y(m, \mathbf{X}^C, \mathbf{Y}^C, \theta_0);$$

$$\text{S.T.} \quad (X^C, Y^C)_i \in R_{C_i}, \quad i = 1, m$$

$$\pi/4 \leq \theta_0 \leq 3\pi/4$$

with $(X^C, Y^C)_i$ as an anchor point for a constraint, m is the number of constraints, R_{C_i} is the region of the workspace available for constraint positioning, and n is the total number of data points used for computing the LSE difference between the attainable and the desired (target) profile. Note that each constraint has a specific constraint positioning region associated with it. This region is derived from the initial CBDM estimation of the plausible solution spaces. Furthermore, the number of the shape profile evaluation points is significantly larger than the number of the constraints, since each of the segmented

elements contains multiple evaluation points. This is done to improve the fidelity of the LSE computations and is especially important at this stage of the analysis since we operate on rigid straight elements (fourbar links). Figure 3.12 depicts the potentially drastic difference in the LSE value computations associated with simply considering the endpoints of the link versus considering the intermediate points along its entire length.

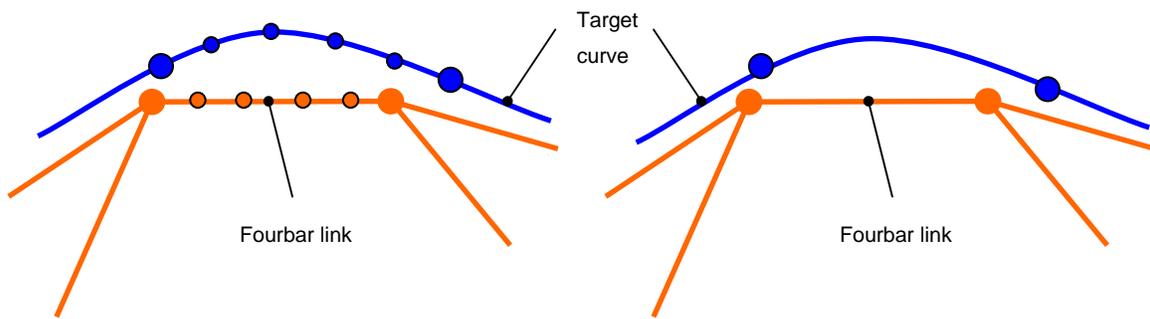


Figure 3.12. LSE computation for a single segmented element with and without intermediate profile points

Each iteration of the optimization cycle considers a unique combination of the anchor positions and the input angle θ_2 , generates the corresponding geometrical configuration of the chain of the fourbar cell using kinematic analysis presented in this section, and computes the associated LSE value. Once the lowest attainable LSE value is achieved, the optimization cycle is terminated, and the anchor positions are forwarded to the FEA-based flexure optimization engine.

3.5 Flexure analysis

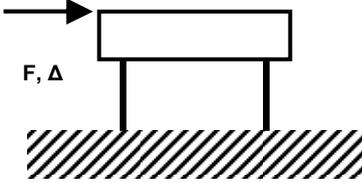
The representation of the shape-morphing profile via a chain of rigid planar fourbar linkages is well suited for approximating the locations of potential anchor candidates in the shape-morphing structures. The associated kinematic analysis requires little system resources, which lends itself to quick convergence of the optimization cycle. However, in order to properly model the response of a compliant structure additional analysis is required. This design framework utilizes a Finite Element Analysis (FEA) solver in order to generate acceptably accurate approximations to the physical response of a compliant shape-morphing structure.

3.5.1 Linear elastic deformation analysis

One of the goals addressed during the development of this design framework is the implementation of a self-contained FEA engine. After some consideration, its intended functionality was restricted to the linear elastic deformation analysis. The reasoning is to retain the basic philosophy behind the CoMeT design tool (described in section 2.2.4) – enabling the end user to rapidly explore multiple solution spaces in order to quickly arrive at the final solution. According to Culpepper and Kim, small-to-moderate motion simulations are much less computationally intensive when compared to large motion simulations; however, they are still fully capable of identifying a mechanism concept as either promising or inappropriate [30]. Table 3.1 lists numerical comparisons between the analysis results of a commercial FEA package and a linear

elastic solver for a compliant beam (1m x .05m x .05m) in cantilever and four-bar configuration and with small and large deformation loading conditions:

Table 3.1. Comparison of large and small deflection results [30]

				
Deformation scale	Small δ [microns]	Large δ [mm]	Small δ [microns]	Large δ [mm]
Linear elastic deformation model	3.127	313	5.621	281
ADINA	3.122	289	5.587	265
% Error	0.16	8.30	0.61	6.04

Error magnitudes listed in the table, while significant for the large deformations, are quite sufficient to narrow the list of possible design topologies down to a few promising concepts, which can then be analyzed in detail [30].

3.5.2 FEA solver setup

The compliant shape-morphing structures primarily operate in two dimensions with regards to the profile changes. The FEA solver utilized in this design framework relies on beam elements arbitrarily oriented in space (Fig. 3.13) [39].

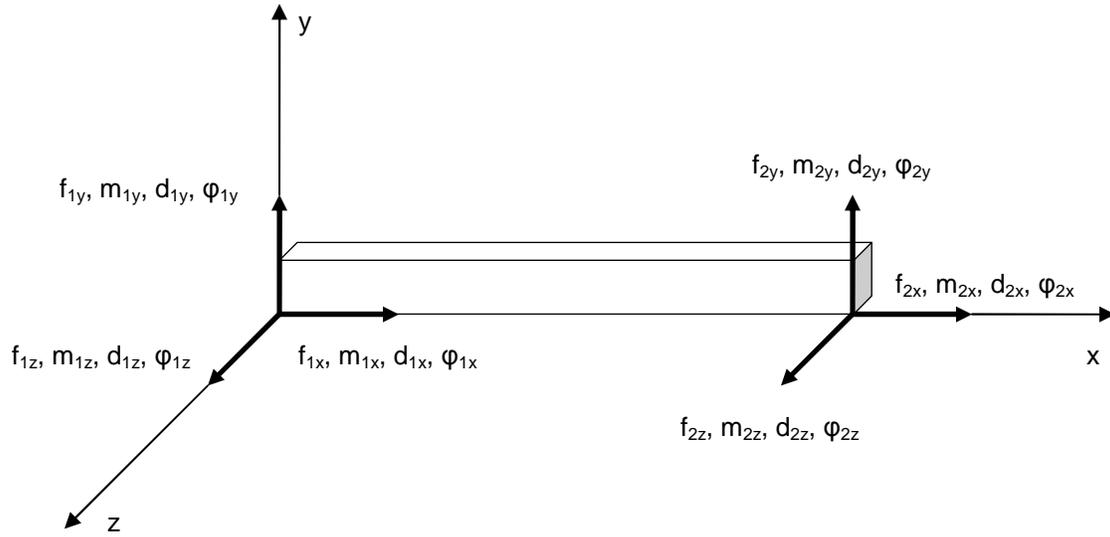


Figure 3.13. 3D beam element [39]

Direct superposition of the stiffness matrices associated with the bending in x - y plane, x - z plane, and the axial stiffness matrix yields the following element stiffness matrix:

$$\hat{k} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{bmatrix} \quad (7)$$

where E is the beam material modulus of elasticity, G is the shear modulus, A is the cross-section area of the beam element, L is its length, and I_y and I_z are the second moments of inertia about the y - and z -axis respectively. Before the global stiffness matrix can be assembled, the individual element stiffness matrices need to be transformed from local to global axis system via the following expression [39]:

$$k = \lambda^T \hat{k} \lambda, \quad (8)$$

where λ is given by:

$$\lambda = \begin{bmatrix} \lambda_{3 \times 3} & & & \\ & \lambda_{3 \times 3} & & \\ & & \lambda_{3 \times 3} & \\ & & & \lambda_{3 \times 3} \end{bmatrix}, \quad (9)$$

and the $\lambda_{3 \times 3}$ is computed via the following expression:

$$\lambda_{3 \times 3} = \begin{bmatrix} l & m & n \\ -\frac{m}{D} & \frac{l}{D} & 0 \\ -\frac{l \cdot n}{D} & -\frac{m \cdot n}{D} & D \end{bmatrix}, \quad (10)$$

where $l = \frac{x_2 - x_1}{L}$, $m = \frac{y_2 - y_1}{L}$, $n = \frac{z_2 - z_1}{L}$, and $D = \sqrt{l^2 + m^2}$ (11)

Following this transformation and eliminating the global stiffness matrix entries associated with the grounded nodes of the individual beam elements, we arrive at the following fundamental expression:

$$K\Delta = V, \quad (12)$$

where K is the global stiffness matrix, Δ is the node displacement vector, and V is the loading vector. The Δ vector can be found using any number of solution methods – in this case and LU-decomposition solver routine was used. Once the node displacements are determined for each of the beam elements we recover the internal forces associated with each one, utilizing the original \hat{k} developed for the given beam. Linear beam deformation theory is then utilized to compute the deflection values at each of the sample points along the length of the beam element.

3.5.3 Optimization details

Similar to the rigid body kinematic approximation optimization problem, the new optimization problem includes the specific combination of the input design variables associated with the segmented rigid-body fourbar cell representation of the shape-morphing profile:

$$\min \quad F(\mathbf{X}^C, \mathbf{Y}^C, \mathbf{F}) = \frac{1}{n} \sum_{j=1}^n \sqrt{(X^D_j - X^T_j)^2 + (Y^D_j - Y^T_j)^2}, \quad (13)$$

where

$$\mathbf{X}^D = G_X(m, \mathbf{X}^C, \mathbf{Y}^C, \mathbf{F}),$$

$$\mathbf{Y}^D = G_Y(m, \mathbf{X}^C, \mathbf{Y}^C, \mathbf{F});$$

$$\text{S.T.} \quad (X^C, Y^C)_i \in R_{C_i}, \quad i = 1, m$$

with $(X^C, Y^C)_i$ as an anchor point for a constraint, m is the number of constraints,

R_{C_i} is the region of the workspace available for constraint positioning, and n is the total

number of data points used for computing the LSE difference between the attainable and the desired (target) profile. Note that once again each constraint has a specific constraint positioning region associated with it; however this time the region is based on the estimated anchor positions obtained from the rigid fourbar chain optimization step. Similarly, the number of constraints is normally not equal to the number of the shape profile evaluation points. Each iteration of the optimization cycle generates the corresponding deformed profile using the linear elastic deformation methods presented in this section and computes the associated LSE value. Once the lowest attainable value of the LSE is achieved the optimization cycle is terminated, and the final anchor positions are presented to the user.

CHAPTER 4. IMPLEMENTATION

4.1 VR design environment

To take advantage of the unique data investigation and interrogation capabilities offered by VR, a scalable compliant shape-morphing structures design framework has been developed. It is currently being used on a desktop VR system, consisting of a computer workstation equipped with a set of stereo glasses and a haptic interface device (Figure 4.1).



Figure 4.1. Design framework in different VR setups

The design framework can also be utilized in a fully immersive multi-screen projection environment. Additional challenges exist when attempting to implement a haptic interface in such an environment, since haptic devices are usually intended for desktop use and have a relatively small physical workspace [40]. Figure 4.2 shows the virtual design environment with a sample mechanism displayed in its original shape and deflected shape with applied loads and constraints.

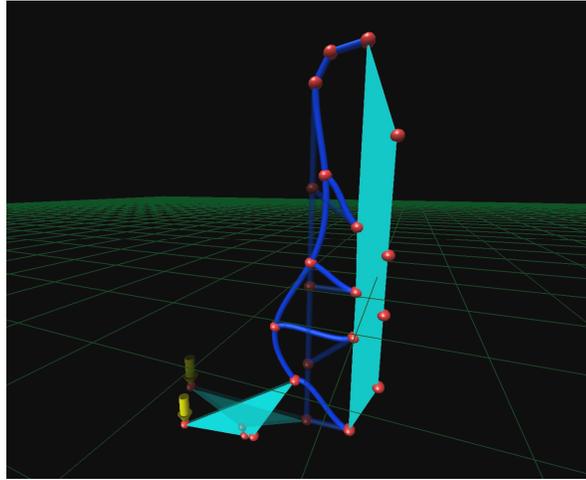


Figure 4.2. Compliant structure in its initial and deformed states, displayed in an immersive VR environment

The framework allows designers to define the problem and view the solution within the virtual environment. An assortment of virtual tools support initialization, positioning and modification of the standard compliant system elements, and input of the loading conditions of the proposed design (forces, anchor points). Design is assisted through force feedback from the haptic interface, which allows precise positioning of the elements via ‘snapping’ to the already-defined features. Furthermore, users have the ability to modify the material properties of the constructed compliant system, change the geometrical configuration of the components (e.g., beam cross-section), and investigate the elastic response of individual beams. An evolved set of haptically-enabled menus provides for effective control over the design framework’s functionality. The design framework is written in C++ using VRJuggler [41]. It can run on any operating system that is supported by VRJuggler, including Microsoft Windows, Linux, and IRIX.

4.2 Design problem solution sequence

This section will outline the individual steps involved in the design of a compliant shape-morphing structure using the design framework.

4.2.1 Problem definition

Figure 4.3 depicts the basic interface to the design framework with the main menu. Users have the ability to navigate the 3D environment and select the operational mode of the framework via a set of haptic (force feedback assisted) menus. Users can also enable ‘snapping’ to one of the sketch surfaces, effectively eliminating one of the degrees of freedom from the shape definition space and restricting the designed profiles to XY , XZ , or YZ planes.

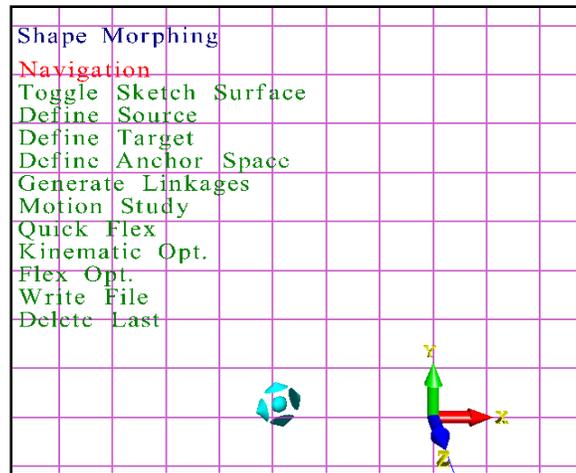


Figure 4.3. Main design environment

Figure 4.4 depicts the first stage of the design sequence, which involves specifying the two distinct profile configurations of the compliant structure. The profile of the structure in its natural (un-flexed) configuration is designated as the source profile

(yellow curve), and the desired configuration of the structure is designated as the target profile (blue curve). The continuous curves are cubic B-splines that pass through the user-defined control points. Users have the ability to specify an arbitrary number of control points for both the source and the target profiles, as well as the ability to modify any existing control points. This allows for specification of any potential profiles. In Figure 4.4 green spheres on each of the curves represent the control points, and yellow cylinders represent the initial estimated locations for the pivots of the segmented rigid-body representation of the compliant structure that will be used in the kinematic motion analysis

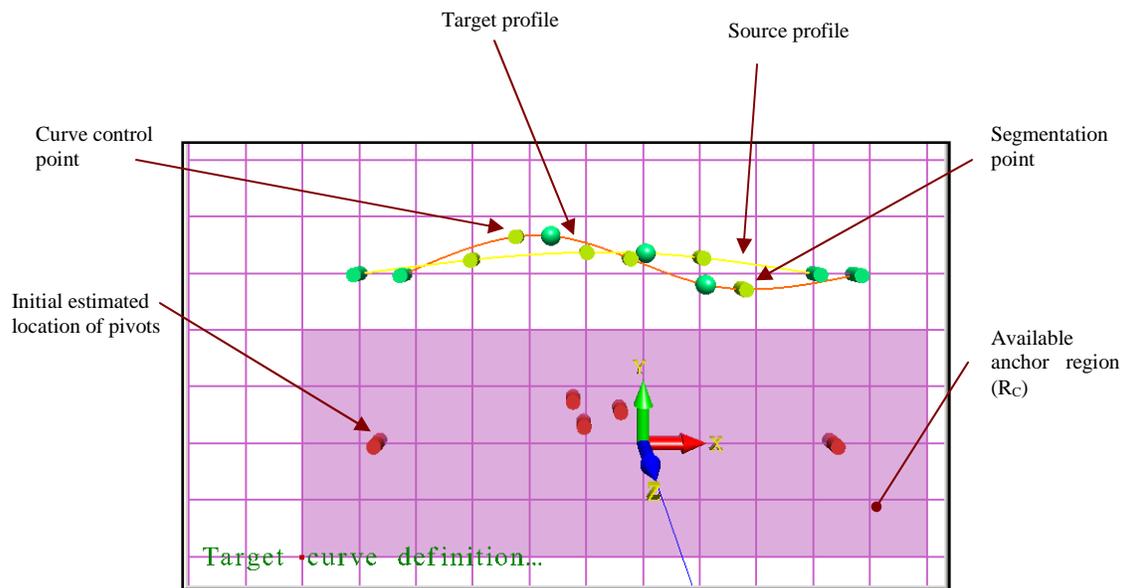


Figure 4.4. Source and target curve specification

4.2.2 Kinematic analysis

Once the user is satisfied with the problem definition, he or she can proceed with the kinematic analysis of the segmented curve. Figure 4.5 depicts the chain of the individual fourbar linkages/cells responding to the motion of the driving link of the first cell in the chain (on the left). The Least-Squares Error is also computed and its value is provided to the user. At this point in the design sequence the user also has the option of viewing the motion of the linkage by moving the first node on the deformable surface (utilizing the haptic interface) and observing the resultant mechanism configuration move in response to the haptic input.

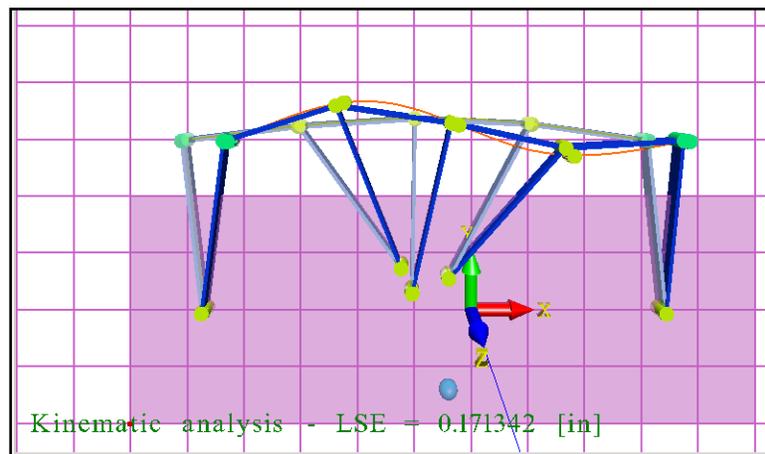


Figure 4.5. Chain of the fourbar linkages in the deflected configuration

4.2.3 Finite element analysis

Following the kinematic analysis of the structure, the FEA analysis functionality of the design framework is performed. Figure 4.6 depicts the shape-morphing structure, subject to an input load, and the resulting deformation of the structure.

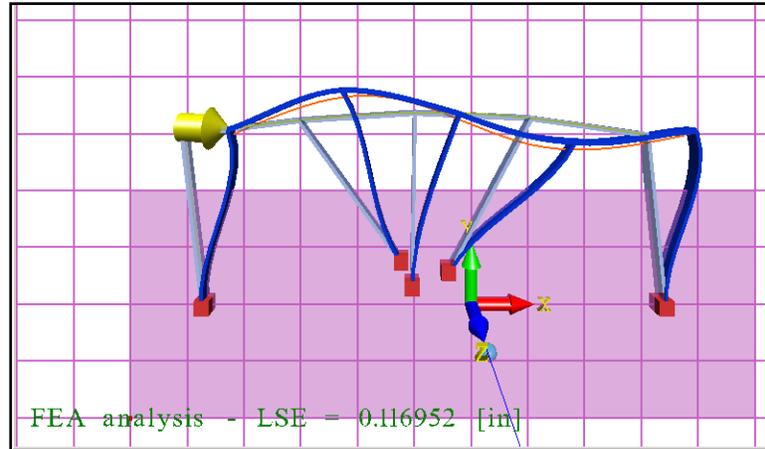


Figure 4.6. Flexed configuration of the shape-morphing structure

4.3 Functionality details

This section outlines some of the details behind the basic functionality of the design framework, including force feedback, interaction options, and mathematical algorithms.

4.3.1 Haptically-assisted menu system

A stand-alone menu object class has been developed. This menu class has the ability to initialize a new instance of the menu object or change contents of an existing menu object at any point in the program's execution, except when the menu is being displayed in the VR environment. The ability to change the menu object's content at runtime is used to update the information displayed in a menu to reflect the current state of the design framework.

During the initialization phase a title is assigned to each menu instance. The menu contents are built by specifying the string designator associated with each option (“Navigation”, “Define source”, etc.). The options are numbered sequentially as they are being added to the menu object. The menu object is automatically resized in order to accommodate the title and option names of different lengths and the different number of options. The menu geometry in the VR environment is created using OpenGL primitives and the GLF library [42]. GLF allows for display of two- and three-dimensional text in OpenGL, with a variety of supported fonts and display options.

Menu interaction is the primary operating state of the design framework, taking precedence over any other activities or states of the application. Users can access the menu system at any time by depressing the corresponding button on the haptic device. Once the menus are displayed, navigation within the available set of menus is performed by moving the haptic end effector. Vertical motion corresponds to selection of the individual entities within the current menu, while horizontal motion cycles through the available menus. Both selection sequences are looped, i.e., upon reaching the end of the available selection options the selection reverts to the first available menu entity.

The entire menu selection operation is assisted by continuous force feedback to the user. Haptics were utilized for menu interaction primarily to reduce the time required to make a particular selection and to increase the fidelity of the interaction (reduce erroneous choices). Upon entering the menu interaction mode, the exact coordinates of the end effector of the force feedback device are determined, and a single haptic attractor point is set at the corresponding spatial coordinates. As a result the user experiences slight resistance as he or she move the haptic device away from the reference point. The

end effector's displacement is continuously computed, and, if it exceeds a predetermined threshold (0.5 inches in the vertical direction for scrolling within the displayed menu, 1.5 inches in the horizontal direction for switching between different menus), the next option/menu is selected/displayed, and a new haptic reference point is set at the haptic end effector's location. As a result the user feels a sequence of distinct "clicks" as he or she navigates the menu system.

4.3.2 Source/target curve specification

It is expected that this design framework will be applied to a diverse spectrum of problems. As such, it is necessary to build an input interface that supports user interaction regardless of the level of complexity. The profile of the structure in its natural (un-flexed) configuration, and the desired configuration of the final structure are two of the primary input tasks. The design framework should provide the user with sufficient control over the geometrical layouts of the two profiles in order to accommodate any potential design problem with arbitrary curve placements and the configurations of the individual curves. Current functionality of the design framework is restricted to in-plane flexures; therefore, a 2-D curve is sufficient to describe any profile. Interpolation of a natural cubic spline was ultimately chosen as the appropriate curve generation method because it can accommodate an arbitrary number of control points, and it provides sufficient control of the curve's profile. Among all twice continuously differentiable functions, natural cubic splines yield the least oscillation about the interpolated function f . Furthermore, unlike other interpolation methods, natural cubic splines actually pass through the associated

control points rather than pass close to them. This allows the users to utilize real-world coordinates and measurements (e.g., coordinate measure machines, etc.) to precisely define the design problem profiles.

A data set $\{x_i\}$ of $n+1$ control points corresponds to a cubic spline with n piecewise cubic polynomials, $S(x)$:

$$S(x) = \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \dots & \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases} \quad (14)$$

In order for these interpolations functions to be classified as a natural cubic spline the following conditions are required:

- The interpolating property is specified $S(x_i) = f(x_i)$ (15)

- The spline segments are continuous, $S_{i-1}(x_i) = S_i(x_i)$, $i = 1, \dots, n-1$ (16)

- The curve is twice continuous differentiable,

$$S'_{i-1}(x_i) = S'_i(x_i), \quad S''_{i-1}(x_i) = S''_i(x_i), \quad i = 1, \dots, n-1 \quad (17)$$

- *Natural* cubic end conditions are satisfied, $S''(x_0) = S''(x_n) = 0$ (18)

Determination of the polynomial coefficients associated with the individual $S(x_i)$ expressions is performed via the tridiagonal decomposition method [43]. The end result is the ability to determine the y-coordinate for any point on the interpolating cubic spline for the given x-coordinate value. This calculation is performed to generate the spline points

for graphical display, as well as to determine the spline segments to be used in the subsequent mechanical synthesis of the compliant structure.

During the source and target profile definition, the coordinates of all available control points associated with either curve are monitored. If any changes in the interpolating spline configuration are detected (e.g., due to moving one of the control points or adding a new one), the polynomial coefficients are recomputed and a new curve profile is generated and displayed on the screen.

4.3.3 Force feedback

The concept of haptics is primarily concerned with acquiring information and manipulating objects through touch [44]. According to Salisbury and Srinivasan, haptic interfaces enable users to touch, feel, and manipulate objects simulated by virtual environments (VEs) and teleoperator systems.[45] A significant portion of the design framework's functionality depends on the ability of the user to experience haptic feedback during its operation. Force feedback is utilized for all aspects of the design process – from interaction with the menu system and setting up the initial problem parameters to the investigation of the potential solution's performance. It should be noted, however, that all of the aforementioned functionality requires only a 3-DOF (Degree-of-Freedom) haptic device, as currently there is no need to provide any torque force feedback data to the user. Therefore, the design framework can be potential utilized on almost any commercially available haptic platform.

Throughout this work a 3-DOF PHANTOM Omni haptic device from SensAble Technologies was utilized (Fig. 4.7) [46].



Figure 4.7. PHANTOM Omni

The device was chosen due to its portability and compact footprint, as well as its industry-standard IEEE-1394 FireWire port interface. Furthermore, it is capable of 6-DOF positional sensing. Some of the operational parameters associated with the device are outlined in Table 4.1:

Table 4.1. PHANTOM Omni specifications [46]

Force feedback workspace	~6.4 x 4.8 x 2.8 [in]
Nominal position resolution	~0.0022 [in]
Maximum exertable force at nominal position	0.75 [lbf]
Continuous exertable force	>0.2[lbf]

One of the more severe limiting factors attributed to the Omni operation is the rather small magnitude of the exertable force it can provide to the operator. Therefore, additional steps may be required in order to scale the range of the forces associated with the operation of the synthesized structure to the force feedback range of the device.

Another major factor to be considered by the haptics user is the significant difference between the refresh rate required to render forces on the haptic device and the refresh rate necessary to display the virtual environment. The frame rate of the graphics part of the application is nominally between 30 and 60 frames per second. If the frame rate drops below 30 frames per second the user tends to experience discontinuities in the visual perception of animated sequences. Haptic refresh rates, on the other hand, are normally fixed around 1000 times per second. If the refresh rate drops below 1000 Hz, the user starts to lose the kinesthetic sense of stiff contact with the haptically-rendered objects resulting a loss in fidelity [47]. To accommodate the distinctly different update requirements, haptics rendering and the graphics rendering are usually performed in separate threads. This requires synchronizing the graphics and haptic events that take place in response to user actions, such as button presses and haptic-specific events, such as touching a constraint, flexing a deformable structure, etc.

This framework utilizes the OpenHaptics™ toolkit from SensAble Technologies to address the aforementioned considerations [47]. Along with a variety of sample code and the hardware drivers, OpenHaptics toolkit includes the Haptic Device API (HDAPI) and the Haptic Library API (HLAPI). The HLAPI enables high-level haptic rendering and is structured similar to OpenGL API programmers. Existing OpenGL code can be reused, simplifying the synchronization of the haptic and the graphics threads. This, however, comes at the cost of having little to no control over the finer operational parameters of the haptic device. As an alternative, the HDAPI allows the user to gain low-level access to the haptic device and to directly render forces of arbitrary magnitudes

and directions. In this work, HDAPI is used to implement the haptic functionality of the framework.

The HDAPI consists of two primary components: the device and the event scheduler [47]. The device abstraction component enables a variety of 3D haptic devices to be used with the HDAPI. The commands that will be performed within the haptic thread are specified via the scheduler callbacks. A typical HDAPI-based application includes the device initialization, generation of the scheduler callbacks that will define the force effects, starting the scheduler, generation of the forces as needed, and, finally, exiting the scheduler once the application is terminated. Appendix A contains a diagram that outlines the typical event sequence for rendering virtual objects via an HDAPI-based program.

State synchronization between the haptic and the graphics rendering loops is accomplished via thread-safe copies of data that contain a snapshot of the state. This provides a better alternative to a mutual exclusion (mutex) approach, where a lower priority thread can fail to release a thread lock in order for the haptics rendering loop to proceed at the necessary 1000 Hz refresh rate [47]. This design framework utilizes two distinct state-management containers: one for the data supplied to the haptic device and one for the data coming from the haptic device. The corresponding framework state variables are outlined in Table 4.2

Table 4.2. Application state synchronization variables

Data obtained from the haptic device <i>hapticDeviceState</i>	Button 1 state (<i>boolean</i>) Button 2 state (<i>boolean</i>) Device coordinates (<i>vector</i>) Device transformation (<i>matrix</i>) Error state
Data supplied to the haptic device <i>hapticDeviceControlState</i>	Anchor point (<i>vector</i>) Render force (<i>boolean</i>) Render node snapping (<i>boolean</i>)

The device state is retrieved as a state snapshot via a synchronous call (*hdScheduleSynchronous*), while an asynchronous call (*hdScheduleAsynchronous*) is used to modify the operational parameters of the device.

4.3.4 Optimization functionality

There are generally two types of optimization approaches: gradient-based, which requires the user to provide the gradient ΔF in addition to the value $F(\mathbf{X})$ for any given combination of the optimization parameters in vector \mathbf{X} , and the derivative-free approach. The gradient computation is often cumbersome, inconvenient, or outright impossible if the function F is not differentiable and is supplied as a complicated evaluation, which is the case in this design framework. Although a finite difference approximation (in one direction) of the form

$$\partial f / \partial x \approx [f(x + \Delta x) - f(x - \Delta x)] / 2\Delta x \quad (19)$$

can be used to compute the gradient of a function, it is normally not advised due to the high cost of the associated operations ($2n$ function evaluations for the gradient using center differences)[48]. Therefore, the framework utilizes a derivative-free algorithm, called NLopt, that requires the user to only supply the values of the objective function $F(X)$ corresponding to a specific set of optimization parameters' values [48]. NLopt is a free/open-source library for nonlinear optimization, developed by Steven G. Johnson, and associate professor of Applied Mathematics at MIT and licensed under GNU LGPL. NLopt provides a variety of gradient-based and derivative-free optimization routines, and is capable of performing global and local optimization with provisions for unconstrained optimization, bound-constrained optimization, and general nonlinear inequality constraints.

Four derivative-free algorithms supported by NLopt were considered for use in this design framework:

- COBYLA (Constrained Optimization BY Linear Approximation) relies on the construction of successive linear approximations of the objective function and constraints with the help of a simplex of $n+1$ points, and optimizes these approximations in a trust region at each step [49].
- NEWUOA, originally developed for unconstrained optimization, seeks the least value of the objective function iteratively utilizing a quadratic model, which is used in a trust region for adjusting the variables [50].
- Nelder-Mead Simplex is a classic optimization algorithm in which a function of n variables is minimized by comparing its values at the $(n+1)$ vertices of a general

simplex, capable of adapting to the local landscape. The vertex with the highest value is replaced by another point [51].

- BOBYQA optimization algorithm performs derivate-free bound-constrained optimization using an iteratively constructed quadratic approximation for the objective function [52].

In order to determine the optimization algorithm suitable for solving the problem specific to this design framework the following timing tests were performed. The compliant lumber support problem (see Chapter 5 for detailed description) was solved using the four aforementioned optimization algorithms for the 5- and 8-anchor shape morphing structure configurations. Table 4.3 outlines the results of the test.

Table 4.3. Comparison of optimization algorithms termination times (in seconds)

	5 anchors	8 anchors
COBYLA	50.63	180.97
NEWUOA	188.19	360.47
Nelder-Mead Simplex	45.63	191.46
BOBYQA	18.86	65.67

Based on the trial runs, the BOBYQA algorithm was chosen as the optimization engine of the design framework.

The name BOBYQA is an acronym for Bound Optimization BY Quadratic Approximation. It requires the user to specify simple bounds for each of the optimization variables and to provide an initial set of optimization variable values that satisfy those bounds. The NLopt implementation of the BOBYQA algorithm also allows the user to

specify multiple termination conditions. These conditions can be specified limits on the tolerances for the function values and/or parameters, limits on the maximum desirable function value, or limits on the bounds for the total number of function evaluations and/or wall-clock time of the optimization cycle [48]. The following termination conditions are used in this framework for both the kinematic and the FEA-based optimization sequence:

- Fractional function tolerance of $1e-6$: the algorithm stops if $|\Delta F| / |F| < 1e-6$
- Maximum wall-clock time of 600 seconds: the algorithm stops when the total elapsed time exceeds 600 seconds.

The latter condition is used primarily as a condition to stop the simulation if the algorithm does not progress to a solution.

The initial set of the optimization variable values is generated via two methods. The user can randomly populate the available global anchor region with a number of data points corresponding to the number of anchors used by the current compliant structure. Alternatively, the initial anchor positions can be selected using the CBDM-based approach and placed along the perpendicular bisectors between points on the segmented target and source curves, while taking care to constrain the initial anchor positions to the global anchor region.

CHAPTER 5. TEST CASES

5.1 Sample design problems

Several design problems were considered during the investigation of this framework's functionality.

5.1.1 Simple curve

The least complex compliant shape-morphing design problem that could be investigated in this design framework is a flexure of a straight profile into a simple convex curve with no inflection points. A sample problem was created based on the shape as described by the following control point coordinate values (Table 5.1).

Table 5.1. Simple convex curve control points (in)

	P1	P2	P3
X	-2.75	-0.5	2.25
Y	3.00	3.50	3.00

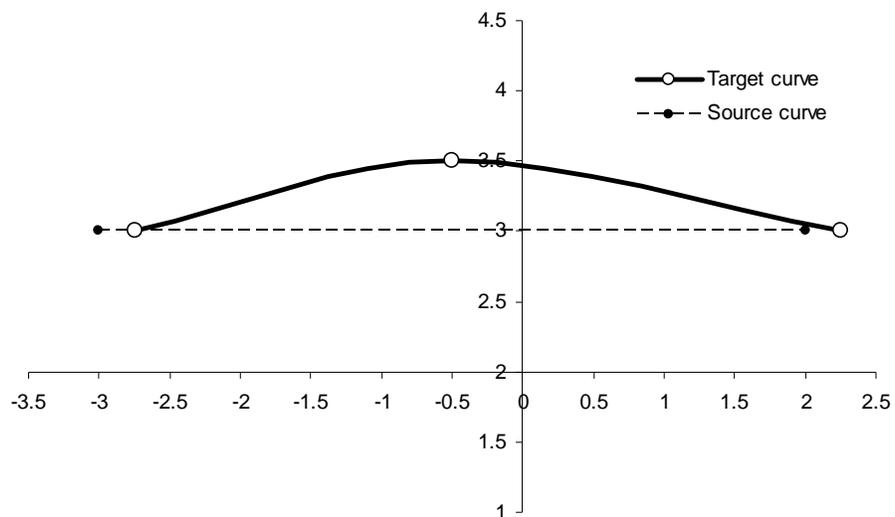


Figure 5.1. Simple convex curve target profile

The curve is segmented into 4 elements, resulting in application of 5 constraint arms. A kinematic rigid body model is created as the first step in the method. The initial locations of the anchor points are determined as described in Section 3.2.2. The kinematic model was then optimized, resulting in new anchor positions. The rigid body model is then replaced with a flexible body model, which is then optimized using the anchor positions from the kinematic analysis step as the initial values. The result is shown in Figure 5.2, where the target curve is represented in red, the unflexed compliant structure in green, and the final solution in blue.

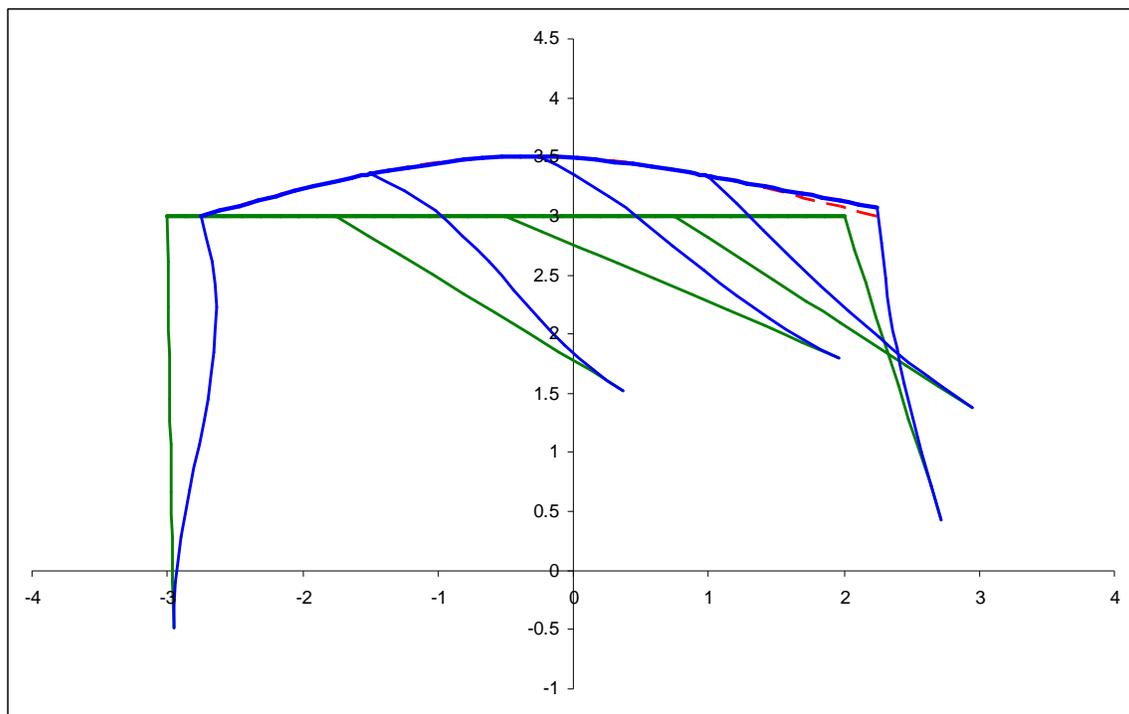


Figure 5.2. Simple convex curve problem solution

This solution provides the least squares error of 0.0091 inches, achievable with the actuation force of 11.86 [lbf]. The solution was generated in 24.22 seconds.

5.1.2 Concave-convex-concave curve

Somewhat more complex compliant shape-morphing design problem to be investigated is a flexure of a straight profile into a concave-convex-concave curve with two inflection points. The associated problem setup is presented in Table 5.2 and Figure 5.3.

Table 5.2. Concave-convex-concave curve target profile (in)

	P1	P2	P3	P4	P5
X	-2.75	-1.5	-0.25	1.00	2.25
Y	3.00	2.75	3.25	2.75	3.00

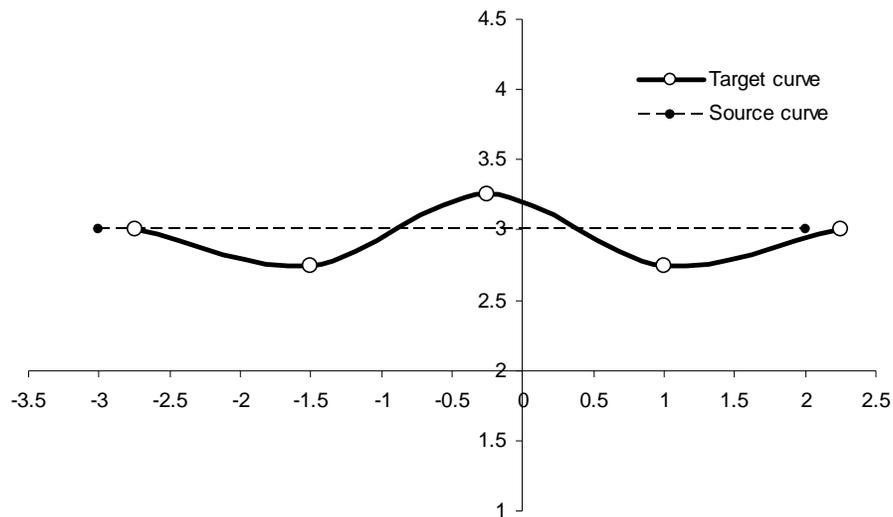


Figure 5.3. Concave-convex-concave curve target profile

For this example 6 constraint arms were chosen, resulting in 5 compliant surface elements. The method resulted in the following solution (Fig. 5.4).

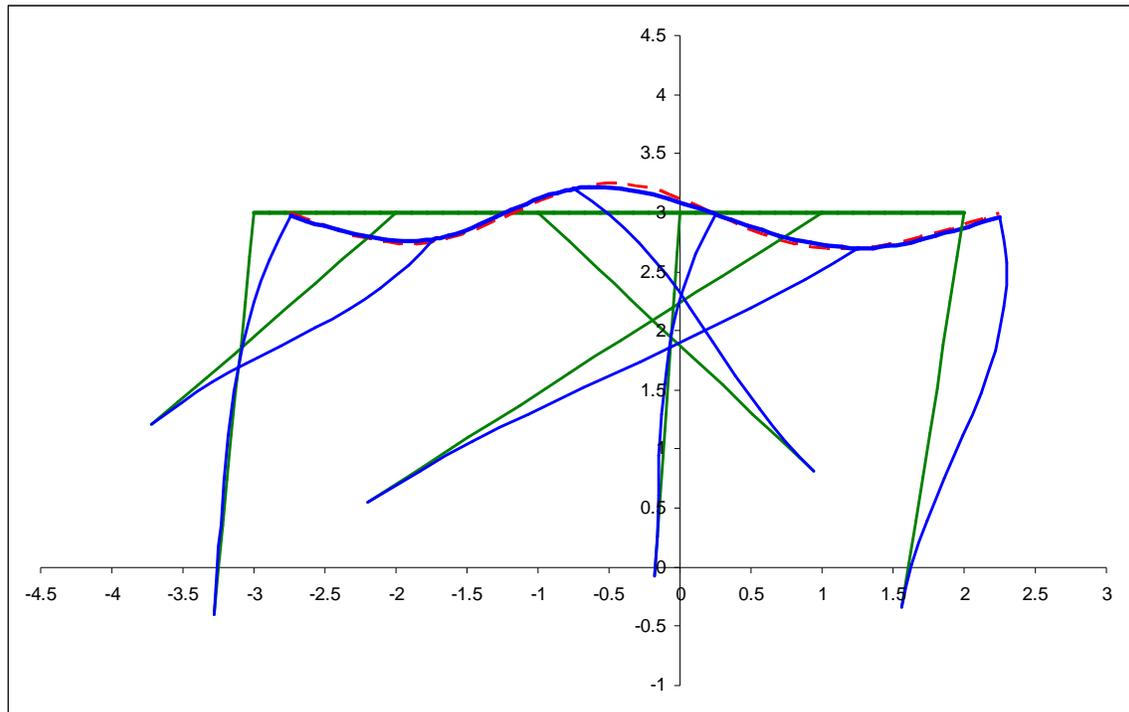


Figure 5.4. Concave-convex-concave curve problem solution

This solution provides the least squares error of 0.0239 inches, achievable with the actuation force of 76.69 [lbf]. The solution was generated in 86.36 seconds.

5.1.3 Compliant lumbar support

It was desired to apply the methods of this research to an already existing problem in the literature. Lu and Kota used the load path approach to synthesize a lumbar support compliant structure [25]. Figure 5.5 shows the target shape, the design shape, and the final compliant mechanism that roughly approximates the natural profile of human spine as a model for a lumbar support in a vehicle seat.

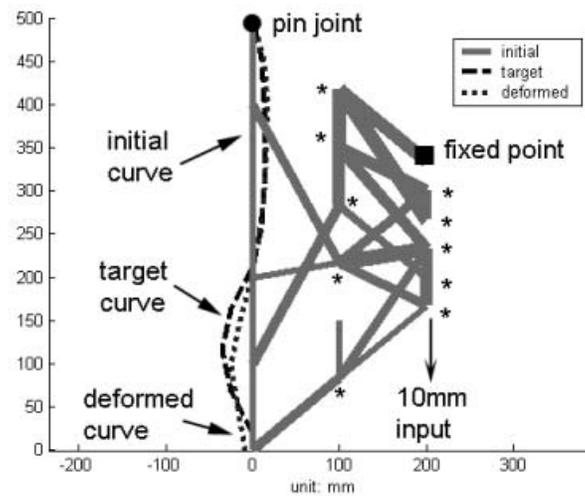


Figure 5.5. Compliant lumbar support [25]

Data for the target curve were not presented in the Lu and Kota paper, so in order to define the target curve to be used in this example, the original profile was scaled at the factor of $100 \text{ [mm]} = 1 \text{ [in]}$, and multiple coordinate points were sampled along the target profile. Table 5.3 contains the coordinate values that serve as the control points for the target profile.

Table 5.3. Compliant lumbar support control points (in)

	P1	P2	P3	P4	P5	P6	P7	P8
X	-2.75	-2.5	-2.0	-1.5	-1.0	0.0	1.0	2.1
Y	3.0	3.1875	3.344	3.3125	3.094	2.875	2.875	3

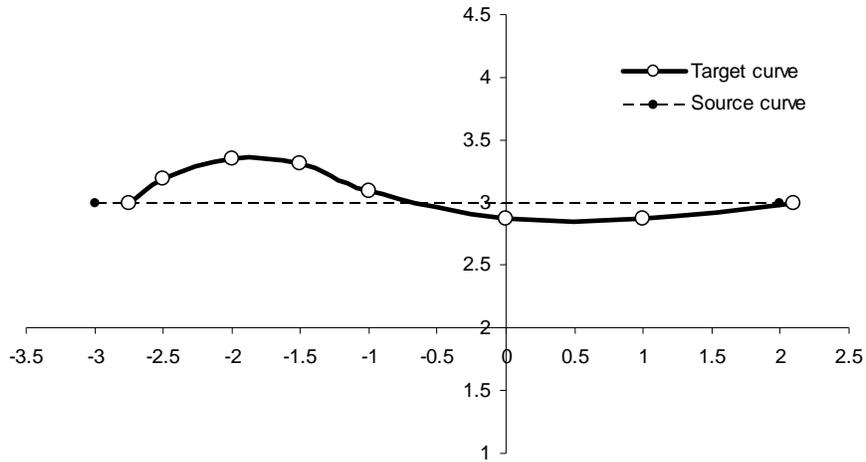


Figure 5.6. Compliant lumbar support target profile

For this example 5 constraint arms were chosen, resulting in 4 compliant surface elements. The method resulted in the following solution (Fig. 5.7):

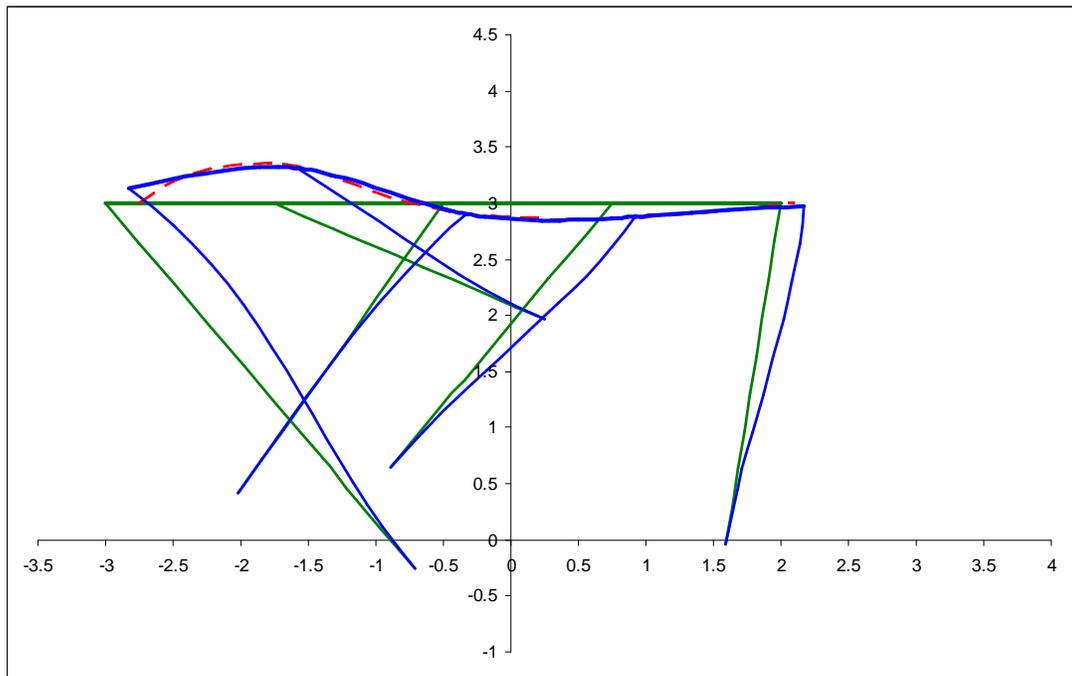


Figure 5.7. Compliant lumbar support solution

This solution provides the least squares error of 0.0468 inches, achievable with the actuation force of 28.97 [lbf]. The solution was generated in 21.27 seconds.

5.2 Detailed analysis of the compliant lumbar support

The compliant lumbar support was used in the in-depth investigation of the design method's performance. According to Lu and Kota, they were able to analyze the problem in the average time of **460 seconds** (7.67 minutes) with the average LSE deviation of 11.24 millimeters (0.44 inches), using the load path approach. Considering the scaling that took place while generating the control points' coordinate data for use in this design framework (100 mm = 1 in), the adjusted average LSE deviation value to be used as a reference is **0.1124 inches**.

The following table and figures contain the synthesis results for the compliant lumbar support generated by the design method outlined in this thesis, including the Least Squares Error values associated with each compliant structure configuration, the actuation force required to achieve the optimum deflection, and the solution time. The material used in the investigation is Delrin 2700, with the individual beam profiles of 0.25 [in] x 0.0625 [in] used for the anchor compliant elements, and 0.25 [in] x 0.0938 [in] beam profiles used for the shape morphing surface elements. The deflected structure configurations corresponding to each design scenario can be found in Appendix C.

Table 5.4. Solutions to the compliant lumbar support design problem

# anchors	LSE [in]	Force [lbf]	Time [sec]
5	0.04678	28.97	21.27
6	0.03795	57.05	26.38
7	0.03841	68.18	41.73
8	0.03763	60.22	85.36
9	0.03763	60.23	124.75
10	0.03348	141.72	375.56

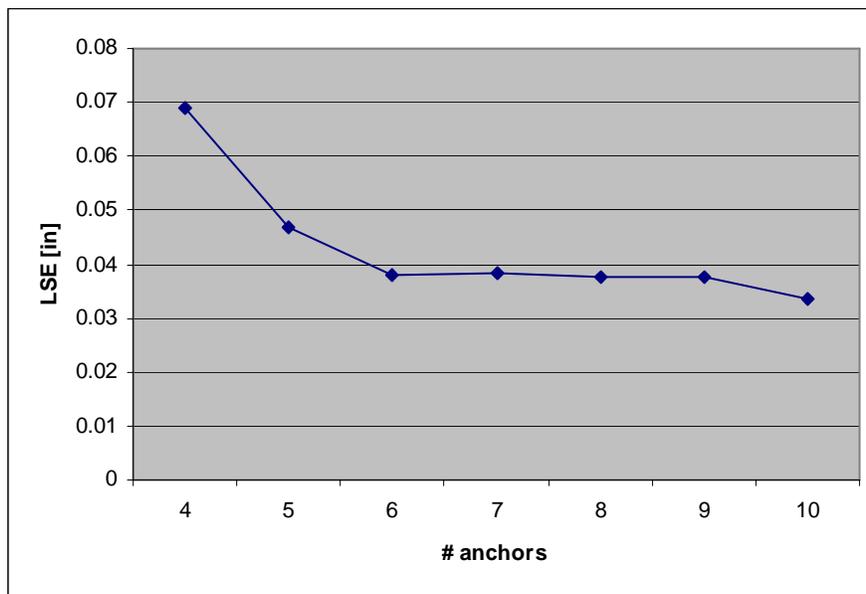


Figure 5.8. LSE values

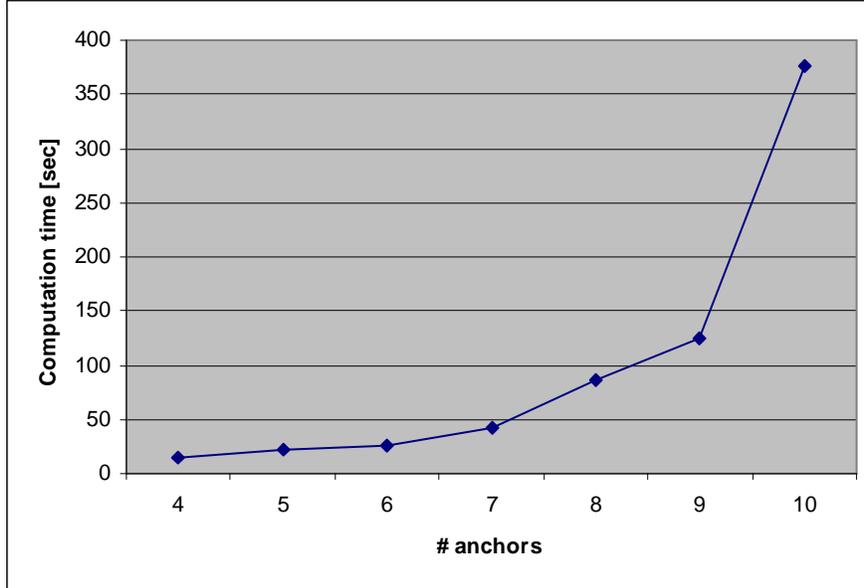


Figure 5.9. Computation time

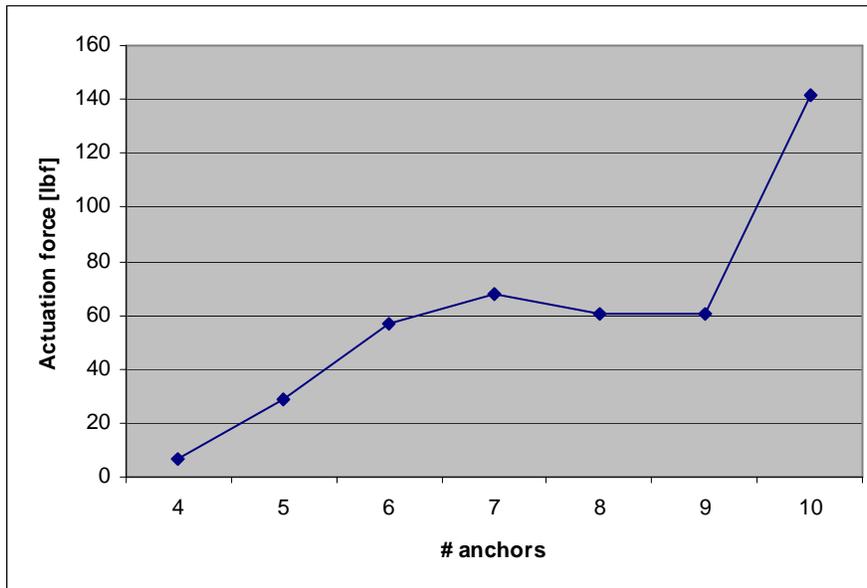


Figure 5.10. Actuation force

Based on these results we can conclude that the proposed design method is capable of generating superior solutions to shape-morphing compliant structure design problems when compared to the existing design paradigm. The resulting structures

possess simpler topology, are capable of higher-fidelity responses, and can be generated quicker.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

A novel approach to the design of compliant shape-morphing structures using constraint-based design method and virtual reality has been developed as an alternative to the two primary methods prevalent in the design community at this time - the pseudo-rigid body method (PRBM) and the topological synthesis (which tend to suffer from either a poor potential solution synthesis capabilities or from susceptibility to overly-complex solutions). A tiered design method that relies on kinematics, finite element analysis, and optimization in order to apply the CBDM concepts to the design and analysis of shape-morphing compliant structures is presented. By segmenting the flexible element that comprises the active shape surface at multiple points in both the initial and the target configurations and treating the resulting individual elements as rigid bodies that undergo a planar or general spatial displacement we are able to apply the traditional kinematics theory to rapidly generate sets of potential solutions. An FEA-augmented optimization sequence establishes the final compliant design candidate. Coupled with a virtual reality interface and a force-feedback device this approach provides the ability to quickly specify and evaluate multiple design problems in order to arrive at the desired solution without an excessive number of design iterations and a heavy dependence on the intermediate physical prototypes.

In the subsequent work we plan to expand the design framework to include the ability to analyze general 3D response of compliant shape-morphing structures (large scale and out-of-plane deformations), to generate methods addressing the secondary

design criteria (interference avoidance, collision avoidance, aesthetics, and ergonomics), as well as to continue improving the design framework interface (e.g., a better method for entering numerical data during the problem specification phase of the design process, which can be addressed by combining virtual menus and voice recognition)

BIBLIOGRAPHY

1. Blanding, D.L., *Exact Constraint: Machine Design Using Kinematic Principles*. 1999: ASME Press.
2. Hale, L., *Principles and Techniques for Designing Precision Machines*. 1999, MIT.
3. Lu, K.-J., Kota, Sridhar. *Synthesis of shape morphing compliant mechanisms using load path representation*. in *Proceedings of SPIE Vol.5049*. 2003.
4. Lu, K.-J., Kota, Sridhar, *Design of compliant mechanisms for morphing structural shapes*. *Journal of Intelligent Material Systems and Structures*, 2003. **14**(6): p. 379-391.
5. Barfield, W., Furness, T.A. III, *Virtual Environments and Advanced Interface Design*. 1995, New York, NY: Oxford University Press.
6. Stuart, R., *The Design of Virtual Environments*. 2001, Ft. Lee, NJ: Barricade Books.
7. Bullinger, H.G., Breining, R., Bauer, W. *Virtual Prototyping - State of the Art in Product Design*. in *26th International Conference of Computers & Industrial Engineering*. 1999. Melbourne.
8. Deisinger, J., Breining R., Rößler, A., Höfle, J., Rückert, D. *Immersive Ergonomic Analyses of Console Elements in a Tractor Cabin*. in *4th Immersive Projection Technologies Workshop*. 2000. Ames, Iowa.
9. Oliveira, J.C., Shirmohammadi, S., Hosseini, M., Cordea, M., Georganas, N.D., Petriu, E., Petriu, D.C. *VIRTUAL THEATER for Industrial Training: A*

- Collaborative Virtual Environment*. in *4th World Multiconference on Circuits*. 2000. Greece.
10. Duncan, T.J., Vance, J.M., *Interactive interrogation of computational mixing data in a virtual environment*. ASME Journal of Mechanical Design, 2007. **129**(3): p. 361-367.
 11. Seth, A., Su, H.J., Vance, J.M., *Development of a dual-handed haptic assembly system: SHARP*. Journal of Computing and Information Sciences in Engineering, 2008. **8**(4): p. 044502-1-8.
 12. Osborn, S.W., Vance, J. M. *A Virtual Reality Environment for Synthesizing Spherical Four-bar Mechanisms*. in *ASME Design Engineering Technical Conference*. 1995. Boston, MA.
 13. Kraal, J.C., *An application of virtual reality to engineering design: synthesis of spherical mechanisms*. 1996, Iowa State University: Ames, IA.
 14. Evans, P.T., Vance, J.M., Dark, V.J., *Assessing the Effectiveness of Traditional and Virtual Reality Interfaces in Spherical Mechanism Design*. ASME Journal of Mechanical Design, 1999. **121**: p. 507-514.
 15. Furlong, T.J., Vance J.M., Larochelle, P.M., *Spherical Mechanism Synthesis in Virtual reality*. ASME Journal of Mechanical Design, 1999. **121**: p. 515-520.
 16. Kihonge, J.N., Vance, J. M., Larochelle, P. M. *Spatial Mechanism Design in Virtual Reality with Networking*. in *ASME Design Engineering Technical Conference*. 2001. Pittsburgh, PA.
 17. Larochelle, P.M. *SPADES: Software for Synthesizing Spatial 4C Mechanisms*. in *ASME Design Engineering Technical Conference*. 1998. Atlanta, GA.

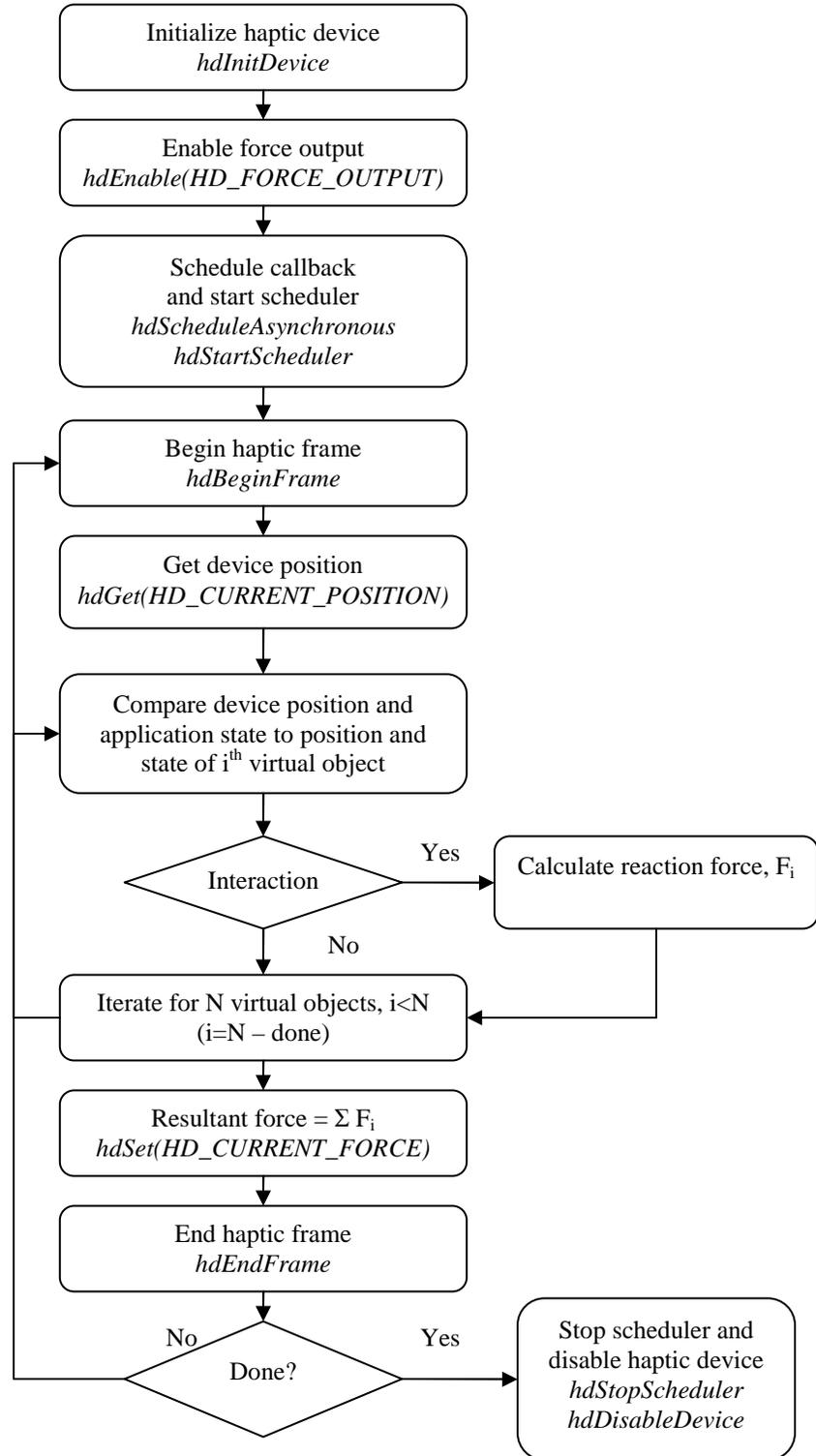
18. Vance, J.M., Larochele, Pierre, Dorozhkin, Denis V. *VRSpatial: Designing spatial mechanisms using virtual reality*. in *ASME Design Engineering Technical Conference*. 2002. Montreal, Canada.
19. Howell, L.L., *Compliant Mechanisms*. 2001, New York, NY: John Wiley and Sons.
20. Maxwell, J., *General considerations concerning scientific apparatus*, in *The Scientific Papers of James Clerk Maxwell (Dover Press)*. 1980.
21. Howell, L.L., Midha, A., *Parametric deflection approximations for end-loaded, large-deflection beams in compliant mechanisms*. *Journal of Mechanical Design*, 1995. **117**(1): p. 156-165.
22. Ananthasuresh, G.K., Kota, S., Kikuchi, N. *Strategies for systematic synthesis of compliant MEMS*. in *ASME Winter Annual Meeting*. 1994.
23. Frecker, M.I., Ananthasuresh, G.K., Nishiwaki, N., Kikuchi, N., Kota, S., *Topological synthesis of compliant mechanisms using multi-criteria optimization*. *Journal of Mechanical Design*, 1997. **119**: p. 238-245.
24. Lu, K.-J., Kota, Sridhar, *Topology and Dimensional Synthesis of Compliant Mechanisms Using Discrete Optimization*. *Journal of Mechanical Design*, 2006. **128**(5): p. 1080-1091.
25. Lu, K.-J., Kota, Sridhar, *An Effective Method of Synthesizing Compliant Adaptive Structures using Load Path Representation*. *Journal of Intelligent Material Systems and Structures*, 2005. **16**: p. 307-317.
26. Hooke, R., *De Potentia Restitutiva*. 1678.

27. Timoshenko, S.P., *History of Strength of Materials*. 1983, New York: Dovers Publications.
28. Su, H.-J., Dorozhkin, D.V., Vance, J.M., *A Screw Theory Approach for the Conceptual Design of Flexible Joints for Compliant Mechanisms*. ASME Journal of Mechanisms and Robotics, 2009. **1**(4): p. 041009.
29. Petri, P.A., *A continuum mechanic design aid for non-planar compliant mechanisms*, in *Mechanical Engineering*. 2002, MIT: Cambridge, MA.
30. Culpepper, M.L., Kim, S. *Framework and Design Synthesis Tool Used to Generate, Evaluate and Optimize Compliant Mechanism Concepts for Research and Education Activities*. in *ASME Design Engineering Technical Conference*. 2004. Salt Lake City, UT.
31. Culpepper, M.L.a.K., S., *CoMeT: Compliant Mechanism Tool Users Guide 1.0*. 2003, MIT Precision Systems Design and Manufacturing Laboratory Publications: Cambridge, MA.
32. McCarthy, M.J., *Geometric Design of Linkages*. 2000, New York: Springer-Verlag.
33. Mallik, A.K., Ghosh, A., Gunter Dittrich, *Kinematic Analysis and Synthesis of Mechanisms*. 1994, Boca Raton, FL: CRC Press, Inc.
34. McCarthy, M.J., *An Introduction to Theoretical Kinematics*. 1990, Cambridge, MA: The MIT Press.
35. Norton, R.L., *Design of Machinery*. 1999: McGraw-Hill.
36. Chase, T.R., Mirth, J.A., *Circuits and branches of single-degree-of-freedom planar linkages*. ASME Journal of Mechanical Design, 1993(115): p. 223-230.

37. Reinholtz, C.F., Sandor, G.N., Duffy, J., *Branching Analysis of Spherical RRRR and Spatial RCCC Mechanisms*. Journal of Mechanisms, Transmissions, and Automation in Design, 1986(108): p. 481-486.
38. Tse, D.M., *Computer-Aided Synthesis of Mechanical Systems for Spatial Motion*. 2000, Florida Institute of Technology: Melbourne, FL.
39. Logan, D.L., *A First Course in the Finite Element Method*. 2002: Brooks/Cole, Thomson Learning.
40. Oliver, J., Dorozhkin, D. *Collaborative Research: Constraint-based Compliant Mechanism Design using Virtual Reality as a Design Interface*. in *NSF Engineering Research and Innovation Conference*. 2008. Knoxville, TN.
41. Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., and Carolina Cruz-Neira. *VR Juggler: A Virtual Platform for Virtual Reality Application Development*. in *IEEE Virtual Reality 2001 Conference (VR'01)*. 2001. Yokohama, Japan.
42. Podobedov, R., *GLF Library*. 2002, Romka Graphics, http://romka.demonews.com/projects/glf/index_eng.htm.
43. Bartels, R.H., Beatty, J.C., Barsky, B.A., *An introduction to splines for use in computer graphics and geometric modeling*. 1987, Los Altos, CA: Kaufmann Publishers.
44. Biggs, S.J., Srinivasan, M.A., *Haptic Interfaces*, in *Handbook of Virtual Environments: Design, Implementation, and Applications*, K.M. Stanney, Editor. 2002, Lawrence Erlbaum Associates, Inc.: Mahwah, NJ.

45. Salisbury, J.K., Srinivasan, M.A., *Sections on haptics, in virtual environment technology for training (BBN Rep. No. 7661)*. 1992, The Virtual Environment and Teleoperator Research Consortium (VETREC), MIT: Cambridge, MA.
46. SensAble, *PHANTOM Omni (R) Haptic Device*. 2010, SensAble Technologies, <http://www.sensable.com/haptic-phantom-omni.htm>.
47. SensAble, *OpenHaptics(TM) Toolkit, Version 2.0, Programmer's Guide*. 1999, SensAble Technologies.
48. Johnson, S.G., *NLOpt*. 2010, MIT, <http://ab-initio.mit.edu/wiki/index.php/NLOpt>.
49. Powell, M.J.D., *A direct search optimization method that models the objective and constraint functions by linear interpolation*, in *Advances in Optimization and Numerical Analysis*, S. Gomez, Hennart, J.P., Editor. 1994, Kluwer Academic Publishers: Oaxaca, Mexico. p. 55-67.
50. Powell, M.J.D., *The NEWUOA software for unconstrained optimization without derivatives*, *DAMTP 2004/NA08*. 2004, Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences: Cambridge, UK.
51. Nelder, J.A., Mead, R., *A simplex method for function minimization*. The Computer Journal, 1965. 7: p. 308-313.
52. Powell, M.J.D., *The BOBYQA algorithm for bound constrained optimization without derivatives*, *DAMTP 2009/NA06*. 2009, Department of Applied Mathematics and Theoretical Physics, Centre for Mathematical Sciences: Cambridge, UK.

APPENDIX A. HDAPI FUNCTIONALITY FLOWCHART



Adapted from [47]

APPENDIX B. INITIAL ANCHOR PLACEMENT STUDY RESULTS

Section B.1 Simple convex curve data

Table B.1. Termination times [sec] – simple convex curve

# anchors	Random											CBDM
	1	2	3	4	5	6	7	8	9	10	Median	
4	0.05	0.06	0.08	0.05	0.05	0.06	0.05	0.06	0.05	0.05	0.05	0.03
5	0.11	0.09	0.16	0.08	0.08	0.08	0.13	0.16	0.14	0.13	0.12	0.05
6	0.11	0.08	0.16	0.11	0.25	0.14	0.13	0.2	0.17	0.23	0.15	0.08
7	0.33	0.41	0.34	0.27	0.25	0.25	0.38	0.2	0.27	0.42	0.3	0.13
8	0.48	0.42	0.16	0.23	0.17	0.61	0.34	0.36	0.17	0.25	0.295	0.14
9	0.33	0.58	0.27	0.25	0.2	0.36	0.19	0.53	0.28	0.66	0.305	0.19
10	0.61	0.31	0.34	0.61	0.53	0.36	0.97	0.3	0.41	0.36	0.385	0.27

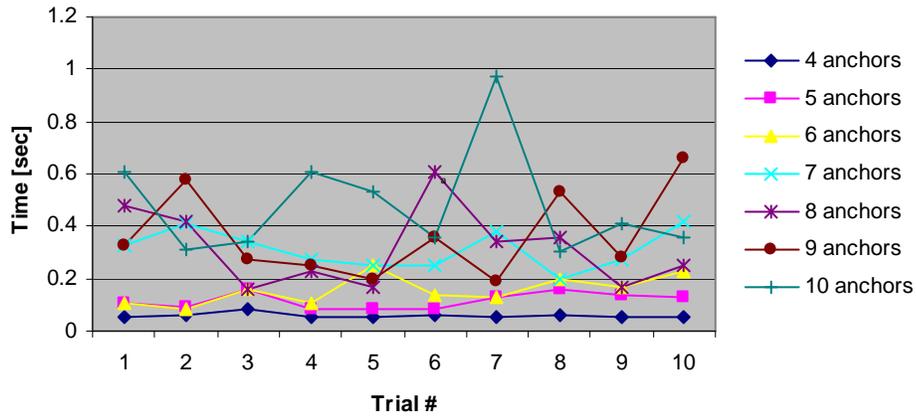


Figure B.1. Termination times – simple convex curve – random data set

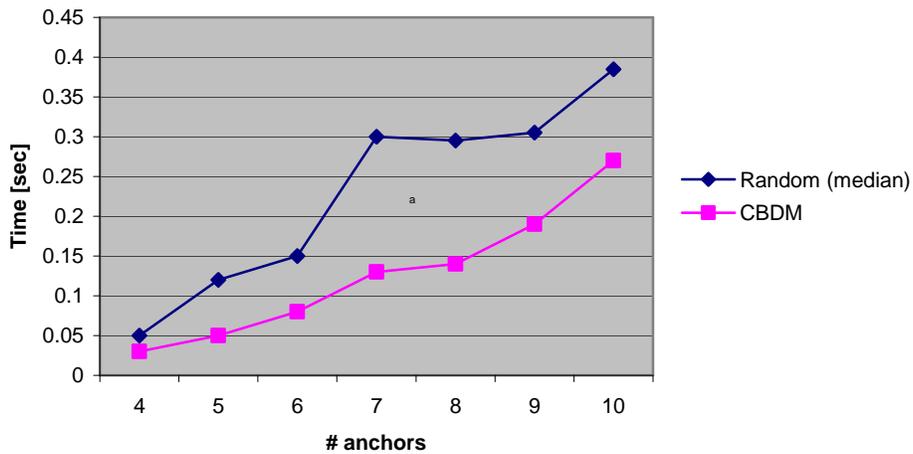


Figure B.2. Termination times – simple convex curve – comparison

Table B.2. Termination LSE values [in] – simple convex curve – simple convex curve

anchors	Random											CBDM
	1	2	3	4	5	6	7	8	9	10	Median	
4	0.0308	0.0308	0.0308	0.0308	0.0308	0.0308	0.0308	0.0309	0.0308	0.0308	0.0308	0.0308
5	0.0251	0.0251	0.0251	0.025	0.1566	0.0251	0.0251	0.025	0.0251	0.025	0.0251	0.0251
6	0.0226	0.1384	0.0224	0.1183	0.0225	0.0225	0.0674	0.0225	0.0224	0.0224	0.0225	0.0224
7	0.0212	0.0212	0.0212	0.0213	0.0212	0.0212	0.0215	0.0212	0.0212	0.0216	0.0212	0.0212
8	0.0226	0.021	0.0283	0.207	0.2622	0.0207	0.2848	0.2073	0.1368	0.0204	0.0826	0.0207
9	0.0586	0.0306	0.0678	0.0322	0.1243	0.1799	0.3016	0.0622	0.1166	0.0202	0.065	0.0203
10	0.1511	0.0198	0.1038	0.0198	0.0199	0.0199	0.1936	0.1197	0.0251	0.0966	0.0608	0.02

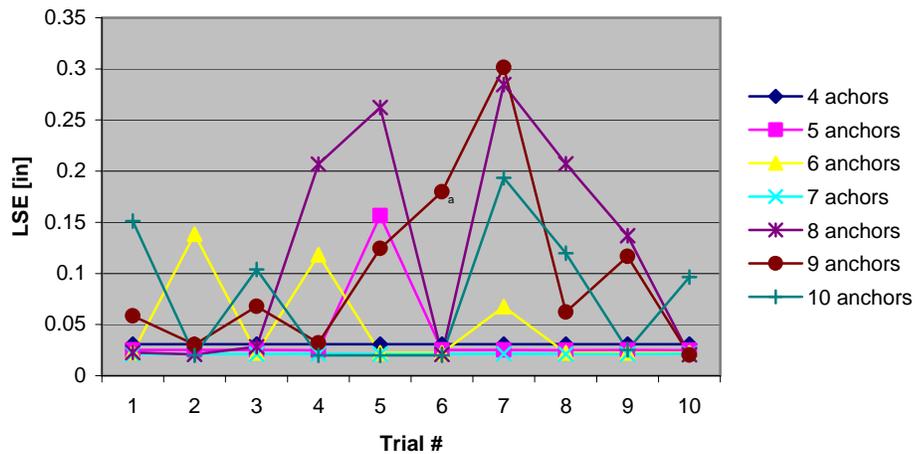


Figure B.3. Termination LSE values – simple convex curve - random data set

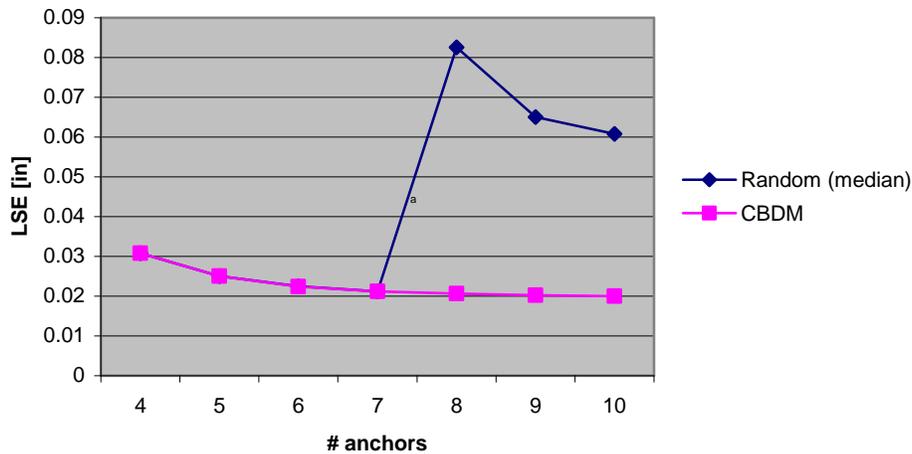


Figure B.4. Termination LSE values – simple convex curve - comparison

Section B.2 Compliant lumbar support data

Table B.3. Termination times [sec] - compliant lumbar support

# anchors	Random										Median	CBDM
	1	2	3	4	5	6	7	8	9	10		
4	0.05	0.08	0.03	0.06	0.05	0.05	0.03	0.08	0.06	0.05	0.05	0.05
5	0.08	0.13	0.23	0.08	0.17	0.09	0.13	0.16	0.16	0.2	0.145	0.08
6	0.2	0.19	0.16	0.14	0.2	0.14	0.25	0.3	0.19	0.16	0.19	0.09
7	0.31	0.39	0.97	0.48	0.55	0.39	0.33	0.08	0.08	0.61	0.39	0.11
8	0.09	0.28	0.48	0.09	0.59	0.09	0.38	0.89	0.61	0.42	0.4	0.17
9	0.94	0.88	0.11	0.17	0.2	0.66	0.36	0.81	1.09	0.11	0.51	0.19
10	0.11	0.34	0.09	0.77	0.42	0.11	0.23	0.28	0.69	0.72	0.31	0.19

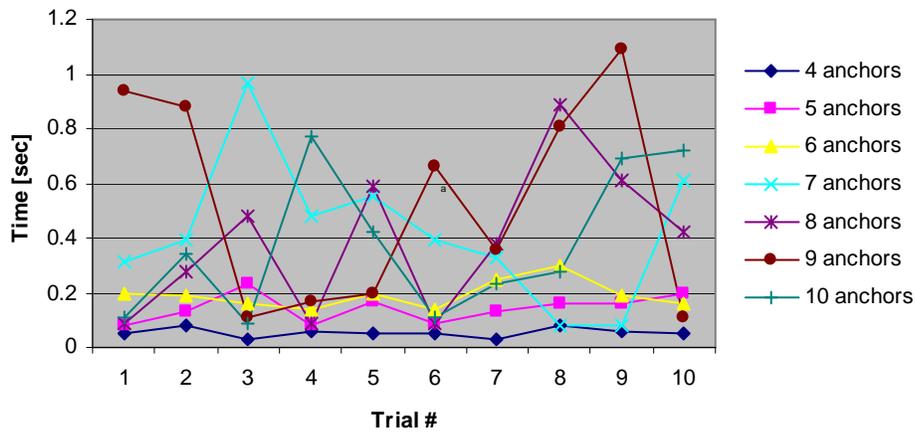


Figure B.5. Termination times] – compliant lumbar support – random data set

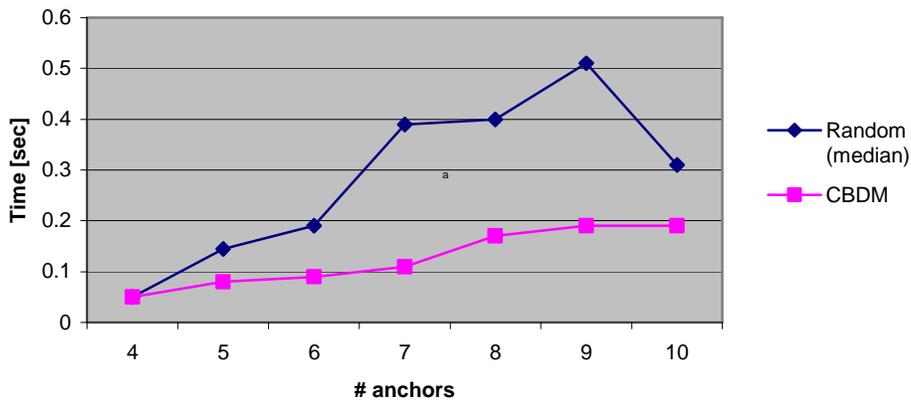


Figure B.6. Termination times – compliant lumbar support – comparison

Table B.4. Termination LSE values [in] - compliant lumbar support

anchors	Random											CBDM
	1	2	3	4	5	6	7	8	9	10	Median	
4	0.0646	0.0646	0.0646	0.0652	0.0646	0.0646	0.0646	0.0646	0.0646	0.0646	0.0646	0.0646
5	0.0325	0.0323	0.0323	0.0323	0.0323	0.0322	0.0323	0.0323	0.0323	0.0323	0.0329	0.0323
6	0.0259	0.026	0.0259	0.0259	0.0259	0.0259	0.0259	0.0259	0.0259	0.0259	0.0259	0.0259
7	0.0245	0.0244	0.0244	0.0244	0.0244	0.0244	0.0245	0.1851	0.1962	0.0245	0.0244	0.0244
8	0.1779	0.0214	0.0214	0.2316	0.0244	0.1978	0.0214	0.0215	0.0214	0.0214	0.0214	0.0214
9	0.0195	0.0195	0.2319	0.1655	0.1627	0.0196	0.1518	0.0195	0.0195	0.2133	0.0857	0.0195
10	0.2197	0.021	0.1942	0.0194	0.0474	0.2348	0.1005	0.0989	0.0196	0.0195	0.0731	0.0195

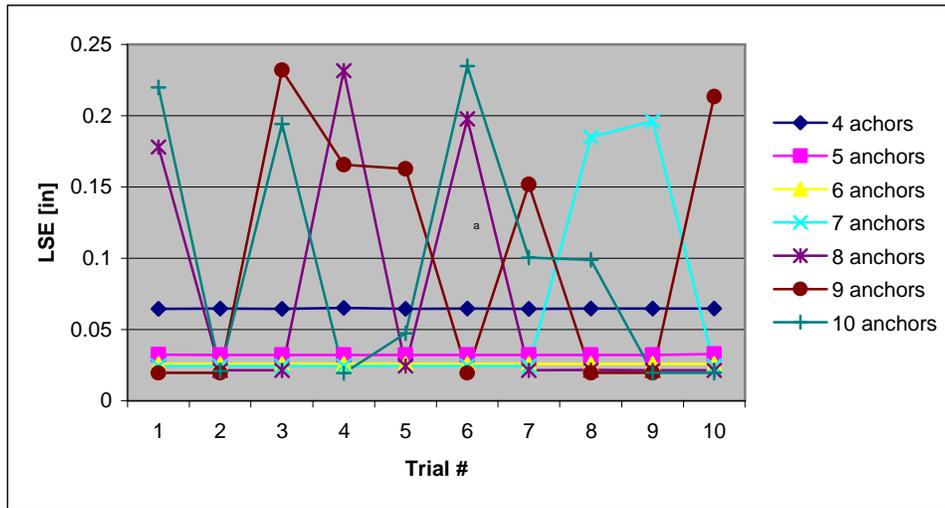


Figure B.7. Termination LSE values – compliant lumbar support - random data set

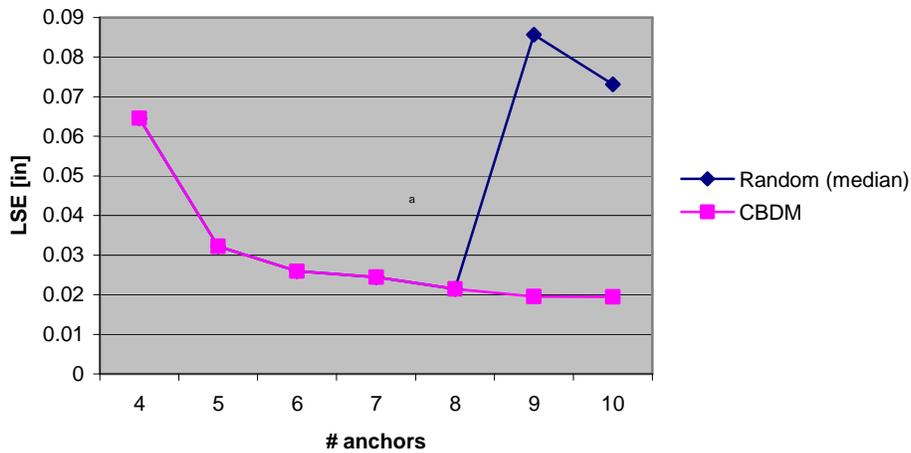


Figure B.8. Termination LSE values – compliant lumbar support - comparison

Section B.3 Concave-convex-concave curve.

Note that the 4-anchor design problem was eliminated from the trial run sequence due to the inability to generate an acceptable compliant structure configuration that would satisfy the design problem criteria.

Table B.5. Termination times [sec] – complex curve

# anchors	Random										Median	CBDM
	1	2	3	4	5	6	7	8	9	10		
5	0.06	0.09	0.05	0.09	0.06	0.06	0.09	0.08	0.06	0.05	0.06	0.05
6	0.09	0.2	0.16	0.17	0.11	0.22	0.13	0.11	0.2	0.11	0.145	0.06
7	0.13	0.08	0.22	0.09	0.28	0.25	0.17	0.27	0.22	0.19	0.205	0.08
8	0.22	0.23	0.33	0.25	0.08	0.27	0.36	0.52	0.09	0.36	0.26	0.13
9	0.3	0.39	0.14	0.28	0.39	0.47	0.5	0.3	0.17	0.34	0.32	0.14
10	0.55	0.45	0.53	0.53	0.17	0.58	0.23	0.56	0.19	0.25	0.49	0.2

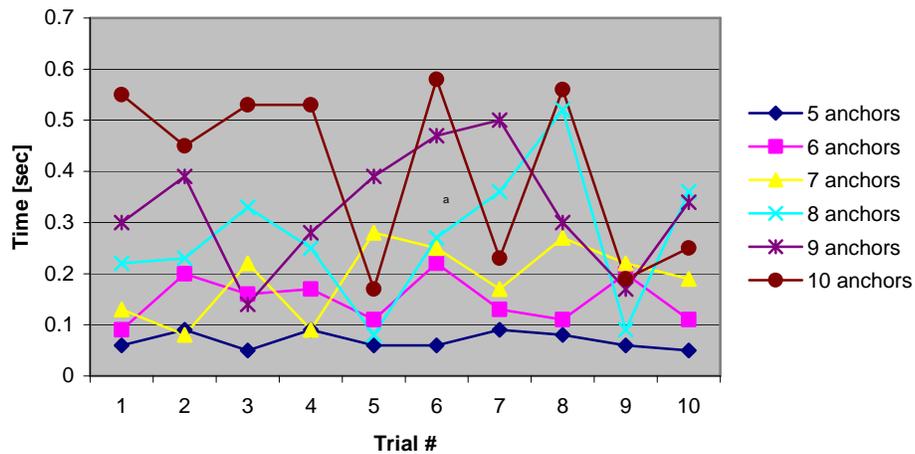


Figure B.9. Termination times – complex curve – random data set

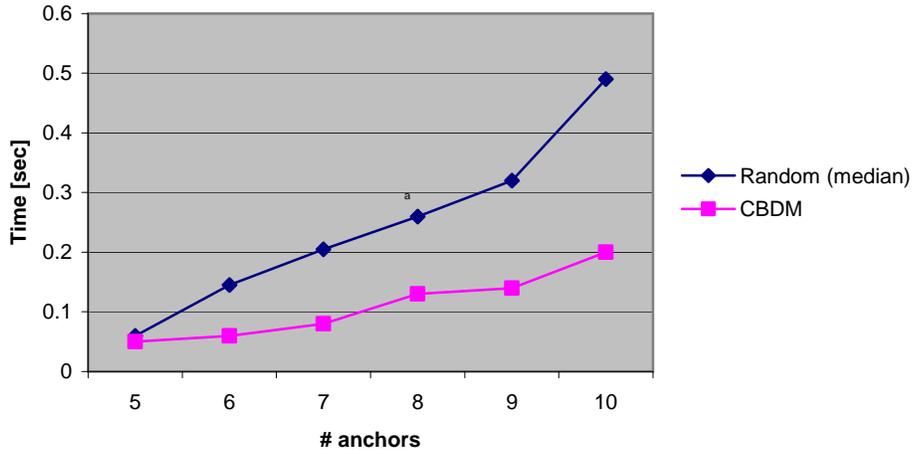


Figure B.10. Termination times – complex curve – comparison

Table B.6. Termination LSE values [in] – complex curve

anchors	Random											CBDM
	1	2	3	4	5	6	7	8	9	10	Median	
5	0.0853	0.0854	0.0853	0.0853	0.0854	0.0854	0.0853	0.0853	0.0853	0.0853	0.0853	0.0853
6	0.0766	0.0767	0.0766	0.0767	0.0766	0.0766	0.0766	0.0766	0.0766	0.0766	0.0766	0.0766
7	0.0737	0.278	0.0737	0.0737	0.0737	0.0737	0.0737	0.0738	0.0737	0.0737	0.0737	0.0737
8	0.0718	0.0718	0.0718	0.0718	0.2959	0.0718	0.0718	0.0718	0.2797	0.0718	0.0718	0.0718
9	0.0705	0.0705	0.1217	0.0705	0.0705	0.0705	0.0706	0.0792	0.255	0.0705	0.0705	0.0705
10	0.0695	0.1033	0.0695	0.0695	0.2471	0.0695	0.2178	0.0776	0.2609	0.2116	0.0905	0.0695

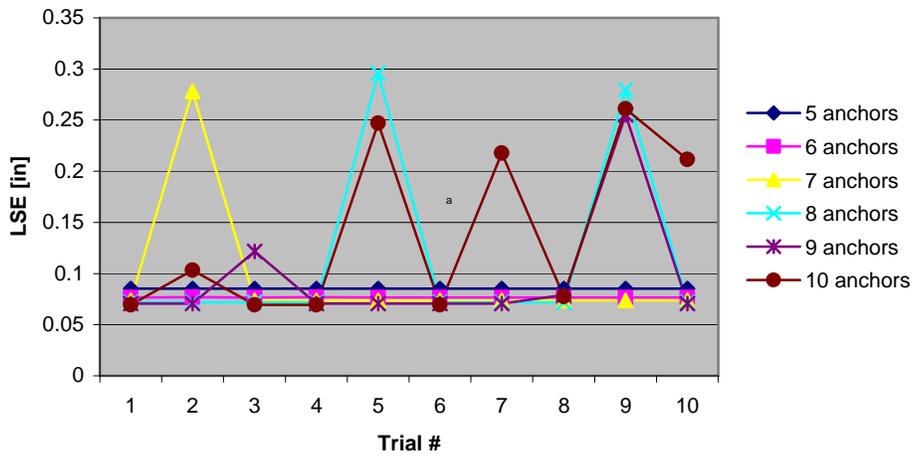


Figure B.11. Termination LSE values – complex curve - random data set

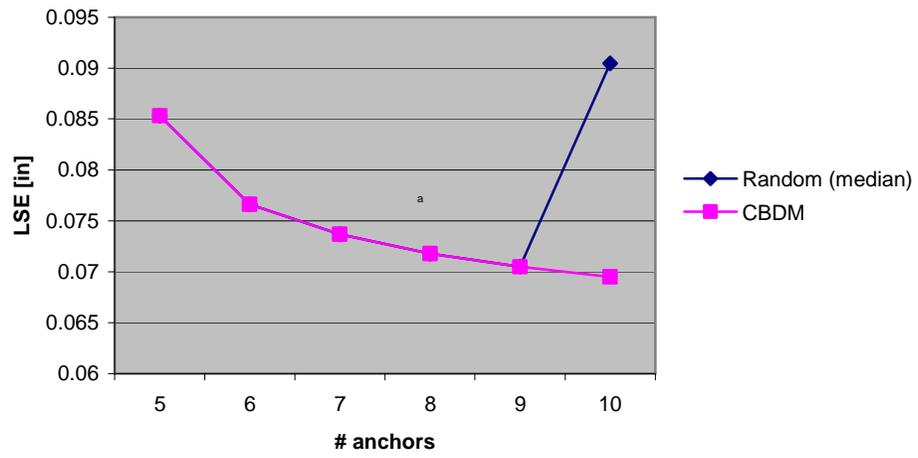


Figure B.12. Termination LSE values – complex curve - comparison

APPENDIX C. COMPLIANT LUMBAR SUPPORT SOLUTIONS

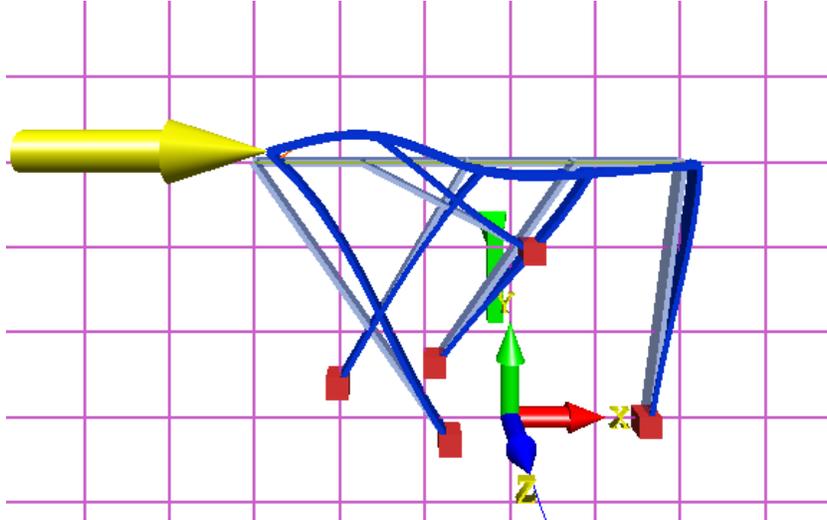


Figure C.1. 5 anchor solution

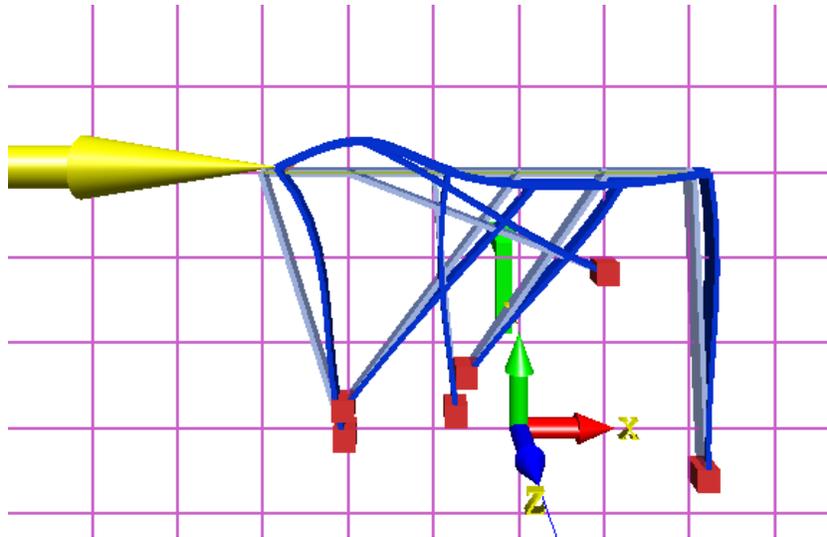


Figure C.2. 6 anchor solution

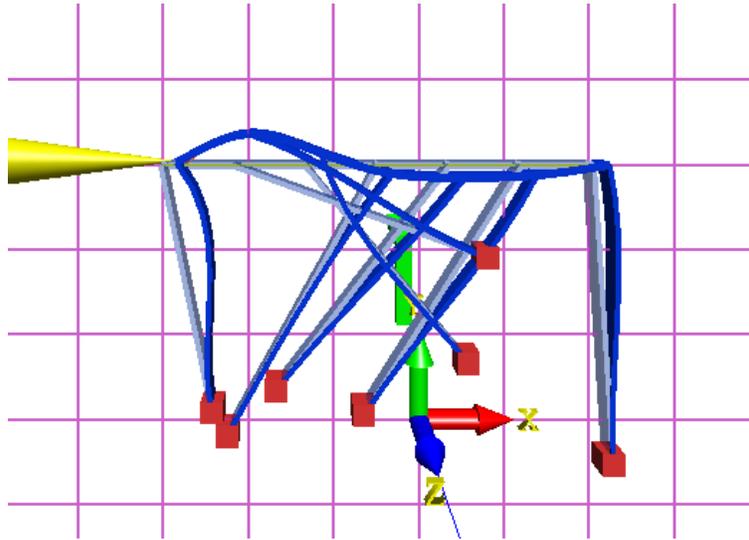


Figure C.3. 7 anchor solution

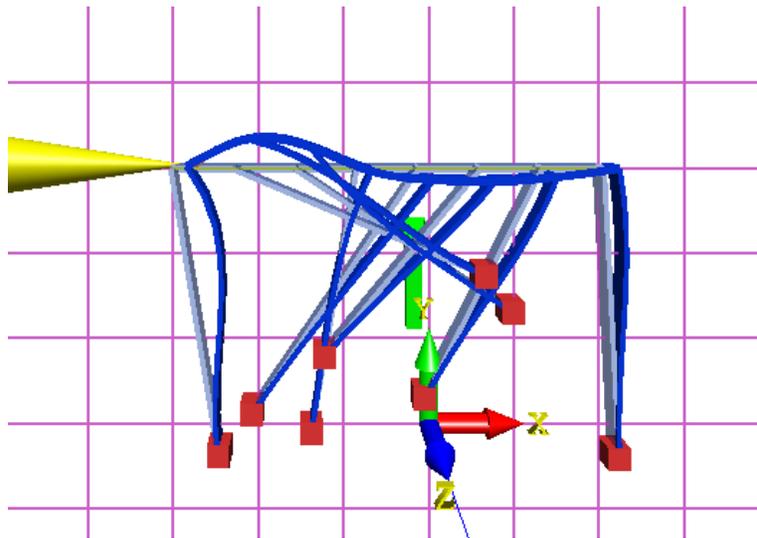


Figure C.4. 8 anchor solution

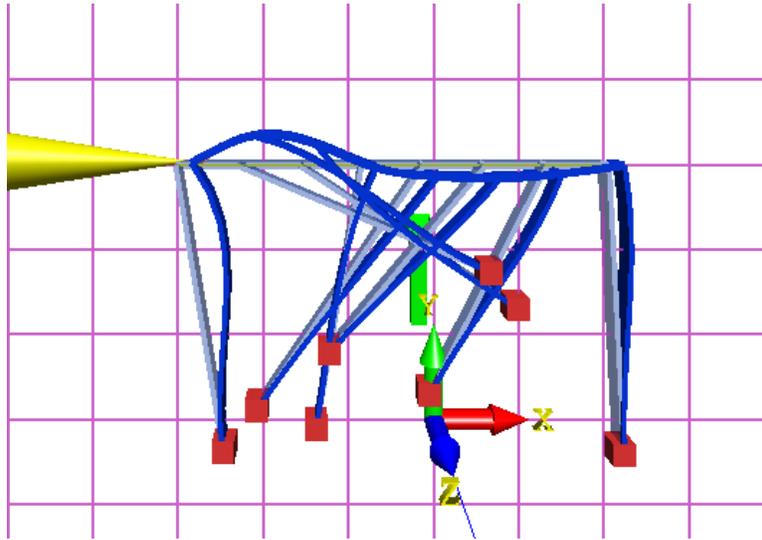


Figure C.5. 9 anchor solution

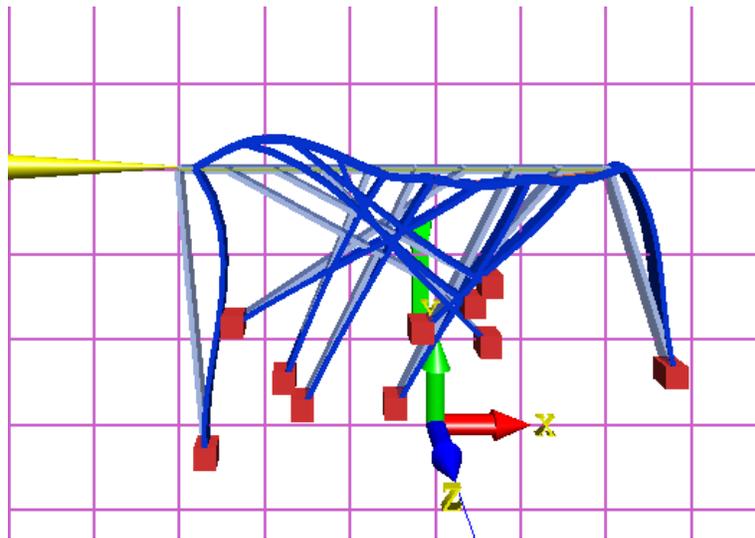


Figure C.6. 10 anchor solution