

**Adaptive Changes in the Morphology of Natural Materials Due to
Alterations in Mechanical Stress**

by

Katharine I. Hsu

BACHELOR OF SCIENCE IN CIVIL AND ENVIRONMENTAL ENGINEERING
UNIVERSITY OF CALIFORNIA, LOS ANGELES, 1995

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN CIVIL AND ENVIRONMENTAL ENGINEERING
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1997

© 1997 Massachusetts Institute of Technology.
All rights reserved.

Author _____

Department of Civil and Environmental Engineering
January 19, 1997

Certified By _____

Lorna J. Gibson
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted By _____

Joseph M. Sussman
Professor of Civil and Environmental Engineering
Chairman, Departmental Committee on Graduate Studies

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JAN 29 1997

Eng.

LIBRARIES

Adaptive Changes in the Morphology of Natural Materials Due to Alterations in Mechanical Stress

by

Katharine I. Hsu

Submitted to the Department of Civil and Environmental Engineering

On January 19, 1997

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Civil and Environmental Engineering

Abstract

Natural materials have shown their ability to adapt their geometry in response to mechanical stress. In examining their shape and structure, it becomes apparent that the major strategic function of structural materials and systems involves mechanical support under changing load. This comparison study examines how the morphology of four biological materials (bone, wood, plant stems, and blood vessels) is influenced by environmental loading conditions. As these materials remodel their shape through adaptive growth, they seek to maintain an optimal structure that is able to sustain a particular mechanical support requirement.

Bone and wood adapt to increased mechanical load by increasing their density while plant stems and blood vessels develop thicker vessel walls. For bone, trabecular bone density and cortical bone thickness are increased with load in order to maintain a roughly constant peak functional strain; wood and blood vessels remodel their shape under elevated levels of stress in order to maintain a constant surface stress and constant circumferential stress, respectively. In addition, plant stems and blood vessels which develop thicker vessel walls due to mechanical stressing show a tendency to reduce their elastic modulus so that they could overcome their stiffness and become more flexible, preventing rupture. The examination of these materials suggests that natural materials remodel functions to maintain a constant stress or strain, giving a constant factor of safety.

Thesis Supervisor: Lorna J. Gibson

Title: Professor of Civil and Environmental Engineering

Acknowledgments

I would like to express my gratitude to Professor Lorna J. Gibson for her guidance throughout this project. It has been a great privilege for me to learn from you.

My graduate school life experience would not have been complete without the shared moments with friends from Boston Chinese Evangelical Church's Graduate Student Small Group. Thank you for sympathizing with me and sharing in my joys as we all learned what it was like to "drink from a fire hose". Your friendship will be treasured. To MIT Chinese Christian Fellowship - you have touched my life through your sincerity and genuine love for our Lord. I will never forget the joy that I see in your hearts.

E-Mail has served as my lifeline to the support of my friends back home. Thank you all for your friendship, laughter, compassion, and long-distance prayers. I especially cherish the honesty, understanding, and prayers of my fellow junk-food aficionado. Your friendship is very precious to me; God is good!

I would have been difficult to persevere without the love and encouragement of my family. Mom and Dad - I appreciate all of your sacrifices. You have always served as godly examples for me. Michael - you have become more than a brother to me, but also a friend. Know that I am always praying for you.

I give my utmost to my Lord, Jesus Christ, who is my strength in weakness. May my small achievements be a reflection of your endless grace. You deserve all praise and glory. I continue to find joy in obedience and contentment in each day because You are the strength of my heart.

*Because of the Lord's great love we are not consumed,
for his compassions never fail.
They are new every morning;
great is your faithfulness.*

Lamentations 3:22-23

Table of Contents

<i>Abstract</i>	3
<i>Acknowledgements</i>	5
<i>Chapter 1: Introduction</i>	11
<i>Chapter 2: Bone</i>	15
2.1. Introduction	15
2.2. Bone Composition and Structure	15
2.2.1. Bone Composition on a Molecular Level	
2.2.2. Bone Microstructure: Woven Bone, Primary Bone, Secondary Bone	
2.2.3. Cortical Bone and Trabecular Bone	
2.2.4. Long Bone Anatomy	
2.2.5. Remodeling	
2.3. Bone Remodeling	49
2.3.1. Historical Remarks	
2.4. Experiments and Morphological Changes	54
2.4.1. Bone Remodeling: Assumptions	
2.4.2. Experiments: Bone Atrophy	
2.4.3. Experiments: Bone Hypertrophy	
2.5. Optimization	76
2.5.1. Bone Density	
2.5.2. Trabecular Orientation	
2.6. Remodeling Theories	82
2.6.1. Trabecular Bone Models: The Fabric Tensor	
2.6.2. Trabecular Bone Models: Fhyrie and Carter's Optimization Function	
2.6.3. Optimization Model	
2.7. Conclusion	95
<i>Chapter 3: Wood</i>	101
3.1. Introduction	101
3.2. Wood Composition and Structure	101
3.2.1. Cell Wall Composition	
3.2.2. Wood Microstructure	
3.2.3. Wood Formation	
3.2.4. Gross Morphology of Wood	

3.4.5.	Reaction Wood	
3.3.	Wood Adaptation	120
3.3.1.	Physiological Response	
3.3.2.	Developmental Response	
3.4.	Constant Stress Hypothesis	124
3.5.	Morphological Changes - Shape Optimization	125
3.5.1.	Branch Stem Joint	
3.5.2.	Tree Wounds and Overgrowth	
3.6.	Re-examination of Uniform Stress Hypothesis	131
3.7.	Other Forms of Adaptive Changes in Wood	141
3.8.	Conclusion	144
Chapter 4: Plant Stems		147
4.1.	Introduction	147
4.2.	Plant Structure and Composition	148
4.2.1.	Vascular System	
4.2.2.	Meristems	
4.2.3.	Roots and Stems	
4.2.4.	Vascular Cambium and Secondary Growth	
4.2.5.	Leaves	
4.3.	Remodeling Theories: The Chemical Response to Thigmomorphogenesis	160
4.3.1.	Ethylene	
4.3.2.	Auxin	
4.4.	Experiments: Morphological Changes and Adaptations	164
4.4.1.	Rubbed and Wind-Exposed Plants	
4.4.2.	Elfin Forests	
4.5.	Optimization	173
4.5.1.	Stem Hardening	
4.5.2.	Reduction of Wind Drag	
4.6.	Conclusion	177
Chapter 5: Blood Vessels		181
5.1.	Introduction	181
5.2.	Organization and Structure of the Vascular System	182
5.2.1.	Vascular Wall Cells	
5.2.2.	Wall Structure of Vascular System	
5.2.3.	Arteries	

5.2.4.	Capillaries	
5.2.5.	Veins	
5.2.6.	Relationship Between Wall Thickness and Vessel Lumen	
5.3.	Experiments: Morphological Changes and Adaptations	193
5.3.1.	Stresses in the Arterial Wall	
5.3.2.	Cellular Alignment	
5.3.3.	Structural Changes in Large Arteries in Hypertension	
5.3.4.	Hypertension Remodeling	
5.4.	Conclusion	218
Chapter 6: Conclusion		221
6.1.	Constant Stress and Strain	223
6.2.	Elasticity	224
6.3.	Conclusion	225

Chapter 1: Introduction

Biological systems often have efficient designs because their structure has been able to evolve and adapt to its environment over time. One example can be seen in wood, which has a strength per unit weight comparable to steels. Antlers, shell and bone have a toughness an order of magnitude greater than engineering ceramics. Many engineering designs look to nature as an inspiration because of its tendency for optimum structural or material design. The Crystal Palace built in London for the 1851 Exhibition was designed by Sir Joseph Paxton who conceived the link-work construction by examining a leaf of a water lily. Sir Marc Brunel studied a ship worm when designing a tunneling shield for the construction of the Wapping Tunnel. It has also been suggested that the 1915 patent by Junkers for a honeycomb or sandwich airplane structure was based on the internal structure of a pheasant feather. Science has come full circle as more and more scientists are looking towards nature's materials for clues about how to make stronger composites, improve building materials, and build more efficient structures. [Ashby *et al.* 1995, Cowin 1986, Gibson *et al.* 1995]

Natural materials have evolved to fulfill the needs arising from the ways animals and plants function. Many of these needs involve mechanical adaptation: the need to support static and dynamic loads created by organism mass, the need to store and release elastic energy, the need to resist fracture. Adaptive optimization has led to the formation of exceptionally efficient materials as this process seeks to minimize the mass of material required for some mechanical requirement (i.e. stiffness, strength, toughness) to minimize the metabolic energy required for synthesizing materials. Natural materials are able to adapt their geometry in response to mechanical stresses. As the material experiences modifications in the type of loading it experiences, the material can change or *remodel* its structure such that the general function of the material is maintained. Functional analysis ventures to describe the way an organic structure or process contributes to the maintenance and survival of an organism.

Biomechanics attempts to understand the mechanics of living systems. The term "mechanics" is used to describe force, motion, and strength of materials. In general,

mechanics is utilized in the analysis of a dynamic system, and the biological world has become an object of inquiry. Biology can provide insights into the structure and function of certain materials and how biological systems work to support one another; for example, how the shape of a bone accommodates the type of loading it experiences, how external environmental conditions influence the growth of a tree, how a body's circulation system obliges elevations in blood pressure.

The focus of this comparative study will be to examine how the shape of biological materials is influenced by the loading conditions of their environment. Studies on bone, wood, plant stems, and blood vessels have all shown morphological adaptations when subjected to a change in mechanical loading. In each of these materials, there is a consistent finding that with increased loading, tissue deposition is increased to adapt to such a change. This thesis will examine what the mechanical criteria are for such remodeling behavior and attempt to determine if there are any similarities in the remodeling of the above materials.

The following chapters will investigate the following natural materials: bone, wood, plant stems, and blood vessels. Each chapter will first look at the basic structure and composition of the material. Subsequently, experiments which focus on morphological adaptations to mechanical loading will be reviewed so as to understand the effects of loading on the material, followed by an evaluation of how the materials' general form is optimized for the given loading condition. The concluding chapter compares the functional adaptive changes in morphology of the four materials.

References

- Ashby, M.F., L.J. Gibson, U. Wegst, and R. Olive. 1995. The Mechanical Properties of Natural Materials - I: Material Property Charts. *Proc. R. Soc. Lond. A.*, v. 450, pp. 123-140.
- Cowin, S.C. 1986. Wolff's Law of Trabecular Architecture at Remodeling Equilibrium. *Journal of Biomechanical Engineering*, v. 108, pp. 83-88.
- Gibson, L.J., M.F. Ashby, G.N. Karam, U. Wegst, and H.R. Shercliff. 1995. The Mechanical Properties of Natural Materials - II: Microstructures of Mechanical Efficiency. *Proc. R. Soc. Lond. A.*, v. 450, pp. 141-162.

Chapter 2: Bone

2.1. *Introduction*

It has been recognized that there exists a relationship between the intensity of mechanical loading to which a skeleton is exposed during daily activities and the mass and strength of the bone. Through casual observation, one notices that large, heavy individuals who are more physically active are inclined to have denser and stronger bones compared to frail, sedentary individuals. Along with bone density variations which appear in different individuals, there are also complex distributions of *bone apparent density* (bone mass per unit volume) within the bone of a specific person or animal. This is a result of bone's ability to adapt to the changes in its loading pattern. It is able to adjust its density to accommodate increasing or decreasing load such that its strength is maintained if not improved.

Remodeling in the context of adaptation to a mechanical environment is used to describe any adaptive changes in bone. Internal remodeling usually refers to trabecular bone remodeling and occurs in a process of activation, resorption, and formation. Remodeling of the endosteal and periosteal surfaces of cortical bone tend to involve more modeling, where formation and resorption of bone tissue may occur more independently, or repair processes. In this chapter, observations in long bone in both trabecular and cortical bone, will be examined based on the theories and experiments put forth in recent publications. This study will look at bone structure and see how this natural material responds to loads through its modeling process.

2.2. *Bone Composition and Structure*

Bone is described as a connective tissue which supports the various structures of the body by supporting it against gravity, acting as a lever system for muscular action, and serving as a protective covering for vital internal organs. Bone is also a metabolic tissue in that it serves as a supply for calcium which is essential for nerve conduction, clot

formation, and cell secretion. In mechanical terms, bone can be considered a composite material with solid and fluid phases.

2.2.1. *Bone Composition on a Molecular Level*

On the molecular level, bone is a composite material composed of a fibrous protein, known as collagen, and stiffened by the dense filling of calcium phosphate. Some of its other components include water, amorphous polysaccharides, proteins, and living cells and blood vessels. Collagen acts as a structural protein; the protein molecule, tropocollagen, consists of three polypeptides of the same length and combines to form the microfibrils of collagen. Tropocollagen molecules are approximately 260 nm long and are staggered along side each other by one-fourth their length. They possess the tendency to combine together to form microfibrils by bonding head to tail with molecules of neighboring fibrils. There is a small gap or *hole region* between the head of one molecule and the tail of the next, and because the tropocollagen molecules are stacked side by side, these gaps produce a 67 nm periodicity. On the whole, the microfibril is stabilized by intermolecular crosslinks as microfibrils gather to form fibrils. The collagen microfibrils, together with glycosaminoglycans, form the organic phase of bone's extracellular matrix. The high collagen content (90%) in the organic phase assists bone in resisting tensile stresses. The inorganic or mineral phase consists mainly of calcium phosphate crystals and gives the bone its exceptional resistance to compressive stresses. It comprises approximately 50% of bone by volume and 75% by weight. The inorganic material in bone corresponds fairly closely to hydroxyapatite, $3\text{Ca}_3(\text{PO}_4)_2 \cdot \text{Ca}(\text{OH})_2$, with small quantities of other ions. It exists as tiny crystals about 200 Å long with an average cross-section of about 2500 \AA^2 . These hydroxyapatite crystals are arranged along the length of the collagen fibrils. Thus, bone is essentially a composite material made up of collagen and hydroxyapatite. The apatite crystals are very strong and stiff while collagen is not; thus the Young's modulus of cortical bone (18 GPa in tension in the human femur) is between that of apatite and collagen. As a composite material, bone's strength is higher

than that of apatite or collagen alone because its softer components prevent the stiff ones from brittle cracking and the stiff components prevent the soft ones from yielding.

Bone cells which are embedded in the extracellular matrix include osteoblasts (bone-forming cells), osteoclasts (bone-destroying cells), and osteocytes (bone-maintaining cells). *Osteoblasts* [Figure 2.1] are cuboidal cells with a single, rounded nucleus surrounded by an extensive rough endoplasmic network associated with the production of protein (collagen and proteoglycan in the organic phase). Osteoblasts also maintain a Golgi¹ apparatus for packaging and changing proteins for secretion. In general these cells can be found as a continuous layer on bone surfaces during active deposition. *Osteoclasts* [Figure 2.2] are characterized as large, multinucleated cells with a ruffled, membrane border located along the bone surface. These particular cells can have a diameter up to 100 μ and can contain up to 50 nuclei. Like osteoblasts, osteoclasts have well-developed Golgi bodies which produce lysosomes that aid in the breakdown of underlying bone while mitochondria² are also abundant in the cytoplasm of osteoclasts. *Osteocytes*, the bone-maintaining cells, [Figure 2.3] are actually osteoblasts trapped in small spaces called lacunae in surrounding osteoid. These cells tend to be flattened with few cellular organelles and a single nucleus. Osteocytes maintain an extensive network of interaction with other cells through cellular processes within openings in the bone known as canaliculae. Gap junctions which join the processes from adjacent cells provide a pathway for metabolic diffusion and are essential for maintaining viable bone. If there is an interference in the canalicular network, it may result in osteocyte death, and that portion of bone becomes necrotic and removed by osteoclasts.

Osteocytes are the remains of osteoblasts which have secreted bone around themselves and become isolated from the extracellular matrix. Approximately 10% of the osteoblasts which deposit bone carry on as osteocytes while the others disappear. It has been observed that osteoblast and osteocyte orientations are parallel to collagen fibers. The result is a fibrous, oriented-composite matrix with cells that can sense stress directions

¹ A Golgi body is an organelle consisting of layers of flattened sacs that take up and process secretory and synthetic products from endoplasmic reticulum. It then either releases the finished products into various parts of cell cytoplasm or secretes them to the outside of the cell.

² Mitochondria are organelles in the cytoplasm of cells that function in energy production.

Chapter 2

Previous Work

Surprisingly little research has been performed in the field of shape changing robots. One notable exception is the class of robots, developed independently by Forrest Bishop [1] and Joseph Michael [2]. These robots consist of large rectilinear agglomerations of independent cuboidal structures. These cubes (proposed to be on the scale of 100nm) cooperate by sliding over each other in an interlocked fashion, thereby changing the shape of the agglomeration to affect motion and manipulate objects. In addition, an agglomeration may fission to divide a task, or fuse with another to form a larger agglomeration for cooperation on a large task. These cubes can be thought of as nano-structures capable of communicating with each other to form a larger structure, much like the “alloy” of the T-2000 from *Terminator 2*. However, though they are an interesting concept, no such nano-structures have been made and much of the proposed interconnection and intercommunication is purely theoretical.

Also of interest to the field of shape changing robots is the work by Mark Yim [3] on Polypod, a reconfigurable modular robot built from chains of independently controllable shape changing modules. These modules come in two varieties (both roughly wedge shaped) and can attach to one another as well as to cubic “nodes”. A number of these modules can be assembled to form various shapes capable of locomotion and manipulation. As opposed to Bishop’s “Shape Shifter”, Polypod has been built and tested and is capable of performing a number of “gaits” as well as manipulating some objects.

Both Polypod and the robots of Bishop and Michael are dynamically reconfigurable (at least in concept), and thus have extreme generality. They should both be able to perform very complex tasks. However their speed and rigidity may be a problem at larger scales. Also, at this point in time, they rely on statically stable gaits to move.

By contrast with these proposed architectures, the Recti-Blob (both the previous and current version) must be regarded as a less general intermediate between dynamically reconfigurable shape

changing robots, and nonmodular, nonreconfigurable robots. The Blob cannot move links with respect to one another, nor can it add links dynamically; however, it can change its shape drastically to interact with and conform to its environment. The Blobs are also fast and can easily perform dynamic tasks, such as flips and jumps.

Chapter 3

Developing an Efficient Control Strategy for the New Recti-Blob

3.1 The Simulation Environment

As with many problems in modern control systems, we developed the movement algorithms for the Blob II in simulation. This simulation environment consists of a physics based simulator, two in-house, graphical front-ends, and a set of proprietary C library routines called the Creature Library [4]. The development process begins with the creation of a C code structural model of the robot using routines from the Creature Library. This C file is then compiled into a number of other files necessary to fully describe the robot to the simulator. At this stage it becomes possible to run the simulator on the structural model, routing inputs to various parameters via a text based interface to the simulator. A time history of the model and its movements can be recorded and then viewed by either of the two graphical front ends (one being a wireframe view, the other an animated solid model). This simulation environment is graphically illustrated by Figure 3-1.

The initial creation of a structural model for the robot is substantially streamlined by the use of the Creature Library. The Creature Library allows a robot to be “built” in code from a tree of links and joints, just as it would be described in the real world. This single file is then compiled to generate a number of other files (containing descriptions of the actuators, ground contacts, initial values, and dynamics of the structure) necessary for the simulator. The code description used to create an eight link Blob can be found in Appendix A. In addition, the Creature Library generates a “control.c” file which contains a code stub that is called at a regular interval by the simulator. This stub, while initially blank, can be replaced with suitable code that will implement an algorithm for locomotion. For a robot such as the Blob II, this may be as simple as a C routine which rotates the desired position of the 8 joints every .5 seconds (see Appendix B).

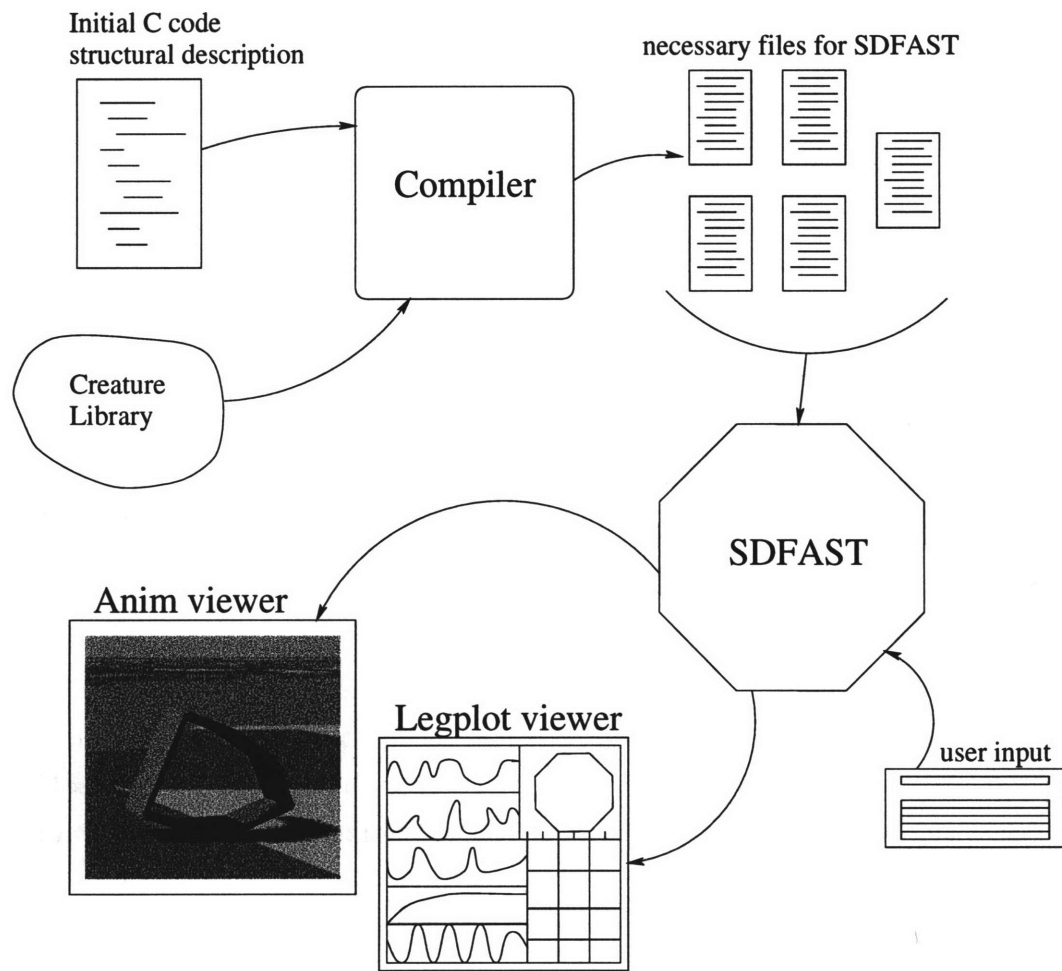


Figure 3-1: A graphical representation of the simulation environment used to develop the control strategies of the Blob II.

SDFAST[5], the physics based simulator, is a commercial modeling program that generates dynamic simulations of rigid body systems. In addition, the program is capable of simulating both free-flying and grounded systems (it can accommodate a uniform gravity field as well as the interactions of the rigid body system with the ground). This makes SDFAST extremely useful for exploring locomotive robots which by definition must interact with the ground. Additionally, most legged robots (including the Blob and Blob II) are characterized by having complex interactions between components which transition from airborne states to “grounded” states.

Because SDFAST does not produce a graphical output of its own, the MIT Legged Locomotion Laboratory has written two proprietary front ends with which to view the output of the simulations. The first of these is Legplot, a simple and quick wire-frame picture of the simulated system. Legplot is also capable of displaying graphical representations of various simulation parameters as a function of time. These are extremely helpful in exploring the detailed behavior of the models as they interact with their environment.

The other front end which has been created by the Leg Lab is called Anim. This program uses the output from SDFAST to generate 3-D renderings of the simulated model. An example of a single frame from Anim can be seen in Figure 3-2. As opposed to Legplot which provides detailed simulation information (but is visually simple), Anim is generally used to create video quality renderings of the various simulated robots and creatures.

3.2 A Model of the Initial Blob

Before creating a model of the Blob II, we simulated a model of the original Blob. Because the Blob has been realized physically as well as in the simulator, this provided a comparison for discrepancies between simulation and the real world. The code description used to model the original Blob is given in Appendix A. This file was then compiled into the necessary files for SDFAST, and the physical structure of the model was checked using Anim (see Figure 3-2).

To model the previously implemented control system, we added the necessary code to the **control.c** file. Just as in the original Blob, the simulated code accepted an initial joint position or “gait” as input, and then rotated this pattern around the joints at an arbitrary interval (approximately .3 seconds). Because it is possible to model the actuators as force sources (which more closely approximates the pneumatic cylinders on the Blob), the pattern used was a set of force output values. This algorithm is implemented with the code fragment shown in Appendix B. A number of the initial patterns or gaits are displayed in Figure 3-4 and a video of the simulation can be found at <http://www.ai.mit.edu/projects/simulations/blob/pref.mpg>

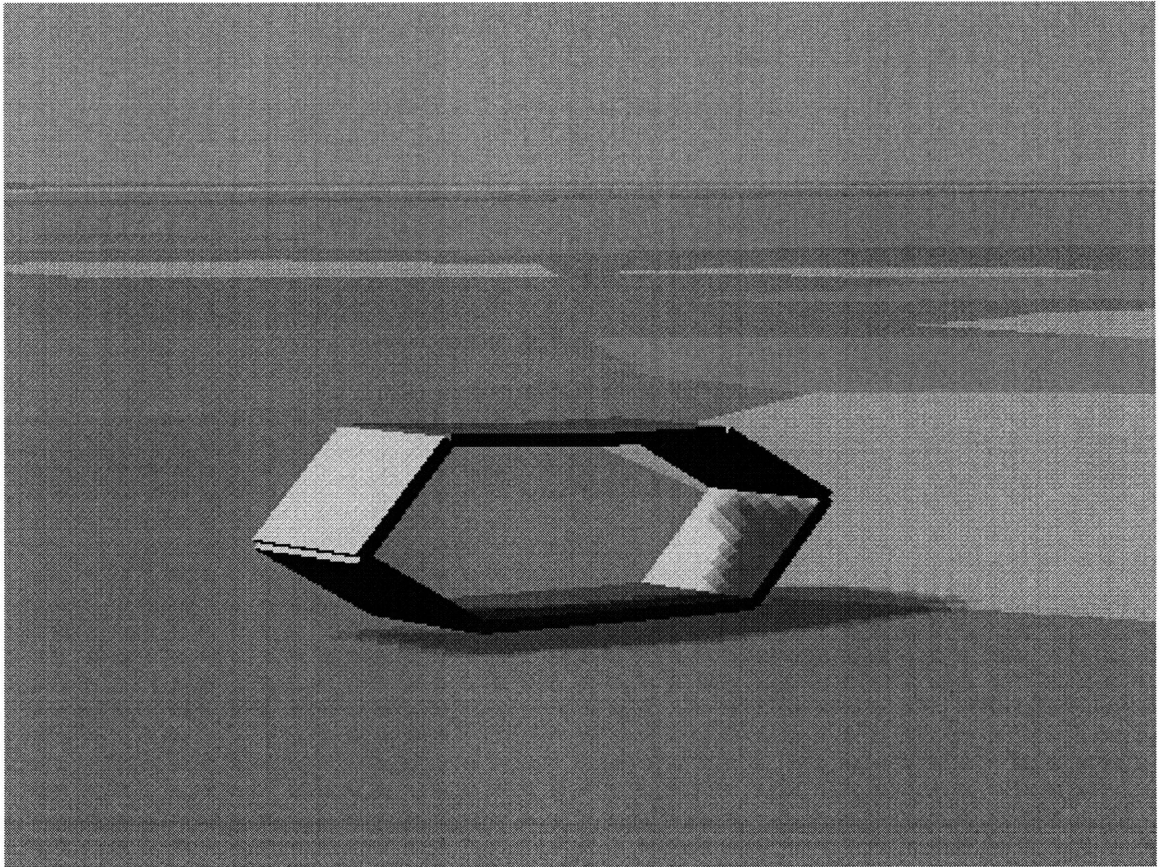


Figure 3-2: A single frame of the initial model of the Blob generated by Anim.

3.3 Discrepancies Between Simulation and the Real World

By comparing the simulations of the Blob with the actual robot, we determined a number of important differences between simulation and the real world that needed to be considered. First, while the simulation can have an arbitrarily strong actuator with zero weight, the Blob must be powered by physical actuators which are limited in the amount of force they can generate and in the fact that they have far from zero mass. Because the simulated actuators have zero mass, this causes a discrepancy between the overall moment of inertia of the Blob in simulation versus the real world. While the two robots are equivalent in total mass, the simulated Blob will have the mass distributed much closer to its exterior. We can compare the Anim frame (fig. 3-2) with the pictures of the actual Blob (fig. 1-1) to see that, while the center of mass of the two structures will be the same, the moment of inertia of the simulated Blob (about an axis normal to the view plane) will be larger. Thus it is easier to “roll” the physical Blob. That is to say, a real actuator which is as strong as the simulated one will be overpowered compared to the simulated Blob.

Another aspect of the simulation which differs from the real world is that the simulated actuators are ideal. An ideal actuator has both a constant torque throughout its range of motion and serves to a position infinitely fast. A physical actuator, while closely approximating a constant torque output over its range of motion, cannot servo infinitely fast. This discrepancy between simulation and the real world must be accounted for in simulations.

This is accomplished by not directly connecting the simulated actuator to the joint. Rather, a spring and damper is inserted (in parallel) between the actuator output and the joint, as shown in Figure 3-3a. We can compare this to a model of a real actuator (Figure 3-3b) connected directly to a joint. For any real actuator, the inherent spring and damper can be tuned (though not eliminated) by using various control techniques. Thus, with added control, a real actuator will behave approximately as if it were an ideal torque/force source in series with a spring and damper — exactly the model of our simulated actuator system.

3.4 An Open Loop Control for the Recti-Blob II

After creating a successful model of the original Blob and exploring the discrepancies between the simulator and real world, we made the necessary changes to the description file to change the actuators from force/torque sources, back to position controlled servos. With this addition, the simulated Blob II was complete in terms of physical constraints, and it simply remained to implement a number of control strategies.

The first control algorithm consisted of a simple open loop strategy which differed only slightly from the original controller. Rather than passing joint forces around consecutive joints, an absolute angle value is passed from joint to joint. This value ranges from $\pi/2$ radians (a 90 degree concave

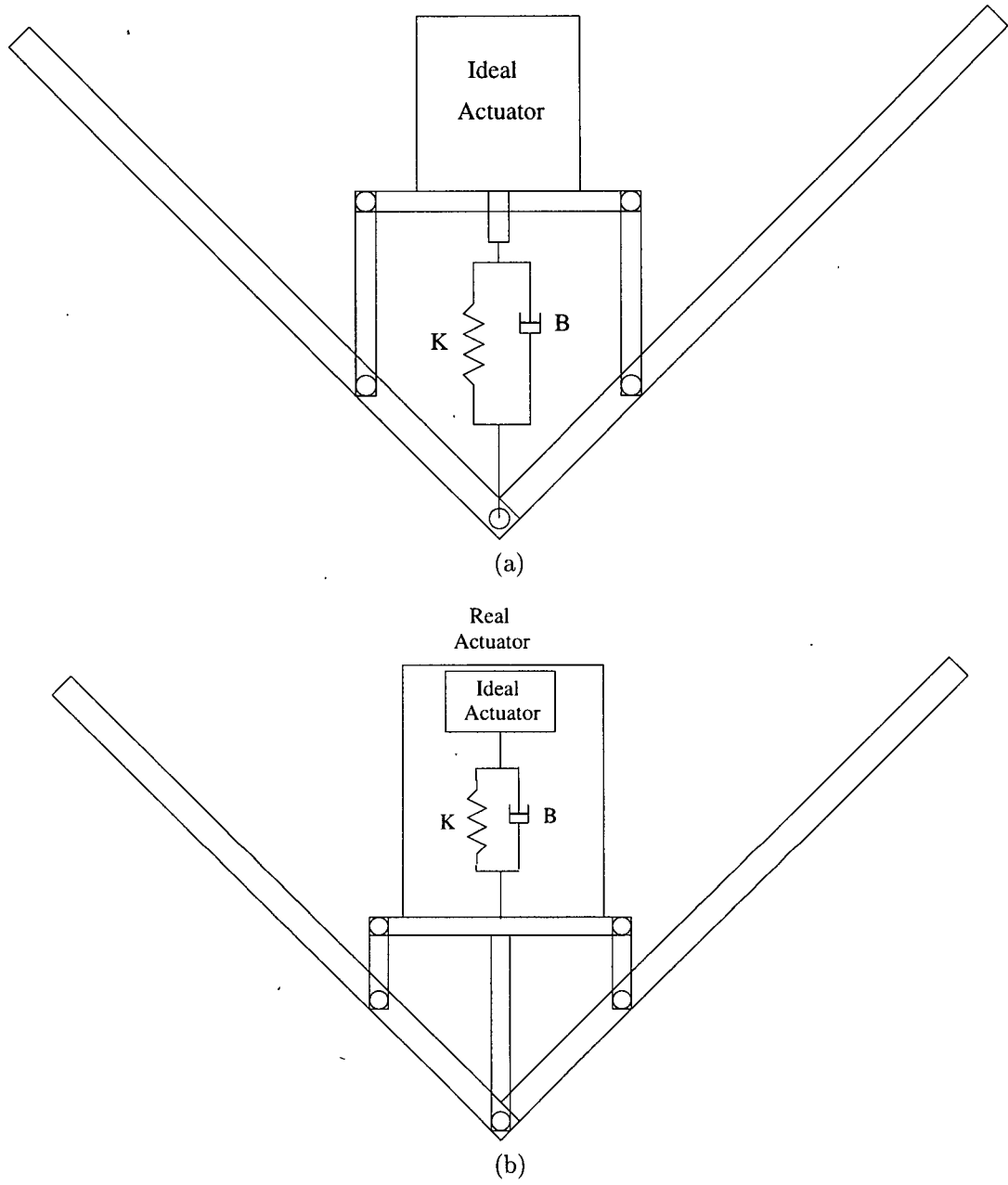
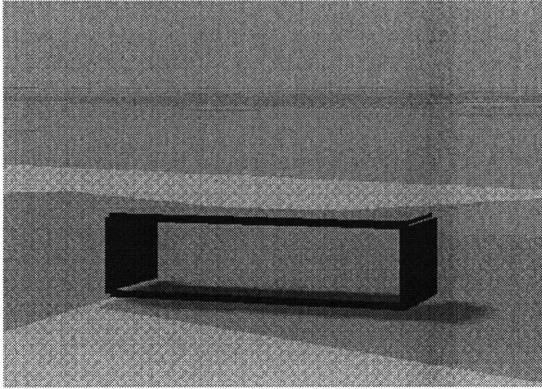
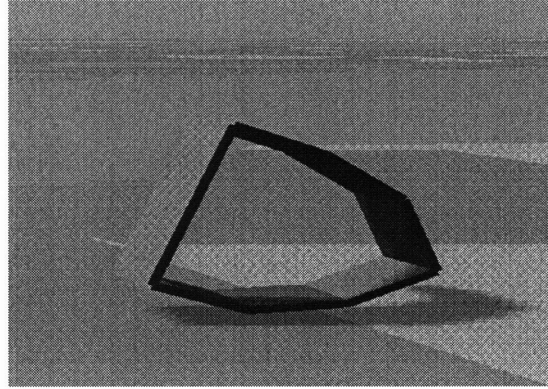


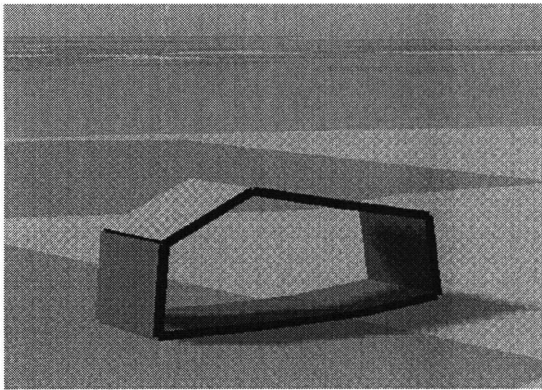
Figure 3-3: A model of a simulated joint (a). A model of a real actuator and joint (b).



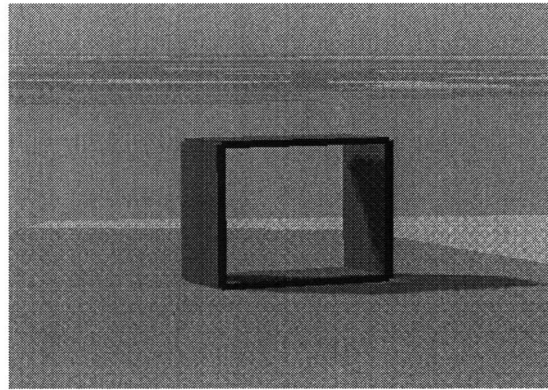
(a)



(b)



(c)



(d)

Figure 3-4: Four initial positions (gaits) for the Blob. The common track style gait (a); a gait which uses only three “on” actuators and is thus known as the triangle (b); a gait which is quite efficient when timed correctly because of the fact that it the most “round” (c); an extremely inefficient (because sustained forward motion seldom happens), square gait (d).

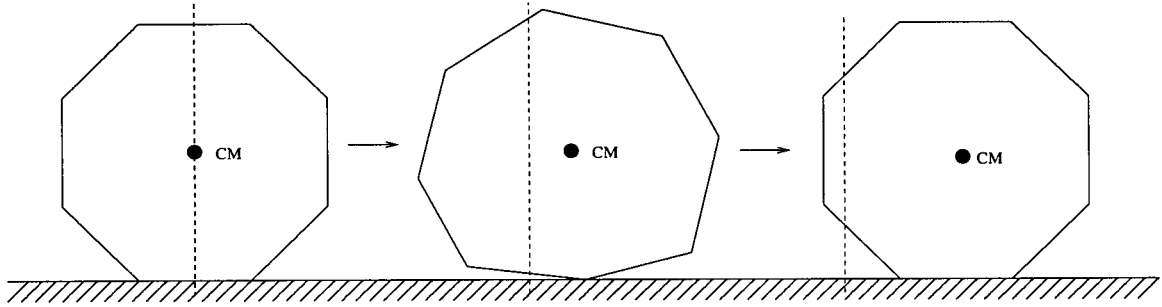


Figure 3-5: The Blob II taking a step.

angle) to $-\pi/2$ radians (a 90 degree convex angle). Also, like the initial controller, the angles are rotated on an arbitrary time step which ranged from .12-.18 seconds.

As with the previous Recti-Blob, an initial position for each joint must be specified. We began by setting the default joint angle to 45° or $\pi/4$ radians, making the Blob II a regular octagon. This seemed to be the logical choice since a regular octagon most closely resembles a circle. With this initial formation, we then alter slightly the two joints on either side of the link on the ground (the “foot”) to move the center of mass in front of the current foot. This will cause the Blob II to roll forward onto the next foot. The angles are then returned to the default values and the process is repeated (see Figure 3-5).

This algorithm represents a good starting point for a number of reasons:

- It utilizes techniques already known to be successful in the previous Blobs (both simulated and real).
- When implemented on the physical robot, it will require no sensory feedback, and thus substantially simplifies the hardware and software used to implement the algorithm.
- Most importantly, it takes advantage of the new actuation. Because the first Blob could only form (for the most part) rectangular shapes, most of the kinetic energy used in making a “step” was lost when the next side of the Blob came down. Now, with infinitely variable joint angles, a more smooth transition can be made from one step to the next. In addition, even when a smooth transition cannot be made, because the Blob II is basically a semi-rigid octagon less forward momentum is lost from step to step (this will be quantified in Section 4.3).

A video of a typical simulation run with the open loop controller can be found at <http://www.ai.mit.edu/projects/leglab/simulations/blob/open.mpg>.

3.5 Closing the Loop: Adding a Foot Switch Feedback Path

While the open loop controller was a substantial improvement over the initial Blob's control strategy, it did not utilize the full capabilities of the Blob II. Because the joint angle rotation occurred at fixed, arbitrary times, the Blob II moved in a "choppy" or mechanical manner. If, instead, the joints were cycled in a manner that were cued from the environment, the motion could be much smoother, more closely approximating a wheel.

To this end, we explored the possibility of using foot switches (simple on-off, contact sensors on each link that indicate when a link is on the ground) to provide feedback for the controller. We began by placing a single switch at the center of each link and then modifying the controller to rotate joint positions based on when the foot switch turns on. In theory, this would prevent the Blob from both "tripping" on itself (when the joint angles are rotated before the next link actually contacts the ground) and from "stalling" (when the angles aren't rotated fast enough).

However, because the foot switches were located flush to the surface of a perfectly flat link, they tended to be unreliable. Often, a link would transfer directly from one edge to the next, giving no reading from that link's foot switch. In addition, because the Blob moves rather quickly, it was often too late in the "step" to rotate angles *after* the center of the link made contact with the ground.

This suggested the addition of multiple foot switches placed at both edges of each link in addition to the one located at the center. However, as with the single switch, the "front" edge switch was activated too late in the step to be of use in the current step. Furthermore, in order to use the trailing edge switch, it was necessary to specify a delay from when contact was lost to when the next rotation should take place. This is exactly the problem of the open loop control: as soon as one introduces an arbitrary time into the system, the robot is no longer basing its motion purely on its environment. We thus abandoned foot switches as a possible method for feedback to the controller of the Blob II.

3.6 The Joint Angle Feedback Controller

With foot switches eliminated as possible feedback sensor for the Blob II, we examined a possible feedback path using the joint angles themselves. Because the actuators are connected via springs and dampers to each joint, it is possible to have motion at a joint even without the actuator moving. Thus, a measure of each joint's angle will provide more information than just the actuator's desired position.

We began by implementing an algorithm that rotated the joint positions based on when the leading (shallow) joint attained its desired actuator position. For example, if the leading joint had a desired position of .3 radians, the Blob II would rotate that value to the next joint only after the current joint actually flattened to .3 radians. The simulation from this strategy tended to look much

like the initial open loop control in that the Blob II was often moving well onto the next step before angle rotation took place.

In order to correct this, we increased the value at which the rotation occurred. Thus, while the leading joint was servoing to .3 radians, the angle rotation would take place after it had reached only .4 radians. This dramatically improved the performance of the Blob II (as determined by simple visual evaluations). By only slightly tuning the value at which rotation occurred (as well as the values of K and B for the actuator), we were able to achieve an extremely efficient locomotive gait (Section 4.3). The video located at <http://www.ai.mit.edu/projects/leglab/simulations/blob/jaf.mpg> shows the Blob II moving using this algorithm.

Chapter 4

Results of Simulating the Blob and Blob II

Creating the Recti-Blob and the Recti-Blob II in a simulated world provided a wealth of information about both robots. By comparing the simulated version of the original Blob to the physical version, we were able to characterize the discrepancies between parameters of the simulations and corresponding parameters in the real world. We were then able to incorporate this knowledge into the simulation of the Blob II to better predict its behavior when it was actually built.

In addition, the simulations of both robots provided us with an accurate and easy way to measure a number of parameters of the robots that would have been virtually impossible to measure in the real world. Variables such as the force the robot exerts on the ground and the exact torque output of each joint are difficult to measure on a physical robot without complicated hardware and measuring equipment. But in simulation, the values can simply be read from a time history graph.

However, the most important information that we obtained from the simulations pertained to the efficiency of the Blob II. Based on numbers obtained from simulating the Blob II with the joint angle feedback control strategy from Section 3.6, we found that, when traveling on level ground, the Blob II could be nearly as efficient as a wheel. To numerically quantify what it means to be “nearly as efficient as a wheel,” we will present an example of two rolling objects, an octagon and a circle/wheel. This example will serve as the foundation for comparing the Blob and Blob II’s efficiencies.

4.1 Example: Efficiency of a Rolling Regular Polygon

Roboticians often speak of the efficiency of a robot or system, but what quantity are we really measuring when we speak of the efficiency of a system? Efficiency is often defined as the ratio of

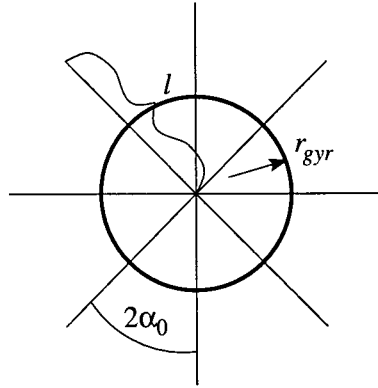


Figure 4-1: A graphical representation of a rimless wheel. The circle marked by the constant, r_{gyr} , represents the moment of inertia of the wheel.

power into the system divided by the work output of the system:

$$\eta = \frac{\text{power in}}{\text{work out}} \quad (4.1)$$

In the case of locomotive robots, however, this quantity has little meaning because on flat ground, no actual mechanical work is done. While a robot may move from point to point, the center of gravity of the robot stays fixed (or in the case of legged robot, it oscillates around a fixed height) and thus no additional energy is gained by the system — therefore, no mechanical work is done by the system. Thus, Equation 4.1 will not suffice to measure the efficiency of many locomotive robots.

In the case of the Blob, Blob II, or any robot who's center of gravity oscillates from one step to the next, it is far more informative to examine a single step. Specifically, we are interested in the amount of energy which the robot retains from step to step. This measure provides us with an idea of the amount of work the actuators must do in order to maintain locomotion.

In order to look at the energetics of the Blob and Blob II we use the derivation by Tad McGeer [7] of the efficiency of a rolling rimless wheel.¹ We begin with the model in Figure 4-1 of a rimless wheel. Because the Blob II has perfectly flat sides, it is equivalent to a rimless wheel with the number of spokes equal to the number of sides of the Blob II. One can visualize this by connecting the ends of the spokes of Figure 4-1 with straight lines. Such a structure would “roll” exactly the same as the original rimless wheel.

If we then imagine this wheel rolling along a level surface, we notice that each time a new “leg” hits the ground, the wheel will lose speed (and equivalently energy). If we assume that each of these collisions with the ground is inelastic and impulsive, then the wheel's angular momentum will be

¹This work was originally intended to represent the inverted pendulum model of a person walking, however it applies equally well to any regular polygon.

conserved from step to step. The loss in speed can be calculated as follows.

Immediately before the collision the angular momentum of the wheel is

$$H^- = (\cos 2\alpha_0 + r_{gyr}^2)ml^2\Omega^- \quad (4.2)$$

where r_{gyr}^2 is the wheel's radius of gyration normalized by the leg length, l (in this case the moment of inertia of the wheel), Ω^- is the angular velocity of the wheel before the collision, and $2\alpha_0$ is (from Figure 4-1) the angle between spokes.

Immediately after the collision, the angular momentum is

$$H^+ = (1 + r_{gyr}^2)ml^2\Omega^+ \quad (4.3)$$

Equating these implies that

$$\eta_v \equiv \frac{\Omega^+}{\Omega^-} = \frac{\cos 2\alpha_0 + r_{gyr}^2}{1 + r_{gyr}^2} \quad (4.4)$$

We have thus defined an efficiency in terms of the ratio of the velocities from one step to the next. We may further define a dimensionless pendulum frequency

$$\sigma^2 \equiv \frac{1}{1 + r_{gyr}^2} \quad (4.5)$$

and combine Equation 4.4 and 4.5 to obtain

$$\eta_v = 1 - \sigma^2(1 - \cos 2\alpha_0) \quad (4.6)$$

This metric for efficiency, cast in dimensionless terms of mass m , length l , and time $\sqrt{l/g}$, is a ratio of velocities; therefore, we must square the result to obtain a measure in terms of energy. In addition, this value of η_v^2 represents the energy lost per step and thus, it must be subtracted from unity to obtain the energy retained from one step to the next. We thus have:

$$\eta_E = 1 - \eta_v^2 = \left(2\sigma^2 - \frac{3\sigma^4}{2}\right) - (2\sigma^2 - 2\sigma^4) \cos 2\alpha_0 - \frac{\sigma^4}{2} \cos 4\alpha_0 \quad (4.7)$$

From this equation, we may calculate $\eta_{E_{octagon}}$, the efficiency of a rolling, rigid octagon. In order to compare this measure to the Blob II, we will assume that the mass of the structure is concentrated at the ends of the spokes, resulting in a value of $r_{gyr} = 1$ (or equivalently, $\sigma^2 = 1/2$). Also, for an octagon, $2\alpha_0 = 45^\circ$. Substituting these values into Equation 4.7, we obtain $\eta_{E_{octagon}} = .72$, thus, 72% of its energy is retained as the octagon moves from one side to the next.

We may also use Equation 4.7 to calculate $\eta_{E_{wheel}}$, the efficiency of an ideal rolling wheel (i.e the frictionless case). This is represented by the case of a rimless wheel with infinitely many, infinitely

close spokes. We will again assume that the mass is concentrated at the end of the spokes giving $r_{gyr} = 1$ and since the spokes are infinitely close, $2\alpha_0$ is equivalently zero. In this case obtain a value for $\eta_{E_{wheel}}$ of 1.0 or 100% efficiency. An ideal wheel loses no energy as it rolls.

4.2 Efficiency Metric for the Blob II

With the theoretical values obtained above, we had benchmarks with which to compare the Blob II's efficiency. However, the simulation does not by default record or measure any quantity like "the efficiency of a system." We thus had to determine the energy retained from step to step based on various other parameters from the system.

This is a relatively simple task because of the fact that many more "basic" parameters of a simulated system *are* accessible to us. We begin by first calculating the work done at each joint during a single step.

$$W_{joint} = \int_0^{t_0} |v_{joint} \tau_{joint}| dt \quad (4.8)$$

Here, v_{joint} is the instantaneous velocity of the joint and τ_{joint} is the instantaneous torque at the joint (both parameters are recorded by the simulation). We use the absolute value in Equation 4.8 because work must be "sourced" or "sunk" by the actuator regardless of it being positive or negative work.

We may then sum these values over all joints to obtain the total work done per step. This is then divided by the kinetic energy and the total number of steps to obtain the energy lost (through work done by the actuators) on each step

$$E_{lost} = \frac{\sum_{joint=1}^n W_{joint}}{steps_{tot} KE}, \quad (4.9)$$

where KE is the instantaneous kinetic energy of the Blob, and $steps_{tot}$ is the total number of steps taken by the Blob (where a step is defined as a transition from one link to the next). As with the rimless wheel, we subtract Equation 4.9 from unity to obtain the energy retained by the Blob II from step to step

$$\eta_{E_{Blob II}} = 1 - \frac{\sum_{joint=1}^n W_{joint}}{steps_{tot} KE}, \quad (4.10)$$

We may then add this formula to the control code of the Blob II (Section 3.1) which will allow us to read this efficiency metric directly from the simulation.

4.3 Comparative Results

After modifying the control code of the Blob II to incorporate the efficiency metric, we began by simulating the most simple of the new control strategies, the open loop controller (Section 3.4).

This algorithm produced the graphs in Figure 4-2(a) and (b) of efficiency versus time and the Blob's kinetic energy versus time, respectively.

While we can see that the efficiency asymptotically approaches a value which is far from even the rolling octagon, it is informative to compare these two plots to similar graphs obtained from simulating the original Blob. The same parameters are plotted for the simulation of the original Blob in Figure 4-3.

As we can see, while the open loop controller is not especially efficient, it is orders of magnitude better than the original Blob. The original Blob does not retain any energy from one step to the next, in fact, its actuators must contribute anywhere from 1 to 20 times the energy of a step just to maintain forward motion.

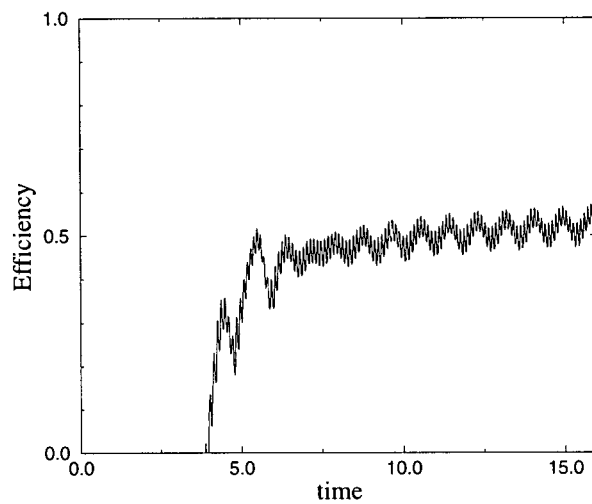
The inefficiency of the original Blob is further evidenced by the respective kinetic energy plots. We see that the Blob II is capable of retaining much of its kinetic energy from step to step, while the original Blob retains almost no kinetic energy from step to step. This can be attributed to the difference in gaits of the two simulated robots. While the Blob II may remain somewhat "round" because of the more sophisticated actuation, the moving of the initial Blob resembles essentially a rolling a square.

It is also helpful to examine a graph of the vertical position of the center of mass versus time of the two robots (Figure 4-4). As we would expect, both the original Blob and the Blob II exhibit an oscillation of the center of mass around a central average, however it is clear that the Blob II uses far less energy on a step displacing its center of mass. Thus it would appear that reducing these oscillations would lead to even more efficient operation.

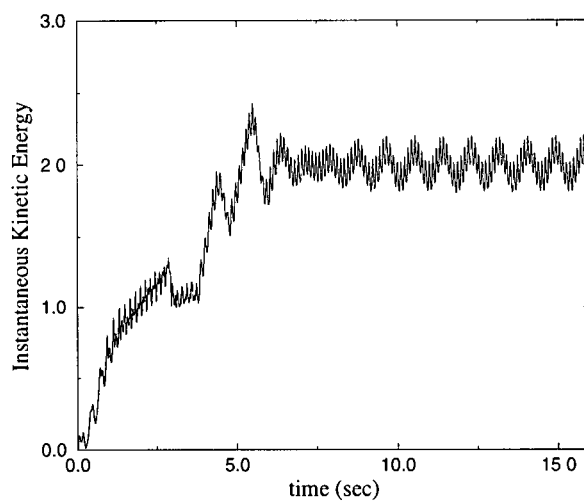
While the open loop strategy was capable of producing sustained forward locomotion in the Blob II, it did not meet the original goals of efficiency approaching that of a wheel. Instead, we examine the efficiency of the Blob II which took advantage of information about its environment to produce more efficient locomotion. Specifically, we examined the joint angle feedback algorithm implemented on a number of variations of the Blob II (structurally larger and more massive).

The graph in Figure 4-5 shows plots of the efficiency, kinetic energy and vertical center of mass displacement of the Blob II using the joint angle feedback controller. We see that the efficiency approaches 75%, a substantial improvement over both the initial Blob and the open loop Blob II. This is due to a number of factors including a smaller oscillation of the center of mass as well as better timed "switching" (the moment at which joint angles are rotated). However, this is only marginally better than the number we obtained by rolling a rigid octagon.

Because of possible numerical artifacts of the simulator (i.e. the simulator must run in discrete time with some minimum sized integration step), we suspected that this 75% limit was a result of the specific size of the Blob II being run in the simulator. In order to test this, we created two more versions of the Blob II, one which was three times larger in the linear dimensions and twice

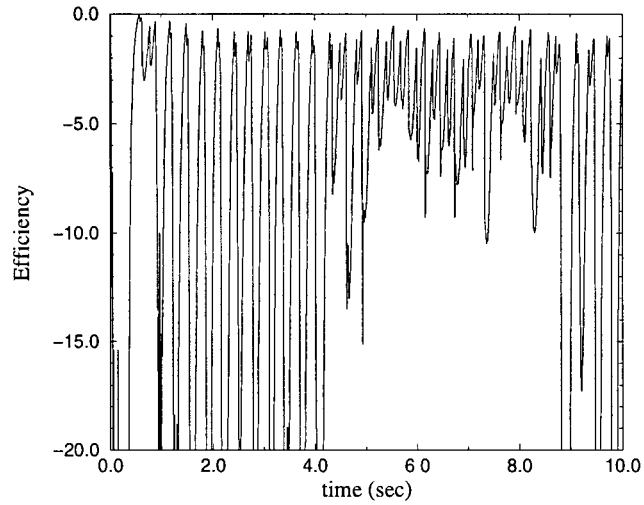


(a)

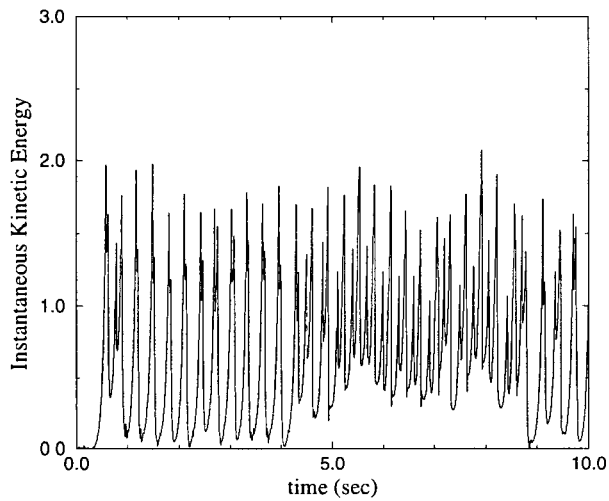


(b)

Figure 4-2: A plot of efficiency versus time based on the value $\eta_{E_{Blob II}}$ calculated by the simulator (a) and instantaneous kinetic energy versus time (b) of the Blob II using the open loop controller.

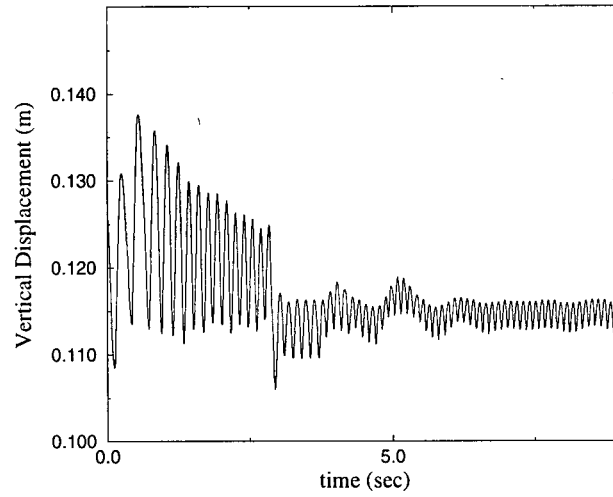


(a)

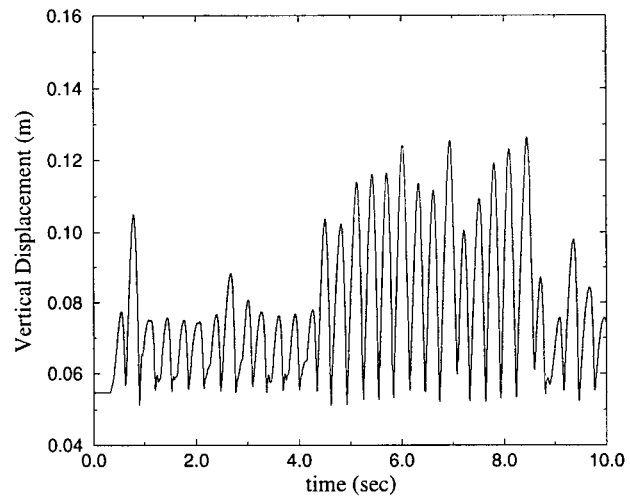


(b)

Figure 4-3: Plots of the efficiency versus time (a) and kinetic energy of the robot versus time (b) based on simulations of the original Blob. The negative value in the efficiency plot simply means that from one step to the next, not only is energy not retained from step to step (which would correspond to 0 efficiency), but a substantial amount of energy must be added to the system just to maintain forward progress.

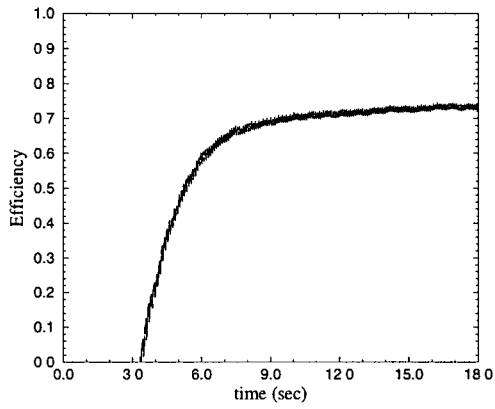


(a)

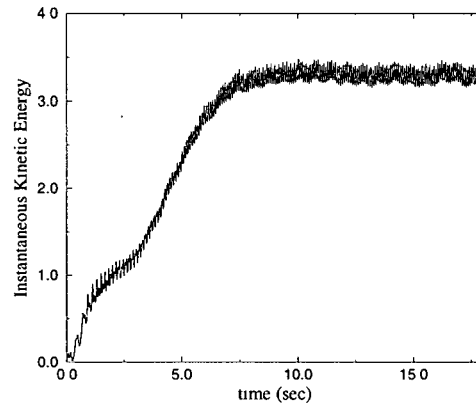


(b)

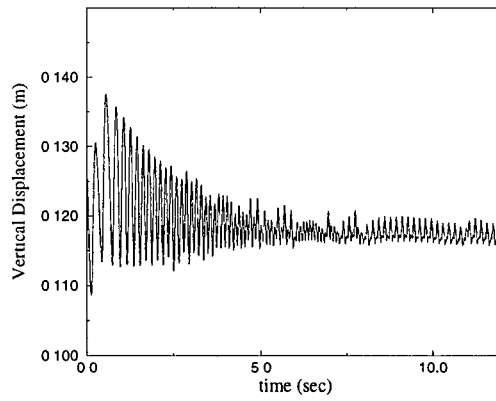
Figure 4-4: Plots of the vertical displacement of the Blob II's center of mass versus time (a) and the vertical displacement of the original Blob's center of mass versus time (b).



(a)



(b)

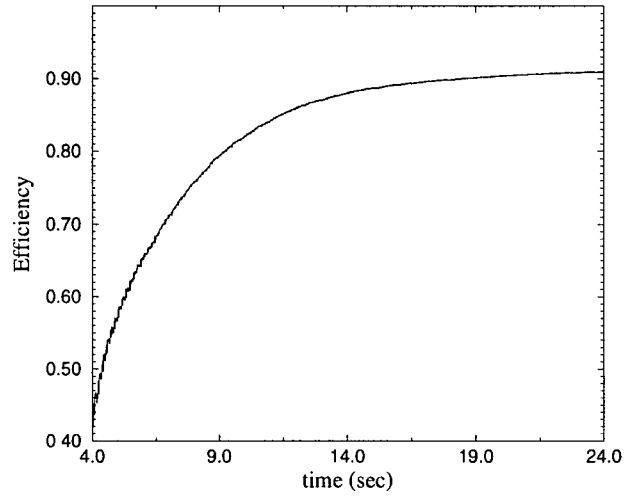


(c)

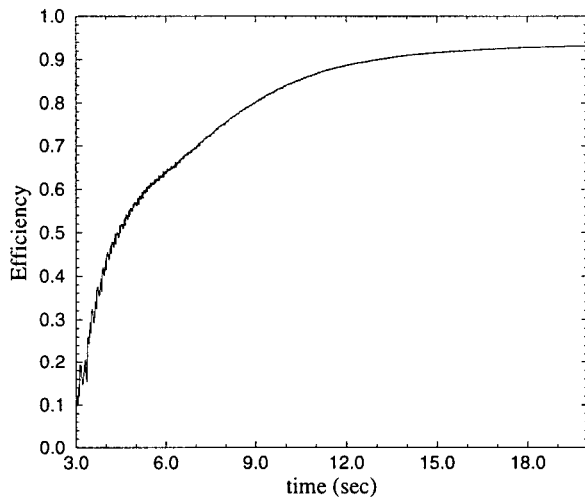
Figure 4-5: Plots of the efficiency (a), kinetic energy (b), and vertical center of mass displacement (c) versus time of the Blob II using the joint angle feedback controller. We note that the efficiency approaches 75% and see that the center of mass oscillations are extremely small.

as massive, and one which was three times larger in both linear dimension and mass. Using these models we obtained the graphs of efficiency versus time in Figure 4-6.

As we can see, the larger Blobs were able to attain an efficiency of approximately 91%. This value suggests that the previous limit of 75% *was* simply an artifact of the simulator and that it is possible to make even an eight link Blob nearly as efficient as a wheel.



(a)



(b)

Figure 4-6: Plots of the efficiency versus time of a Blob II three times larger in linear dimension and twice as massive(a) and a Blob II three times larger in mass and linear dimension (b).

Chapter 5

Designing and Creating the Recti-Blob II

The main goal in designing the Recti-Blob II was to maintain the architecture of the original Blob while eliminating its substantial drawbacks. This required (as a minimum) replacing the actuators and modifying the joint to accommodate both concave and convex angles (see Figure 5-1). However, because of the size and architecture of the Blob, these two requirements are intimately related. Thus, a problem which appeared to be simply finding a powerful actuator and changing the position of two rods quickly became a problem in redesigning the entire robot.

5.1 The Actuators

To find a suitable actuator for the Blob II, we considered a number of constraints. First, we required that the actuators be continuously variable (as opposed to the on-off pneumatic pistons of the original Blob). Additionally, they must be small¹ and capable of being self-contained. Being self-contained, implies that the actuator could somehow derive its power from a source on-board the Blob II, making feasible the goal of an autonomous Blob. For the case of hydraulic and pneumatic actuators, this would entail a pressurized fluid source on the Blob II (usually in the form a gas powered compressor) and for an electric motor, this would require some type of on-board batteries.

These constraints essentially eliminated the possibility of using pneumatic actuators since they require a prohibitive amount of hardware to be infinitely positionable. Similarly, a hydraulic actuator that is both small and self-contained is simply not commercially available. We were thus left with the option of electric motors.

¹Larger Blobs are not impossible (see Section 4.3), but it was decided to create a Blob II which was roughly the same size as the original Blob

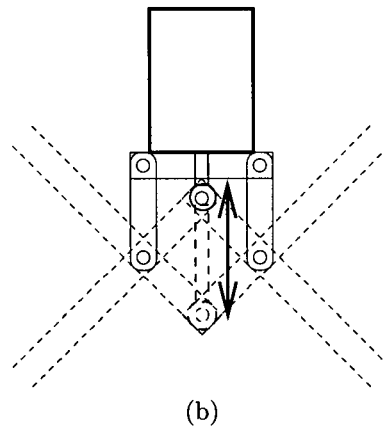
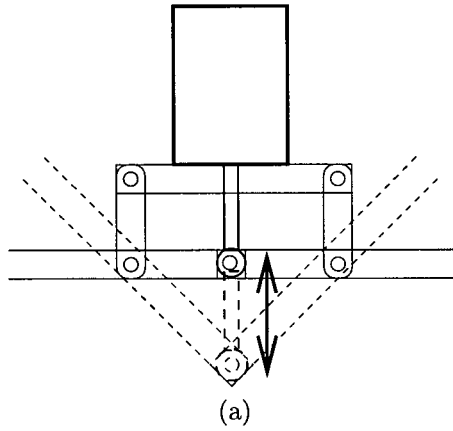


Figure 5-1: The required modification of the previous joint (a) to accommodate concave angles—the link rods are moved closer to the joint and the actuator throw is increased (b).

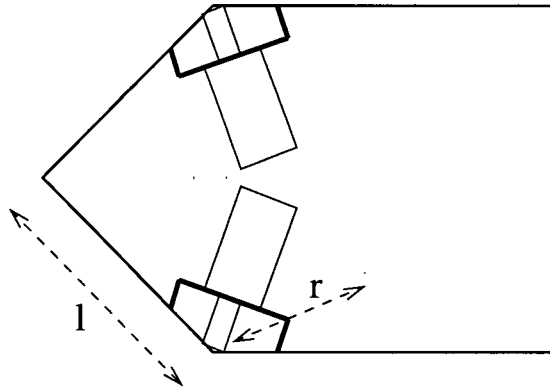


Figure 5-2: Three joints of the Blob (or Blob II), rolling through a tight corner. The actuators, each inscribed in a cylinder of radius r , are not colliding.

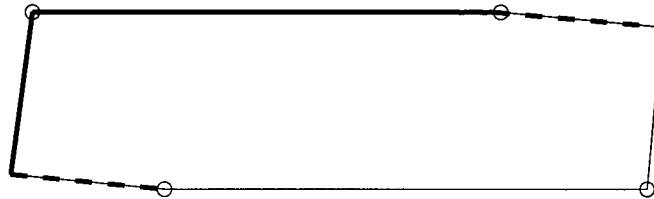


Figure 5-3: A ten link Recti-Blob starting to execute a step. The solid bold lines are links being lifted. The dashed bold lines are links being half lifted. The circles indicate the joints doing the lifting.

A number of different motor options were considered including linear stepper motors and DC servo motors (the standard radio control variety). Again, the physical characteristics of the Blob and Blob II constrained which actuators could be considered. Specifically, the actuators must be small enough to clear each other when an inside tight corner of the robot is rolling past itself (Figure 5-2), yet also deliver enough force to (together) lift the half the mass of the robot (Figure 5-3).

5.1.1 Determining a Quantitative Measure to Evaluate the Actuators

In order to choose a suitable actuator/motor, consider the following figure of merit. First, let n be the number of links, l be the length of each link and m be the mass of each link. Consider the actuator to be inscribed in a cylinder with radius r , coaxial with the joint. For any given “tightest corner,” such as that shown in Figure 5-2, r must be related to the link length, l , by the constant factor, k_1 :

$$r = k_1 l \tag{5.1}$$

This constraint must prevent the actuators from colliding with one another during a step of this gait.

During a step, a portion of the Blob is lifted a fixed distance by a specific number of actuators. The exact distance and number of actuators doing the work depend on the gait being executed. However, for any given gait the work done per actuator per step is:

$$W = nmgk_2l = M g k_2 l \quad (5.2)$$

where M is the mass of the robot, and k_2 is a proportionality constant which captures the gait geometry. This gait geometry measures what fraction of mass M is being lifted what fraction of distance, l , by how many actuators. The dependence of the work on M and l is made explicit so that k_2 depends only on the chosen gait, not on the absolute size or mass of the robot.

The work done per actuator per step may also be written in terms of the capabilities of the chosen actuator type:

$$W = F \Delta X, \quad (5.3)$$

where F is the maximum force delivered and ΔX is the actuator excursion, assuming the actuator is linear. Finally, because the Blob and Blob II are only two dimensional in this analysis, the mass of a link may be written in terms of a one dimensional mass density, ρ :

$$m = \rho l. \quad (5.4)$$

Combining Equations 5.1- 5.3 and grouping properties of the actuator on the left side, and properties of the Blob on the right, we obtain:

$$\nu = \frac{F \Delta X}{r^2} = \frac{n \rho g k_2}{k_1^2}. \quad (5.5)$$

When obtained by substituting values into the right hand side of Equation 5.5, ν represents the minimum possible value that the actuator properties must attain in order to execute the gait described by the factor k_2 . Thus, we may place a lower bound on the quantity $\frac{F \Delta X}{r^2}$ and select actuators that either meet or surpass this value.

5.1.2 Evaluating of the Original Blob

The initial Blob was characterized by the following numbers:

Actuator force output, F 24.5 lbs.

Actuator throw Δ, X .75 in.

Coaxial cylinder radius, r 2.75 in.

These result in value of ν (from the left side of Equation 5.5) of 2.43 lbs/in.

This can be compared to the value obtained on the right side of Equation 5.5 by plugging in other known values for the Blob.

Total mass, M .165 slugs
Number of links, n 8
Length of a link, l 4.0in.

Based on these values, and assuming that $k_2 = .01$ (representing the most demanding gait), we may evaluate the following quantities.

Corner constant, k_1 .6875
One dimensional mass density, ρ .02 slugs

Plugging these numbers into the right side of Equation 5.5, we find that $\nu = 1.34$.

This suggests three things about the previous version of the Blob:

1. The actuators are much more powerful than they need to be in order to execute even the most demanding gait.
2. The actuators are moving much further than they need to in order to move the Blob.
3. The Blob could be substantially larger and the actuators could still provide enough locomotive power.

Because the actuators on the original Blob were selected to be more than powerful enough to “just” execute the most demanding gait, the Blob is capable of both moving at high speeds and performing acrobatics. More importantly, this “cushion” of power allowed some flexibility in the initial construction and design of the original Blob. In general, the actuators selected (or designed) for a robot should be overpowered to ensure that any performance goal can be achieved.

5.1.3 Selecting a New Actuator

With the above figure of merit to provide a gauge of the power necessary for a new Blob, we began evaluating potential actuators. Initially, we selected a linear stepper motor from Airpax which used an internally threaded commutator to move a threaded shaft in a linear motion (Figure 5-4). The important characteristics of the motor are as follows:²

²The value of r is assumed to be the same as above. This is due to the fact that the motors are similar in size to the pneumatic pistons and thus should fit in a similar sized cylinder.

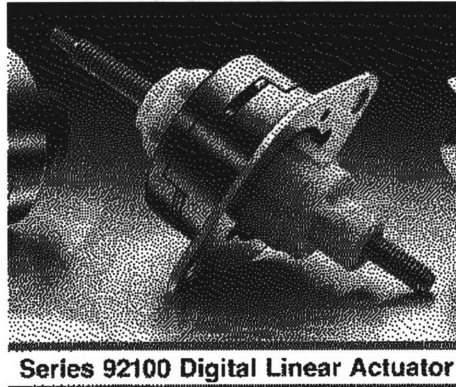


Figure 5-4: A photograph of the Airpax stepper motor which was initially chosen as the actuator for the Blob II

Actuator force output, F	2.8125 lbs. (45 oz.)
Actuator throw Δ, X	1.5 in.
Coaxial cylinder radius, r	2.75 in.

These values result in a value of ν of .56. While this is substantially lower than the value of the pneumatic actuator used on the original Blob, it is not necessarily a problem for the Blob II. The stepper motors are substantially lighter than the pneumatic pistons and therefore the value of ν from the right side of Equation 5.5 for the Blob II is approximately .52. Thus, while the stepper motors did not have the substantial “cushion” of the pistons, they should have (in theory) been powerful enough to move the Blob II.

We proceeded to design a single joint for the Blob II (Figure 5-5) which incorporated the stepper motors, as well as a series elastic element.³[9][8] However, after prototyping the joint, we found that, while the motors were almost powerful enough to move the Blob II, they simply did not have the necessary power or speed to accomplish the goal of truly robust locomotion (not to mention acrobatics).

In addition to being underpowered, the stepper motors were expensive and prone to breaking under stresses that would easily be encountered by the Blob II. Thus, we concluded that the best actuator for the Blob II was one that was commercially available (and thus replaceable) and had a much greater force output than the Airpax stepper motors. The most logical choice which fit these criteria, as well as being small and continuously controllable was the “standard” Futaba style radio control (R/C) servo.

³The benefits of series elastic actuation are well documented elsewhere. Basically, they were used here to provide an explicit energy storage element from which the Blob II could draw on from step to step. A detailed reference can be found in [8].

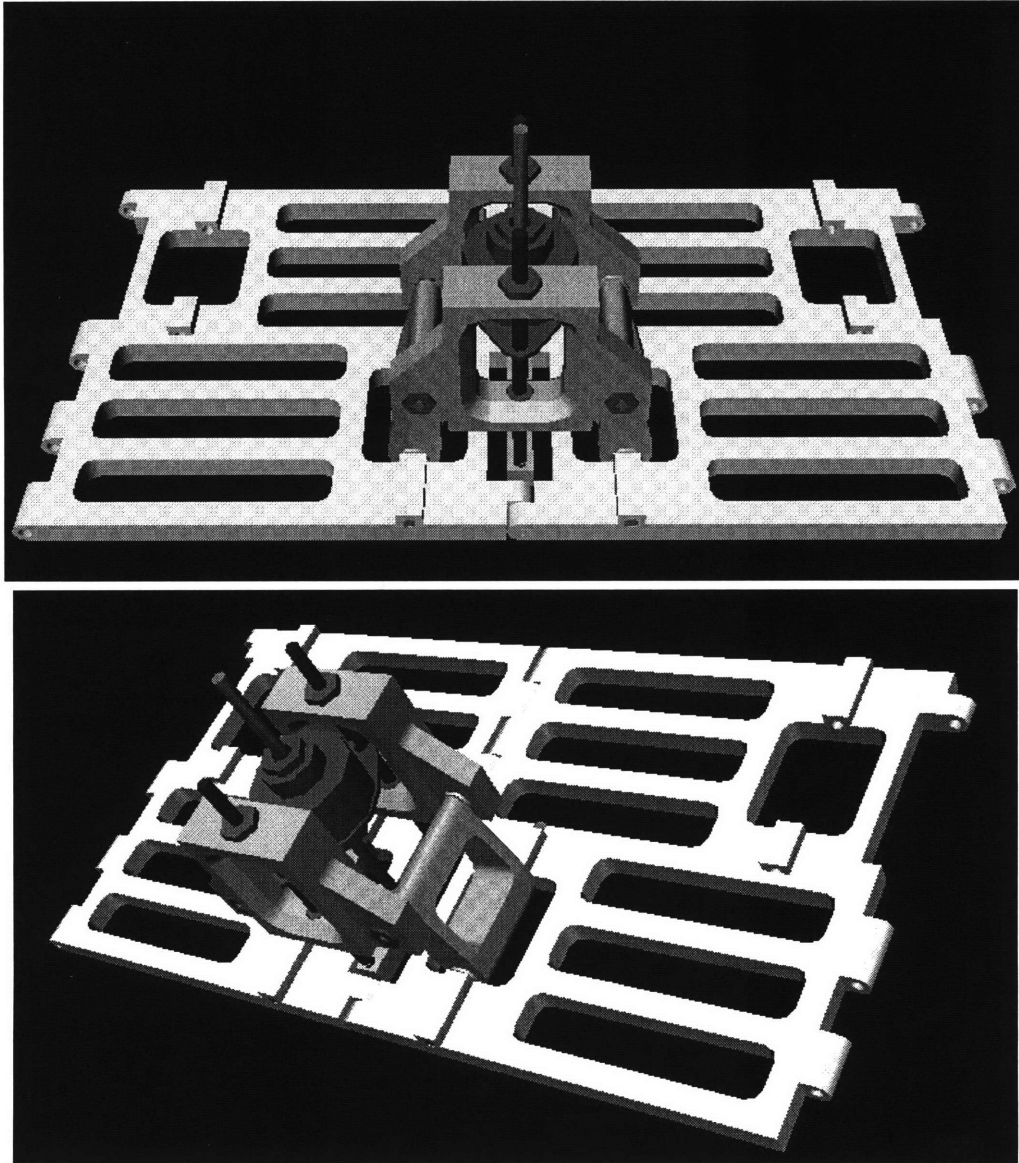


Figure 5-5: Two renderings of the initial joint design of the Blob II using the Airpax stepper motor.

Ultimately, we chose a servo made by Futaba which had the highest torque output for its size and which was also capable of traveling its range of motion in only .11 seconds. We again perform the figure of merit analysis on the Futaba servo using the manufacturer supplied numbers.⁴

$$\begin{aligned} \text{Actuator force output, } F & \frac{129\text{oz-in.}(\text{torque output})}{.78\text{in.}(\text{actuationradius})} = 10.3 \text{ lbs. (approx. linear)} \\ \text{Actuator throw } \Delta, X & 120^\circ \text{ (rotation angle)} = 1.35 \text{ in. (linear excursion)} \\ \text{Coaxial cylinder radius, } r & 3.0\text{in.} \end{aligned}$$

From these numbers, we obtain a value for ν of 1.55. Again, this value is less than the value obtained from the original Blob's pneumatic pistons, however, it is a major improvement over the stepper motors. In addition, based on previous experience with these types of actuators, we were confident that they would withstand the external stresses placed on them by the Blob II.

However, switching to an R/C servo had a substantial drawback: servos are rotary rather than linear. Thus, it was impractical to simply modify the existing joint design to accommodate the use of a rotational actuator rather than a linear one.

5.2 Structural Design of a New Joint

As with the actuators, building a Blob II that captured the architecture and size of the original Blob placed many constraints on the physical structure as well. It was necessary to design a Blob II based on much the same platform as the Blob. However, because of the major change in actuation (a rotational actuator versus a linear one), little of the original or first prototype joint design could be used.

The original Blob's joint is based on the idea of a five bar linkage with one of the degrees of freedom being constrained by the actuator itself (see Figure 1-4). The main link axis is fixed to the end of the piston which is constrained to move only in a linear fashion (this is a feature of the pneumatic cylinder and was accomplished in the stepper motor joint design by using a set of vertical shafts on either side of the motor). Without this constraint, the actuator platform would have been free to move from side to side and reliable operation of the joint would be impossible.

After initially exploring the idea of using a similar design with the servo actuators, we quickly found that a five bar linkage was inappropriate for the application. A rotational actuator simply does not lend itself to that type of mechanism. In essence, we would be converting a rotational motion to a linear motion and then back to a rotational one—which is inefficient. Rather, the rotational motion of the servo should be used to directly control the main axis joint angle.

It would be impossible to design the Blob II with each main joint axis coincident with a servo's

⁴With the servos, we assume that because of their slightly larger size, the coaxial cylinder radius will be slightly larger. This will simply mean that the links will have to be larger as well.

axis of rotation and still maintain the active skin architecture of the Blob. Thus, we turned to another mechanical linkage to place the actuators inside the “skin” of the robot while maintaining the direct connection of the rotational actuator to the rotational joint.

The final design of the joint is illustrated in Figure 5-6.⁵ This design uses a parallel linkage to translate the servo arm angle directly into an angle on the main joint axis. By making the two parallel distances the same (e.g. the distance from the center axis to the link rod insertion point on the link and the length of the servo arm) we guarantee that the servo angle of deflection will be exactly half the angle between the two links. Coupling this with the fact that two servos are used at every joint, we can have each actuator move from $+45^\circ$ to -45° (assuming that 0° corresponds to the servo being perpendicular to the vertical plane in Figure 5-6) and thus accomplish a full range of 90° convex and concave motion.

A second advantage to the new design is the fact that it uses two servos at each joint to provide actuation. This is beneficial because it doubles the force output at each joint. In other words, we have substantially increased the value of ν from the left side of Equation 5.5 while only increasing the right side by a slight amount (due to the fact that the weight of the robot, and thus the value of ρ , is only increased rather than doubled). This translates to an increased “power cushion” which, as in the original Blob, should allow the robot to be more robust and even perform acrobatics.

It is interesting to note that even with two servos at each joint, the control hardware necessary for the new joint design is not substantially greater than that of the original Blob. This is accomplished by orienting the two servos in opposite directions. Because of the orientation, a clockwise rotation of both servos (which can be commanded by sending the same signal going to both servos) will cause the joint to form a 90° concave angle. Contrast this to a case where the servos are oriented in the same direction. With both of them now turning clockwise, the joint angle will remain flat, and the central mounting plate will be at a 45° angle to the links.⁶

5.3 The Blob II

As with the stepper motor joint design, we prototyped the servo joint to provide a real world test of the concept. As expected, the joint performed well. We were able to achieve the full $\pm 90^\circ$ concave and convex angles and we were even able to lift approximately five pounds with the joint (this is at least a pound more than the estimated total weight of the robot). We thus felt confident in completing the design of the entire Blob II.

⁵The renderings in the figure, as well as the design model itself, were created using Pro/Engineer software from Parametric Technologies[6]. ProE is a suite of design tools capable of modeling and drawing complicated 3-D parts and assemblies.

⁶This feature may eventually be useful, allowing the Blob II to achieve formations it could not achieve with a mounting plate that is restricted to be bisecting the main joint angle. Initially however, the problem is most easily avoided and the two servos should be given the same command signal.

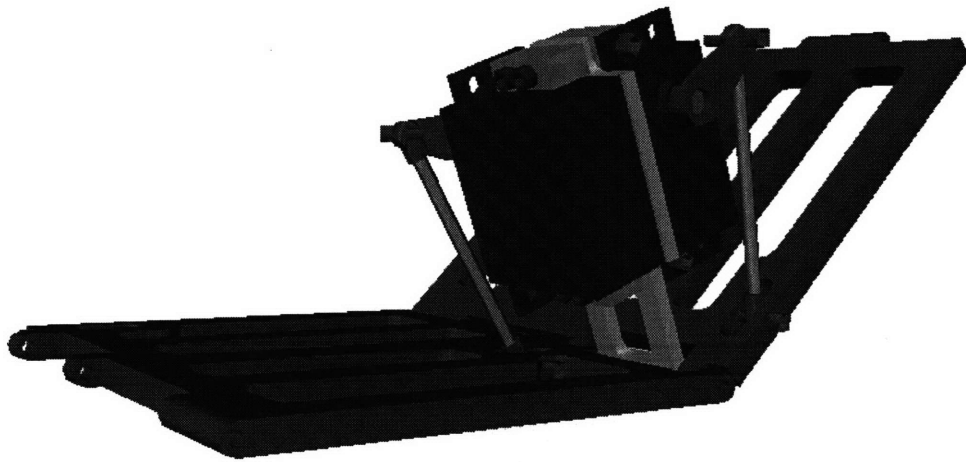
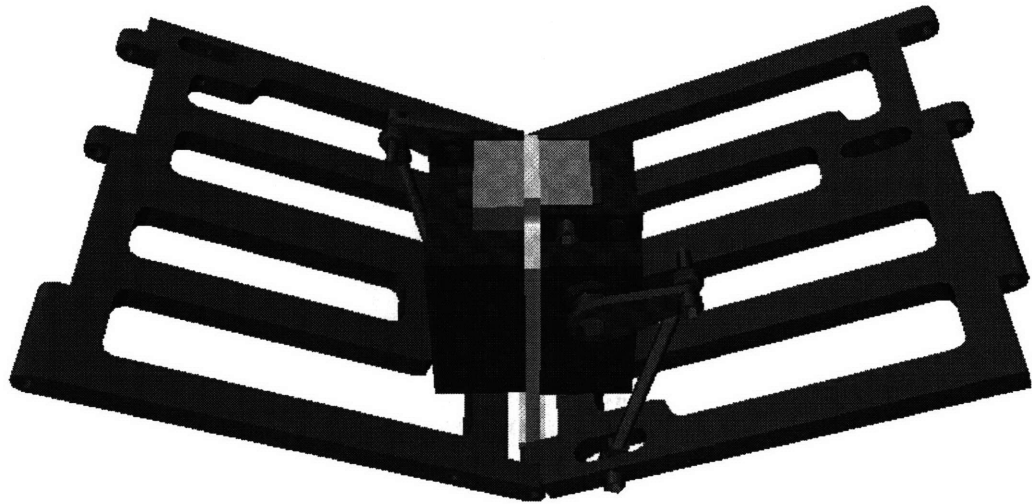


Figure 5-6: Two renderings of the joint design used in the final version of the Blob II. Note the parallel linkage and the dual servos.

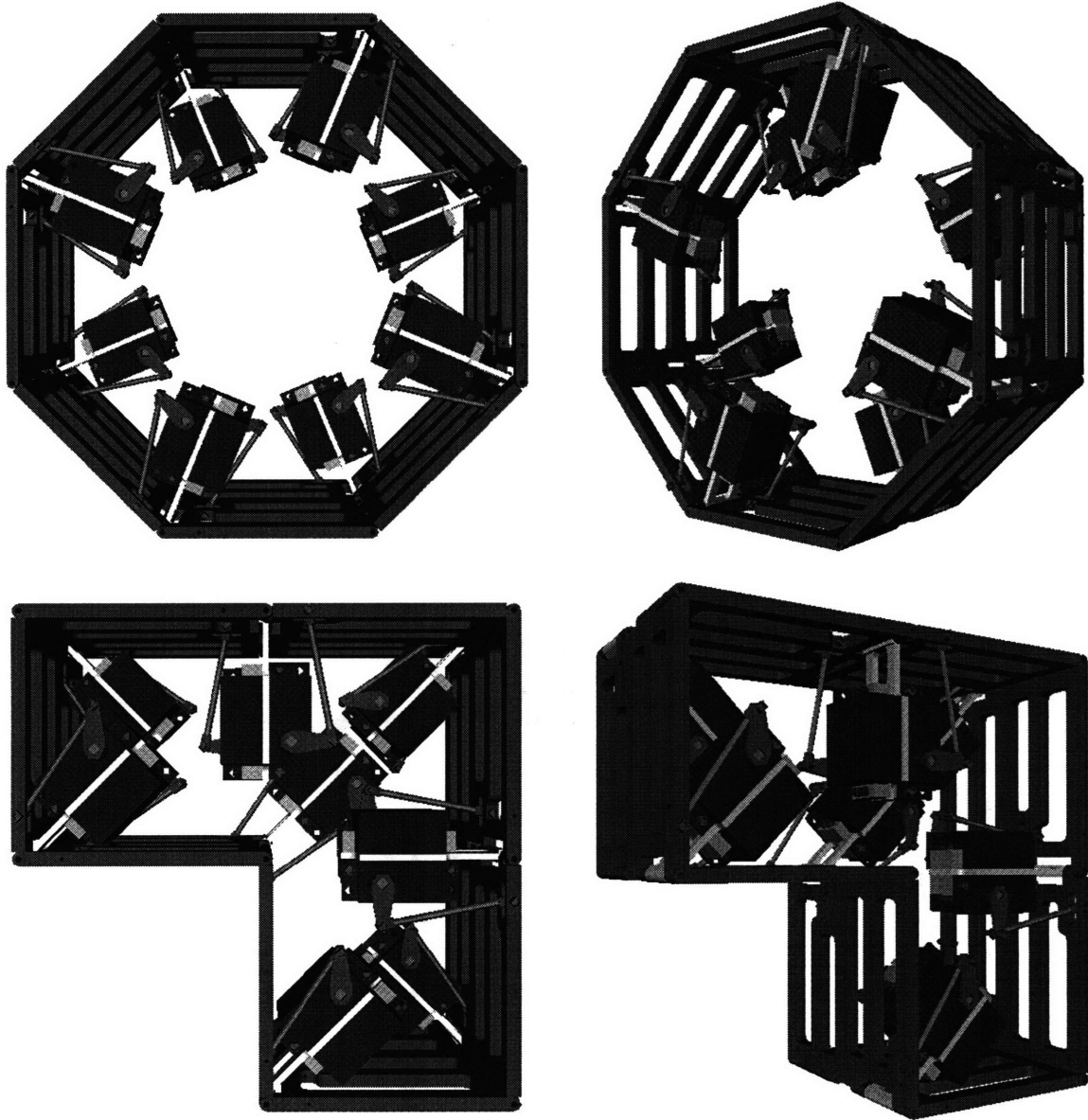


Figure 5-7: Two renderings of the Recti-Blob II. Note the similarity in structure (i.e. the “active skin”) to the original as well as the two servos at each joint.

Four renderings of the Blob II can be seen in Figure 5-7. The top pictures represent the Blob II as it will usually operate — in an octagon shape with only the two joints nearest the ground moving to any significant degree. The bottom pictures show the Blob II forming the newly possible concave shape which should allow it to surmount obstacles that are comparable to half the Blob II's size. It is important to note that the actuators are capable of passing each other with out colliding when the Blob II forms such a tight concave angle.

We also expanded the original design of the Blob II, adding four links to make a 12 link Blob. Two renderings of this design can be seen in Figure 5-8. This design illustrates the flexibility of the Blob II's architecture the ability to expand and build on the current platform.

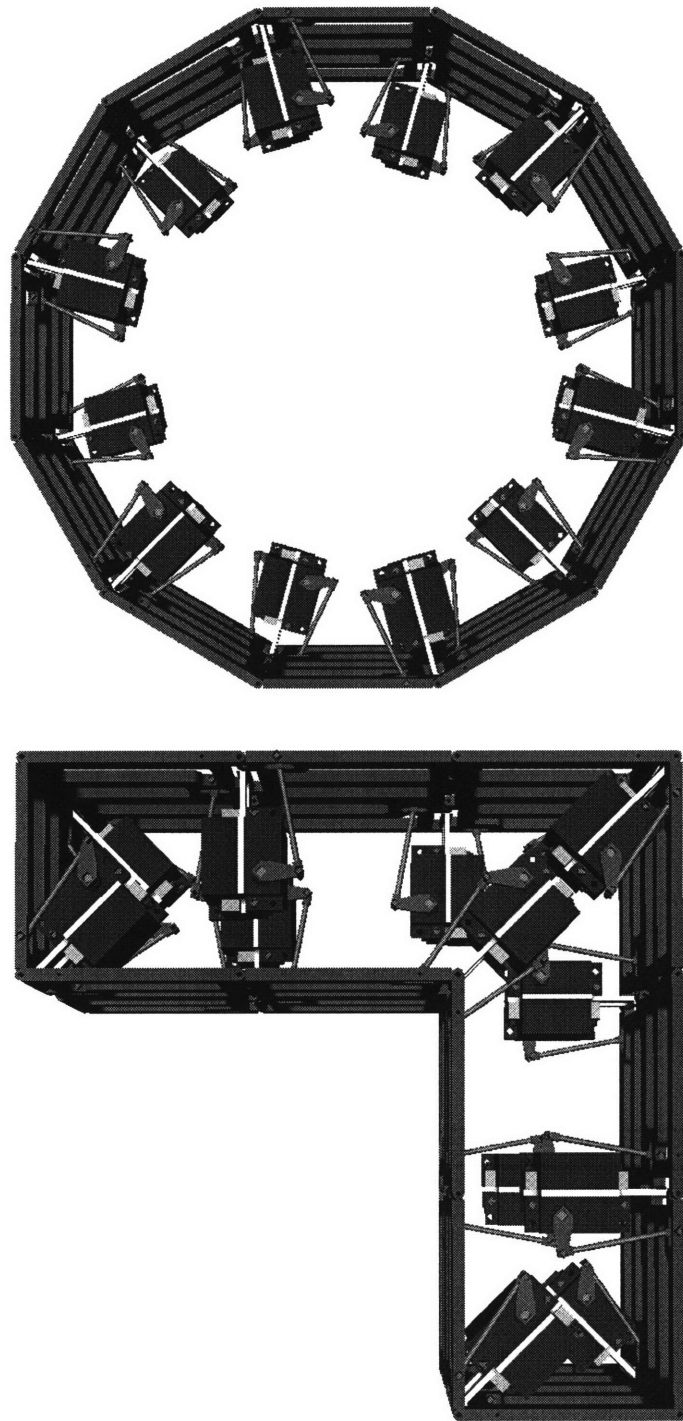


Figure 5-8: Two renderings of a 12 link Recti-Blob II. Again, the first shows the Blob in a normal “rolling” formation and the second, a concave conformation capable of surmounting stairs. The structure itself is easily expanded to any number of links based on the original 8 link design.

Chapter 6

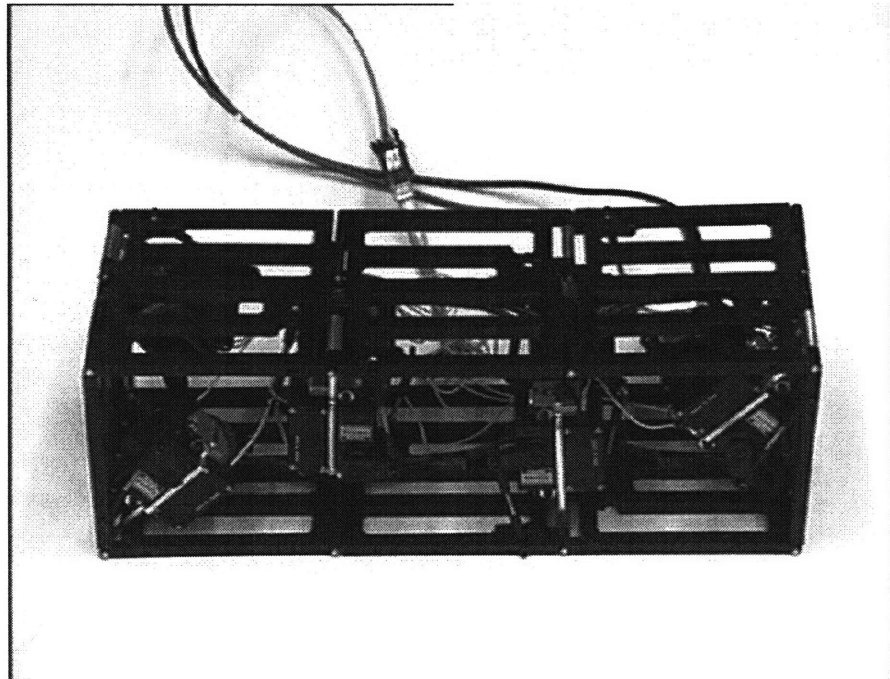
Results of Building the Recti-Blob

II

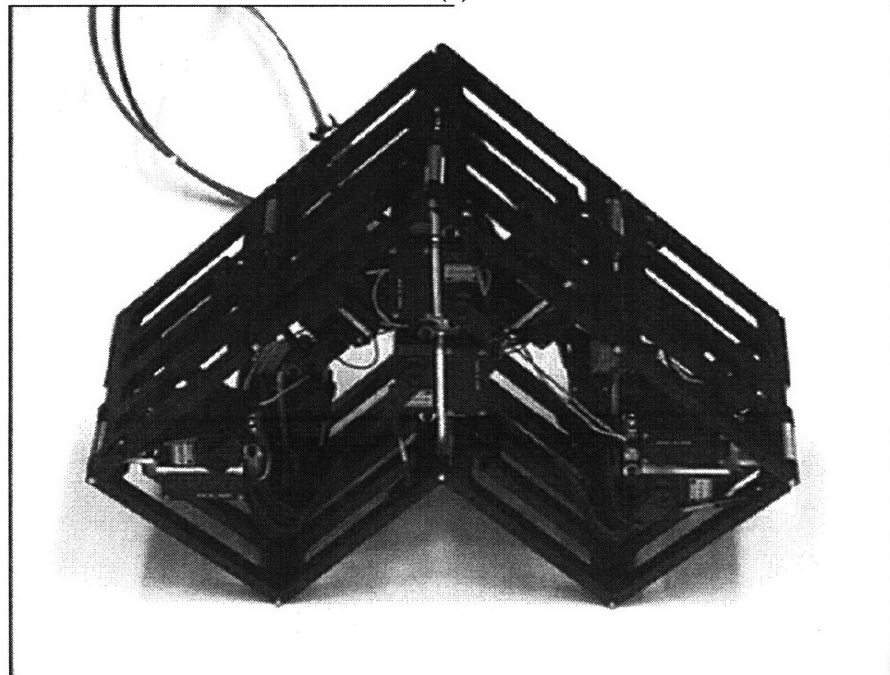
Based on the detailed model which we created using ProE, we were able to generate the necessary drafted drawings from which to create the parts of the Recti-Blob II. After machining all the parts in house, we were able to assemble the robot itself. The links of the Blob II are made of delrin (as opposed to aluminum in the original Blob) and are drilled to save weight. We chose to use delrin over aluminum for the main links because of the substantial weight savings and because of delrin's ability to absorb shock without deformation. We used aluminum for the remainder of the structural components (e.g. the plate on which the actuators are mounted) because it is easy to work with and because of its structural rigidity. The actuators are completely unmodified Futaba S9402 high torque servos. The completed Blob II can be seen in Figure 6-1.

With a completed physical structure, we focused on the hardware necessary to control the Blob II. We selected a controller made by Digital Design & Systems, Inc. (<http://www.dideas.com>) called the ServoX8. This controller consisted of an HC11 microcontroller, a PAL with which to generate the correct PWM signals to control the servos, a serial input port and 8 outputs which can control 8 servos. A single output was connected to the pair of servos at each joint, assuring that the servo tower would bisect the angle between the two links. This controller is currently powered by a separate power supply, but we hope to migrate to a set of 7 standard R/C car ni-cad battery cells (one mounted on each link except for the link which holds the controller).

We have not yet been able to coerce the Blob II to roll; however initial testing is positive. The Blob II is capable of lifting its own weight easily, and the servos appear to be fast enough to enable acrobatics similar to the original Blob. The Blob II is also fully capable of forming both convex and concave joint angles. With some further debugging, we are confident that the Blob II will be able to achieve all of our original goals.



(a)



(b)

Figure 6-1: The Recti-Blob II in a configuration matching that of the original Blob (a) and in a newly possible configuration (b). The second of these should allow the Blob to climb large obstacles such as stairs.

Chapter 7

Conclusions

The simulation, design and construction of the Recti-Blob II was a unique and challenging project. The Blob II, as well as the original Blob, are novel solutions to the problem of robotic locomotion. In the case of the Blob II, we combine the efficiency of a wheel with the ability to navigate rough terrain previously restricted to legged robots. While we have only initially tested the Blob II, we are confident that it will eventually be capable of navigating terrain such as stairs and large objects as well as be able to maintain efficient locomotion completely autonomously.

This research also demonstrated the following:

- Simulation provides an excellent opportunity to explore numerous possible physical realizations before a single design or strategy is chosen. Additionally, careful attention to the discrepancies between simulation and the real world substantially streamlines the physical implementation of a system. In the case of the Blob II, we were able to compare an existing robot with a simulated version to determine possible limitations of the simulator. We were then able to explore a variety of different Blob II designs rather than physically creating each of them.
- A thorough and detailed design facilitates a straightforward and easily built physical realization. Before a single part was machined, the Blob II was modeled in the SDFAST simulator and designed using Pro/Engineer. As a result, the final construction of the Blob II was completed in under five days and after the original unsuccessful prototype, no unexpected problems were encountered.
- Any system or robot which will be created physically should be overdesigned. Overdesigning implies that all aspects of the physical system which will be stressed should be designed to tolerate stresses substantially above the expected levels. Specifically, it is important to overdesign the actuators because they most directly affect the performance of locomotive robots. This principle was evident in the Blob II after the prototyping of the original joint which was only

barely able to lift its own weight.

The Blob II is a significant step towards the goal of efficient, robust locomotion over varying terrain. Through simulation, we found that the Blob II is capable of sustained locomotion that is only slightly less efficient than a wheel on level ground. Additionally, the Blob II should be capable of navigating terrain such as stairs that are on the order of half the Blob's extended height.

These achievements suggest a number of applications for the Blob II as well as future versions of the Blob. With the addition of more links, the Blob should become more efficient at overcoming obstacles. Such a robot might be used as an autonomous device for clearing rugged areas of mines and hazards. In addition, increasing the size and weight of the Blob tends to increase rolling efficiency (on level ground), suggesting that larger Blobs may more closely approach the efficiency of a wheel. If a Blob were designed with the capability to carry cargo, we could imagine replacing the wheels on a motorized wheel chair with two, independently controlled Blobs. Such a device could be powered by current battery technology because of the efficiency of the "wheels." However, it would allow the rider access to terrain far more rugged than currently possible. Ultimately, the conclusions drawn from simulating and building the Blob II should provide the necessary framework to achieve all of these proposed applications.

Appendix A

C Code Describing an Eight Link Recti-Blob

```
/* create blob.c */

#include <clib.h>

#include "blob.h"

main(argc, argv)
    int argc;
    char *argv[];
-
command_line(argc, argv);

set_track_offset("ls.c.m.x", "ls.c.m.y",
    "ls.c.m.z");

begin_species("blob");

/* The definition of a new link with mass
    LINK'MASS, width LINK'W, and length LINK'L */

new_link("link1");
set_mass(LINK'MASS/2);
begin_shape();
use_color("blackmetal'material");
translate(0.0, LINK'L/2, -0.005);
shape(SBRICK, LINK'W, LINK'L, 0.01);
end_shape();

/* A hack to place the 1st and 9th link in the
    same physical space because the simulator
    cannot handle closed loops of links */

make_actuator_connection("weld1", 1, 0.0,
    0.01, 0.0);

make_actuator_connection("weld2", 2, 0.0,
    (LINK'L - 0.01), 0.0);

/* defining places on the link that will interact
    with the ground */

ground_contact("l1'1", LINK'W/2, 0.0, -0.005);
ground_contact("l1'2", -LINK'W/2, 0.0, -0.005);
ground_contact("l1'3", 0.0, LINK'L/2, -0.005);
ground_contact("l1'4", LINK'W/2, LINK'L, -0.005);
ground_contact("l1'5", -LINK'W/2, LINK'L, -0.005);

/* The definition of a joint...each joint
    also has an actuator associated with it */

joint_pin("joint1", 'x');
limit_pd("joint1", 0.0, 1.57079327, "ls.k'stop",
    "ls.b'stop");
servo(1, "joint1", PD'FF'SERVO, "ls.k'joint",
    "ls.b'joint", "ls.ff'joint1");
set_joint_offset(0.0, LINK'L, 0.0);

/* And we repeat the process...7 more times */

new_link("link2");
set_mass(LINK'MASS);
begin_shape();
translate(0.0, LINK'L/2, -0.005);
shape(SBRICK, LINK'W, LINK'L, 0.01);
end_shape();
ground_contact("l2'1", LINK'W/2, 0.0, -0.005);
ground_contact("l2'2", -LINK'W/2, 0.0, -0.005);
ground_contact("l2'3", 0.0, LINK'L/2, -0.005);
ground_contact("l2'4", LINK'W/2, LINK'L, -0.005);
ground_contact("l2'5", -LINK'W/2, LINK'L, -0.005);
```

```

joint`pin("joint2", 'x');
limit`pd("joint2", 0.0, 1.570796327, "ls.k`stop",
"ls.b`stop");
servo(1, "joint2", PD`FF`SERVO, "ls.k`joint",
"ls.b`joint", "ls.ff`joint2");
set`joint`offset(0.0, LINK`L, 0.0);

```

```

new`link("link3");
set`mass(LINK`MASS);
begin`shape();
translate(0.0, LINK`L/2, -0.005);
shape(SBRICK, LINK`W, LINK`L, 0.01);
end`shape();
ground`contact("13`1", LINK`W/2, 0.0, -0.005);
ground`contact("13`2", -LINK`W/2, 0.0, -0.005);
ground`contact("13`3", 0.0, LINK`L/2, -0.005);
ground`contact("13`4", LINK`W/2, LINK`L, -0.005);
ground`contact("13`5", -LINK`W/2, LINK`L, -0.005);

```

```

joint`pin("joint3", 'x');
limit`pd("joint3", 0.0, 1.570796327, "ls.k`stop",
"ls.b`stop");
servo(1, "joint3", PD`FF`SERVO, "ls.k`joint",
"ls.b`joint", "ls.ff`joint3");
set`joint`offset(0.0, LINK`L, 0.0),

```

```

new`link("link4");
set`mass(LINK`MASS);
begin`shape();
translate(0.0, LINK`L/2, -0.005);
shape(SBRICK, LINK`W, LINK`L, 0.01);
end`shape();
ground`contact("14`1", LINK`W/2, 0.0, -0.005);
ground`contact("14`2", -LINK`W/2, 0.0, -0.005);
ground`contact("14`3", 0.0, LINK`L/2, -0.005);
ground`contact("14`4", LINK`W/2, LINK`L, -0.005);
ground`contact("14`5", -LINK`W/2, LINK`L, -0.005);

```

```

joint`pin("joint4", 'x');
limit`pd("joint4", 0.0, 1.570796327, "ls.k`stop",
"ls.b`stop");
servo(1, "joint4", PD`FF`SERVO, "ls.k`joint",
"ls.b`joint", "ls.ff`joint4");
set`joint`offset(0.0, LINK`L, 0.0);

```

```

new`link("link5");
set`mass(LINK`MASS);
begin`shape();
translate(0.0, LINK`L/2, -0.005);
shape(SBRICK, LINK`W, LINK`L, 0.01);
end`shape();

```

```

ground`contact("15`1", LINK`W/2, 0.0, -0.005);
ground`contact("15`2", -LINK`W/2, 0.0, -0.005);
ground`contact("15`3", 0.0, LINK`L/2, -0.005);
ground`contact("15`4", LINK`W/2, LINK`L, -0.005);
ground`contact("15`5", -LINK`W/2, LINK`L, -0.005);

```

```

joint`pin("joint5", 'x');
limit`pd("joint5", 0.0, 1.570796327, "ls.k`stop",
"ls.b`stop"),
servo(1, "joint5", PD`FF`SERVO, "ls.k`joint",
"ls.b`joint", "ls.ff`joint5");
set`joint`offset(0.0, LINK`L, 0.0);

```

```

new`link("link6");
set`mass(LINK`MASS);
begin`shape();
translate(0.0, LINK`L/2, -0.005);
shape(SBRICK, LINK`W, LINK`L, 0.01);
end`shape();
ground`contact("16`1", LINK`W/2, 0.0, -0.005);
ground`contact("16`2", -LINK`W/2, 0.0, -0.005);
ground`contact("16`3", 0.0, LINK`L/2, -0.005);
ground`contact("16`4", LINK`W/2, LINK`L, -0.005);
ground`contact("16`5", -LINK`W/2, LINK`L, -0.005);

```

```

joint`pin("joint6", 'x');
limit`pd("joint6", 0.0, 1.570796327, "ls.k`stop",
"ls.b`stop");
servo(1, "joint6", PD`FF`SERVO, "ls.k`joint",
"ls.b`joint", "ls.ff`joint6");
set`joint`offset(0.0, LINK`L, 0.0);

```

```

new`link("link7");
set`mass(LINK`MASS);
begin`shape();
translate(0.0, LINK`L/2, -0.005);
shape(SBRICK, LINK`W, LINK`L, 0.01);
end`shape();
ground`contact("17`1", LINK`W/2, 0.0, -0.005);
ground`contact("17`2", -LINK`W/2, 0.0, -0.005);
ground`contact("17`3", 0.0, LINK`L/2, -0.005);
ground`contact("17`4", LINK`W/2, LINK`L, -0.005);
ground`contact("17`5", -LINK`W/2, LINK`L, -0.005);

```

```

joint`pin("joint7", 'x');
limit`pd("joint7", 0.0, 1.570796327, "ls.k`stop",
"ls.b`stop");
servo(1, "joint7", PD`FF`SERVO, "ls.k`joint",
"ls.b`joint", "ls.ff`joint7");
set`joint`offset(0.0, LINK`L, 0.0);

```

```

new`link("link8");
set`mass(LINK`MASS);
begin`shape();
translate(0.0, LINK`L/2, -0.005);
shape(SBRICK, LINK`W, LINK`L, 0.01);
end`shape();
ground`contact("l8`1", LINK`W/2, 0.0, -0.005);
ground`contact("l8`2", -LINK`W/2, 0.0, -0.005);
ground`contact("l8`3", 0.0, LINK`L/2, -0.005);
ground`contact("l8`4", LINK`W/2, LINK`L, -0.005);
ground`contact("l8`5", -LINK`W/2, LINK`L, -0.005);

joint`pin("joint8", 'x');
limit`pd("joint8", 0.0, 1.570796327, "ls`k`stop",
    "ls.b`stop");
servo(1, "joint8", PD`FF`SERVO, "ls.k`joint",
    "ls.b`joint", "ls.ff`joint8");
set`joint`offset(0 0, LINK`L, 0.0);

new`link("link9");
set`mass(LINK`MASS/2);
begin`shape();
translate(0.0, LINK`L/2, -0.005);
shape(SBRICK, LINK`W, LINK`L, 0.01);
end`shape();
make`actuator`connection("weld1", 2, 0.0,
    0 01, 0 0);
make`actuator`connection("weld2", 1, 0 0,
    (LINK`L - 0.01), 0.0);

end`species();

```


Appendix B

C Code Fragments Used to Control the Blob and Blob II

The code which was used to implement the controller on the previous Blob.

```
case 3:
    if((ls.t - ls.temp't != ls.step't)
        -
ls.temp't = ls.t;
ls.temp'force = ls.ff'joint8;
ls.ff'joint8 = ls.ff'joint7;
ls.ff'joint7 = ls.ff'joint6;
ls.ff'joint6 = ls.ff'joint5;
ls.ff'joint5 = ls.ff'joint4;
ls.ff'joint4 = ls.ff'joint3;
ls.ff'joint3 = ls.ff'joint2;
ls.ff'joint2 = ls.ff'joint1;
ls.ff'joint1 = ls.temp'force;
    "
    return,
    break;
```

The three code fragments used in the open loop controller of the Blob II.

```
/* the routine which rotates the angles */

void
rotate()
-
ls.temp'pos = q'd.joint8;
q'd.joint8 = q'd.joint7;
q'd.joint7 = q'd.joint6;
q'd.joint6 = q'd.joint5;
```

```

q'd.joint5 = q'd.joint4;
q'd.joint4 = q'd.joint3;
q'd.joint3 = q'd.joint2;
q'd.joint2 = q'd.joint1;
q'd.joint1 = ls.temp'pos;

"

/* a routine used to return the value of
   a given joint angle from the simulator */

double
ith'joint(i)

    int i;

-
switch (i) -

case 1:
    return q.joint1;
    break;

case 2:
    return q.joint2;
    break;

case 3:
    return q.joint3;
    break;

case 4:
    return q.joint4;
    break;

case 5:
    return q.joint5;
    break;

case 6:
    return q.joint6;
    break;

case 7:
    return q.joint7;
    break;

case 8:
    return q.joint8;
    break;
"

```

```
/* the actual controller */

case 1:
    if (ith_joint((cur_index % 8) + 1) > .41)
        -
rotate();
cur_index = cur_index + 1;
ls.cur_index = (double) cur_index;
ls.steps = ls.steps + 1.0;
    "
    return;
    break,
```

