Evaluation of Modeling Procedures for a Soft-Bodied Robot

A thesis

submitted by

Steven E. McHugh

In partial fulfillment of the requirements

for the degree of

Master of Science

in

Mechanical Engineering

TUFTS UNIVERSITY

August 2011

Principal Advisor: Jason Rife

Committee Member: Rob White

Cross-Department Committee Member: Soha Hassoun

I certify that I have read this thesis and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

(Jason Rife) Principal Advisor

I certify that I have read this thesis and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

(Rob White)

I certify that I have read this thesis and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

(Soha Hassoun [Computer Science])

Approved for the University Committee on Graduate Studies

Abstract

This thesis details two major contributions in the design of highly deformable robots used for Urban Search and Rescue missions. The first contribution of the thesis is a proposed method for obtaining optimized 3D model generation from abstract simulation outputs, while leveraging the benefits of both optimization techniques and human-centric design. The second contribution is a demonstration that a properly constrained simulation is highly beneficial to finding realizable models, as well as a series of tests that ensure the proposed constraints do not significantly hinder the genetic algorithm's search space. To verify the proposed methods, a prototype was designed and tested. The result of the experiment was a partial verification of these methods, in that it showed the methods are feasible to implement, but the experiment also highlighted some challenges that remain in accounting for additional complexities, especially modeling actuator attachment and the effects of contact between the actuators and the body of the robot.

Acknowledgements

I would like to thank, first and foremost, my primary advisor, Jason Rife, for his support, understanding and guidance. I would also like to thank Frank Saunders, whose prior work was an inspiration for this thesis and whose data went into much of the experimental portions of this work, as well as for his guidance in understanding the tools he had previously developed. I would also like to thank Robert White for lending me his Pneumabot prototype control setup for my own prototype experiments. I would also like to thank the entire Chembot team at Tufts University for allowing me to gain insight from their monthly meetings as well as for allowing me to use some of their proprietary pictures for my papers and this thesis. Finally, I would like to thank my parents for their love and support and my husband for bearing the brunt of the emotional support needed to complete this work.

Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Motivation

Urban search and rescue (USAR) utilizes many methods for locating and extracting people trapped in unpredictable and dangerous terrain caused from damaged structures. Time is a critical factor in this task because missing persons can only survive a limited period of time before succumbing to injuries or a lack of basic food and water. However, moving through damaged buildings, such as those encountered in New York in 2001 or Haiti in 2010, is hazardous to rescuers and may even cause further collapse. For the safety of both rescue workers and victims, alternative means of rescue are desperately needed. One proposed solution is the use of highly deformable, light-weight robots (hereafter referred to as "soft robots") to help locate possible survivors, and possibly even bring them limited supplies. Soft robots would be ideal for searching areas inaccessible by either humans or rigid robots by leveraging their flexibility to enter buildings through unconventional means [1].

In general, soft robots have several advantages over rigid robots. In addition to using their unique flexibility and structural dynamic properties to squeeze into small places and traverse a wide variety of terrains, they are easier and cheaper to manufacture due to decreased part count. Furthermore, if resonance phenomena inherent to soft materials were utilized for forward motion in addition to that independently generated from controller actuation, soft robots may even be ideal for missions requiring low power consumptions, such as in space or other remote locations [2]. Even beyond USAR applications, soft robots are ideal for human body contact, such as in medical applications, because their inherent compliance reduces the risk of damaging sensitive tissue [3, 4].

Soft-robotics is a relatively new field, and so looking to nature for inspiration may help identify effective morphologies and control structures for desired task specifications [5, 6]. Work done by some scientists has shown that soft invertebrates are able to accomplish rather difficult tasks with only simple neural architectures. Specifically, caterpillars are capable of crawling, climbing, burrowing, and traversing a wide range of terrains. These capabilities make caterpillars an ideal biological inspiration for soft-robot applications such as search and rescue robotics, for which the ability to locomote in confined spaces is essential [7]. Furthermore, caterpillars are highly deformable with few rigid components, and they are able to utilize a large number of degrees of freedom. Looking towards the *Manduca sexta*, Figure 1, for inspiration, an early prototype of a soft-robot was designed that used shape-memory alloy (SMA) actuators, Figure 2. While this prototype met with limited success, it helped identify some of the key challenges to designing soft-robots that have inspired this thesis [8].

Figure 1: *Manduca Sexta*
(Courtesy of Barry Trimmer)

Figure 2: *Example Early Prototype*
(Courtesy of Barry Trimmer)

The design of a soft robot differs greatly from the more traditional design methods used for rigid robots. While some principles from rigid mechanics can be used, they must be carefully examined and abstracted to account for the large number of degrees of freedom available to the dynamic actuation of soft components. A major obstacle for the designer is the non-intuitive nature in which a soft robot may locomote. For example, the highly deformable nature of soft components can lead to resonant dynamics that affect degrees of freedom not specifically actuated. This means that an actuation scheme that intends to create simple wave-like dynamic patterns may actually cause the robot to move in a highly unanticipated manner as the waves propagate and interact with each other along the body wall.

The principal challenge in designing a soft robot is coordinating the robot's many degrees of freedom to generate efficient locomotion. Soft robots are highly deformable, continuum structures that exhibit an essentially infinite number of degrees of freedom. Since it is impractical to introduce an equivalent number of actuators into the robot, the number of actuated degrees of freedom is significantly lower than the total number of degrees of freedom of the robot. As underactuated structures, soft robots must therefore exploit tuned structural dynamics to enable efficient locomotion. Such tuning has been observed in nature. As an example, the caterpillar Manduca Sexta uses a remarkably simple neuromuscular system to control the motion of its highly deformable, essentially infinite-degree-of-freedom body. The inherent high-degree-of-freedom nature of these soft robots requires a novel method for generating morphologies that will act according to desired control schemes, as well as a means for determining what actuation patterns will most efficiently generate desired movements [9].

1.2 Background

For the purposes of this thesis, a "soft robot" is taken to be any autonomous or remotely operated robotic platform that consists of highly deformable components. These components must give the robot's body a larger number of degrees of freedom than traditional rigid robotic systems [2]. While rigid actuators may be used, the body must still be highly deformable, and the actuators cannot significantly hinder the number of degrees of freedom accessible to the robot's dynamics. While a SMA actuated robot, like the one in Figure 2, is an entirely non-rigid robot, a robot with a soft polymer body and rigid actuators, like the pneumatic ones attached to the outside of the body in the prototype shown in Figure 3, is still considered a soft robot.

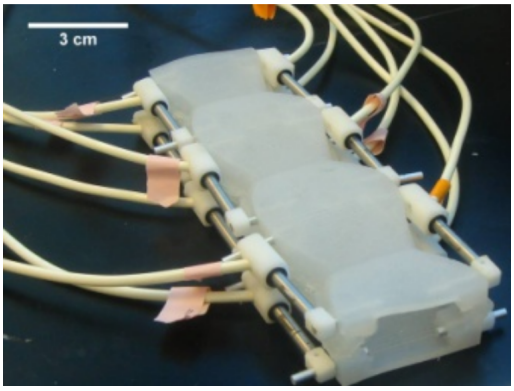


Figure 3: *Soft Robot Prototype with Pneumatic Actuators (Courtesy of [2])*

Rather than employing systematic actuator strategies, current research in soft robotics has tended to utilize deformations in the body for forward motion [10]. For example, one proposed method of actuation is to utilize electro-active polymer gels that simultaneously deform any activated section of the robot [11, 12]. Another group made use of a toroidal flexure actuation technique in their soft robot to allow for a wide range of motion [13]. While these studies help provide insight into the behavior of soft materials, the methods used do

not result in the precision of motion that one would hope to gain from coordinated activation techniques using distributed actuators.

While snake-like robots are made from entirely rigid components, they are designed with a plethora of segments to achieve high degree of freedom motion, which allows for cross-pollination from research in this field and that of soft robots. In addition to the property of flexibility, the typical research objectives of traversing a wide variety of terrains make snake-like robots similar to soft robots [14]. Furthermore, the field of developing snake-like robots has been extensively studied and a large part of this prior work has been in designing these robots to mimic biological lifeforms with notable success [15, 16, 17]. However, the rigid nature of snake-like robots hinders them from deforming or extending axially, while soft robots can use these properties to their benefit. While results from research on snake-like robots can provide some good insight into methods for designing soft-robots, it is unlikely that the control schemes used for these rigid robots would easily apply to soft robotic actuation [18]. Therefore, new methods for designing a soft robot morphology with reliable control patterns are in need of further investigation.

There are a number of labs that have investigated the use of simulation techniques for modeling soft structures with a high number of degrees of freedom. Commonly in these studies, continuum models were used for the soft material models [19, 20]. The work in Walker's paper specifically used Partial Differential Equations (PDEs) to solve for accurate modeling of probing and gripping octopus arm movements. However, this work does not directly apply for the case of Search and Rescue soft robots for two interrelated reasons. Firstly, the octopus model was only in 2D, whereas 3D models for a crawling robot are more desirable to account for complex terrain. Secondly, continuum models would dramatically complicate the evolution of soft robot morphology, especially in 3D.

In work preceding this thesis, a lumped parameter dynamic model was proposed as a modeling scheme for simulating a soft robot [2]. While it will be more computationally efficient than a continuum model, this modeling approach will also prove to be less accurate at resolving internal forces of the robot. However, it was seen in further work, that the gains in efficiency outweighed the loss of accuracy, which proved to be minimal [21]. For these reasons, this thesis seeks to implement, and hopefully improve upon, the lumped parameter approach, as described in much more detail in Section 2.2. Furthermore, success has been found using lumped elements as building blocks for evolutionary algorithms that seek to simultaneously tune structure and controls of robots and other autonomous vehicles with unintuitive dynamics [22, 23, 24]. Utilizing the approach of concurrently evolving structure and controls circumvents the need for extraneous testing of control schemes that could never work with certain morphologies. Therefore, concurrent evolution is a tool that this thesis and prior work has sought to incorporate into our design schemes [2].

The central problem for existing concurrent design methods is that they produce abstract models and not full-realized designs. While efficient gaits for static morphology may be efficiently achieved [21], finding realizable morphology may pose challenges. Without proper constraints on the genotype parameters, physically unrealizable models may emerge from the genetic algorithm [25]. Lumped dynamic modeling is a new scheme that has not yet been fully tested for all possible challenges to its implementation [2].

The focus of this thesis is addressing the central problem facing existing concurrent design methods. Specifically, this thesis seeks to provide a means for converting genetic algorithm parameters from lumped dynamic models into fully realized CAD models that can be constructed. Furthermore, this thesis addresses the inherent need for determining a proper set of constraints on the morphological evolution of the lumped dynamic models to ensure

the optimized output always meets the criteria of being physically realizable. In addressing these problems, this thesis also proposes a need for human involvement in the simulation process to properly set the constraints and smooth an evolved model into a manufacturable one. Finally, this thesis seeks to provide experimental evidence that its contributions to the field of soft robot design improve upon existing methods.

1.3 Contributions

The contributions detailed in this thesis enable models generated from optimized lumped parameter dynamic models to be converted directly into 3D CAD drawings and, subsequently, into fully realized prototypes. The thesis also proposes an alternative formulation for the lumped parameter optimization problem in order to simplify the subsequent 3D realization process. There are two major contributions:

1. *A method is proposed for obtaining optimized 3D model generation from abstract simulation outputs, while leveraging the benefits of both optimization techniques and human-centric design.* At first the functionality of abstract element groupings was explored. Then a pyramidal-block deconstruction scheme was developed that lends itself to further optimization techniques for finding a model with desired dynamical parameters. Furthermore, a strategy is outlined to utilize human-centric design skills to smooth models and quickly reconfigure block initial conditions.

2. *This thesis demonstrates that a properly constrained simulation is highly beneficial to finding realizable models, and that the proposed constraints do not significantly hinder the search space.* A new approach to defining the lumped element genotypes was proposed to allow for geometric constraints. The costs and benefits of basing constraints

on geometrical and dynamical parameters, instead of only dynamical parameters, were explored, with the added knowledge that at least one realizable model exists. Additionally, experiments were performed to analyze if any significant decrease in the search space could be observed.

1.4 Thesis Organization

The remainder of the thesis illustrates the methods and reasoning behind these two major contributions. Chapter 2 describes the design tools used throughout this thesis, such as lumped parameter abstraction, a genetic algorithm and a dynamic simulation utility. Chapter 3 goes on to describe methods for leveraging optimization and human-centric design techniques to transform simulation outputs into a fully realized 3D model (Contribution One). Chapter 4 then describes the methods and motivation for further constraining the genetic algorithm. The chapter also outlines a study on the resulting effects of these new constraints on the genetic algorithm's search space (Contribution Two). Chapter 5 then describes the work done to develop a fully realized prototype designed using the aforementioned contributions, and compares the prototype's motion to gaits of another optimized prototype. The overall results are then discussed in a brief concluding chapter, Chapter 6.

Chapter 2
Design Tools

2.1 Introduction

In this thesis, a genetic algorithm and a rigid-body dynamic solver were the two primary software tools used to optimize soft robot morphologies and control patterns [2]. To utilize a rigid-body dynamic solver to simulate a soft-bodied robot, a lumped parameter model was needed to convert the soft body system into a format compatible with a rigid body dynamics solver. This chapter will explain how these three tools (lumped parameter abstraction, genetic algorithms and rigid-body solvers) functioned in the context of this project.

2.2 Lumped Parameter Modeling Approach

The lumped parameter model was developed to capture lower-order flexibility in a robot's structure. While the tool is a general one applicable to many implementations, the work of this thesis focuses on the example of applying the lumped parameter modeling approach to a caterpillar-like robot that is axis-symmetric and consists of a single repeating segment design. Because the caterpillar, seen in Figure 1, is made up of several distinct segments, an early prototype mimicked this characteristic, as seen in Figure 2. In the lumped parameter representation used in this thesis, segments were modeled as rigid bodies joined by spherical-prismatic joints to enable inter-segmental bending and compression. Linear viscoelastic forcing was applied to each degree of freedom of these joints to represent structural stiffness and internal damping. These segment representations were further subdivided into two rigid elements, also joined by spherical-prismatic joints, in order to model additional bending

and compression within individual segments. Figure 4 illustrates the basic structure of the resulting lumped parameter model. In the figure, each pair of "minor" elements makes up a segment (or "major" element, as labeled in the figure).

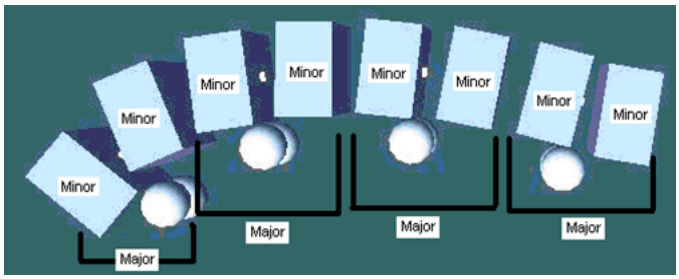


Figure 4: *Simulation Model of the Caterpillar (Courtesy of [2])*

In this thesis, each sequential pair of "major" elements is assumed to be joined by a linear actuator. For example, that actuator may be an SMA spring (as in Chapter 3) or a pneumatic actuator (as in Chapter 4-5). Applying an electrical current to any SMA spring causes resistive heating. The resulting temperature increase, alters the phase of the nickel-tin alloy from which the SMA spring is fabricated. In its higher temperature phase, called Austenite, the SMA spring contracts, while in its lower temperature phase, called Martensite, the SMA spring relaxes. In the lumped parameter model, the x-y-z locations of these springs are adjustable assuming dual bilateral symmetry in the front-view plane (four springs in all, based off of one attachment point). Springs are attached between every other "minor" element (specifically, between the anterior "minor" elements of successive "major" elements). Other adjustable model parameters include "minor" element mass and geometric parameters such as foot separation (w), "minor" element length (L), and "minor" element radius (R), as depicted in Figure 5.

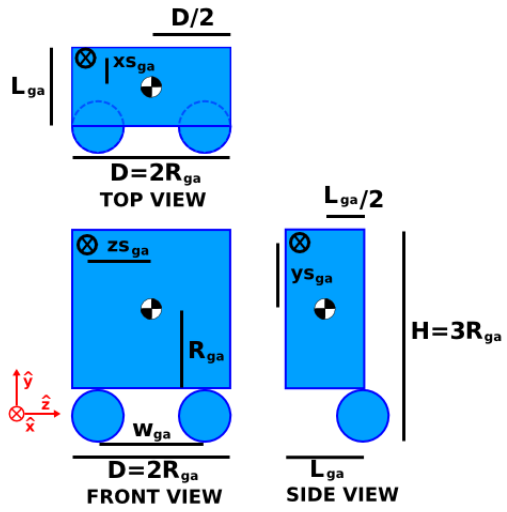


Figure 5: *Minor Element Morphological Parameters Passed into Simulation (Adapted from [2])*

2.3 Rigid-body Dynamic Solver

A rigid-body dynamic solver was used to simulate the performance of a particular lumped parameter model give a particular open-loop control scheme, such as a specified gait pattern. The specific software utility chosen for calculating the rigid-body dynamics was the PhysX© package. PhysX©, like other dynamic simulation utilities, calculate the dynamics of the simulated environment based on models that consist of objects, joints and forces. The equations of motion are calculated based on how the forces applied to one set of bodies move all the bodies joined to, or colliding with, the actuated set of bodies. PhysX© uses the Newton and Euler equations for each segment k to calculate to resultant forces on all the elements in the simulated environment. These equations are described in Equations 1–4.

$$m\dot{v}_k = F_k \tag{1}$$

$$\boldsymbol{I}_k\dot{\omega}_k + \omega_k \times \boldsymbol{I}_k\omega_k = T_k \tag{2}$$

$$F_k = W_k + \sum_l R_l + \sum_m S_m + \sum_n A_n \tag{3}$$

$$T_k = \sum_l \boldsymbol{r}_{l/k} \times R_l + \sum_m \boldsymbol{r}_{m/k} \times S_m + \sum_n \boldsymbol{r}_{n/k} \times A_n + \sum_o \tau_o \tag{4}$$

Description of Variables for Equations 1–4

m is mass
v_k is velocity for segment k
ω_k is the angular velocity
$\boldsymbol{I}$ is inertia matrix about the center of mass
F and T are sums of forces/torques on segment k
W is weight on k
R_l are reaction forces, of which there may be many
S_m are the viscoelastic forces from adjacent elements
A_n are the actuator forces
τ_o are the applied torques
$\boldsymbol{r}_{l/k}$ is the vector to the point of application of reaction force l from the center of mass of segment k
$\boldsymbol{r}_{m/k}$ is the vector to the point of application of viscoelastic force m from the center of mass of segment k
$\boldsymbol{r}_{n/k}$ is the vector to the point of application of actuation force n from the center of mass of segment k

The primary benefit of using a rigid-body dynamics model, rather than a continuum model, is that the rigid-body problems can be described with ordinary differential equations whereas a continuum model necessitates the use of partial differential equations. Partial differential equation solution methods are far less computationally efficient than ordinary differential equation solutions, so many more generations of a genetic algorithm can be simulated and evolved using rigid-body solvers than if classic Finite Difference or Finite Element approaches were used [21].

The PhysX© simulation utility can simulate the interactions, or independent actions, of multiple objects (Actors) in any defined environment (referred to as a Scene). All these Actors can have Shapes and Materials assigned to them, and they may be connected to other Actors with Joints. Figure 6 gives a graphical overview of how a typical PhysX© Scene might be implemented.

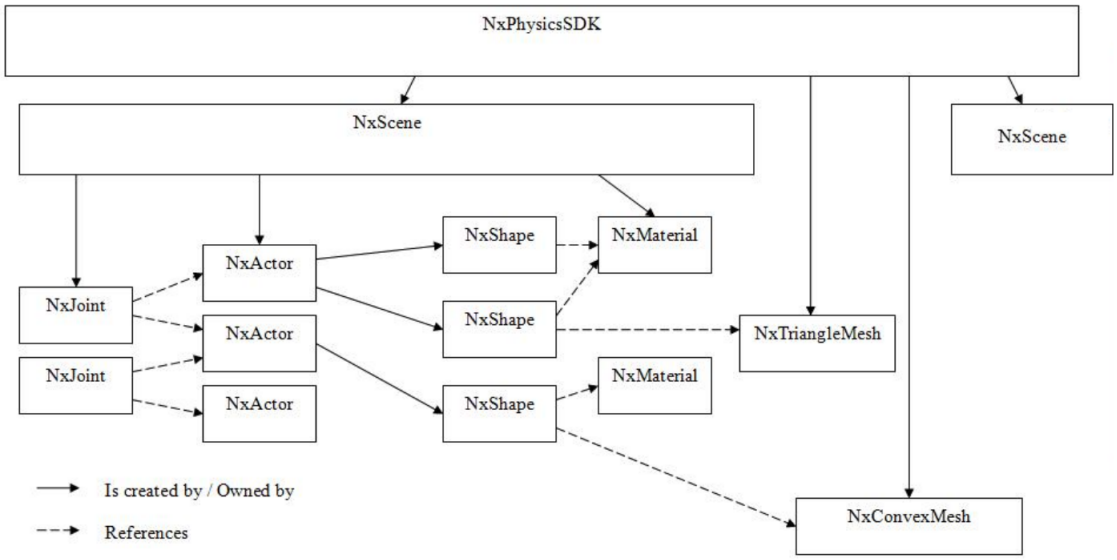


Figure 6: *PhysX© Architecture Diagram (Courtesy of PhysX© Documentation)*

The Scene for PhysX© contains environmental parameters that are applied to all Actors in the Scene, such as friction coefficients, and gravity's relative orientation to the single flat plane provided in the Scene. The Actors are added as sets of predefined Shapes with their relative starting positions to the Scene. When multiple Shapes make up one Actor their positions relative to the first shape are also specified. The Shapes also store properties, such as the Material for each Shape. The Material will inform the PhysX© Scene of the mass and inertia of each Shape, and consequently each Actor, for the dynamic calculations to be performed. Multiple Actors may also be connected to eachother through the use of: revolute,

prismatic, spherical or six-degree-of-freedom Joints. The Joints can also be defined with certain constraints, such as preventing translation or rotation for certain axes, or defining minimum and maximum values for a desired translation or rotation. This property of the Joints allows for the six-degree-of-freedom Joint to be quite adaptable at being defined as any real-world Joint the designer might desire.

The most computationally expensive aspect of calculating the dynamic simulation for these rigid-bodies is the detection and resolution of collisions. PhysX© takes the approach to this problem of defining a "skin width", which translates to the distance one object is allowed to travel into another, such as a body hitting the primary plane of the Scene or two bodies colliding. This method helps give tolerance for PhysX© to iteratively solve for the resulting reaction force of these bodies coming into contact. The motion paths of the bodies are then redefined based on the new forces, and allowed to continue on their way. A diagram depicting the application of "skin width" in the PhysX© environment is displayed in Figure 7.

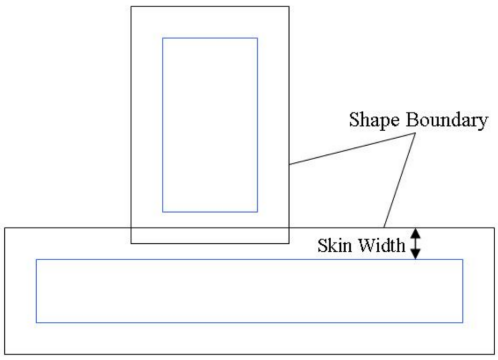


Figure 7: *PhysX© Skin Width Implementation (Courtesy of PhysX© Documentation)*

2.4 Genetic Algorithm

An optimization routine is needed to effectively complete the task of concurrently designing morphology and control patters in a large multi-dimensional search space. Genetic algorithms were chosen as the method of optimization for this problem due to their unique ability to avoid local maximums and minimums in non-convex, multi-dimensional fitness landscapes. Given the complexity of the search space for soft robots, genetic algorithms were seen to be an ideal choice [2]. A genetic algorithm was used for this work as a means for concurrently optimizing the control and morphological parameters that together define the soft robot's gait [2]. A general description of this optimization problem can be expressed by Equations 5–7 where $\boldsymbol{g}$ is the genotype (a vector of control and morphology parameters).

$$\text{MAX}(fitness(\boldsymbol{g})) \tag{5}$$

$$\text{S.T. } c(\boldsymbol{x}, R) \leq 0 \text{ for all time} \tag{6}$$

$$g_i \in [g_{i,min}\ ,\ g_{i,max}] \tag{7}$$

The fitness function for the genetic algorithm optimization favored parameter combinations that caused the robot to travel a straight path on a level surface. Specifically, the fitness of the robot was measured by how far it traveled in an allotted time, provided it stayed within specified dynamic constraints listed in Table 1, as per the segment illustration in Figure 8. The vector of dynamic constraints (c) is dependent on a state vector $\boldsymbol{x}$ (from Equation 6), which describes the center of mass location and orientation, and a radius parameter R, which has several dimensional meanings for the lumped model as seen in Figure 5.

There are also constraints imposed on the genotype elements (g_i) themselves, such that they can only exist between a predetermined minimum and maximum value, as stated in

Constraint to Prevent Cases of:	Equation
Jumping	$c_1 = \min_k (y_k - 2R)$
Tipping	$c_2 = \max_k (2R - y_k)$
Vertical Reaching	$c_3 = \max_k (y_k - 3R)$
Yaw Rotation	$c_4 = \max_k \left(z_k - \frac{1}{10}R\right)$
Lateral Drifting	$c_5 = \min_k \left(z_k - \frac{1}{20}R\right)$
Backward Motion	$c_6 = \max_k (-x_k)$
Rolling	$c_7 = \min_k \left(-\hat{b}_{y,k} \cdot \hat{w}_y\right)$

Table 1: *Dynamic Constraints (Adapted from [2])*

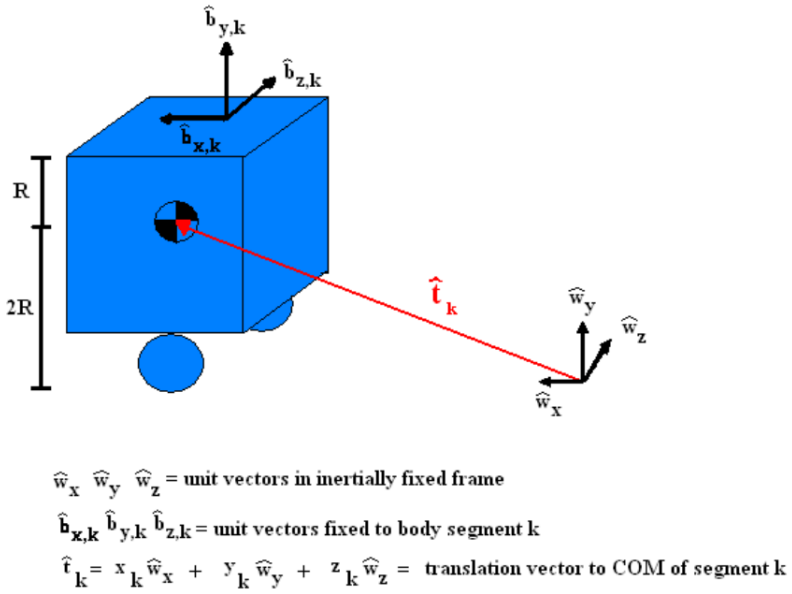


Figure 8: *Constraint Diagram for Typical Segment (Courtesy of [2])*

Equation 7. The ranges of realizable values used in the optimization were inferred from experience with the prototype robot illustrated in Figure 2, and the ranges are further explored in Section 4.5's Table 4.

The genetic algorithm used for this project and its prior work follows a standard system of processes: fitness evaluation, crossing and mutation. The work flow diagram of the genetic algorithm used for this thesis is illustrated in Figure 9. The first step shown in Figure 9

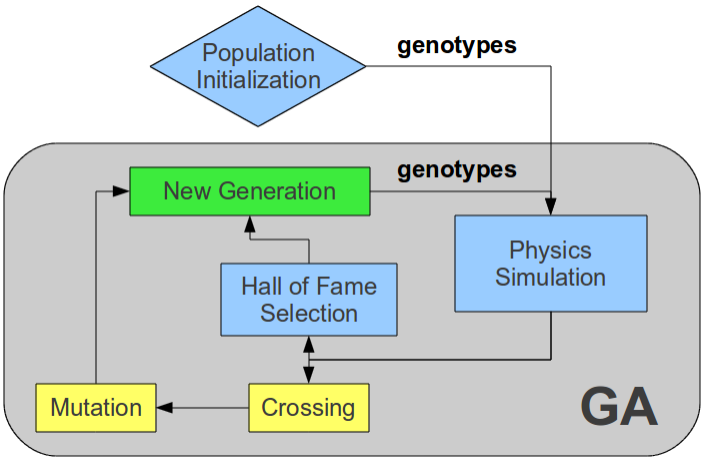


Figure 9: *Constraint Diagram for Typical Segment (Adapted from [2])*

is the initialization of the population at a discrete size [26]. This population size will be maintained throughout the genetic algorithm, and is an important component to the rate of convergence [21]. For all the tests in this project, the population has 100 members. Each member of the population is initialized at this first step with a unique combination of genotype parameters that are each picked randomly within the confines of Equation 7 [27]. The next step is to pass each of the genotypes (population members) into a dynamic simulation utility (described in Section 2.3) and generate a fitness value for each of these genotypes based, in this case, on how far they were able to move in a discrete time [14]. The top ten percent of fitness-ranked genotypes are designated members of the "Hall of Fame", which means they are automatically passed on to the next generation [26].

The next step is to send the remaining members of the population into a stage of genotype crossing and mutation. A roulette selection style is used to determine which population

members will undergo genotype crossing (also known as breeding) and which members will undergo mutation [27]. The roulette method favors the ability of genotypes with higher fitness values to pass their genes onto the next generation, which helps ensure less than optimal designs are driven out of the population over time. Crossing occurs by randomly selecting a cut location (element of the genotype) and swapping all genotype elements following this cut with the severed genotype elements of some other chosen member of the population. Every population member chosen for crossing is paired with only one other population member for the breeding step. The remaining members of the population undergo mutation, in which ten percent of each member's genotype elements are assigned new random values within the confines of Equation 7. The crossed and mutated genotypes then join the "Hall of Fame" genotypes in the new generation. The whole process then repeats itself until a specified number of generations have been generated [27]. Typically the simulation is stopped when it seems that that genetic algorithm is converging to a set of models with a certain performance, which is indicated by a lack of major advancements being introduced after many generations.

Chapter 3
3D Model Generation

3.1 Introduction

Chapter 3 outlines the approach taken in this thesis to convert simulated models into realizable 3D CAD designs. The design approach utilized in this thesis attempts to tackle the difficulties of working with soft materials described in Section 1.2 by leveraging the tools described in Chapter 2. Due to the interdependence of controls and morphology for soft robots, this thesis, along the lines of prior research[2], chose to optimize approximate system dynamics with a lumped parameter model prior to building a detailed three-dimensional CAD design.

This proposed dynamics-based design approach introduces a transformation that exchanges the difficult problem of coordinating the soft robot's degrees-of-freedom for three easier problems. These three easier problems consist of converting the design concept into a lumped parameter model, optimizing the structural and control parameters associated with that model, and finally transforming the lumped parameter model into a fully resolved three-dimensional CAD design. A flow chart for the proposed dynamics-based design method is illustrated in Figure 10.

Conceptually, this transformation approach is not unlike the Eigenvalue transformation which can be used to simplify the solution for a set of coupled ordinary differential equations (ODEs). The Eigenvalue transformation also exchanges a difficult problem - solving a set of coupled ODEs - for three easier problems, which include a forward matrix transformation into a space in which the ODEs are decoupled, the solution of those decoupled ODEs, and the subsequent reverse transformation of the ODE solutions to recover the original set of system variables.

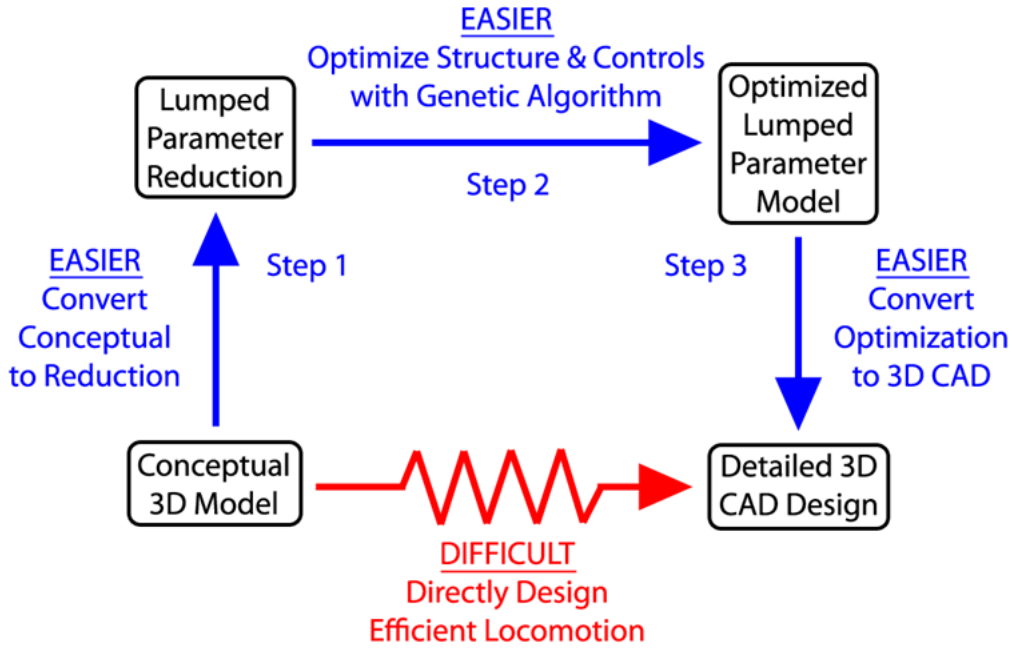


Figure 10: *Dynamics-based Design Approach*

In the lumped parameter domain, the number of structural degrees of freedom is significantly reduced. The lumped parameter model incorporates only a small subset of desirable degrees of freedom, which abstract the soft robot's flexibility and compressibility. Consequently, the problem of optimally coordinating the robot's morphology and control parameters is concentrated to a relatively low-dimensional search space. The lumped parameter model described in Section 2.2, allowed for a more manageable search space for locomotion optimization, which was achieved using the genetic algorithm approach described in Section 2.4, in conjunction with the dynamic simulation utilities described in Section 2.3. The genetic-algorithm approach was able to simultaneously evolve morphology and control parameters to maximize the distance traveled by the robot in a fixed interval of time.

In simplifying the coordination of the robot's degrees of freedom, the proposed dynamics-based design approach introduces the new problem of converting between the lumped parameter and three-dimensional CAD representations of the robot. The process of converting between these two representations is a "projection" process, in which only the most important details of the fully resolved CAD design appear in the lumped parameter model. In this sense, multiple solutions exist for converting the optimized lumped parameter model into a three-dimensional CAD design, since this process must generate missing information that was not essential for simulation of the system's dynamics.

In this thesis, I propose a three-step process that sub-divides the last step of the dynamics-based design procedure illustrated in Figure 10 into three new steps. This approach leverages the strengths of both design automation and human-centric design and provides guidance on how to jump from "Optimized Lumped Parameter Model" to a "Detailed 3D CAD Design". In the first sub-step, related elements of the lumped parameter model are grouped together. This step reduces the number of three-dimensional components that must be designed. In a second sub-step, an automated design method to develop rough three-dimensional models for each element grouping is implemented with optimization criteria. This procedure leverages the ability of computer automation to compute rough shapes (such as rods and prisms) that embody the desired physics for each element grouping. In a third sub-step, a human designer smoothes and joins the rough three-dimensional designs to create a manufacturable CAD representation of the full robot. Generating a smooth design, with fewer stress concentrations or corners that might snag in the field, can be a challenging task for computer automation, but a simpler one for a human designer. The three sub-steps are summarized in Figure 11.

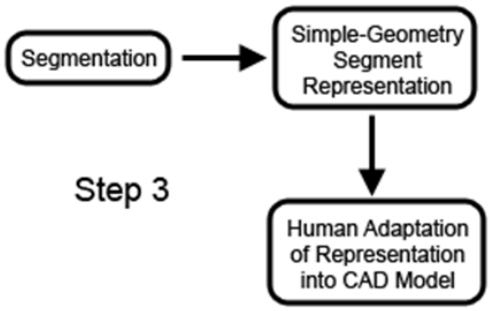


Figure 11: *Semi-Automated Conversion Process*

3.2 Abstract Element Grouping

For the case of the caterpillar robot, it is natural to group together "minor" elements into "major" elements or "segments," as described in Section 2.2. The connections between major segments only simulate the stiffness properties experienced between body segments, but minor segments are introduced so that their connection, in a dynamic solver, can represent the dynamical parameters within the segmental bodies themselves. However, the proposed conversion process assumes the soft-robot body can be constructed as a union of smaller three-dimensional components. In general, slicing a soft structure into arbitrary components does not ensure that those components can be analyzed in a decoupled manner, since the distortion of the overall structure may result from interactions among neighboring components. As an example, it may not be possible to conduct an accurate finite-element analysis for a single component, since boundary conditions may depend on the deformation of neighboring components. The justification of designing the full robot structure as a union of decoupled components requires additional investigation. In particular, it is desirable for the stiffness parameters for the lumped-element model to be faithfully recreated by the refined three-dimensional design. To investigate this property of our soft material, we looked at an example case of a four-segment robot with uniform segments.

It was hypothesized that, at least for caterpillar-like soft robots, it is possible to obtain the desired high-level stiffness properties (e.g. stiffness for the entire body) by applying design automation at the segment level. To justify this hypothesis, at least in a linearized sense, calculations were performed to show that the full-body stiffness predicted by a finite-element analysis matched that of a theoretical calculation for a series of decoupled components, modeled as springs in series. The following is the equation for the stiffness of a set of springs in series [28]. The stiffness of the set of N springs k_{total} depends on the stiffness of each individual spring, k_i.

$$k_{total}(N) = \left(\sum_{i=1}^{N} \frac{1}{k_i} \right)^{-1} \tag{8}$$

For comparison to this theoretical result as applied to our lumped parameter model, a series of finite-element analyses were performed using CosmosWorks^{TM}. The analyses were performed on two representative segment shapes (with elliptical and rectangular cross-sections as shown in Figure 12). For the purpose of this validation analysis, several different body lengths were considered, including integer lengths of one, two, three and four segments. The four-segment geometries are illustrated in Figure 13. In each finite element simulation, the end points of the soft structures were constrained to lie on a plane.

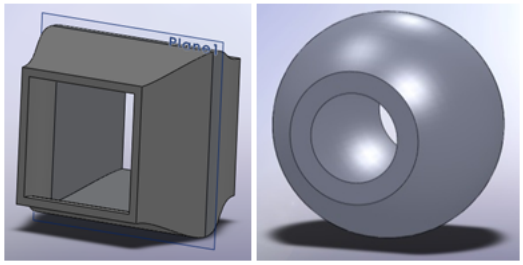


Figure 12: *Segmental Base Shapes*

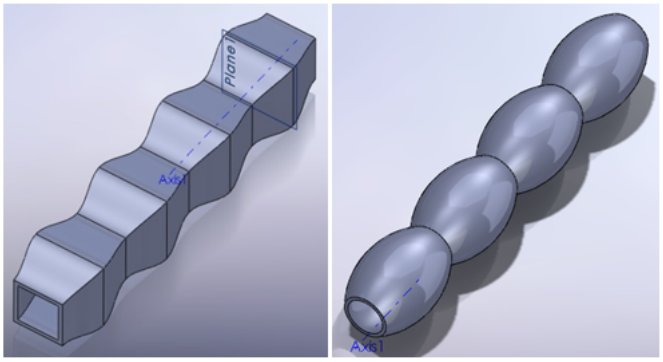


Figure 13: *Conceptual 3-D CAD Designs*

To relate the theoretical result for the springs in series to the finite element result, k_i was first computed, both for segments of square and elliptical cross-section. To do this the body stiffness k_{total} was computed for the M-segment geometries (such as those of Figure 13) and then the inverse of Equation 8 was applied to compute the segment k_i values, which were assumed equal for all segments, resulting in Equation 9, where $N = M \cdot L$, M is the total number of body segments (for instance, four in Figure 13), L is the number uniform subdivisions in each segment, N is the total number of subdivisions made in the entire body.

$$k_i = k_{total}\frac{N}{i} \tag{9}$$

Equation 9 was subsequently applied to compute a theoretical curve for stiffness as a function of body length. For example, the stiffness of 1.4 segments of a M-segment model can be determined with $L = 5$, and $i = 7$. The impact of boundary condition variations can be assessed by identifying at what point the theoretical springs-in-series curve diverges from the Finite Element Analysis (FEA) results. For the case of perfect decoupling, the theoretical curve should not diverge from the FEA data.

When the axial stiffnesses for the full-body analyses are graphed, it is seen that these stiffnesses are very close to theoretical values, as depicted in Figure 14, for the elliptical case. The bending stiffness, however, was not entirely accurate and diverged for the single segment case, as seen in Figure 15. This divergence indicates that decoupling is not perfect. Although the error between the FEA data and the theoretical curve is relatively small (less than 20%), it seemed most reasonable to not automate the design of bending stiffness, leaving the process of tuning bending stiffness to the human designer in the final design stage.

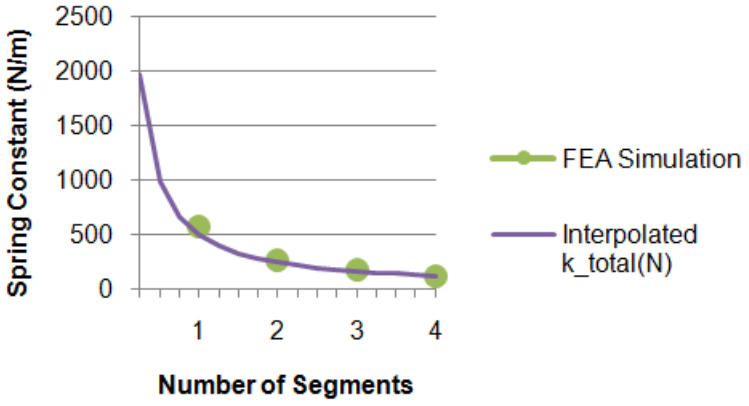


Figure 14: *Axial Stiffness for Elliptical Segments*

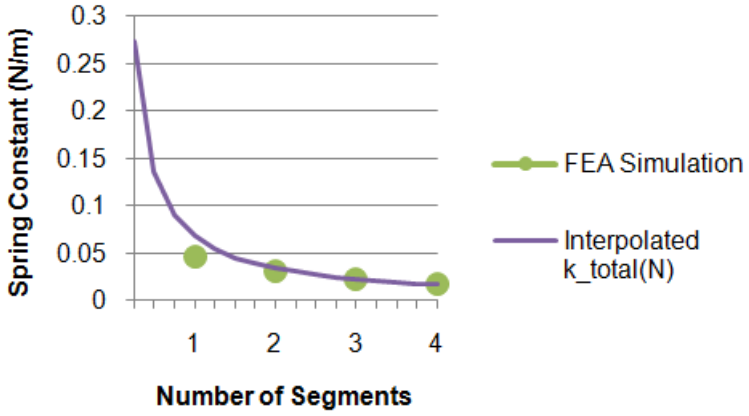


Figure 15: *Bending Stiffness for Elliptical Segments*

3.3 Rectangular-prism Deconstruction with Optimization Routine

This section introduces and optimizes an intermediate modeling step to streamline the conversion of the dynamical simulation results into a detailed design. This step is beneficial because it helps overcome the challenges encountered with nonlinear coupling among dynamical parameters (including mass, center of mass, contact points, moments of inertia, and component stiffness requirements), by simplifying the geometry to allow for eased human-centric manipulation of the design. A problem introduced by this type of nonlinear coupling is the difficulty in developing simple rules for iterative design refinement, since design decisions affect many dynamical parameters at the same time. The proposed approach intends to address this issue by enabling designers to quickly define and test parameterized shapes to assess whether or not they satisfy dynamical requirements prior to developing a refined CAD design.

Specifically, this approach allows the designer to evaluate and tune component configuration through the use of a rough model constructed solely from rectangular prisms. These prisms may be solid or hollow; hollow prisms are created by subtracting a second prism from the first to create a void. The simple structure of this modeling approach allows for quick model definition, because prisms are easy to visualize and place, and for quick optimization, because prism parameters can be described by straightforward closed form equations amenable to optimization. An illustration of this concept is shown in Figure 16.

For the caterpillar design, three rough models for the shape of a body segment were considered. (All body segments were assumed to be identical, as they were in the genetic algorithm.) The progression of the models illustrates the utility of the prism approach in

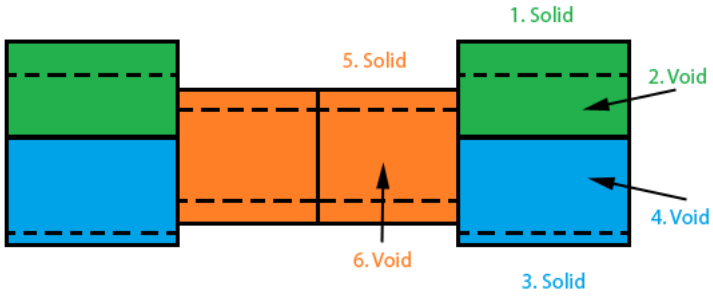


Figure 16: *Diagram of Model for Combining Prisms*

supporting the conversion of a dynamical simulation into a detailed design. The three models (seen in Figure 17) are labeled the simple pyramidal model, the detailed pyramidal model, and the flexible hitch model. Though the basic topology is the same for all models, consisting of a body and two feet, refinements in the prism layout resulted in progressive improvements in the ability of each model to match the dynamical parameter specifications defined in Table 2.

The process began by specifying the first (simple pyramidal) model, which coarsely represents the segment geometries illustrated in Figure 12. The optimized parameters were not able to be matched with this model, so a less constrained (detailed pyramidal) model was then implemented by adding more prisms to the body. Optimization of this model slightly improved parameter values but the detailed pyramidal model still did not match the desired inertia and stiffness values simultaneously. To address this issue a third (flexible hitch) model structure was introduced to reduce coupling between the axial stiffness parameter and the inertia parameters. This third rough design configuration proved very effective at matching the desired dynamical parameters, as indicated by Table 3, discussed later in the chapter.

To demonstrate the method for optimizing a block model, the simple pyramidal model is considered. Figure 18 provides a detailed sketch of the model. The model consists of five hollow prisms (three for the main body plus one for each foot). Each prism is defined with

	Variable	Description	Optimized Value
Form	R_{ga}	Radius of minor segment (units: meters)	0.0143
	L_{ga}	Length of minor segment (units: meters)	0.0193
	m_{ga}	Mass of minor segment (units: kilograms)	0.0103
	w_{ga}	Foot separation	$0.54R_{ga}$
	y_{ga}	Vertical distance from the bottom of the foot to the major segment's center of mass	$2R_{ga}$
	xs_{ga}	SMA attachment point; x distance from center of mass	$-0.08L_{ga}$
	ys_{ga}	SMA attachment point; y distance from center of mass	$0.04R_{ga}$
	zs_{ga}	SMA attachment point; z distance from center of mass	$0.90R_{ga}$
	$I_{xx,ga}$	Inertia along the major segment x-axis, as derived for a solid prism (see Figure 5; units: kg/m^3)	2.81e-6
	$I_{yy,ga}$	Inertia along the major segment y-axis, as derived for a solid prism (see Figure 5; units: kg/m^3)	3.96e-6
	$I_{zz,ga}$	Inertia along the major segment z-axis, as derived for a solid prism (see Figure 5; units: kg/m^3)	3.96e-6
	k_{ga}	Axial Stiffness of minor segment (units: N/m)	304
Control	–	Actuation start time	Six values for each actuator pair
	–	Actuation stop time	

Table 2: *Parameters of An Optimized Design*

a uniform wall thickness, with one exception. To help the designer place the vertical center of mass, the wall thickness of the large central prism was defined to be larger on top than

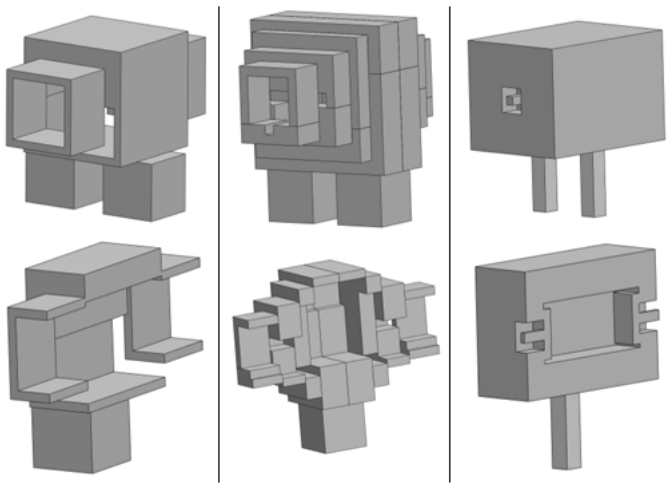


Figure 17: *Simple Pyramidal Model (Left); Detailed Pyramidal Model (Center); Flexible Hitch Model (Right)*

on bottom, as shown in the Front View of Figure 18. The main body was assumed to be a hollow continuous tube capable of holding a payload. Consequently, the prisms of the body were open in the lengthwise direction (along the x-axis).

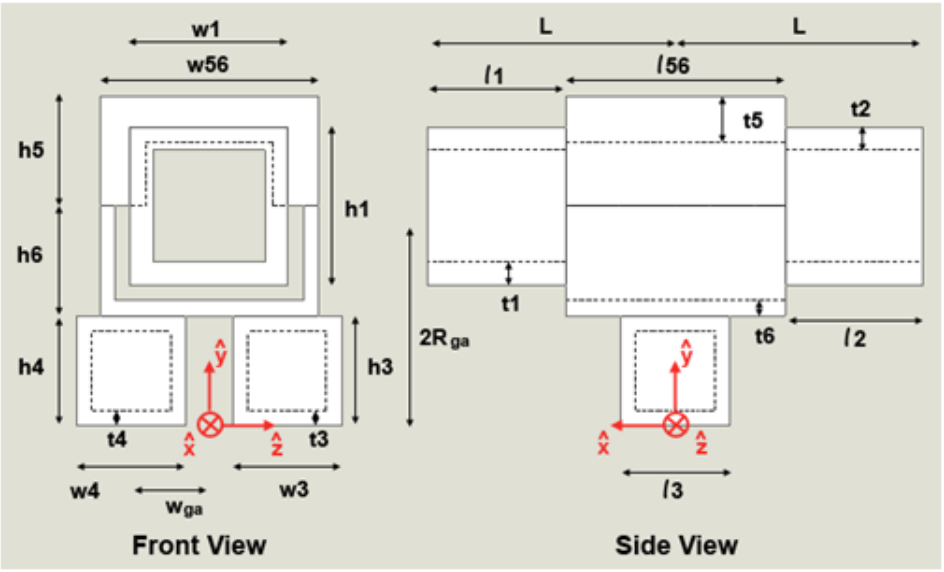


Figure 18: *Detailed Sketch of Simple Pyramidal Model*

An optimization routine was used to drive model parameters as close as possible to their specified values (as summarized in Table 2). Specifically, the optimization tuned five parameters for each prism. The basic parameters for each prism were its edge dimensions and centroid location: height h, width w, length l, x-centroid x_c, y-centroid y_c. Each prism was also defined by two parameters specified by the designer. The first of these was the z-centroid location of the prism z_c, which was set to zero for all blocks except the feet (whose separation distance w_{ga} is defined as a parameter in Table 2). Also, each prism was given a sign s, which was positive for prisms which added additional material to the model and negative for prisms which removed material from the model.

The length of the parameter vector $\boldsymbol{p}$ is $7n$ because there were seven parameters defined for each of the n prisms in the model. The parameter vector has the following form:

$$\begin{aligned}\boldsymbol{p} = [&h_1\ w_1\ l_1\ x_{c,1}\ y_{c,1}\ z_{c,1}\ s_1 \ldots h_j\ w_j\ l_j\ x_{c,j}\ y_{c,j}\ z_{c,j}\ s_j \\ &\ldots h_n\ w_n\ l_n\ x_{c,n}\ y_{c,n}\ z_{c,n}\ s_n]\end{aligned} \tag{10}$$

The optimization problem is formulated to tune this parameter vector (specifically, the first five parameters of each prism were tuned and the other two were set constant) in order to minimize the difference between the rough model and the dynamical simulation. This difference is evaluated by considering a subset of only six of the most highly coupled dynamical parameters: mass, vertical center of mass, axial stiffness, and the principal moments of inertia. The optimization seeks to minimize the cost function, $f(\boldsymbol{p})$, which is defined below as the sum of the relative errors between the rough model parameters and the desired parameter values (indicated by the "ga" subscript). Functions for each of the rough model parameters $m(\boldsymbol{p})$, $y(\boldsymbol{p})$, $k(\boldsymbol{p})$, $I_{xx}(\boldsymbol{p})$, $I_{yy}(\boldsymbol{p})$, $I_{zz}(\boldsymbol{p})$ are described in more detail below.

$$f(\boldsymbol{p}) = \frac{(m(\boldsymbol{p}) - m_{ga})^2}{m_{ga}^2} + \frac{(y(\boldsymbol{p}) - y_{ga})^2}{y_{ga}^2} + \frac{(k(\boldsymbol{p}) - k_{ga})^2}{k_{ga}^2} + \frac{(I_{xx}(\boldsymbol{p}) - I_{xx,ga})^2}{I_{xx,ga}^2} + \frac{(I_{yy}(\boldsymbol{p}) - I_{yy,ga})^2}{I_{yy,ga}^2} + \frac{(I_{zz}(\boldsymbol{p}) - I_{zz,ga})^2}{I_{zz,ga}^2} \tag{11}$$

The optimization is subject to inequality and equality constraints, with a general form given by equations 12 and 13.

$$\boldsymbol{A}\boldsymbol{p} \leq \boldsymbol{b} \tag{12}$$

$$\boldsymbol{A}_{eq}\boldsymbol{p} = \boldsymbol{b}_{eq} \tag{13}$$

The detailed form of these constraints varied for each of the three rough designs considered. In all cases, these constraints included the following:

Equality Constraints

- The sum of the lengths (l_j) of the body segments is constrained by the dynamical simulation (to be $2L_{ga}$).
- Centroid locations and edge dimensions are constrained for prisms with adjacent surfaces.
- Prism lengths (l_j) are constrained to ensure both fore-aft symmetry about the y-z plane, and bilateral symmetry about the x-y plane.

Inequality Relationships

- Wall thicknesses are constrained to exceed 2 mm.
- Prisms representing the segment body are ordered in a sequence of decreasing edge heights (h_j) and widths (w_j), in accordance with distance from the midline.

The optimization cost function depends on closed-form equations for mass, center of mass, axial stiffness, and inertias derived for a union of rectangular prisms. The mass parameter,

for example, is the sum of the masses of the n individual prisms.

$$m(\boldsymbol{p}) = \sum_{j=0}^{n} m_j \tag{14}$$

The mass of an individual prism is volume multiplied by density (ρ), which was set to a representative value for the intended elastomer design materials (DragonSkin™ or a DragonSkin™ foam).

$$m_j = \rho h_j w_j l_j s_j \tag{15}$$

The y-location of the center of mass for a component can be found by summing the contributions of all the individual prisms' to the y-centroid location.

$$y_j = y_{c,j} \tag{16}$$

$$y(\mathbf{p}) = \frac{\left(\sum_{j=0}^{n} m_j y_j\right)}{m} \tag{17}$$

The axial stiffness parameter is computed by applying a series Hooke's Law model in the axial direction. When cross-sectional area is constant, Hooke's Law can be used to define a spring constant k of the form used in the dynamical simulation to represent stiffness (see Table 2).

$$k = \frac{EA}{l} \tag{18}$$

This formula is based on the baseline length of the part l, its cross-sectional area A, and its Young's Modulus E. The last parameter is set based on representative material properties for DragonSkin™. When the cross-sectional area changes in discrete steps along the length

of the component, this formula for spring stiffness must be modified slightly. Specifically, the total spring stiffness for the component can be approximately modeled with the typical springs in series summation. This equation has the following form [28]:

$$k(\boldsymbol{p}) = \left[\sum_{u=1}^{q} \frac{l_u}{EA_u}\right]^{-1} \tag{19}$$

This model is dependent on the number of unique cross-sectional areas (q) perpendicular to the x-axis. The total area of each cross-section is calculated as the sum of the contributions (positive and negative) for each of the c prisms associated with that cross-section.

$$A_u = \sum_{i=1}^{c} h_i w_i s_i \tag{20}$$

The feet are assumed to not contribute to axial stiffness, so their cross-sectional area is not evaluated.

Lastly, the principal inertias are the summation of the individual prism inertias.

$$I_{xx}(\boldsymbol{p}) = \sum_{j=1}^{n} I_{xx,j} \tag{21}$$

$$I_{yy}(\boldsymbol{p}) = \sum_{j=1}^{n} I_{yy,j} \tag{22}$$

$$I_{zz}(\boldsymbol{p}) = \sum_{j=1}^{n} I_{zz,j} \tag{23}$$

Equations for the principal inertia of each prism are based on formulae commonly derived in introductory dynamics textbooks [28], with an appropriate application of the parallel axis theorem.

$$I_{xx,j} = \frac{m_j}{12}\left(h_j^2 + w_j^2\right) + m_j\left[(y_j - y)^2 + zc_j^2\right] \tag{24}$$

$$I_{yy,j} = \frac{m_j}{12}\left(w_j^2 + l_j^2\right) + m_j\left[xc_j^2 + zc_j^2\right] \tag{25}$$

$$I_{zz,j} = \frac{m_j}{12}\left(h_j^2 + l_j^2\right) + m_j\left[xc_j^2 + (y_j - y)^2\right] \tag{26}$$

The optimization problem defined by equations 11 - 26 was solved using MATLAB™ (and specifically using the *fmincon* function), which is designed for solving general nonlinear constrained optimization problems.

3.4 Human Centric Smoothing and Defining of Block Initial Conditions

Sub-step three from Figure 11 relies on the designer to adapt the optimized geometry into a 3D CAD model. In addition to smoothing the simple-geometry segments into clean, continuous shapes (such as those in Figure 13) and connecting these segments together with flexures, it is also necessary to ensure the model matches all dynamical parameters. As noted in the previous section, the optimization only considered the most highly coupled dynamical parameters; thus, matching other parameters, such as bending stiffness and actuator attachment points, is left to this final design step.

To generate a detailed design, a designer drafts 2-D cross sections based on the rough model and lofts these together to form the segment. Specifically, the designer defines planes for each unique cross-section. Some interpolation is done to ensure the amount of material is approximately maintained and the constraining dimensions, such as maximum segment length, are respected. After the lofting process, the designer adds attachment points for

actuators at the locations specified by the genetic algorithm and removes material that interferes with actuator motion. Lastly, the segments are connected at their flexures to form the full model.

In this process which involves human-centric steps, the model's dynamical parameters will change slightly from those of the block model. Therefore, further manipulation of the geometry is needed to bring these dynamic properties back to their target values. An FEA program, such as (CosmosWorks^{TM}) in our examples, can be used to determine the resulting stiffness, inertias, mass and center of mass, and to compare these properties to the respective values determined from the dynamic simulation. The designer must continue to iterate through minor modifications to the segment's morphology to tweak these parameters to better match the dynamical specifications. The final design found in this paper's test-case of the described method, as applied to one caterpillar-like segment based on the parameters from Table 2, is illustrated in Figure 19. The full caterpillar system, consisting of four segments, is illustrated in Figure 20.

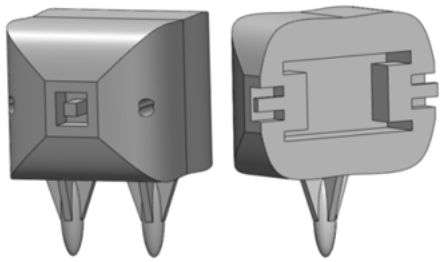


Figure 19: *Smoothed Flexible Hitch Model with Actuator Attachment Points*

A comparison of the dynamical parameters for each design step is summarized in Table 3. The first line of the table lists the specifications generated by the genetic algorithm. The second line lists the parameter values obtained for the prism-based rough model. The third line lists the parameter values that were achieved for the final design illustrated in Figure 19.

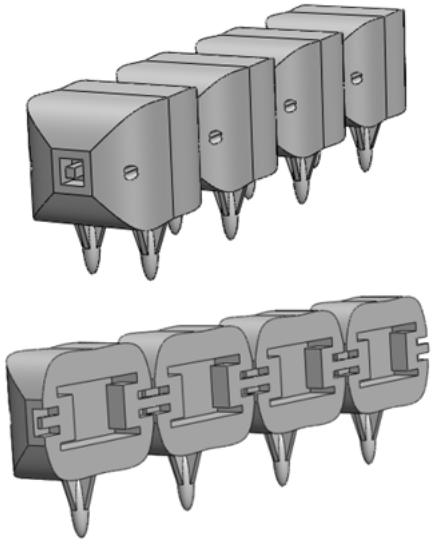


Figure 20: *Four-Segment Flexible Hitch Model*

A percentage parameter-matching error is evaluated for each of the last two lines. This percentage error was computed as a mean of the relative errors for each of the six parameters listed in the table.

	m (g)	**y (mm)**	**k (N/m)**	$\mathbf{I_{xx}}$ **(kg/m²)**	$\mathbf{I_{yy}}$ **(kg/m²)**	$\mathbf{I_{zz}}$ **(kg/m²)**	***Error (%)***
Target	20.6	28.6	152	2.81e-6	3.96e-6	3.96e-6	–
Rough	20.6	28.6	150	2.82e-6	3.99e-6	3.97e-6	*0.5*
Smoothed	20.9	28.7	152	3.17e-6	3.74e-6	3.74e-6	*4.3*

Table 3: *Comparison of Optimized Parameter Values and the Smoothed 3-D Model*

3.5 Discussion

Obtaining a three-dimensional design which closely matched the desired lumped parameter model was challenging. The final model matched the specifications within a mean relative error of 4.3%. In comparison, the mean relative error of an initial model found from a

primitive form of the procedure described in this chapter was 261% including axial stiffness and even 19.7% when ignoring axial stiffness. For the example used in this paper to illustrate the use of the described procedure, the final model's bending stiffness was lower than the target value, but this could be corrected for by adding thin lengthwise flanges to the flexible hitches in future iterations. This modification would increase bending stiffness while having almost no effect on the other parameters.

In part, some of the difficulty in matching dynamical specifications can be traced back to the definition of the conceptual lumped parameter model simulated by the genetic algorithm (Step 1 in Figure 10), which were described in Section 2.2. For example, the conceptual model used by the genetic algorithm assumed the inertia and mass of the feet were negligible. In realizing a three-dimensional design (as shown, for example, in Figure 17), this assumption introduced problems, in that it was very difficult to simultaneously achieve the desired vertical center of mass location and x-inertia. Furthermore, to achieve any solution, the design specification of a long hollow central void, to deliver a payload, had to be abandoned to allow for the implementation of a flexible hitch, which was required to achieve the example's desired stiffness while maintaining the desired inertia. In the next chapter, methods for improving the genetic algorithm by constraining its search space are explored.

Chapter 4

Enhancements Made to Optimization Procedures: Enhancement of Optimization Procedures Through the Addition of Geometric Constraints

4.1 Overview

In this chapter, methods for improving dynamics-based design of a soft caterpillar-like robot space are investigated. An important method explored is the imposition of constraints similar to the geometric definitions laid out in Equations 12–26 on the genetic algorithm to simplify the subsequent process of generating rough three-dimensional designs. It is important to understand and properly implement constraints when using a genetic algorithm to develop designs. This conclusion arose from the discovery of conflicting geometrical requirements for designs due to uncoupled determination of the optimized dynamical parameters. Although a design that closely matched the dynamical parameter requirements was found, it defies some overall design specifications of the project (such as holding a payload and embedding control circuitry) and required many weeks of iterations to be found. In the interest of helping the designer find a realizable model from simulation as quickly as possible, minimizing the impact of these geometrical conflicts was prioritized. Additionally, implementing the geometric definitions laid out in Equations 12–26 in the genetic algorithm, integrates sub-steps one and two from Figure 11 into Step 2 of Figure 10, leaving only sub-step three to the designer.

4.2 Introduction

One possible reason that conflicts between geometrical constraints and dynamical parameters arose in outputs of the genetic algorithm is that there is no common origin for these parameters. In other words, there's no assurance that any design could be found with all possible combinations of parameter values. Instead, each parameter (mass, center of mass, axial stiffness, bending stiffness, and the mass moments of inertia) were all found randomly and combined without thought as to their physical relations. For example, it was impossible to achieve the optimized mass and rotational inertia because even a solid block (which minimized rotational inertia for a given mass) had too high of a rotational inertia. Furthermore, a hollow prism is the desired shape to allow for an embedded payload in the caterpillar-like robot, so clearly the parameters from Table 2 are unacceptable for our desired design, and we showed earlier that only a design that violated the specification of a hollow interior could be used to achieve a model with parameters close to those that were optimized.

4.3 Background

As mentioned in Section 3.5, it is challenging to obtain a three-dimensional design which closely matches the desired parameters, such as those in Table 2. Although a model was found that matched the specifications within a mean relative error of 4.3% (Table 3), some design specifications were sacrificed, such as a hollow inside capable of delivering payloads and improved bending stiffness. It seems clear that the difficulty in matching dynamical specifications was due to the manner in which the conceptual lumped parameter model simulation was defined by the genetic algorithm. A new means for defining the genotypes of the genetic algorithm, in order to ensure realizable designs, is clearly a much needed

improvement to the Dynamics-based Design Approach.

4.4 Morphology-based Lumped Element Constraints

This new approach sought to optimize geometrical parameters (such as length, width, height and relative offsets of solid or hollow blocks), rather than dynamical parameters (such as mass, inertias and stiffnesses). Furthermore, the random generation of these geometrical parameters were constrained based on an overall design concept to ensure that only realizable designs are chosen. The dynamical parameters were then calculated based on these random geometries, using Equations like 14–26 and then these parameters were supplied to the dynamic simulation for determination of genetic fitness. This method ensured that at least one physically realizable design could obtain these dynamical parameters. Additionally, the optimization technique described in Section 3.3 can be used to uncover other unique geometries that also match these realizable set of dynamical parameters.

The use of a genetic algorithm, specifically, introduces difficulties in maintaining geometrical constraints. Many of the geometrical properties are interrelated. For example, a void prism cannot exceed the width of its corresponding solid prism (plus some wall thickness), but both values are subjected to independently random changes from crossing and mutation that might violate this intuitive constraint. The crossing stage of the genetic algorithm may therefore lead to genomes that describe abstract morphologies that are physically unrealizable. Consequently, an additional constraint check was implemented after each of the crossing and mutation stages. At this additional step the modified parameters are checked to ensure that they fit within the geometrical constraints for the specific genome. When a violation of constraints is encountered, a constrained mutation is performed on the gene to randomly generate an element that is within the allowable geometrical constraints of the

particular model. For a full list of the interrelated geometrical constraints added to the genetic algorithm, see Appendix A.

4.5 Search Volume Comparisons: Experimental Verification of Effects

It was hypothesized that preventing the dynamical parameters from spanning their entire search space by constraining how they are found (i.e. from evolved geometries) would limit the number of designs found in this search space. To begin to understand what fraction of the search space the designs spanned, a quantitative method for describing this fraction was desired. A process was undertaken in which a seven-dimensional volume (mass, center of mass, axial stiffness, bending stiffness, xx-Inertia, yy-Inertia and zz-Inertia) was evaluated with each axis representing a dynamical parameter and the dimensions of the volume spanning from the minimum to the maximum allowed values of that parameter. The genetic algorithm already constrained the search space of the dynamical parameters between minimum and maximum values (Table 4); the process began by defining that volume. However, some parameters were set as constant, so these parameters were expanded based on experience with the values garnered from previous design iterations ("Interpolated" in Table 4).

Given this *super-volume* of all possible dynamic property combinations with each dynamic parameter explored entirely from its minimum to maximum values (Table 4), the question remained as to what fraction would be occupied by the possible search space of the morphology-based designs. The newly constrained search space can be visualized as a blob within the super-volume. Given that defining the search space volume of morphology-based designs would be mathematically infeasible due to non-linear relations between the geometric

		GA	Interpolated
Mass (kg)	Min	0.007	0.007
	Max	0.013	0.013
CoM (m)	Min	0.0320	0.0320
	Max	0.0374	0.0374
Axial Stiffness (N/m)	Min	304	100
	Max		500
Bending Stiffness (N·m)	Min	0.047	0.020
	Max		0.080
xx-Inertia ($kg \cdot m^2$)	Min	1.19e-6	1.19e-6
	Max	3.03e-6	3.03e-6
yy-Inertia ($kg \cdot m^2$)	Min	8.20e-7	8.20e-7
	Max	2.00e-6	2.00e-6
zz-Inertia ($kg \cdot m^2$)	Min	8.20e-7	8.20e-7
	Max	2.00e-6	2.00e-6

Table 4: *Minimums and Maximums for the Dimensions of the 7D Search Space*

parameters in dynamical parameter calculations, the blob's volume was explored by producing random samples. An example of a two-dimensional representation of the super-volume and *sample-based design volume* (as defined by random samples) is seen in Figure 21. In this figure, the dashed box represents the super-volume, the red dots are the random samples, and the green area represents the design volume.

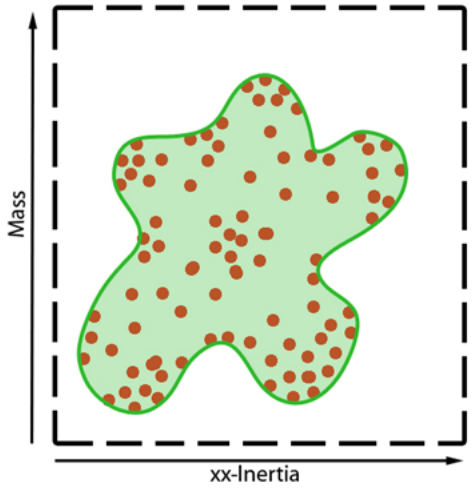


Figure 21: *2D Representation of the Super-Volume and the Sample-Based Design Volume*

With a large number of dimensions, it's not a trivial task to calculate what percentage of the super-volume is filled by the design volume, subsequently referred to as the *volume fraction*. One method used was to divide the super-volume into a discreet number of 7-dimensional sub-volumes that are referred to as bins. A two-dimensional representation can be seen in Figure 22. These bins are formed by dividing each axis into the same number of equal-length segments (in terms of the min-to-max range fraction for that parameter), and then evaluating every possible combination of unique segments. As a result there are $(\#\text{ of segments per axis})^{(\#\text{ of dynamical parameters})}$ total bins.

After determining how many random samples fall into each bin, the volume fraction is then found by dividing the number of bins with more than zero samples by the total number of bins in the super-volume. A depiction of the counting process can be seen in Figure 23, though all the computation was actually done using MATLAB^{TM}. Because this is a discrete problem, any sample will only fall into one bin, despite the fact that in this figure some samples span multiple bins. Upon performing this analysis with the volume

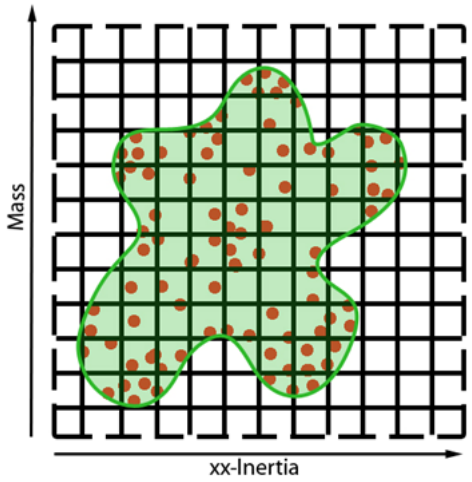


Figure 22: *2D Representation of the Super-Volume Divided into Bins*

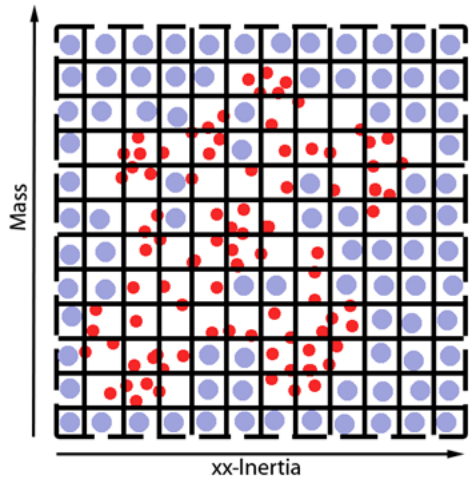


Figure 23: *Volume Fraction Found by Counting Number of Bins with More Than Zero Samples (Empty Bins Shown in Blue)*

defined by the minimum and maximum parameter values used in the genetic algorithm, it was found that zero bins contained samples in them. This result was also checked by running the optimization from the prior work with the geometrical constraints for a set number of samples and a set number of optimization iterations, and the program was asked to report how many samples fell into any bins. The result was again that of zero filled bins.

There are two possible explanations for this result (referred to as scenario A and scenario B), as depicted in Figure 24. In scenario A, the genetic algorithm search space is any size volume that does not overlap the design search space. In scenario B, the genetic algorithm search space does overlap the design search space, but is such a small volume that it's statistically improbable that a sample would fall within its boundaries. The inverse case, where the design volume is very small compared to the genetic algorithm search space, is not possible because the volume fraction would be small, but must be inside at least one bin and therefore can not explain the results.

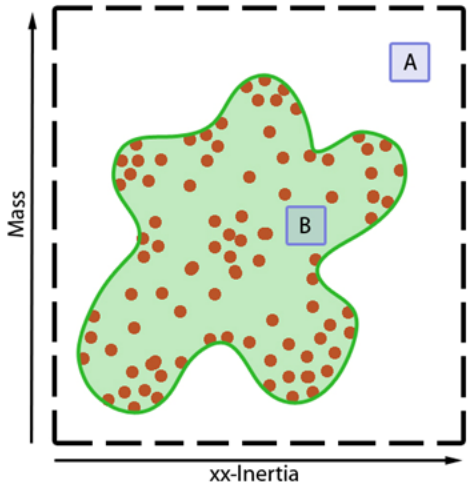


Figure 24: *Depiction of Scenarios for 0% Volume Fraction*

A solution to Scenario A is evaluating the search space of multiple designs, so that a larger total volume is explored, as seen in Figure 25. A solution to Scenario B is to explore a different search space. The motivation for determining the volume fraction was to get some information about the shape of the design volume. Therefore, the understanding of this volume can still be improved by determining what fraction the blob-volume occupies of a super-volume defined by the global maximum and minimum values for each parameter. These global values were found by manipulating the geometric parameters for each design,

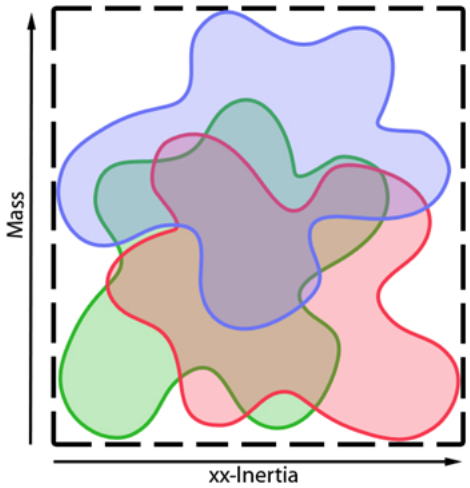


Figure 25: *Multiple Designs Help to Explore a Larger Fraction of the Super-Volume*

within their respective constraints, to calculate the maximum and minimum possible values for each of the dynamic parameters. Evaluating the volume fraction of each design within the span of the maximum and minimum possible values known to be obtainable from the constrained equations will help further understanding of the effect of the constraints on the evaluated search space.

One observation was that a significant number of samples were required in order to properly define the design volumes. To gain an understanding of how many samples are needed to feel confident that the data exhaustively represents the global design volume, the maximum and minimum parameter values obtained for logarithmically increasing sets of random samples were explored. As the number of samples grows, and the chances of having larger maximum or smaller minimum value increases, the bounding box of the design volume grows, as seen in Figure 26. This growth was graphed to see how many samples were necessary to approach the global minimums and maximums. The entire compilation of these graphical results can be seen in Appendix B. Additionally, a comparison of the global maximum and minimum parameter values with the maximum and minimum parameter

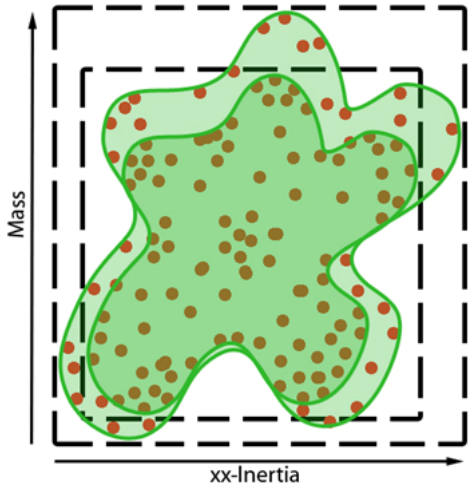


Figure 26: *Depiction of Effects on Defined Super-Volume Due to Increased Sampling*

values obtained with random sampling can be seen in Table 5. The three designs used for testing were the previously discussed Simple Pyramid, Detailed Pyramid and Flexible Hitch models (see Section 3.3). Furthermore, this table shows that the original dynamical parameter ranges were a subset of the global ranges, implying that Scenario B was the cause of the 0% volume faction.

		Interpol.	Global 1	Samp 1	Global 2	Samp 2	Global 3	Samp 3
Mass (kg)	Min	0.007	8.44e-5	1.2e-3	9.12e-5	2.0e-3	2.3e-3	6.5e-3
	Max	0.013	31.9184	23.8207	31.9184	23.1022	47.2960	37.9538
CoM (m)	Min	0.0320	1.02e-3	2.37e-3	1.02e-3	2.14e-3	2.30e-3	5.90e-3
	Max	0.0374	0.3452	0.3160	0.3923	0.3303	0.3000	0.2987
Axial Stiffness (N/m)	Min	100	4.1429	5.3594	4.1429	5.9610	4.1988	4.4765
	Max	500	869913	628881	869913	454931	243870	286714
Bending Stiffness (N·m)	Min	0.020	1.04e-6	5.65e-6	1.04e-6	1.74e-6	4.20e-6	4.59e-6
	Max	0.080	6.69e3	4.19e3	2.18e3	1.53e3	2.97e3	1.85e3
xx-Inertia ($kg \cdot m^2$)	Min	1.19e-6	1.06e-10	8.86e-8	1.27e-10	6.49e-8	1.37e-7	9.92e-7
	Max	3.03e-6	0.6517	0.4418	0.6517	0.4237	0.8583	0.5673
yy-Inertia ($kg \cdot m^2$)	Min	8.20e-7	2.56e-9	2.89e-8	2.58e-9	1.92e-6	9.18e-8	3.08e-7
	Max	2.00e-6	0.6953	0.4592	0.6953	0.4493	2.2948	1.6631
zz-Inertia ($kg \cdot m^2$)	Min	8.20e-7	2.61e-9	9.99e-8	2.62e-9	5.15e-8	9.33e-8	5.53e-7
	Max	2.00e-6	0.8948	0.5622	0.8948	0.5647	2.5981	1.7434

Table 5: *Comparison of Minimum and Maximum Parameter Values for All Three Designs*

As can be seen in Appendix B, all the parameters for all three designs appear to come close to approaching their respective global maximum values in less than 10^7 samples. However, only the stiffness parameters seem to come close to approaching their global minimum values, but interestingly this is the case for all three designs. It should be also noted that both axes on these plots are in log format, so the numerical difference between the sample and global minimums is actually less than the difference between sample and global maximums. If the y axis is not plotted logarithmically, the opposite conclusion, that the minimums converge, but the maximums do not, is reached. An example is shown in Figure 27 and the percent difference of the range for every parameter and all three designs can be seen in Table 6. These results indicate that at least 10^6 samples are required to accurately represent the global design volume. It is important to find the correct number of required samples to properly represent the volume to ensure that the least required computation time is used.

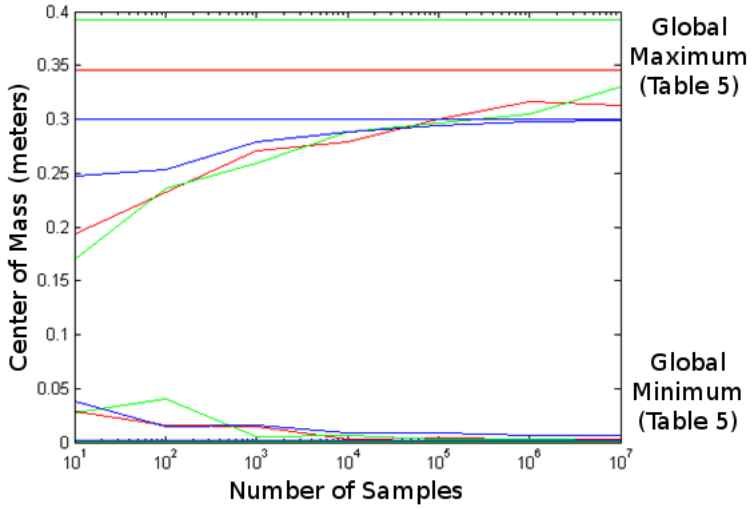


Figure 27: *Example Semi-Log Plot: Center of Mass for All Three Designs (Red = Simple Pyramid; Green = Detailed Pyramid; Blue = Flexible Hitch)*

Based on these findings, 10^6 samples could be used to determine the volume fraction for the super-volumes based on global minimum and maximum parameter values for this

		Simple Pyramid		Detailed Pyramid		Flexible Hitch	
		%	Samples	%	Samples	%	Samples
Mass	min %	3.50e-3	10^7	5.98e-3	10^7	8.87e-3	10^7
	max %	25.4	10^7	27.6	10^7	19.8	10^6
Center of Mass	min %	0.393	10^7	0.285	10^7	1.21	10^7
	max %	8.49	10^6	15.8	10^7	0.446	10^7
Axial Stiffness	min %	1.40e-4	10^7	2.09e-4	10^7	1.14e-4	10^7
	max %	27.7	10^6	47.7	10^7	-17.6	10^7
Bending Siffness	min %	6.89e-8	10^7	3.23e-8	10^7	1.31e-8	10^7
	max %	37.4	10^7	29.6	10^7	37.7	10^7
xx-Inertia	min %	1.36e-5	10^7	9.94e-6	10^7	9.96e-5	10^7
	max %	32.2	10^5	35.0	10^7	33.9	10^7
yy-Inertia	min %	3.78e-6	10^7	2.39e-6	10^7	9.42e-6	10^7
	max %	34.0	10^5	35.4	10^7	27.5	10^7
zz-Inertia	min %	1.09e-5	10^7	5.47e-6	10^6	1.77e-5	10^7
	max %	37.2	10^7	36.9	10^7	32.9	10^7

Table 6: *Percent Difference of Sample Min/Max from Global Min/Max*

particular search volume. To see the effects of having too few samples, the volume fractions for 10^4 and 10^5 samples were also explored. All of this data can be seen in Table 7. From this table it seems as though a significantly large percentage of the super volume is represented even with only one design, such as the Flexible Hitch model or the Detailed Pyramid model. This means that any dynamic parameter combinations for the second or third design should result in a realizable model, and that the probability of finding a realizable model with all the designs is high when dynamic parameters are randomly combined. However, this insight into the volume fraction question does not answer the question of how many samples are in each bin. It's very possible that most of the bins are filled in the second design, for example, but only a small percentage of the bins themselves have samples in them, decreasing the probability of finding a realizable sample within a specific bin. To gain a better understanding of how important constraints on the dynamic parameters are, it is

important to evaluate the density of samples in each bin.

Design	Samples		
	10^4	10^5	10^6
Simple Pyramid	0.09	0.42	0.57
Detailed Pyramid	0.05	0.13	0.88
Flexible Hitch	0.21	0.56	0.75

Table 7: *Volume Fraction for All Three Designs with 8 Bins per Axis*

Before considering bin density, however, it's important to understand the effect of using different number of bins per axis. It is rather obvious that having too few bins could skew the data. For example, as seen in Figure 28, the difference between using two bins per axis versus four bins per axis can have a dramatic effect on the volume fraction. Therefore, a quick analysis of different numbers of bins on a small sample set was performed, as seen in Table 8.

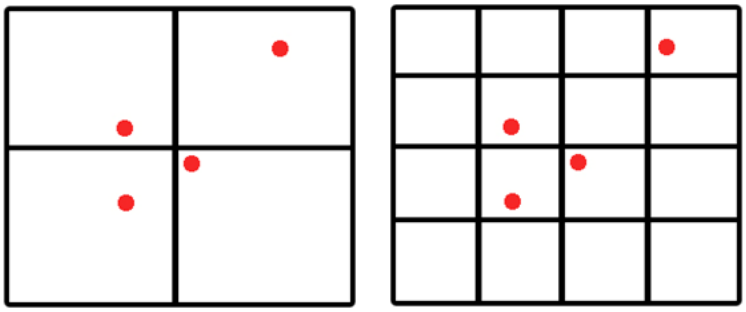


Figure 28: *Effects of Bin Concentration*
(Left: 2 Bins, 100% Volume Fraction; Right: 4 Bins, 25% Volume Fraction)

As can be seen, a large number of bins is necessary to gain a good understanding of the design volume's presence within the super-volume.

To determine the density of each bin, the number of samples that occur in each bin were recorded, rather than just recording a binary full or empty value, as before. As seen in a

Design	Bins		
	4	8	12
Simple Pyramid	0.75	0.57	0.51
Detailed Pyramid	1.00	0.88	0.58
Flexible Hitch	1.00	0.75	0.76

Table 8: *Effects of Bin Concentration on Volume Fraction for All Designs with* 10^6 *samples*

2D example on the left of Figure 29, you can just as easily count the number of samples. The right side of Figure 29 shows how this data can be used to gain an understanding of the actual shape of the design volume. This test was performed on a varying number of bins with

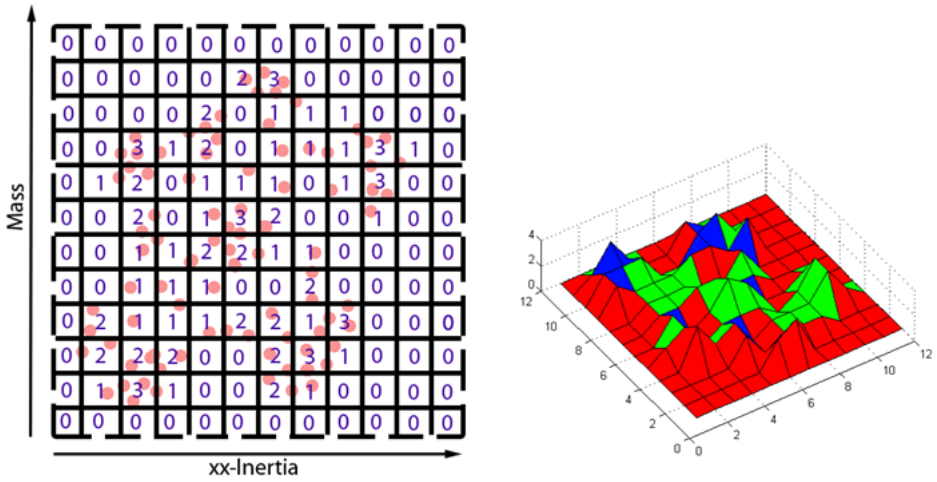


Figure 29: *Determining the Density of Samples Throughout the Super-Volume*

three different sample sizes. Table 9 is an overview of the data collected. Rather than use the global minimum and maximum to define the volume fraction, these data indicate that defining the super-volume on the minimum and maximum values of all the samples would better depict the density of where the samples are falling relative to one another. The bin fraction represents what fraction of the total number of bins had samples in them, for each parameter, and Table 9 shows the average for each design's set of seven parameters for all

nine tests.

Bins	**Samples**	Simple Pyramid		Detailed Pyramid		Flexible Hitch	
		Average Bin Fraction	Volume Fraction	Average Bin Fraction	Volume Fraction	Average Bin Fraction	Volume Fraction
4	10^4	0.7857	0.1780	0.0000	0.0000	0.0000	0.0000
	10^5	0.0000	0.0000	0.9286	0.5625	0.0000	0.0000
	10^6	0.9643	0.7500	1.0000	1.0000	1.0000	1.0000
8	10^4	0.7321	0.0890	0.6607	0.0451	0.8214	0.2094
	10^5	0.8929	0.4187	0.7679	0.1346	0.9286	0.5625
	10^6	0.9286	0.5742	0.9821	0.8750	0.9643	0.7500
12	10^4	0.6310	0.0297	0.6071	0.0186	0.7024	0.0720
	10^5	0.7738	0.1432	0.7500	0.1208	0.8929	0.4279
	10^6	0.9167	0.5135	0.9286	0.5777	0.9643	0.7639

Table 9: *Overview of Bin Density Data*

4.6 Discussion

One interesting observation is that there was only one dense region per parameter for all designs. This means that the samples did not cluster in one area closer to the minimum and another distinct area near the maximum. The fact that multiple peaks per parameter never occurred suggests that all the samples are clustering in a single area. Furthermore, in almost all cases, the number of samples was largest near the minimum and slowly declined as the bins approached the maximum. Even when the first bin did not have the largest number of samples, the dense region started on the first bin due to a significant number of samples existing within it. As for the effect on numbers of bins and samples, an increase in samples, with bins held constant, increases the volume fraction, while an increase in bins, with samples held constant, leads to a decrease in volume fraction. This indicates that the sample distribution is not uniform across the dense regions and that an increasing number

of samples increases the probability of more bins being filled. This analysis on the design volume seems to indicate that constraining the dynamic parameters has little effect on what possible designs are found. Therefore, the modifications to the optimization procedures discussed in this chapter produce a significant benefit to the modeling methods outlined in chapter 3.

Chapter 5
Experimental Prototype

5.1 Overview

In this chapter, investigations will be made into the reliability of the new procedural methods outlined in chapter 4. In addition to finding simulated models, this chapter will also explore the construction of a prototype and compare its dynamics to that of a similar robot prototype. Additionally, comparisons are made between the new simulation data to data of a competitive prototype, but some lessons are learned about the difficulty of ensuring a simulation matches reality.

5.2 Introduction

After all the modifications that were made to the genetic algorithm, it was prudent to see if models could be found that surpassed the fitness of prior models. It would also be important to see if any negative consequences were observed due to the more extensive constraint algorithms, and if models produced were indeed physically realizable. Furthermore it was desired to have an actual prototype built so that real-world dynamics could be compared to those of the simulations.

5.3 Background

Although the design tools described in Chapter 2 were focused on evolving the morphology and gait patterns for a soft robot that would use SMA actuators, it was decided to modify

the code to account for actuation using pneumatic pistons instead, in the interest of prototype work. At the time, little success had yet been achieved converting simulations of SMA actuated soft robots into realizable prototypes that moved in accordance to their simulated gaits, even within reasonable expectations. Several problems in dealing with SMA attachment and effective actuation had not yet been resolved by other teams working on similar efforts. However, a pneumatically actuated soft robot (hereafter referred to as Pneumabot) had been constructed (seen in Figure 3, and shown again in Figure 30 unattached to its actuation platform) and was moving according to predicted gaits, within reasonable margins of errors. Full examination of this robot's gait patterns was later published in a work by Saunders, et al. [29], and those results were used here for benchmarking purposes.

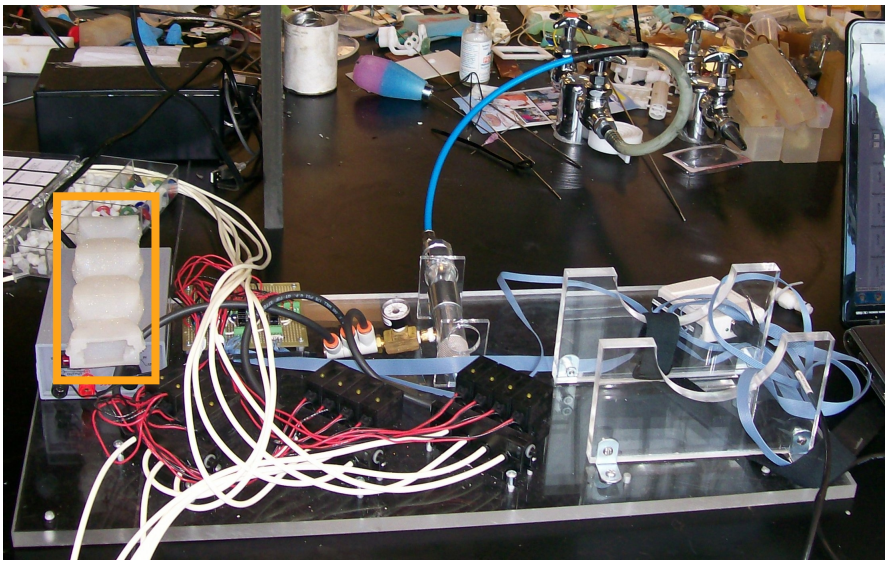


Figure 30: *Pneumabot (Orange Box) and Its Actuation Setup*

Therefore, in hopes of having a baseline to compare the performance of a newly evolved prototype, models were sought that would be physically similar to the Pneumabot (three major flexure-based segments, rather than four, and no feet) and that would be simulated

with actuation patterns mimicking those of the same pneumatic pistons. The dynamic simulation code was modified to accept the force application mechanisms of pneumatic actuators, and the genotype of the genetic algorithm, and input to the dynamics simulator, were also modified to account for the different morphology and actuation parameters. For further illustrative purposes, Figure 31 shows an example of the lumped model used for simulation in the code up to this point (also seen in Figure 4), whereas Figure 32 shows an example of the lumped model used in simulations to find a pneumatically actuated soft robot.

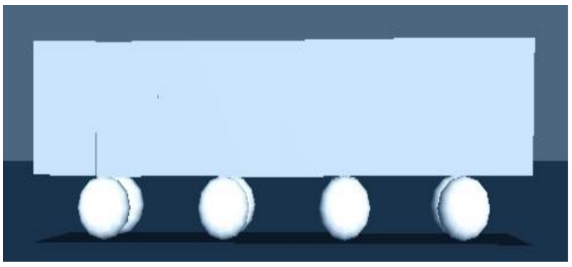


Figure 31: *Simulated Model Used to Generate Optimized SMA Actuated Model*

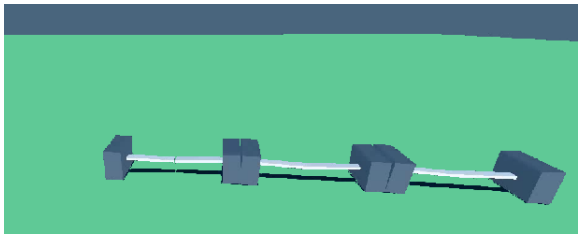


Figure 32: *PhysX© Simulation Used to Generate Optimized Pneumatically Actuated Model*

5.4 Identified Models with Improved Simulated Performance

After evaluating the simulated performance of models found from several generations of genetic algorithm runs, several models with exceptionally high fitnesses were identified. The computational runs went through several iterations due to trouble encountered with the "skin width" property of PhysX© allowing certain models to build up energy and then jump forward a large distance. These models resulted in a false positive for the distance test because their dynamics would not be repeatable in the physical world due to the exploitation of skin-width anomalies. Five models with realistic dynamics, described in Table 10, were confidently identified. Also in this table are the specifications for the two gaits that were tested on the Pneumabot prototype [29]. Of the five new models in this table, Design D5

Design	**Time (sec/GP)**	**Distance (cm/GP)**	**Velocity (cm/sec)**	**Max Height (cm)**	**Max Width (cm)**	**Segment Length (cm)**
D1	9.15	4.04	0.44	4.0	3.0	11.9
D2	2.33	2.12	0.91	7.0	1.1	6.8
D3	1.12	1.26	1.12	3.7	11.4	5.0
D4	9.22	4.70	0.51	9.8	12.8	21.1
D5	7.78	9.97	1.28	2.6	8.8	4.6
Pneu1	1.62	0.65	0.40	2.5	5.1	6.4
Pneu2	0.83	0.43	0.52	2.5	5.1	6.4

Table 10: *Five Promising Designs*

has the best overall fitness value, and one that was 2.5 times better, at least in simulation, than the best prior gait found for the Pneumabot. Another interesting fact about Design D5 is that its physical parameters were the closest match to those of the Pneumabot in comparison to the other four simulated models.

5.5 Development of Fully Realizable CAD model

Upon finding a desirable model, the first step was to convert the lumped prismatic design into a smoothed CAD model, as described in Section 3.4. Using those techniques a more realizable physical model was achieved, as illustrated in Figure 33. As discussed in Section 3.4,

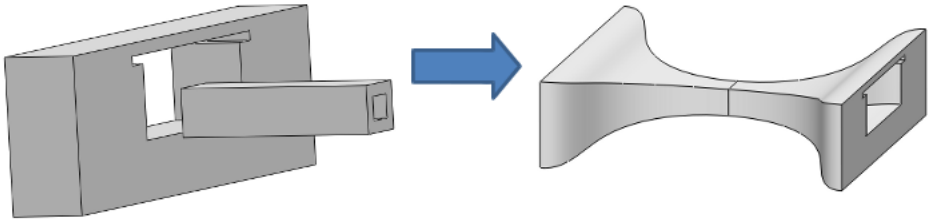


Figure 33: *Conversion of Rough CAD Model to Smoothed CAD Model of Chosen Design (D5)*

certain errors are introduced into the dynamical properties of the model during this process. However, the unique talents of human iterative capabilities were successfully leveraged to further tweak the design and reduce those errors, much like those described in Section 3.4. Table 11 shows the specific values achieved for each step of the process, and how the errors were reduced with proper application of the smoothing design techniques.

	Mass (g)	**CoM$_x$ (mm)**	**CoM$_y$ (mm)**	**I$_{xx}$ (kg·m^2)**	**I$_{yy}$ (kg·m^2)**	**I$_{zz}$ (kg·m^2)**	**Axial Stiffness (N/m)**	**Bending Stiffness (N/m)**	**Error (%)**
Simulation	22.6	11.9	8.0	1.8e-5	1.8e-5	2.6e-6	708.4	0.015	
Smoothed	22.1	12.2	9.6	1.2e-5	1.3e-5	2.8e-6	956.6	0.018	
Error (%)	*2.1*	*2.6*	*19.4*	*31.4*	*26.9*	*6.85*	*35.0*	*22.9*	*18.4*
Tweaked	19.0	12.1	8.9	1.1e-5	1.1e-5	2.4e-6	668.9	0.015	
Error (%)	*15.9*	*1.31*	*10.7*	*40.4*	*36.1*	*5.9*	*5.6*	*1.9*	*14.7*

Table 11: *Error Due to Smoothing, and Subsequent Error Reduction*

With a realizable CAD design with reasonable dynamic properties achieved (shown in Figure 34), rather than simply idealistic simulated models, it is possible to build a physical prototype of the model.

5.6 Fabrication and Testing of Fully Realized CAD Model

Just as the Pneumabot is constructed with DragonSkin™, that design parameter was held constant by building the new prototype from DragonSkin™ as well. Actually, DragonSkin™ is the elastomer of choice for most of the soft robots being designed, so its physical properties were already those used in the simulations for the SMA and pneumatic actuated bots, as was discussed in Section 3.3. To build the design from DragonSkin™, molds such as those shown in Figure 35 were generated from the inverse of the smoothed design in Figure 34 that were then constructed from plastic using a commercial-grade 3D printer. Due to the rather intricate inner-geometry of the design, the top and bottom halves of the robot were molded separately. This would allow for painting the molds with liquid DragonSkin™ and forcing excess out through escape holes in the mold, rather than injecting a larger mold with a separate center that would later need to be removed. Due to the tiny necks of the design, as seen in Figure 34, there was concern that the thicker innards of the plastic body mold, could be slid out through these thin areas without damaging the prototype.

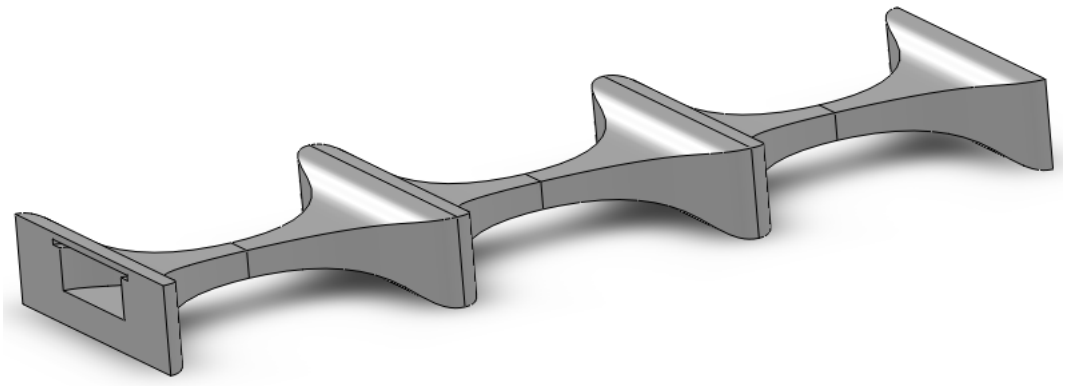


Figure 34: *CAD Drawing of Full Design for Prototype*

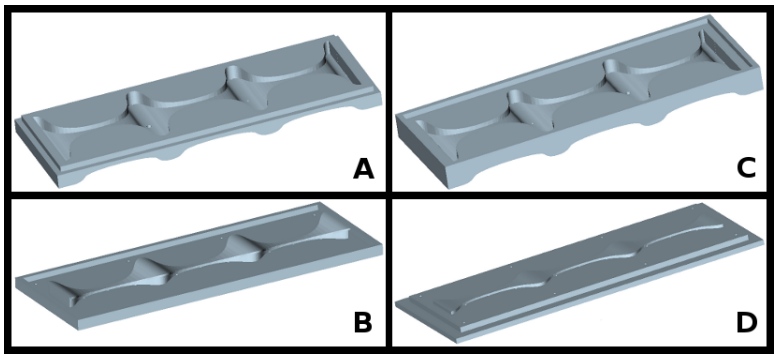


Figure 35: *CAD Drawings of Mold that was 3D Printed to Create Prototype from* DragonSkin™
(A: Top Half of Mold for Top Section of Prototype;
B: Bottom Half of Mold for Top Section of Prototype;
C: Top Half of Mold for Bottom Section of Prototype;
D: Bottom Half of Mold for Bottom Section of Prototype)

DragonSkin™ casting was created using the following procedural steps. First of all, DragonSkin™ is generally only in stable liquid form at room temperature as two separate substances, A and B. Once A and B are combined at room temperature, they begin to form into the solid elastomer known as DragonSkin™. However, mixing these two very viscous liquids, A and B, together to form a uniform mixture inevitably introduces bubbles, that will become air pockets in the solid elastomer and change the properties of the DragonSkin™ to values different than those of a pure DragonSkin™ body wall, which is what is assumed in the simulations to be true. So, to ensure the properties of the DragonSkin™ body are uniform in the prototype, an additional step is included to the mixture process. After substances A and B are mixed thoroughly, they are put in a vacuum chamber and exposed to decreasing air pressures until all the air bubbles have been sucked from the liquid DragonSkin™.

Finally, the top and bottom halves for each mold are painted with the viscous DragonSkin™ liquid while trying to ensure all nooks and crannies are well covered, and then the top and bottom halves of the mold are squeezed together, allowing all excess material to ooze out of the cleverly placed escape holes, designed to ensure that no air pockets form in the mold

that the liquid DragonSkin^{TM} would not otherwise reach, like outer edges. The top and bottom halves are then clamped together until the DragonSkin^{TM} solidifies. When the top and bottom halves of the prototype are thoroughly dried (about 24 hours later), they are carefully removed from the molds and then put together using a strong adhesive substance. The model is then punctured at the evolved actuation attachment points, and the pneumatic cylinders are inserted into place. The resulting completed prototype is see in Figure 36.

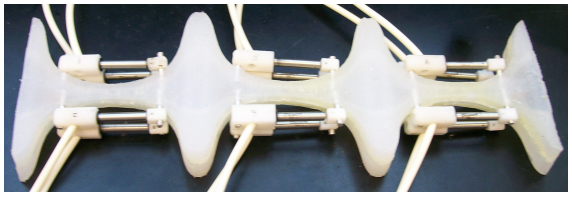


Figure 36: *DragonSkin^{TM} Prototype*

5.7 Experimental Testing of the Prototype

The control platform used is the same as that which was used for the Pneumabot, seen in Figure 30. This setup uses a basic control box to either turn on or off air flow to any of the twelve specified pneumatic actuators as it is told to do so. The software interacting with this control box is a program written in LabView^{TM} that takes in a gait period (seconds) and then at what point, as a fraction of gait period, each actuator is turned on (sent air so it extends to its maximum length) and turned off (removed from constant air flow so it retracts to its minimum length). For example, if you wanted pneumatic piston 5 to start the gait period as off, then turn on one fifth of the way through the gait period and then be off for the last quarter of the gait period, you would enter 0.2 into the "on" field, and 0.75 into the "off" field. The gaits evolved for Design D5 were entered into the LabView^{TM} program and then the robot's motions were observed. A still of the resulting video is shown

in Figure 37, in which you can clearly see some actuators on, and others off. The distance traveled from a starting point defined as the edge of a floor tile was continually examined to determine if improvements in mobility had been made over previous prototypes.

Figure 37: *DragonSkin™ Prototype in Motion*

5.8 Discussion

The simulated dynamics of Design D5 were based on a relatively unique motion in which the front of the soft robot would lift itself up and smack back down to create a slight bouncing wave that propagated throughout the rest of the body. As it repeated this action, it was able to move quite significant distances. However, the prototype was unable to mimic this motion in reality. Through observation, it was clear that the robot was attempting to lift up its front half and would then smack it down, but the lift was significantly less than in simulation and the slight-bounce wave observed in simulation never propagated through the rest of the prototype's body.

The hypothesis for this failing was an oversight in the modifications made to the dynamic simulator for the evolutions run that found this morphology and gait, when transferring the code from using SMA actuators to pneumatic actuators. While SMA actuators are negligibly light, and therefore only modeled as applied point forces in the dynamic simulation, the pneumatic actuators are actually a significant portion of the weight of the soft robot. While

modifications were made to the dynamic simulator to account for the differences in applied forces and timing with a switch from using SMA actuators to pneumatic actuators, additional code was never written to add the weight of these actuators to the physical model. Clearly, these additional objects in the code, even as just additional weight if not also introducing motion constraints, would dramatically affect the dynamics of any model simulated with the PhysX© simulation utility. It seems that this is a likely reason for seeing far less than ideal movement in the actual prototype.

Chapter 6
Conclusions

This thesis outlined the approaches taken to develop the contributions outlined in Section 1.3 and analyzed their effectiveness on each other and their potential benefits to the scientific community as a whole, in as much as they are interested in developing soft robots for Urban Search and Rescue missions, or similar tasks. The first contribution, as described in Chapter 3, identified a methodology to convert lumped parameter models into a detailed three-dimensional designs using a three step process. It was also shown empirically, for a caterpillar-like robots geometry, that the axial stiffness parameters for individual segments could be analyzed in a manner decoupled from the full body. This contribution also identified an automated method to create a rough model for individual caterpillar segments, composed of a set of rectangular prisms, which enabled rapid model development and optimization to ensure close matching of the physical characteristics of the lumped parameter model. The first contribution also outlined how the unique talents of a human designer could be used to convert lumped prismatic structures into smooth segment designs. However, finding equations that accurately predict the stiffness behavior of flexure-to-flexure segments, as well as sub-segments, should be further investigated in future work beyond what was discussed in Section 3.2, to allow for better evolution of stiffness properties.

Furthermore, the work done for the first contribution also identified problems with how constraints were set up in the genetic algorithm used to find optimized lumped parameter models. This discovery necessitated the work done in the second contribution (Chapter 4) to find a new method for constraining the search space, while ensuring that search space was not constrained to a point of impeding the discovery of unique and valid designs. While the new constraints certainly helped ensure that a realizable model could be evolved that

was physically realizable and met certain design specifications, such as a hollow mid-section for payload delivery, there remain concerns about the new search space. Although a lot of analysis was performed on the modifications to the search space that did quell some worries about its potential over-constraining nature, the new constraints result in a very complex search space that is only really discovered when a large number of bins are used. Given the potential benefits of ensuring only realizable designs are proposed in simulation, it is desirable to use the proposed constraints. However, further work must be done to analyze what effect these constraints have on the fitness values.

Finally, in Chapter 5 there were lessons learned about the difficulty in ensuring that a simulation is well defined in order to gain expected dynamics from a prototype. While the methods for developing the prototype described in both contributions were well executed, and the construction of the prototype was carried out effectively, further work needs to be done to redefine the code for the physical simulation utility.

Despite a less than desirable resulting prototype gait, the author still feels that unique contributions were made that will help further ease the development of soft robots. To the best knowledge of the author, everything proposed as unique work has yet to be documented by other teams in the literature on these subjects, and so the author hopes this report will help others further their own research and find effective solutions for the dire problems posed from disasters requiring efficient Urban Search and Rescue.

Appendix A

Full List of Geometric Constraints

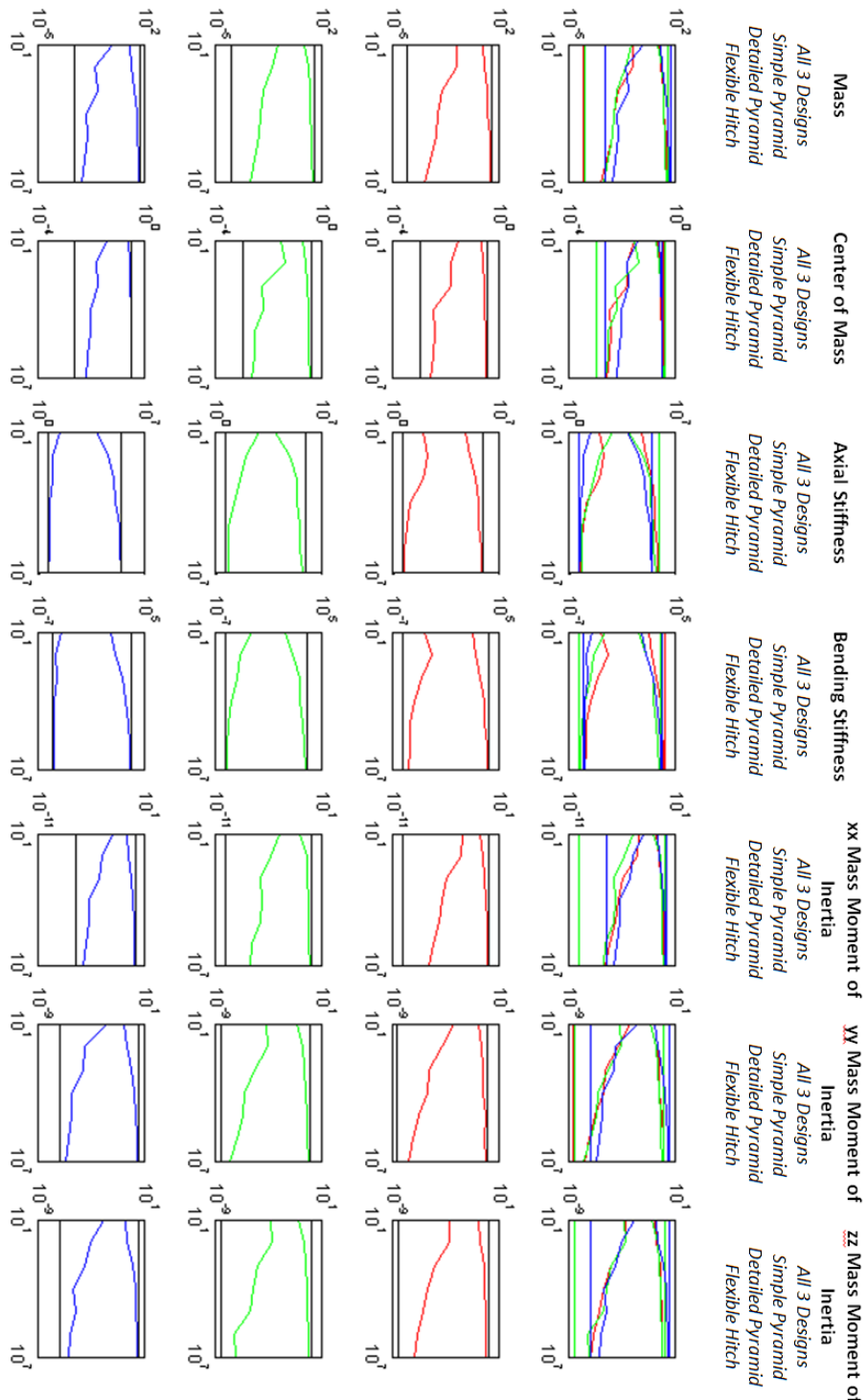
Parameter	Minimum (meters)	Maximum (meters)
segLength	0.027	0.227
midSolidBottomHeight	0.003	0.100
midSolidTopHeight	0.003	0.100
midSolidWidth	0.006	0.200
midVoidBottomHeight	0.001	(midSolidBottomHeight - 0.002)
midVoidTopHeight	0.001	(midSolidTopHeight - 0.002)
midVoidBottomWidth	0.002	(midSolidWidth - 0.004)
midVoidTopWidth	0.002	(midSolidWidth - 0.004)
midLength	(segLength - 0.026)	(segLength - 0.002)
endCapSolidHeight	0.006	(midSolidBottomHeight + midSolidTopHeight)
endCapSolidWidth	0.006	midSolidWidth
endCapVoidHeight	0.002	(endCapSolidHeight - 0.004)
endCapVoidWidth	0.002	(endCapSolidWidth - 0.004)

Descriptions of Parameters

segLength	Length for Full Segment
midSolidBottomHeight	Height for Bottom Mid-Section Prism (Solid)
midSolidTopHeight	Height for Top Mid-Section Prism (Solid)
midSolidWidth	Width for Mid-Section Prism (Solid)
midVoidBottomHeight	Height for Bottom Mid-Section Prism (Void)
midVoidTopHeight	Height for Top Mid-Section Prism (Void)
midVoidBottomWidth	Width for Bottom Mid-Section Prism (Void)
midVoidTopWidth	Width for Top Mid-Section Prism (Void)
midLength	Length for Mid-Section Prism (Solid & Void)
endCapSolidHeight	Height for End-Cap Prism (Solid)
endCapSolidWidth	Width for End-Cap Prism (Solid)
endCapVoidHeight	Height for End-Cap Prism (Void)
endCapVoidWidth	Width for End-Cap Prism (Void)

Appendix B

Global Min/Max Convergence



PERCENT DIFFERENCE OF SAMPLE MIN/MAX FROM GLOBAL MIN/MAX FOR INCREASING SAMPLES OF THE SIMPLE PYRAMID DESIGN

# Samples =>		10^1	10^2	10^3	10^4	10^5	10^6	10^7
Mass	min %	0.453	0.457	5.67e-2	2.78e-2	1.89e-2	8.54e-3	3.50e-3
	max %	76.5	65.9	49.3	47.7	26.4	31.1	25.4
Center of Mass	min %	8.04	4.21	3.81	0.632	0.735	0.644	0.393
	max %	44.2	32.9	21.6	19.3	13.1	8.49	9.41
Axial Stiffness	min %	1.10e-2	1.88e-2	1.03e-2	8.94e-4	3.73e-4	1.80e-4	1.40e-4
	max %	93.6	86.2	67.3	53.5	48.3	27.7	28.1
Bending Stiffness	min %	4.95e-6	2.98e-5	2.75e-06	2.86e-7	9.53e-8	8.29e-8	6.89e-8
	max %	98.7	97.5	90.5	73.6	69.1	44.5	37.4
xx-Inertia	min %	9.64e-2	6.16e-2	1.56e-3	4.89e-4	2.06e-4	4.39e-5	1.36e-5
	max %	90.1	79.5	70.0	63.7	32.2	42.6	32.9
yy-Inertia	min %	6.97e-2	4.98e-3	3.22e-4	2.19e-4	3.81e-5	9.59e-6	3.78e-6
	max %	86.3	72.9	64.1	56.6	34.0	40.3	34.7
zz-Inertia	min %	2.03e-2	2.17e-2	8.13e-4	1.81e-4	8.22e-5	2.09e-5	1.09e-5
	max %	88.9	83.5	58.8	63.9	38.7	42.9	37.2

PERCENT DIFFERENCE OF SAMPLE MIN/MAX FROM GLOBAL MIN/MAX FOR INCREASING SAMPLES OF THE DETAILED PYRAMID DESIGN

# Samples =>		10^1	10^2	10^3	10^4	10^5	10^6	10^7
Mass	min %	0.380	0.163	3.85e-2	2.79e-2	1.92e-2	1.08e-2	5.98e-3
	max %	82.5	60.3	53.7	48.2	42.0	38.7	27.6
Center of Mass	min %	6.82	10.1	1.16	1.35	0.489	0.429	0.285
	max %	56.9	39.9	34.0	26.6	24.6	22.3	15.8
Axial Stiffness	min %	8.13e-2	1.01e-2	3.82e-3	1.28e-3	3.05e-4	2.35e-4	2.09e-4
	max %	99.1	91.8	77.8	74.9	65.9	64.2	47.7
Bending Stiffness	min %	5.14e-5	2.79e-6	1.25e-6	2.34e-7	8.97e-8	3.73e-8	3.23e-8
	max %	99.7	98.1	87.2	86.3	76.1	56.7	29.6
xx-Inertia	min %	2.34e-2	2.32e-3	1.67e-4	2.60e-4	1.73e-4	1.43e-5	9.94e-6
	max %	95.1	74.1	69.2	54.5	56.5	49.6	35.0
yy-Inertia	min %	8.64e-3	1.12	6.22e-4	7.25e-5	4.75e-5	1.26e-5	2.39e-6
	max %	94.3	77.6	69.2	62.1	46.1	47.5	35.4
zz-Inertia	min %	1.31e-2	2.45e-2	8.57e-4	3.53e-4	1.96e-4	5.47e-6	7.88e-6
	max %	94.1	80.0	69.6	53.0	55.3	52.0	36.9

PERCENT DIFFERENCE OF SAMPLE MIN/MAX FROM GLOBAL MIN/MAX FOR INCREASING SAMPLES OF THE FLEXIBLE-HITCH DESIGN

# Samples =>		10^1	10^2	10^3	10^4	10^5	10^6	10^7
Mass	min %	1.72	0.111	0.171	2.50e-2	3.03e-2	1.60e-2	8.87e-3
	max %	80.6	74.0	55.5	42.0	35.8	19.8	24.4
Center of Mass	min %	12.2	4.06	4.49	2.03	2.16	1.39	1.21
	max %	17.5	15.6	7.09	3.75	2.10	0.747	0.446
Axial Stiffness	min %	8.55e-3	155e-3	1.72e-3	5.96e-4	3.06e-4	2.59e-4	1.14e-4
	max %	97.4	85.1	63.2	51.6	18.5	8.32	-17.6
Bending Stiffness	min %	1.26e-6	1.57e-7	2.14e-7	7.45e-8	3.46e-8	3.60e-8	1.31e-8
	max %	99.5	98.6	90.1	77.4	56.3	42.4	37.7
xx-Inertia	min %	0.283	1.37e-2	7.88e-3	4.37e-4	5.25e-4	2.50e-4	9.96e-5
	max %	91.4	86.3	72.5	52.6	40.6	42.6	33.9
yy-Inertia	min %	0.116	9.73e-4	7.33e-4	4.45e-5	4.27e-5	2.40e-5	9.42e-6
	max %	96.1	91.6	79.4	55.9	47.9	30.3	27.5
zz-Inertia	min %	5.09e-2	3.10e-3	8.14e-4	5.51e-5	1.09e-4	3.39e-5	1.77e-5
	max %	95.2	91.5	76.5	54.4	51.1	34.4	32.9

PERCENT DIFFERENCE OF SAMPLE MIN/MAX FROM GLOBAL MIN/MAX IN SUMMARY FOR ALL THREE DESIGNS

		Simple Pyramid		Detailed Pyramid		Flexible Hitch	
		%	Samples	%	Samples	%	Samples
Mass	min %	3.50e-3	10^7	5.98e-3	10^7	8.87e-3	10^7
	max %	25.4	10^7	27.6	10^7	19.8	10^6
Center of Mass	min %	0.393	10^7	0.285	10^7	1.21	10^7
	max %	8.49	10^6	15.8	10^7	0.446	10^7
Axial Stiffness	min %	1.40e-4	10^7	2.09e-4	10^7	1.14e-4	10^7
	max %	27.7	10^6	47.7	10^7	-17.6	10^7
Bending Stiffness	min %	6.89e-8	10^7	3.23e-8	10^7	1.31e-8	10^7
	max %	37.4	10^7	29.6	10^7	37.7	10^7
xx-Inertia	min %	1.36e-5	10^7	9.94e-6	10^7	9.96e-5	10^7
	max %	32.2	10^5	35.0	10^7	33.9	10^7
yy-Inertia	min %	3.78e-6	10^7	2.39e-6	10^7	9.42e-6	10^7
	max %	34.0	10^5	35.4	10^7	27.5	10^7
zz-Inertia	min %	1.09e-5	10^7	5.47e-6	10^6	1.77e-5	10^7
	max %	37.2	10^7	36.9	10^7	32.9	10^7

Bibliography

[1] R. Murphy, "Marsupial and shape-shifting robots for urban search and rescue," *Intelligent Systems and their Applications*, 2000.

[2] F. Saunders, "Simulating soft bodied robots: Methods, benefits, and potential applications," masters thesis, Tufts University, 2009.

[3] M. Hakozaki, K. Nakamura, and H. Shinoda, "Telemetric artificial skin for soft robot," in *Transducers '99*, pp. 1042–1045, 1999.

[4] T. Noritsugu, D. Sasaki, and M. Takaiwa, "Application of artificial pneumatic rubber muscles to a human friendly robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2188–2193, 2003.

[5] J. Clark, J. Cham, S. Bailey, E. Froehlich, P. Nahata, R. Full, and M. Cutosky, "Biomimetic design and fabrication of a hexapedal running robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3643–3649, 2001.

[6] J. Yu, L. Wang, and M. Tan, "A framework for bio-mimetic robot fish's design and its realization," in *Proceedings of American Control Conference*, vol. 3, pp. 1593–1598, 2005.

[7] B. Trimmer, A. Takesian, B. Sweet, C. Rogers, D. Hake, and D. Rogers, "Caterpillar locomotion: A new model for soft-bodied climbing and burrowing robots," in *7th International Symposium on Technology and the Mine Problem*, 2006.

[8] B. Trimmer, "New challenges in biorobotics: Incorporating soft tissue into control systems," *Applied Bionics and Biomechanics*, 2008.

[9] B. Jones and I. Walker, "Practical kinematics for real-time implementation of continuum robots," *IEEE Transactions on Robotics*, vol. 22, pp. 1087–1099, 2006.

[10] Y. Sugiyama and S. Hirai, "Crawling and jumping by a deformable robot," *The International Journal of Robotics Research*, vol. 25, pp. 603–620, 2006.

[11] M. Otake, M. Inaba, and H. Inoue, "Development of a gel robot made of electro-active polymer pamps gel," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 788–793, 1999.

[12] M. Otake, Y. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Inverse kinematics of gel robots made of electro-active polymer gel," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3224–3229, 2002.

[13] J. Cuttino, B. van Dijck, and A. Brown, "Design and development of a toroidal flexure for extended motion applications," *Precision Engineering*, vol. 29, pp. 135–145, 2005.

[14] I. Tanev, T. Ray, and A. Buller, "Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot," *IEEE Transactions on Robotics*, vol. 21, pp. 632–645, 2005.

[15] A. Dobrolyubov, "The mechanism of locomotion of some terrestrial animals by traveling waves of deformation," *Journal of Theoretical Biology*, vol. 119, pp. 457–466, 1986.

[16] J. Keller and M. Falkovitz, "Crawling of worms," *Journal of Theoretical Biology*, vol. 104, pp. 417–442, 1983.

[17] S. Ma, N. Tadokoro, B. Li, and K. Inoue, "Analysis of creeping locomotion of a snake robot on a slope," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2073–2078, 2003.

[18] L. Chen, S. Ma, Y. Wang, B. Li, and D. Duan, "Design and modeling of a snake robot in traveling wave locomotion," *Mechanism and Machine Theory*, vol. 42, pp. 1632–1642, 2007.

[19] I. Walker, D. Dawson, T. Flash, F. Grasso, R. Hanlon, B. Hochner, W. Kier, C. Pagano, C. Rahn, and Q. Zhang, "Continuum robot arms inspired by cephalopods," in *Proceedings of SPIE Conference on Unmanned Ground Vehicle Technology VII*, pp. 303–314, 2005.

[20] G. Debunne, M. Desbrun, M.-P. Cani, and A. Barr, "Adaptive simulation of soft bodies in real-time," in *Computer Animation 2000 Proceedings*, pp. 15–20, 2000.

[21] F. Saunders, J. Rieffel, and J. Rife, "A method of accelerating convergence for genetic algorithms evolving morphological and control parameters for a biomimetic robot," in *Proceedings of IEEE International Conference on Autonomous Robots and Agents*, pp. 155–160, 2009.

[22] H. Lipson, "Evolutionary robotics and open-ended design automation," in *Biomimetics*, pp. 129–156, CRC Press, 2005.

[23] T. Zang, M. Hemsch, M. Hilburger, S. Kenny, J. Luckring, P. Maghami, S. Padula, and W. Stroud, "Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles," in *NASA/TM-2002-211462 technical report series*, Langley Research Center, 2002.

[24] H. Lipson and J. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, pp. 974–978, 2000.

[25] D. Orvosh and L. Davis, "Using a genetic algorithm to optimize problems with feasibility constraints," in *Proceedings of the First IEEE Conference on Evolutionary Computation: IEEE World Congress on Computational Intelligence*, vol. 2, pp. 548–553, 1994.

[26] P. Fleming and R. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol. 10, p. 12231241, 2002.

[27] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. The MIT Press, 2000.

[28] R. Norton, *Design of Machinery*. Boston, Mass.: McGraw-Hill, 3rd ed., 2004.

[29] F. Saunders, E. Golden, R. White, and J. Rife, "Experimental verification of soft-robot gaits evolved using a lumped dynamic model," *Robotica*, 2011.