

ABSTRACT

Title of Thesis: DESIGN, ANALYSIS, AND FABRICATION OF A SNAKE-INSPIRED ROBOT WITH A RECTILINEAR GAIT

Degree Candidate: Brent William Spranklin

Degree and Year: Master of Science, 2006

Thesis Directed By: Associate Professor Satyandra K. Gupta
Department of Mechanical Engineering

Snake-inspired robots display promise in areas such as search, rescue and reconnaissance due to their ability to locomote through tight spaces. However, several specific issues regarding the design and analysis must be addressed in order to better design them. This thesis develops kinematic and dynamic models for a class of snake-inspired gait known as a rectilinear gait, where mechanism topology changes over the course of the gait. A model using an Eulerian framework and Coulomb friction yields torque expressions for the joints of the robot. B-spline curves are then used to generate a parametric optimization formulation for joint trajectory generation. Exact gradient computation of the torque functions is presented. A parametric model is used to describe the performance effects of changing system parameters such as mass, length, and motor speed. Finally, a snake-inspired robot is designed and fabricated in order to demonstrate both the vertical rectilinear gait and a modular, molded design aimed at reducing the cost of fabrication.

DESIGN, ANALYSIS, AND FABRICATION OF A SNAKE-INSPIRED ROBOT
WITH A RECTILINEAR GAIT

By

Brent William Spranklin

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2006

Advisory Committee:

Associate Professor Satyandra K. Gupta, Chair/Advisor
Associate Professor Hugh A. Bruck
Dr. Greg Schultz

© Copyright by
Brent William Spranklin
2006

Acknowledgments

First, I would like to thank my advisor, Dr. S.K. Gupta, for his continued guidance and support during both my undergraduate and graduate education. He taught me a lot, and was always there to motivate me, and for that I will always be grateful.

I would also like to thank Dr. Bruck for his assistance and guidance throughout my education, and Dr. Schultz for taking the time to serve on my committee.

I would like to thank all of my lab-mates and people who collaborated on this project. I had a terrific time collaborating with Lawrence “J.R.” Gyger and James Hopkins on this work, and would like to wish them luck as they complete their academic careers. I would also like to thank my undergraduate assistants Navid Charoosheh, Chris Jinks, and Ben Bernstein for the many hours they contributed. In addition, I would like to thank the rest of the people I had the pleasure to work with; especially those who helped me proofread this thesis.

Finally, I would like to thank my family and friends. I would like to thank my parents for instilling the value of education in me, and my brothers for always being there for me. I would also like to thank Lauren for always sticking by me, and sometimes accompanying me on those late evenings and weekends in the lab.

Table of Contents

Acknowledgments.....	ii
Table of Contents.....	iii
Table of Figures.....	v
Chapter 1 - Introduction.....	1
1.1 Background.....	1
1.1.1 Snakes.....	1
1.1.2 Advantages and Disadvantages of Snake-Inspired Locomotion.....	2
1.1.3 What Can Snakes Offer to Engineers?.....	5
1.1.4 Possible Applications of Snake-Inspired Robots.....	6
1.2 Motivation.....	10
1.2.1 Kinematic and Dynamic Model.....	10
1.2.2 Gait Design.....	12
1.2.3 Improved Mechanical Design.....	12
1.3 Thesis Goal and Scope.....	14
1.4 Organization.....	15
Chapter 2 - Related Work.....	17
2.1 Overview.....	17
2.2 Snake Locomotion.....	17
2.3 Snake-Inspired Robot Designs for Search and Rescue.....	25
2.4 Snake-Inspired Robot Mechanics: Kinematics and Dynamics.....	44
2.5 Rectilinear Gait Generation.....	52
2.6 Robot Trajectory Optimization.....	54
2.7 Summary.....	57
Chapter 3 - Gait Development and Analysis.....	59
3.1 Introduction.....	59
3.2 Background.....	60
3.2.1 Basic Robotics.....	60
3.2.2 Joint Trajectories.....	63
3.2.3 Exact vs. Redundant Actuation.....	64
3.2.4 B-Spline Functions.....	65
3.3 Gait Overview and Rationale.....	67
3.3.1 Drawbacks of a Serpentine Gait.....	68
3.3.2 Advantages of a Rectilinear Gait.....	70
3.3.3 Gait Sequence.....	71
3.4 Kinematics and Dynamics Model.....	74
3.4.1 Gait Kinematics.....	75
3.4.2 Dynamics.....	83
3.4.3 Mechanism 2 (M_2).....	89
3.4.4 Mechanism 3 (M_3).....	92
3.4.5 Steps 4 and 5.....	93
3.4.6 Mapping Solutions.....	94
3.5 Trajectory Generation.....	96
3.5.1 B-Spline Trajectory Formulation.....	97

3.5.2 Trajectory Optimization.....	97
3.5.3. A Heuristic-Based Trajectory Generation Approach.....	105
3.6 Results.....	111
3.7 Validation.....	123
3.8 Summary.....	125
Chapter 4 - Parametric Study.....	127
4.1. Introduction.....	127
4.2. Mission Requirements	127
4.3. Discussion of Varied Parameters	129
4.4 Discussion of Results.....	132
4.5 Summary.....	143
Chapter 5 - Snake-Inspired Robot Realization	145
5.1 Introduction.....	145
5.2 Design Goals and Objectives	145
5.3 Structural Design	148
5.3.1 Molded Structure	149
5.3.2 Multi-Material Molding.....	151
5.4 Component Selection.....	157
5.4.1 Actuators	157
5.4.2 Power	159
5.4.3 Control System.....	162
5.5 Overall Module Mechanical Design and Manufacturing.....	167
5.6 Locomotion Results	177
5.7 Next Generation Snake: Embedded Components.....	179
5.8 Summary.....	183
Chapter 6 - Conclusions.....	184
6.1 Contributions.....	184
6.2 Anticipated Benefits.....	185
6.3 Future Work.....	186
6.3.1 Model Improvement and Validation	186
6.3.2 Improved Prototype Design	187
6.3.3. Generation of Optimal Gaits	189
Appendix A: Equations.....	190
Appendix B. Design Information.....	197
References.....	203

Table of Figures

Figure 1.1 Python regius, a common snake.	2
Figure 1.2. Rescuers search for survivors at a disaster scene [1].	7
Figure 1.3. Example of an industrial pipe structure [2].	9
Figure 1.4. Chart of a possible overall system model.	11
Figure 2.1. Serpentine Locomotion [9].	19
Figure 2.2. Snake exhibiting serpentine locomotion through a series of vertical pegs. The snake alters its curve proportionally to peg spacing [3].	19
Figure 2.3. Steps 1-7 show the progression of snake as it exhibits concertina locomotion. Portions colored black are stationary while portions colored white are sliding [3].	20
Figure 2.4. Sidewinding Locomotion. The shaded portions are at rest with respect to the ground [3]	21
Figure 2.5. Points on the ventral surface of the snake are used to demonstrate rectilinear motion. Heavier points denote static contact while the lighter points denote sliding contact [4].	23
Figure 2.6. Rectilinear Motion. In step one, region A is at maximum contraction, region B is contracting, region C is at maximum elongation, and region D is elongating. The progression of the wave is seen through the motion steps of the gait [4].	24
Figure 2.7. Active Cord Mechanism by Hirose [9].	27
Figure 2.8. ACM traveling through a track using sensors [9].	29
Figure 2.9. Hirose's ACM-R3 [10].	30
Figure 2.10. One link of Dowlings snake-inspired robot [13].	33
Figure 2.11. Dowlings snake-inspired robot [13].	34
Figure 2.12. Actuated universal joint design by Wolf et al [18].	36
Figure 2.13. Gavin Miller's S5 snake-inspired robot [19].	37
Figure 2.14. Omnitread robot climbing up a step [21].	41
Figure 2.15. Nomenclature for Hirose's equations [9].	46
Figure 2.16. Model of muscle structure in snake joint [9].	47
Figure 2.17. Parameters in Chirikjian and Burdick's approach [24].	49
Figure 2.18. Burdick and Ostrowski model [25].	50
Figure 2.19. Merino and Tosunglu's Gait [34].	53
Figure 2.20. Initial path, 2-link manipulator [37]	56
Figure 2.21. Optimal solution, 2-link manipulator (working against gravity) [37].	56
Figure 3.1. Representation of gait analysis and development procedure.	60
Figure 3.2. Basic robot link representation.	61
Figure 3.3. Closed-loop (left) and open-loop (right) configurations of a snake-inspired robot.	62
Figure 3.4. Gait presented by Merino and Tosunoglu [34].	72
Figure 3.5. Modified traveling wave gait, steps and mechanisms are identified.	74
Figure 3.6. The first gait step.	76
Figure 3.7. Representation of the first mechanism as a open-loop, planar, kinematic chain.	77

Figure 3.8. Representation of the first mechanism as a closed-loop, planar, kinematic chain.....	79
Figure 3.9. Free body diagrams for the links in M_1	87
Figure 3.10. Gait step 2, which is modeled with mechanism 2 (M_2).....	90
Figure 3.11. Model of M_2	91
Figure 3.12. Gait step 3, which can be modeled with mechanism 3 (M_3).....	93
Figure 3.13. Architecture of fitness evaluation function – COMPUTE_DYN(C)..	109
Figure 3.14. Organization of search algorithm.	110
Figure 3.15. Effort vs. iterations for step 1 search.....	112
Figure 3.16. Resultant trajectory function for joint 1.	114
Figure 3.17. Resultant trajectory function for joint 2.	114
Figure 3.18. Snapshots of gait step 1, shown at 0.1s intervals.	115
Figure 3.19. Torque versus time for joint 1.	116
Figure 3.20. Torque vs. time for step 2.....	117
Figure 3.21. Effort vs. iterations for step 2.	118
Figure 3.22. Resultant trajectory for joint 1.....	119
Figure 3.23. Resultant trajectory for joint 2.....	119
Figure 3.24. Resultant trajectory for joint 3.....	120
Figure 3.25. Snapshots of gait step 2, shown at 0.1s intervals.	121
Figure 3.26. Torque vs. time for joint 1.....	122
Figure 3.27. Torque vs. time for joint 2.....	122
Figure 3.28. Torque vs. time for joint 3.....	123
Figure 3.29. Snapshots of gait.....	125
Figure 4.1. Mapping between design parameters and performance.	129
Figure 4.2. Varied parameters. The gait angles occur at the transition points, or “steps.”	132
Figure 4.3. Trajectories generated for step 2 at different step times.....	134
Figure 4.4. Torque 2 would be greater than Torque 1.	134
Figure 4.5. Sample two-link manipulator [44].....	136
Figure 4.6. Effort versus step time. Feasible design space for sample exercise shown.	138
Figure 4.7. Effort versus velocity, changing step times. Feasible design space for sample exercise shown.....	138
Figure 4.8. Gait angle versus effort, gait step time = 1s.	139
Figure 4.9. Effort versus velocity, achieved by changing the gait angle. Gait step time is 1s.	140
Figure 4.10. Effort vs. angle when gait step time is 0.1s.....	140
Figure 4.11. Effort vs. velocity (achieved by changing gait angle) when gait step time is 0.1s.	141
Figure 4.12. Effort versus length.	142
Figure 4.13. Effort versus velocity, varying length.	142
Figure 4.14. Effort versus mass.	143
Figure 5.1. Example of a fully-modular snake-inspired robot.....	147
Figure 5.2. Overmolding [52].	152
Figure 5.3. Multi-Shot molding [52].....	153
Figure 5.4. Multi-material molded gimbal mechanism [51].....	155

Figure 5.5. Original rotor assembly [51].	155
Figure 5.6. Redesigned rotor structure: Only 8 parts [51].	156
Figure 5.7. Servomotor.	158
Figure 5.8. Battery technology trends (adopted from [54]).	160
Figure 5.9. Lithium polymer battery.	161
Figure 5.10. Modular circuit architecture.	165
Figure 5.11. Circuit layout. Components IC1 and IC2 are the voltage regulator and the PIC, respectively. C1, C2, and C3 are .1, .33, and .1 μ F capacitors, respectively.	166
Figure 5.12. Circuit board layout, key components are labeled.	167
Figure 5.13. Completed design module.	169
Figure 5.14. Side views of modules.	170
Figure 5.15. Stage 1 of molding for part 1.	172
Figure 5.16. Stage 1 molding part 2.	173
Figure 5.17. Stage 2 molding.	174
Figure 5.18. Second Stage exploded view.	174
Figure 5.19. Robot module.	175
Figure 5.20. Fully assembled modular snake-inspired robot.	176
Figure 5.21. End module.	177
Figure 5.22. Snapshots of locomotion.	179
Figure 5.23. Solid Model of embedded module.	182
Figure 5.24. Embedded module.	182
Figure 5.25. A pager motor that has been partially embedded in polyurethane.	182

Chapter 1 - Introduction

1.1 Background

1.1.1 Snakes

Snakes are diverse creatures that occupy a wide range of habitats. They also have a wide range of locomotive capabilities, ranging from crawling and burrowing to climbing and even swimming. While snakes all have a similar structure, they do exist in a variety of sizes and aspect ratios. For example, snakes such as the Boidae family (Boas and Pythons) tend to have thicker, heavier bodies (Figure 1.1), while snakes in families such as the Leptotyphlopidea family (Thread snakes and Worm snakes) tend to have thinner body types. Snakes also range in length from more than 20 feet for reticulated pythons and anacondas, to substantially less than 1 foot long for many of the smaller varieties.

The design of a snake is a simple structure that is repeated many times. Snakes bodies are elongated forms that consist of a long backbone made of many vertebrae. In fact, there are only three different kinds of bones in the entire snake skeleton: the skull, the vertebrae, and the ribs. Snake backbones consist of 100-400 vertebrae, and the design of each vertebra allows small motions in both the lateral and vertical directions. They do not allow any twisting, however, and thus act as compliant universal joints. Each vertebra itself only allows a very small amount of angular motion, but the motions of many vertebrae allow snakes to drastically curve their bodies. Each vertebra allows rotation of 10-20 degrees in the horizontal plane, and between 2-3 degrees in the vertical plane.

While some of the subtleties and details of snake locomotion are not fully understood, the basic forms of forward locomotion can be grouped into several different classes of gaits. The gait that a particular snake may use depends on the type of snake, the terrain that it is traveling over, and the speed at which the snake desires to travel. While a gait known as the serpentine gait is most commonly thought of as snake locomotion, this thesis focuses on rectilinear snake locomotion because of the potential benefits of rectilinear locomotion that are discussed in Section 3.3.



Figure 1.1 Python regius, a common snake.

1.1.2 Advantages and Disadvantages of Snake-Inspired Locomotion

Snake locomotion provides several advantages over traditional forms of locomotion in both animals and machines. Due to their elongated form and lack of legs, snakes have compact cross-sections and thus can move through very thin holes and gaps. Likewise, snake-inspired devices have much thinner cross sections than other robots with equivalent sizes and capabilities. In addition to the thinner cross section, snakes

also have the ability to climb up and over obstacles that are much taller than their body height. This is done by lifting the front half of their long bodies. Similarly, a snake-inspired robot can lift its body up and over obstacles much larger than most legged or wheeled devices. These properties are very desirable when moving through complex and cluttered environments. Additionally, snakes are also stable. Because their bodies are constantly in contact with the ground at many different points, it is difficult to knock them over, especially since they have a low center of mass and do not lift their bodies off the ground much during locomotion.

Snakes have redundant designs that rely on the same kind of joint (and structure) that is repeated many times. This means that if one joint fails, the snake can continue to locomote. The simplicity of the design also means that the snake does not have any fragile appendages that can easily break.

The form of locomotion that snakes use also relies on a large amount of contact between the ground and the posterior. This large surface area gives the snake good traction characteristics in variable environments. Whereas one wheel or leg in a traditional kind of robot may slip, the large contact surface of a snake-inspired robot would make this occurrence less likely.

Snakes are very versatile and can act as both locomotors and manipulators, as they can use their bodies to wrap around objects to grasp them. This can be seen in the climbing action across tree branches, or when a constrictor is clenching its prey. Since one structure can do both things, the need for different mechanisms to achieve different tasks is eliminated.

Finally, despite frictional opposition to their locomotion, snakes actually have been shown to consume a comparable amount of energy to other biological forms with similar sizes, weights, and speeds. This can be explained by the fact that snakes do not perform a lot of lifting of their body in their motion, and they also do not consume a lot of energy by moving different appendages like legged animals (this is elaborated on in Chapter 2).

With all of these advantages, why are snakes unique in their form of locomotion among animals? Why do the majority of different animals use different forms of locomotion, and why are all mobile robots not based on snakes? The answer is that there are also many disadvantages to snake locomotion as well.

A major disadvantage to snake locomotion is that it is often slower than other forms of locomotion. The fastest snake has a maximum forward speed of 3 m/s (Black Mamba) and many snakes travel much slower. Other wheeled devices and organisms with legs that are similar-sized have the ability to travel much faster. For example, the Prairie Racerunner, a species of lizard, has been clocked at speeds up to 8 m/s.

Another deficiency of snake structure and snake locomotion of interest to the robot designer is that a snake does not have many convenient locations to carry a payload like many other wheeled and legged locomotion platforms. Mobile robots often carry a suite of sensors and actuators, as well as power components. Typically, for a wheeled or legged robot the majority of the body provides a place to carry payload, and the legs or wheels require a comparatively small amount of the volume. With

snake-like robots, however, the entire form is used in the locomotion process, and thus must conform to the shape of the snake.

A final notable drawback of snake locomotion is related to control of the many degrees of freedom. As was previously mentioned, natural snakes can contain as many as 400 vertebrae, each of these constituting two degrees-of-freedom. Many robotic implementations of snake locomotion have had as many as 20 actuated joints. The control (as well as engineering analysis) of a system such as this is not trivial. Current control architectures have difficulty dealing with systems with many degrees-of-freedom.

1.1.3 What Can Snakes Offer to Engineers?

Snake-like robots can offer engineers a novel means of locomotion and a versatile platform design for unmanned robotic systems. The functional requirements for a snake are similar to the functional requirements of many tasks that we wish to achieve with robotics. Snakes must be able to move in environments with many obstacles and be able to place their bodies into tight spaces. Some snakes also must be able to burrow, climb, and even swim. Similarly, unmanned robots must be able to travel in both tight and cluttered environments where humans cannot go for reasons of either safety or size. Users also often want unmanned robots that are stealthy, for example in the case of espionage and reconnaissance. Finally, robots need to be robust enough to complete their task.

1.1.4 Possible Applications of Snake-Inspired Robots

Urban Search and Rescue (USAR)

With the world's population increasing, and more people living in urban environments, both man-made and natural catastrophes are becoming more and more common. This has been illustrated in recent events such as the terrorist attacks on the World Trade Center in 2001, and hurricane Katrina in the United States. Other examples of catastrophes would be the tsunami in southeast Asia that killed almost 200,000 people in December of 2004, and the 2005 earthquake in the Kashmir region of Pakistan.

During disasters in urban areas, it is common for buildings to collapse and for debris to be present, complicating the search and rescue process. Such environments make survivors difficult to find amongst the debris, because they may be buried. Additionally, debris and partially-collapsed buildings can also make the search environment dangerous for both human rescuers and even trained dogs because of the potential for collapse. Even more dangerous would be the case where toxic chemicals or radiation are present in the cleanup site, which could be the result of an event such as a terrorist attack.

In an event where lives are at risk, time is always critical. Rescue workers need up-to-the-minute information about the hazards they face and where the survivors may lie. Other critical information includes the structural integrity of the disaster area, what hazardous materials are present, and the presence of fires or flammable gasses in the rubble.

Yim et al. [1] outline a typical search and rescue operation. A typical search and rescue team consists of approximately 10 people. The team consists of a structural engineer, canine handlers, and “various specialists in handling special equipment to find and extract a victim”. The engineer oversees and judges the structural integrity of the building where the rescue is taking place, while the canine handlers use their dogs to search with their keen senses. Equipment that is commonly used in USAR efforts consists of video cameras mounted on poles that are inserted into crevices to look for survivors (Figure 1.2). Listening devices are also used to hear calls for help, and thermal imaging is used to search for body warmth.



Figure 1.2. Rescuers search for survivors at a disaster scene [1].

Snake-inspired robots would be a valuable aid to such rescue crews in many situations. Robots could be mounted with sensor equipment and allowed to search the rubble in order to relay up-to-the-minute information back to rescuers. Cameras,

microphones, and a variety of sensors could be mounted on snake-inspired robots, and the robots could travel deep into the rubble, obtaining information that might otherwise be difficult for rescuers to obtain. Snake-inspired robots could be used to find survivors in voids deep in the rubble, or to survey the structural integrity of the damaged building.

Snake-inspired robots could travel in areas where it would be impossible, impractical or dangerous to send humans or even dogs. This would be the case because of both their unique attributes, and simply because they are expendable. If a piece of rubble falls on a segment of a snake-inspired robot, damaging it, the robot could still function due to its hyper-redundant characteristics. Because of their characteristics, they could search the rubble faster and more effectively than human rescuers. Furthermore, if many robots are used, the effort can be accelerated even further. This would especially be the case if autonomous robots could be developed that operate in swarms.

Inspection

Another important and beneficial use of snake-inspired robots is inspection tasks. In many circumstances, it is necessary to inspect environments that are either too small for humans to inspect, or are too dangerous to send a human into. Examples of such tasks would vary from the inspection of the ballast in a Navy ship or submarine to the inspection of portions of a reactor in a nuclear power plant. Other examples could range from the unstructured environment of a space station to the tight environment of a pipe network.

The same attributes that would make snake-inspired robots desirable in USAR tasks would also make them desirable in tasks of inspection. Robots could again carry sensor equipment to record and relay information about the use environment. The benefits of a small cross section and stability would come into play when navigating the unstructured and tight environments that may be required in such inspection tasks.

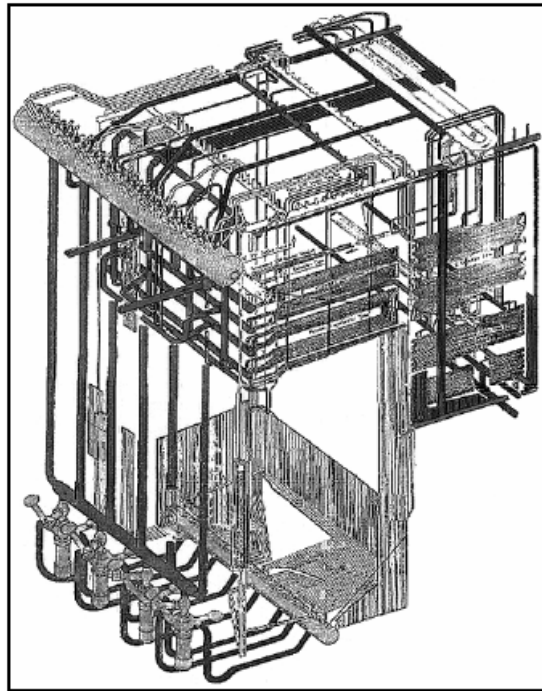


Figure 1.3. Example of an industrial pipe structure [2].

Routing of Cables

In addition to inspecting environments such as pipes, serpentine robots could also be used to route cables through complex pipe structures (Figure 1.3). Cables could be attached to the robot, and the robot could then travel the desired path of the routing, placing the end of the cable at the final destination of the routing.

1.2 Motivation

The motivation for this work can be organized into three areas of improvement for snake-inspired robots: a) *modeling*, b) *gait design*, and c) *mechanical design*.

1.2.1 Kinematic and Dynamic Model

In order to design a snake-inspired robot that utilizes a vertical rectilinear gait, it would be advantageous to have a kinematics and dynamics model for a multi-linked snake-inspired robot that is moving in such a fashion. By being able to compute the torque as a function of time during the entire course of motion for the gait, relevant information about the snake-inspired robot design (metrics such as effort) can be extracted.

In order to develop functional snake-inspired robots, they must be designed to meet certain functional requirements and performance characteristics. These may be characteristics such as maximum forward velocity, range, operation time, payload, and many more. If a design is to be developed to meet such parameters, then a model should be devised to determine how certain design parameters, such as module mass, length, and the gait parameters, affect the performance of the robot. A dynamic model must be developed (with a kinematic model preceding it) that will allow the designer to ascertain the necessary parameters to compute the performance. Such a model could then facilitate optimization, feasibility studies, and scaling of snake-inspired robot designs. For example, if one were designing a robot for a particular mission that required the robot to travel a certain distance in a certain amount of time,

the use of the dynamics calculations would allow one to determine the amount of effort required by the robot, and thus the power supply could be selected accordingly. A kinematics and dynamics model would be a central component of a possible overall system model for the design of a snake-inspired robot like the one shown in Figure 1.4.

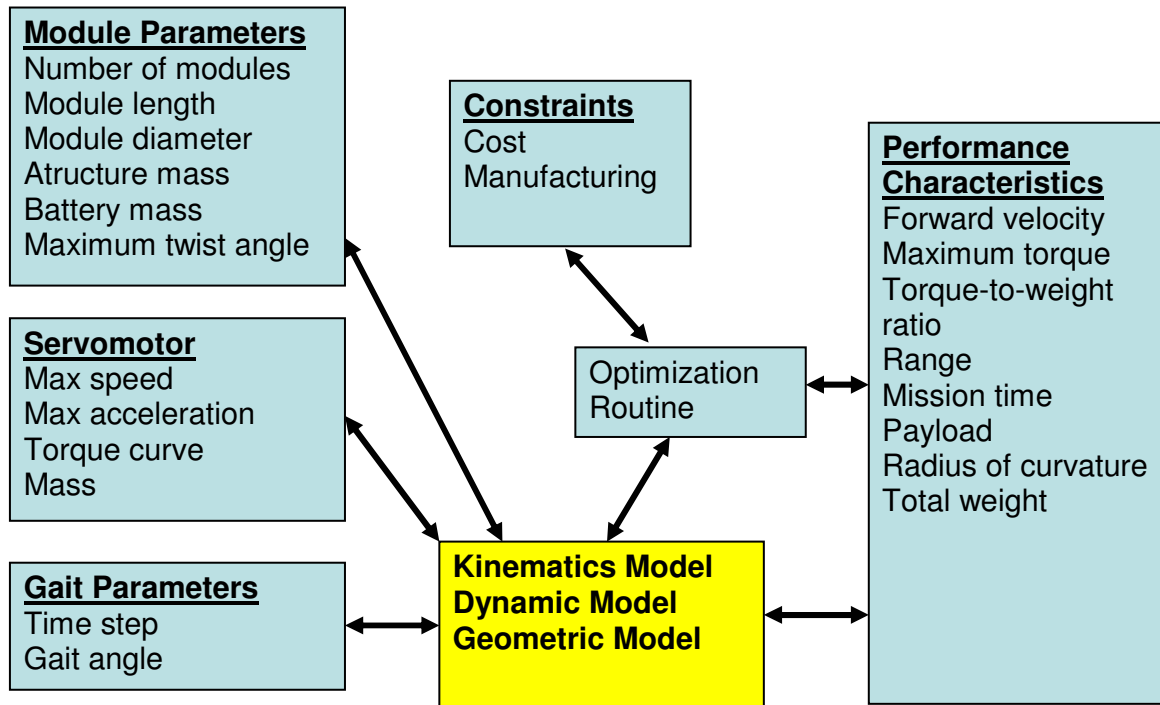


Figure 1.4. Chart of a possible overall system model.

Most of the work in snake-inspired robot kinematics and dynamics has been conducted on snake-inspired robots that move with a serpentine gait (lateral undulation) using wheels or other features on the ventral surface to provide the needed anisotropic friction to achieve forward locomotion. Other robots have been developed and demonstrated that use a rectilinear gait without wheels and have been shown to move through varying terrain. However, a detailed analysis of their

locomotion kinematics and dynamics have not been thoroughly developed. This is due to the fact that the dynamically changing ground contact causes the system to morph during the gait.

1.2.2 Gait Design

While rectilinear gaits have been implemented, it is desirable to have a framework to generate gaits with desirable performance characteristics, or even optimal performance characteristics. A motivation for this work is to develop a highly-controllable means for gait design with an optimization model. Thus, gaits can be easily synthesized that provide desirable characteristics such as requiring less effort.

Earlier gait designs have mostly been based on linear joint trajectories, or joint trajectories that are based on sinusoids. A better joint trajectory representation needs to be developed such that the gaits can be highly tuneable. A parametric gait design is needed such that trajectories can be easily controlled. By using joint trajectories that are highly tunable, the gait design can then be integrated with the dynamics model to design gaits that are optimal.

1.2.3 Improved Mechanical Design

A considerable barrier in the development of snake-inspired robots is that they are currently expensive to produce, as they require manual assembly of many small parts because of the many sections and joints. This is especially important since one of the probable uses for snake-inspired robots, dangerous search and rescue applications, would place the robots in locations where it is likely that they would be destroyed.

Further, it would be desirable to have many snake-inspired robots in such a situation so that many can be deployed at the same time, increasing the amount of ground that is covered by the search. A snake-inspired robot design that is comprised of identical modules could lead to reduced costs if the modules could be mass produced and easily assembled. This means that they could be rapidly assembled for deployment. They could also be assembled to custom requirements such as length. Such modules should be designed and manufactured in a means that reduces both parts and assembly operations, yet still has a rugged enough design to withstand hazardous environments. Most prior work in the field of snake-inspired robots has paid scant attention to part counts and assembly operations, resulting in robots that contain many parts that are manually assembled. Large amounts of assembly operations can lead to high manufacturing costs.

If snake-inspired robots could be developed and manufactured in a low-cost means, they could be considered to be expendable and possibly disposable. This means that search and rescue workers could send many of them into a location to retrieve data, and not have to be concerned with whether or not they are recovered. This would allow search and rescue workers to obtain more data in less time, and allow more focus to be spent on rescuing survivors and securing dangerous areas, instead of retrieving their search and rescue robots.

The final motivation for this work is to develop and demonstrate the operation of a fully-modular snake-inspired robot that has been constructed using a concurrent assembly and fabrication process, multi-stage multi-material molding, aimed at low-cost fabrication. The use of multi-stage multi-material molding would significantly

reduce the part count and assembly operations required to produce each module. The use of mechanically-embedded components could further improve on the design goals by encasing components to protect them from the environment, and eliminating a host of fasteners that are typically required to hold such components in place.

1.3 Thesis Goal and Scope

The goals and scope of this thesis can be organized into the several topics laid out in Sections 1.2.1-1.2.3. First, a formulation for the kinematics and dynamics of a 6-link snake-inspired robot locomoting with a particular class of rectilinear gait on flat terrain is presented. A means of analysis is presented that is conducted by breaking up the motion into separate mechanisms, and a set of equations using an Eulerian framework with a Coulomb friction model is presented to calculate joint torques and effort values. Second, a means of optimizing a gait is presented using the dynamic model that was developed and a B-spline curve representation of the joint trajectories. A simple discrete-sampling based approach is used to present how an optimization procedure would work, and a continuous strategy based on gradients is discussed. Third, a parametric study is conducted to demonstrate how several relevant design parameters affect gait design and performance. Finally, a fully-modular snake-inspired robot is developed that meets the following goals:

1. The robot will demonstrate that the class of locomotion gait that has been studied in this thesis will provide forward locomotion.

2. The robot will demonstrate the desired overall design architecture of a robot system that can be built out of identical modules.
3. The design will lay the foundation for a modular robot constructed with components that are embedded during the molding process, reducing parts, assembly operations, and also resulting in a more ruggedized module.

1.4 Organization

The organization of this thesis can be divided into three major technical sections, in addition to the related work section:

Chapter 2 is a literature survey of subjects relevant to this thesis. The major topics of discussion are snake-inspired robot design, snake-inspired robot kinematics and dynamics, rectilinear gait design for snake-inspired robots, and effort-based optimization for robot joint trajectories.

Chapter 3 contains complete description of the gait synthesis and analysis problem. The general form of the gait must be selected, and the model developed. Once the rigid-body model with morphing topology is defined, then the kinematic and dynamic relations and constraints are developed to produce the complete system of Newton-Euler equations for each step of the gait problem. Utilizing this set of equations, which predicts the torque in each joint, the effort of the gait can be computed. This effort calculation is then used to develop a framework for optimizing the gait as a function of the joint trajectory parameters, by searching for minimum-

effort solutions. The results from a discrete sampling-based solution are presented, with the framework for a continuous solution using direct gradient computation.

In **Chapter 4** of this thesis, a parametric study of the bulk design parameters of the robot is discussed. Physical parameters of the robot and the gait are varied and the effects that they have on the gait performance is evaluated and discussed. This section is important for studying how snake-inspired robots can be modified and optimized for specific mission parameters.

The final technical chapter, **Chapter 5**, describes the physical realization of a snake-inspired robot. The complete design and manufacturing process for the robot is presented. Emphasis is placed on the modular architecture of the design, as well as the novel, low-cost means of manufacturing utilizing multi-material molding. Furthermore, a design that may further reduce assembly costs by using physically embedded components is discussed.

Chapter 2 - Related Work

2.1 Overview

The development and analysis of a snake-inspired robot is a multidisciplinary task. Thus, the body of literature and the previous work that has been done that pertains to snake-inspired robot design, development, analysis, and gait generation can be grouped into several different topics. First, biological snake locomotion is described. Authentic snake locomotion serves as the inspiration for this snake-inspired robot concept, and thus biological snake locomotion must be understood in order to mimic snakes using robotic devices. Second, the design of current snake-inspired robots and their differences, advantages, and disadvantages are discussed. In order to analyze the kinematics and dynamics of a snake-inspired robot, which must be done in order to develop and characterize gaits, the body of work that exists in the area of kinematic and dynamic analysis of snake-inspired robots is investigated. The third section addresses different models. The robot described in this thesis progresses using a rectilinear gait, and this approach to motion builds on prior work. This work is discussed in the fourth section. Finally, because this thesis seeks to provide a framework for optimizing the gait, prior work and different approaches to robot joint trajectory optimization are discussed.

2.2 Snake Locomotion

There are four common and distinct gaits that snakes typically use in terrestrial locomotion. These are known as: a) *serpentine* (lateral undulation), b) *concertina*, c)

sidewinding, and d) *rectilinear progression* [3]. Most snakes are capable of executing all or several of these different forms of locomotion, and typically switch as the conditions require. In certain circumstances, a snake may even use a hybrid combination of more than one gait.

Serpentine Locomotion

Serpentine locomotion, also known as lateral undulation, is the most common form of locomotion used by snakes (Figure 2.1). All snakes are capable of serpentine locomotion, and they frequently use serpentine locomotion when moving through terrain such as grass, stones, and sand. The snake's body moves laterally in a sinusoidal curve that propagates down the snake. In this form of locomotion, every part of the snake's body follows the same path as the snake moves along. The forward propulsion occurs due to forces pushing laterally (normal) against the snake's body. These forces are mostly achieved by the snake pushing its body against obstacles located along its path (Figure 2.2). Obstacles can be large, such as a stick or rock, or small, such as small pebbles and sand. Studies have shown that snakes will alter the curvature of their serpentine waves dependent on the terrain that they are moving over and the location of obstacles in that terrain [3].

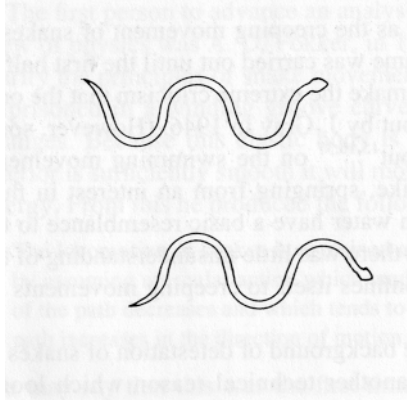


Figure 2.1. Serpentine Locomotion [9].

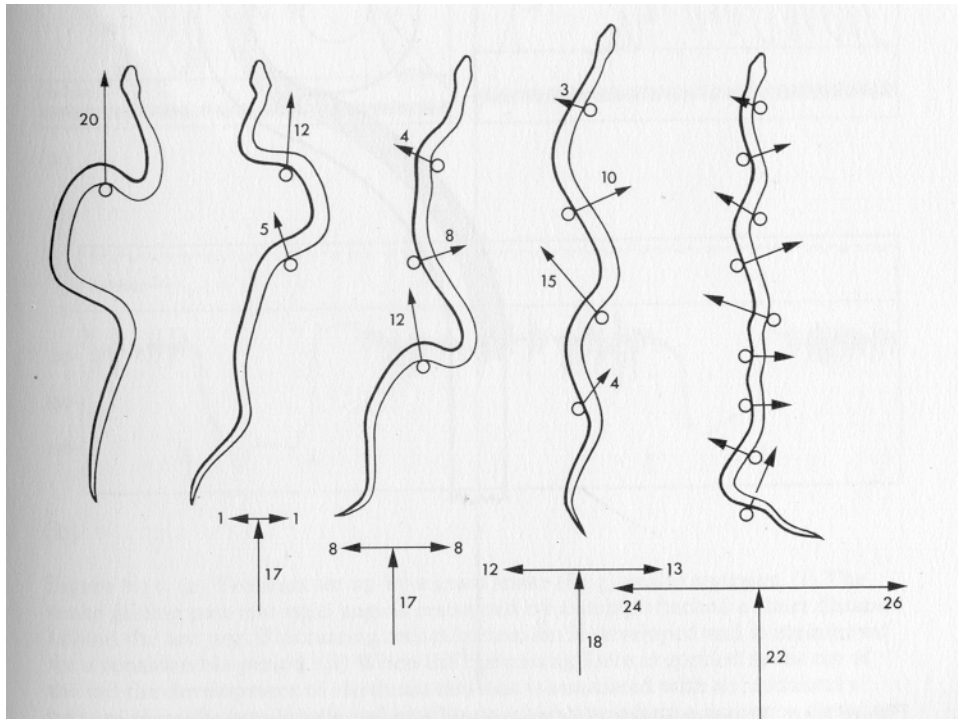


Figure 2.2. Snake exhibiting serpentine locomotion through a series of vertical pegs. The snake alters its curve proportionally to peg spacing [3].

Concertina Locomotion

A second form of locomotion in snakes is known as concertina locomotion. Concertina locomotion is used less frequently than serpentine locomotion, but it is often used in situations where the snake is moving through a thin tunnel or channel. It is also used, although less frequently, on terrain that is rough but uniform. Concertina locomotion is shown in Figure 2.3, and consists of the body configuring itself into short curves. The curvature is then increased, propelling part of the snake forward. The snake then recompresses to repeat the motion. The forward propulsion exists because a portion of the snake remains static with the ground while the other portion is compressing or uncompressing forward. These static contact points are achieved by either the snake exerting pressure on neighboring obstacles or by the ventral scales preventing backward slipping against the ground.

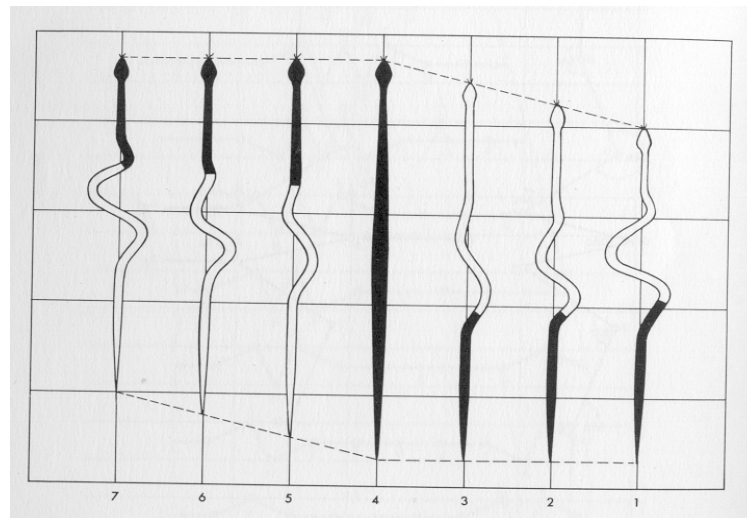


Figure 2.3. Steps 1-7 show the progression of snake as it exhibits concertina locomotion. Portions colored black are stationary while portions colored white are sliding [3].

Sidewinding (Crotaline) Locomotion

A third, and very distinct form of snake locomotion is known as sidewinding. Sidewinding is a form of locomotion that is typical of a certain group of rattlesnakes that live in sandy deserts. Sidewinding is similar to serpentine locomotion in that the sidewinder propagates waves of curvature along the body. However, in sidewinding locomotion, the resultant movement of the snake is sideways with respect to the axis of the body. Figure 2.4 shows tracks that are produced by the sidewinding locomotion that point in the direction of the travel. The snake lifts and rolls its body between the tracks to achieve advancement, as the sections that lie within the tracks are in static contact with the surface. The weight is appropriately transferred to these points to ensure proper friction with the ground. Sidewinding can be considered a specialized gait that is only used on slippery surfaces such as sand.

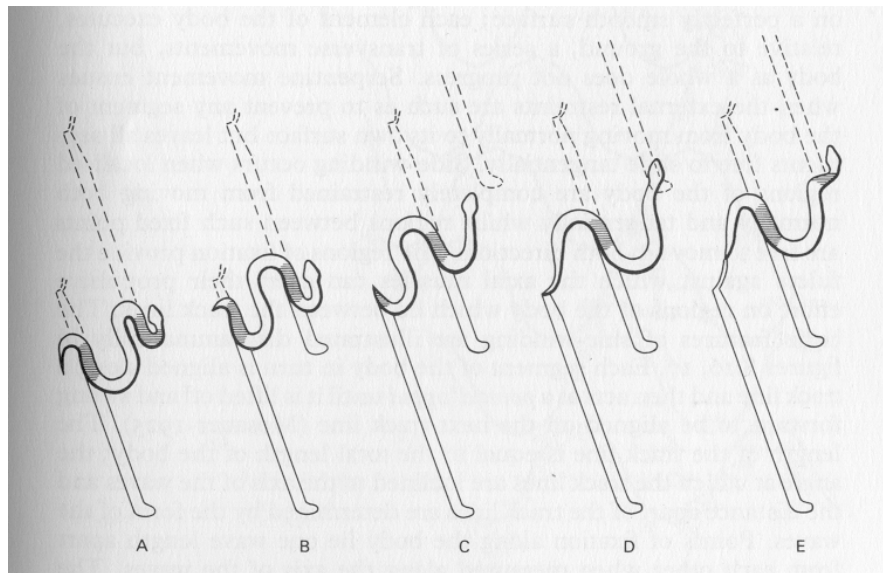


Figure 2.4. Sidewinding Locomotion. The shaded portions are at rest with respect to the ground

[3].

Rectilinear Locomotion

The final common form of snake locomotion is known as rectilinear locomotion. Rectilinear locomotion is a form of locomotion that is common to large snakes with well-developed muscles such as boas and pythons [4]. This form of movement is distinct from the other forms of locomotion in that the snake progresses with its body fully aligned with the direction of movement. Movement is achieved by waves of muscular contraction and expansion passing along the body of the snake. This form of locomotion is best understood by imagining two points located on the ventral (bottom) surface of the snake. With the waves of muscular contraction and expansion, the distance between the two points is oscillating. When the distance between the two points is at a minimum, that segment is at rest. When the distance between the points is either increasing or decreasing, the moving point is moving forward. This is achieved by the frictional characteristics between the snake and the surface, and can be thought of as a “ratcheting” action. The points on the ventral surface move forward in discrete steps. However, the top of the body moves continuously because of the changing geometry of the muscular segments. Figure 2.5 shows marked points on the ventral surface as the snake progresses forward. Figure 2.6 illustrates the muscular contraction and expansion that occurs during rectilinear motion.

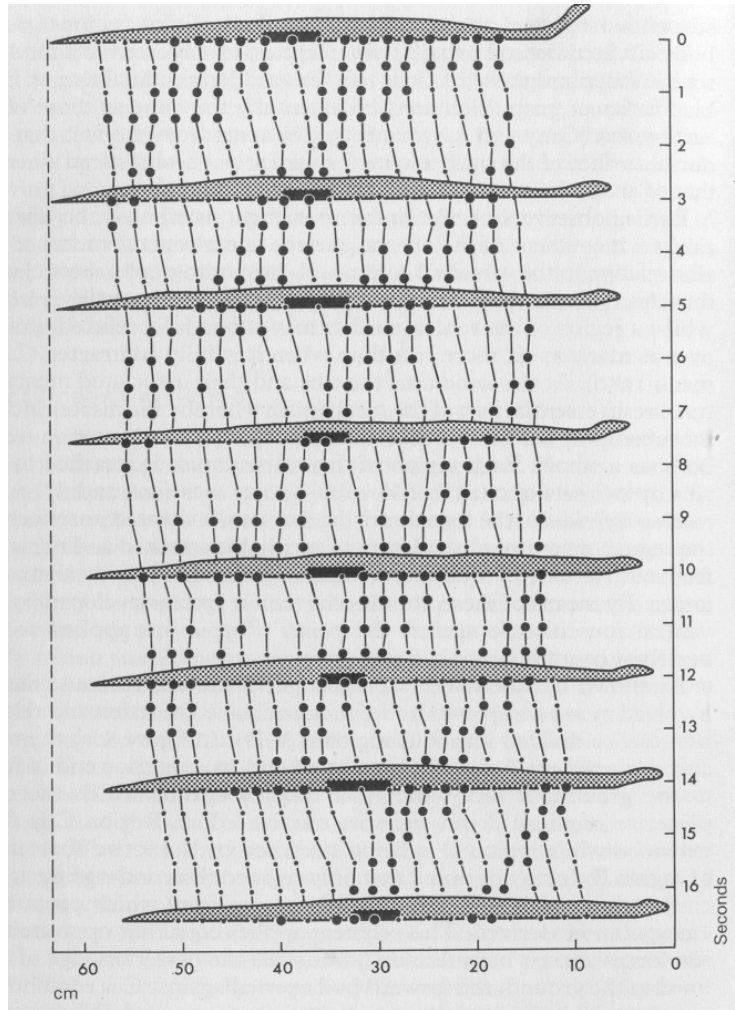


Figure 2.5. Points on the ventral surface of the snake are used to demonstrate rectilinear motion.

Heavier points denote static contact while the lighter points denote sliding contact [4].

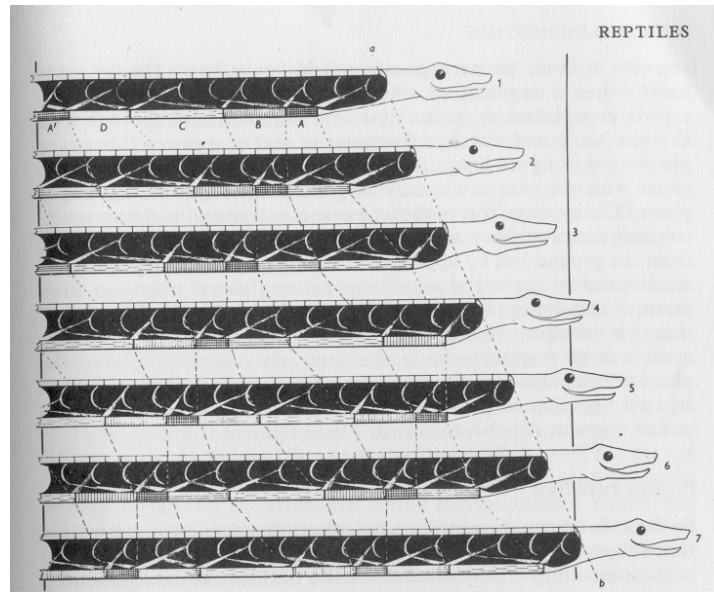


Figure 2.6. Rectilinear Motion. In step one, region A is at maximum contraction, region B is contracting, region C is at maximum elongation, and region D is elongating. The progression of the wave is seen through the motion steps of the gait [4].

Energy Consumption

From a biomimetic perspective, it is easy to see why snake-like locomotion would be of interest to the robot designer. Snake-inspired robots have many advantages over other forms of legged and wheeled locomotion, such as redundancy and a small cross section. However, it is widely believed that snake locomotion is costly from an energy perspective because of its reliance on slipping and the fact that it uses significant lateral motion to achieve a much smaller amount of forward motion. Studies conducted on a black racer (*Coluber constrictor*) using the standard oxygen consumption test have indicated that this is a misconception [5]. These studies have shown that the net cost of transport for a snake locomoting via lateral undulation is similar to the net cost of transport for similar-sized lizards and mammals. However,

concertina locomotion was shown to be significantly more costly. The tradeoff lies within the fact that concertina locomotion is often used in tight environments such as tunnels. Likely explanations for the similar cost of transport between limbless and limbed locomotion are that while snakes lose energy to friction, limbed creatures expend energy lifting their center of mass and accelerating limbs.

Discussion and Relevance

The understanding of snake locomotion in nature gives us several candidate means of locomotion to use for a snake-inspired robot. By observing snakes in nature, we can determine the mechanism they use to achieve propulsion, and how they modify these mechanisms as the terrain changes. In this thesis, a gait directly inspired by rectilinear locomotion in snakes is selected for study because it depends on static friction and allows robots to fit through narrow passageways.

2.3 Snake-Inspired Robot Designs for Search and Rescue

This section seeks to shed light on the field of snake-inspired robots designed for information-gathering applications, such as urban search and rescue and inspection. A considerable amount of work has been done in the field of snake-inspired robotics. However, many of the designs are somewhat similar and many of the goals of the robot realization are different (for example, to provide a platform for a new actuation technology). Therefore, this section is not a complete review of snake-inspired or serpentine robots. Rather, the discussion is limited to designs that can be considered “pioneering” works, designs that have been specifically designed for meeting the

requirements of a search and rescue mission, and designs that can be considered somewhat “novel”. It should be noted, however, that as of this point, snake-inspired robots have not been fully developed and used in real search and rescue environments.

The case for snake-inspired robots has been made by several different researchers. Hirose envisioned a search and rescue paradigm which he calls “snakes and strings” [6]. Similarly, Gavin Miller presented a similar case for such robots [7]. He envisioned a detailed search and rescue scenario whereby a snake-inspired robot could be equipped with an array of sensors such as infrared, pyroelectric, cameras, and microphones. The robot was teleoperated by a search and rescue worker and used to find a survivor. As discussed in Chapter 1, snake-like robots can traverse into areas that humans and dogs can not, obtaining information that is vital to the search and rescue effort, such as site structural integrity and survivor locations. Yim et al. discussed the same mission as a need to develop self-reconfigurable robots [1]. They developed a robot called the PolyBot that could locomote using a snake-like gait and discussed how it could be used in search and rescue applications. While the literature search focuses only on snake-like robots, it is important to note that many researchers developing this other class of robots have also shown their robots to be able to locomote with snake-like gaits [8].

Active Cord Mechanism

Shigeo Hirose is considered one of the pioneers in snake-like robots, and his original robot called the Active Cord Mechanism (ACM) was the first functional snake-like locomotor (Figure 2.7) [9]. The purpose of Hirose's first ACM was to understand the mechanism of locomotion in real snakes. Following many studies on real snakes to ascertain the mechanisms of locomotion, Hirose developed the ACM to validate the work.

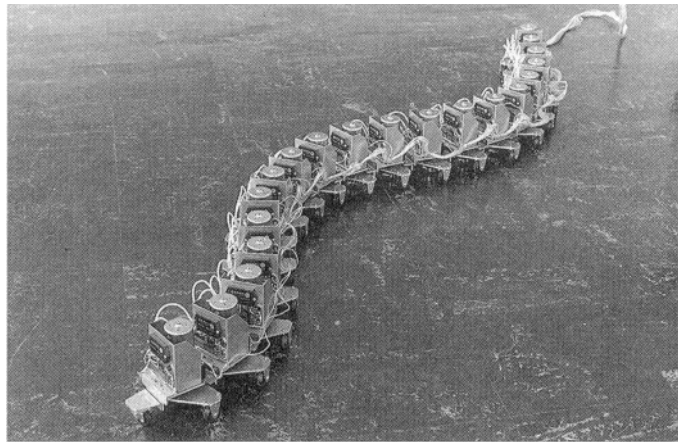


Figure 2.7. Active Cord Mechanism by Hirose [9].

The first ACM consisted of 20 links, and had movement in only two dimensions. This means that it glided along the floor using a serpentine gait. At the core of Hirose's theories about snake locomotion was the fact that snakes produce an anisotropy in friction coefficients between the lateral and tangential frictions on their ventral surface. This is what causes the propulsion in the serpentine gait. In order to realize this in a robot, Hirose placed small wheels on casters on the bottom of each link, facing in the tangential direction of the length of the robot. This resulted in a

very low friction coefficient in the tangential direction, and a high coefficient in the normal direction that the snake uses to propel itself. The links were connected using joints that allow rotation to each other, and locomotion was accomplished by rotating the wheel-base mechanism back and forth. This meant that locomotion was only accomplished through shape changing, like a real snake. The entire robot weighed 28 kg, and was 2 meters long. Each joint was actuated using a servosystem that consisted of a motor and a potentiometer. Control was achieved via a system whereby a command was sent to the first motor, executed, and then sent to the next motor to be executed.

After demonstrating locomotion on a flat surface with no obstacles, Hirose demonstrated how snakes alter their path when obstacles are present by conforming to their environment, and how they use obstacles to propel themselves. This was achieved by adding binary tactile sensors on the lateral sides of each link. In these experiments, Hirose demonstrated how the robot could fully conform to a shape. Work also was done to demonstrate how a snake could propel itself through an abstractly-winding track using only pressure from the walls (Figure 2.8). In this experiment, casters that could roll in 360 degrees were placed on the bottom of the links so that there was not anisotropy of friction coefficients.

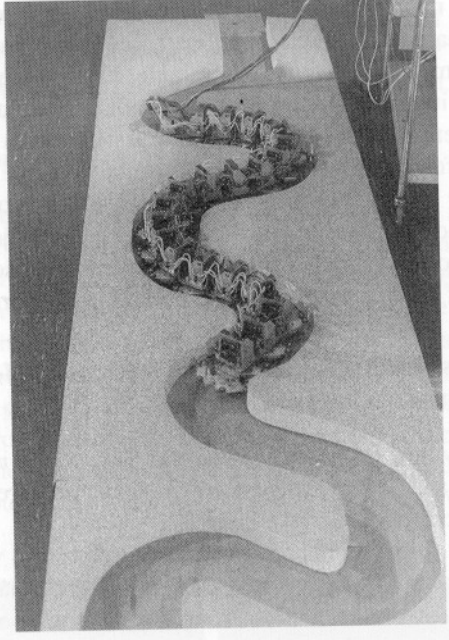


Figure 2.8. ACM traveling through a track using sensors [9].

The original ACM, although not practical for applications other than locomotion on a smooth floor, was a significant contribution to the field of snake-inspired robots. It provided a proof of concept for Hirose's models of snake locomotion, and provided much information on the control of large, hyper-redundant robots. It also generated interest in the field, and was the first step in a line of ACMs by Hirose.

Hirose later applied his results with the first ACM to a much improved snake called the ACM-R3 (Figure 2.9) [10]. The ACM-R3 was designed to be more functional in an actual search and rescue application. Unlike the first ACM, which ran off of an electrical tether cord, the ACM-R3 contained batteries for power, and the servomotors were radio controlled. Each unit contained its own battery and

controller, making the design fully modular. The modular architecture means that there were 6 wires running between modules, as opposed to many more.



Figure 2.9. Hirose's ACM-R3 [10].

The largest differences in the ACM-R3 were that it was capable of 3D motion, unlike the first ACM, and it had large wheels on all sides of the body. These large passive wheels have diameters of 110mm, and add functionality to the system because they can roll against contacted obstacles. The design of the links is such that everything is contained in a shell that has orthogonal axes of rotation on each end. A key requirement in the development of this design was that the snake be able to lift its body weight up. Hirose accomplished this by using servomotors that provided 19 Nm of torque, and the design could lift 8 units up into the air. The overall specs for the ACM-R3 are as follows:

Dimensions: 110 X 110 X 1755 mm

Mass: 12.1 kg

Maximum Twist Angle: 62 degrees (each direction)

Like the first ACM, the ACM-R3 executed a serpentine locomotion gait using the passive wheels. However, in addition to the serpentine locomotion, it could lift its body up to move over obstacles. Hirose also experimented with other gaits on this mechanism, including a lateral rolling gait.

This design provided a marked improvement over the first ACM because it was self-contained, meaning that it had on-board power and can be radio-controlled. The design also showed an improvement in ruggedness, with all of the components mounted inside a shell. Also, an extra degree-of-freedom was added such that the robot could lift up to maneuver over obstacles. The design, however, still required a flat surface on which the wheels could roll to allow locomotion.

The design of the ACM-R3 has since been improved in the generation of the ACM-R5. The ACM-R5 has the added capability that it can move on both land and in water, due to a rugged, waterproof packaging. The ACM-R5 also has wheels on six different sides as opposed to four. Finally, it can operate for 30 minutes without recharging, and has an integrated camera mounted on the “head” unit. To date, no work has been published on the ACM-R5, but the robot was presented at The 2005 World Exposition in Aichi, Japan [11], and detailed information about the ACM-R5 is also available [12].

Robot Developed by Kevin Dowling

Another early implementation of a snake-inspired robot was developed by Dowling at Carnegie Mellon University [13]. Dowling developed a snake-inspired robot while studying gait generation using machine learning. Dowling took a comprehensive look at a wide range of possible technologies that could be used in snake-inspired robots, and designed a snake-inspired robot that could move in three dimensions around a servomotor actuator. Unlike the work of Hirose, Dowling's robot did not require passive wheels in order to move.

Dowling looked at the geometric design of a snake-inspired robot as it related to mission parameters. He determined the dimensions of curved and right-angle pathways that a snake could fit into as a function of link geometry and twist angle. Dowling found that the angle of motion is not as important as the link length. The link length should be as short as possible.

The mechanical design of this robot consisted of an aluminum sheet with servos mounted to it. The servos were mounted orthogonally, so that each end of the link contained an actuated revolute joint. The rotating sections were mounted directly to the servo horn, and adjacent links were attached to each other such that orthogonal servos connect to each other. A sample link can be shown in Figure 2.10. The robot contained 10 links for 20 degrees-of-freedom, and had an overall length of 102 cm. The mass of the robot was 1.32 kg, and each link had a diameter of 6.5 cm.

The robot was controlled using centralized control and powered using a tether. The servos were controlled using a DCC bus and wires that runs the length of the

robot. The control circuitry was located in the “head” of the snake. NiCad batteries were proposed as a power source, but external power was used in the actual implementation. Additionally, a CCD camera was mounted on the head unit. The entire robot is shown in Figure 2.11.

An interesting feature of this robot was that use of “skin” was investigated to provide desirable friction characteristics. Dowling proposed covering the entire robot in a fabric or material that would provide good friction characteristics in order to propel the snake forward. Several candidate materials were discussed and evaluated.

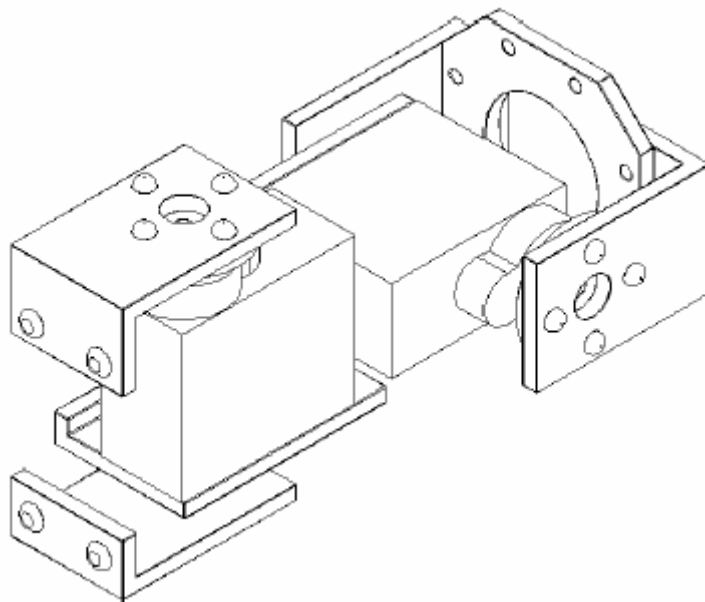


Figure 2.10. One link of Dowling's snake-inspired robot [13].

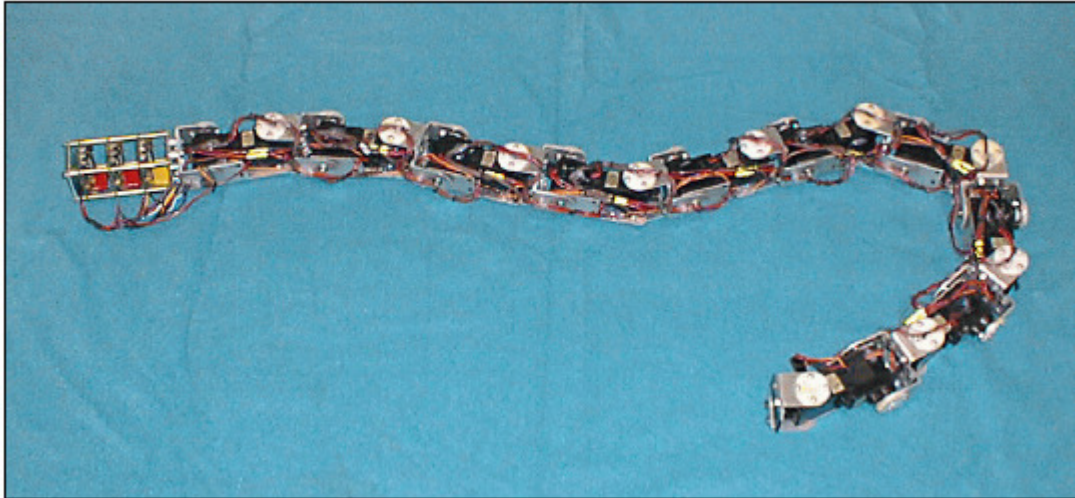


Figure 2.11. Dowlings snake-inspired robot [13].

Other Early Works

Other interesting early snake-like robots include those developed by Chirikjian and Burdick at Caltech [14], Shan [15], and the “Kaa” robot designed for climbing by IS robotics [2]. An important aspect of snake-inspired robot design is the design of actuated joints. Ikeda and Takanashi of NEC developed an innovative joint for serpentine robots and manipulators [16]. The joint was based on an actuated universal joint and was to be used in a snake-inspired robot called the “Quake Snake”.

NASA’s Jet Propulsion laboratory used a modified version of NEC’s joint in the design of their 12 degree-of-freedom hyper-redundant manipulator that could be used for spacecraft applications, functioning in a crowded workspace [17]. The joint used by JPL consisted of a u-joint within a gear-head and bearing assembly. The difference between the JPL joint and the NEC joint was that the NEC joint was on the outside of the assembly.

An innovative search and rescue solution that utilized a hyper-redundant robot with an actuated universal joint was developed by Wolf et al (Figure 2.12) [18]. This robot consisted of a hyper-redundant manipulator robot mounted on a mobile base. Seven actuated, serially-chained, 2-DOF joints constituted one component of the robot which had a camera mounted on the end. This allowed for the end to be inserted into hard-to-reach locations to gather data in a search and rescue effort. The structure was mounted on a mobile base that utilized a standard four-wheel vehicle configuration.

The design of the hyper-redundant chain consisted of actuated universal joints with orthogonal axes of rotation. Actuation of the universal joints was accomplished by linear ball screw actuators that push and pull against the joint. The joints also contained an innovative “snubber” mechanism to prevent damage when a load is placed on the structure, or the stops of the movement have been reached. The joints allow for 55 degrees of motion in each direction.

Like other designs, Wolf et al. realized that it was impractical to run wires through the structure to each of the 14 actuators. Therefore, the control was accomplished using an I2C control bus that runs along the length of the structure from the mobile base. Each link component contained its own H-bridge, decoder, and PIC microcontroller. PWM signals were used to control the motion of each link.

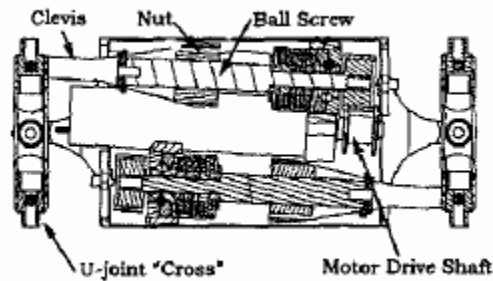


Figure 2.12. Actuated universal joint design by Wolf et al [18].

Robots Developed by Gavin Miller

Another body of work that is of interest in the area of snake-inspired robots is that of Dr. Gavin Miller [19]. Dr. Miller's snake-inspired robot effort is entirely self-funded, yet he has been able to produce a family of sophisticated and life-like snake-inspired robots that are complete with power, control, and sensors. Miller's robots were inspired by the work of Hirose, and used passive wheels on the bottom to assist in movement using a lateral serpentine gait. Realism and aesthetics were a major portion of the design goals. Miller's serpentine robots culminated in his most sophisticated design, called the S5 (Figure 2.13). The S5 was inspired by real snakes and built on an earlier design, the S3. The basic design used universal joints on the top of the robot and two servos on opposite sides of each link that were used to produce lateral and vertical motion. The complete design consists of a head unit plus 32 actuated links that were actuated using 64 servos. Control was achieved using one Basic Stamp II microprocessor (20 MHz), one Scenix Microprocessor, and 8 servo control units. Power was supplied using 42 batteries, and the snake was controlled using a radio controller.



Figure 2.13. Gavin Miller's S5 snake-inspired robot [19].

In addition to the S5 and its predecessors, Dr. Miller also made several attempts to create a snake-inspired robot that only used one motor, and is currently working on a new design of a snake-inspired robot without wheels that locomotes using a rectilinear gait. In order to develop a snake-like robot with only one motor, he developed a system of gears and rockers that would propagate undulations down the robot. The design, however, failed. The rectilinear design is based on a python and is currently under development. It reportedly has a sensor suite that contains sonar, a compass, and pyroelectric heat sensors.

AmphiBot

A recent snake-inspired robot developed with the use-environment in mind is called the AmphiBot [20]. AmphiBot was created as an amphibious robot capable of anguilliform swimming and snake-like undulation. It was produced as a bio-inspired robot to investigate how the central nervous system implements locomotion in animals using a central pattern generator. The AmphiBot was designed to be modular so that individual elements could be quickly replaced and also added and subtracted. Each element contained its own motor, battery, and microcontroller. The robot was remotely controlled and moves in a lateral serpentine gait, assisted by passive wheels. In the water, the lateral sides of the robot generate the forward forces.

The design was developed to be rugged and waterproof. Each individual module was constructed to be waterproof, as opposed to just having a waterproof coating around the entire snake. The structural components were molded using polyurethane. The LiPo batteries were molded directly into the casing and the battery charging circuit was built into the design. The battery with a 600mAh capacity was used, which allowed for 2 hours of continuous use. Each link consisted of four components: a body, two covers, and a connection piece. O-rings were placed in between the covers and the body to form a seal. Five wires that are molded directly into the connection piece were used to supply the control bus and the power bus (when charging) to the entire snake. Each link has a length of 7 cm and a cross section of 5.5 cm by 3.3 cm.

OmniTread

Up to this point the discussion has been limited to robots that locomote purely by snake-inspired means, whether that be serpentine, concertina, sidewinding, or rectilinear. There is a different class of robots that are designed for search and rescue applications that are inspired by snakes, but use active means of progression as opposed to body undulations. These robots were designed to be shaped like snakes and have many degrees of freedom. What separates them from the snake-inspired robots that were previously described is that they use active wheels or tank treads to generate forward motion.

An example of one such robot was the OmniTread robot developed at the University of Michigan [21]. The first OmniTread robot was called the OT-8 (Figure 2.14). This robot consisted of five links that were connected by four, two-degree-of-freedom joints. The propulsion of the robot was achieved by an innovative means: Using tank treads on the four sides of every link. The tank tread design maximizes the “propulsion ratio”, the ratio of surface area that is active in propulsion to the surface area that is not. In order to maximize this ratio, tank treads cover as much of the sides as possible and the gap size between the links are minimized. The idea behind the maximization of this ratio is that any environmental feature that contacts the robot at a location not covered by treads will impede the motion. Treads on each side also make the design indifferent to falling over.

The second innovative feature of the OmniTread was that it is designed with pneumatic bellows that acted as the actuators between the links. These allowed for dual functions. The bellows allowed compliance between the links, allowing the

robot to passively conform to the terrain to maximize traction. The pneumatic bellows meant that stiffness could be adjusted “on the fly”. An example of when this would be needed is when the robot is climbing over a gap. Thus, the bellows were used to both actuate the joints and adjust the compliance. A total of 16 bellows were used, giving the robot 16 position parameters and 16 stiffness parameters. Two valves were used to control each bellows. A universal joint was located in the center of the space between links (between the bellows) in order to maintain structural rigidity.

One motor provided the power to all of the tracks in the robot using a central drive shaft spine running the entire length of the robot, using universal joints. In the next iteration of the OmniTread (OT-4) [22], each link contained small clutches that can engage and disengage each tread as is needed.

The dimensions of the OmniTread OT-8 links were 20X18.5X18.5cm and the entire robot was 127 cm long. The complete robot weighed 13.6 kg. Performance testing has been completed on the OmniTread design. The robot has been shown to be able to climb up a curb more than 36% of its length, and 240% of its height. Additionally the robot can lift up two of its head or tail segments. The OT-8 operated off of a power and pneumatic tether, but the newer OT-4 has built in CO₂ cartridges and batteries for up to one hour of continuous operation. The OT-4 can also fit through a hole with a diameter of only 4 inches. In order to have such a compact design, the OT-4 was designed with complex links that are rapid prototyped using a stereolithographic resin process. The links are reinforced with aluminum.



Figure 2.14. Omnitread robot climbing up a step [21].

Overall, the OmniTread robots would appear to be much more functional than other snake-inspired robots that have been produced to date, but the improved functionality comes with a cost. The mechanisms in the OmniTread, including all of the drive belts and drive shafts make it much more complicated to manufacture than the other robots. The OT-4 is so complicated that its components must be produced using an SLA technique.

GMD-Snake

Another, earlier snake-like robot that used driven wheels was developed by Klaassen and Paap [23]. This robot was called the GMD-Snake2 (Only the second iteration is reviewed in this thesis, as it builds on the first). The GMD-Snake2 consisted of cylindrical links that were connected by universal joints and had wheels on the bottom that are driven by small motors. Additionally, the position of each joint was controlled by three motors that used small ropes to move the joint.

The robot was designed to be a rugged design for practical applications. Links were built around an aluminum cylinder with holes on the surface. The device

could be operated on a tether, or the last section could be entirely filled with batteries. The diameter of this robot was 18 centimeters and the length was 1.5 meters long. The robot had a mass of 15 kg. Like many of the other robots, each section contained its own processor and communications were achieved via a bus.

Discussion and Evaluation

There are several common themes in the design of many of these snake-inspired robots. Upon reviewing the designs, it can be seen that distributed control is often used, with processors and chips located in each of the links. This is done for two reasons: simplicity and modularity. In the case of the AmphiBot, the processors are located locally so that the design is modular and links can be easily added. The control system based on the central pattern generator allows for this scheme. On the other hand, in the hyper-redundant robot by Wolf et al. the processors were distributed locally so that a large number of wires do not need to travel the length of the robot.

Another common design feature in many of the robots discussed is that they rely on either a universal joint or two revolute joints in an orthogonal configuration. This is taken from the inspiration of snakes. Snake vertebrae allow for lateral and vertical twisting, and snakes locomote by using both means to move their bodies. The robots discussed that only have motion in the lateral direction were designed for the laboratory to demonstrate gaits and control architectures. In the case of Hirose's robots, the extra degree-of-freedom was later added.

In general, it can be seen that the majority of snake-like robots have been developed to demonstrate gaits, control schemes, and validate mathematical theory in the laboratory environment. Many of them rely on small passive wheels to locomote on smooth surfaces. In an actual search and rescue environment, the surface may not be smooth enough for the wheels to roll. In addition, many of the robots have been designed for the physical environment of the laboratory instead of the actual search and rescue environment. More recent robots, such as the AmphiBot and the ACM-R5 have been developed with ruggedness in mind with their waterproof design.

The OmniTread can be considered the most functional robot for search and rescue application, however, it illustrates another major drawback in snake-like robot design. The design of the OmniTread contains many parts and is complicated from a mechanical standpoint. Considering that each bellows requires 2 valves to actuate, the entire design has 48 different valves for actuation. Since the actuation of the bellows requires air lines, a manifold is built into the chassis. The shape of the chassis is so complicated that it must be built using the SLA process. The drive train also requires many parts including worm gears and universal joints. The OmniTread is not alone in its large part count. Looking at Figure 2.10, it can be seen that assembling just one link of Dowling's robot would require the relative placement of 9 parts and more than 16 screws. Large part counts, large number of assembly operations, and specialized manufacturing processes would make these snake-inspired robots costly to produce.

2.4 Snake-Inspired Robot Mechanics: Kinematics and Dynamics

The first analysis of the mechanics of snake-inspired robot locomotion was done by Shigeo Hirose, along with the development of the ACM [9]. Hirose began by posing a snake mechanism as a series of serial robot links with infinitesimal length. He assumed that the snake moved by undulation in 2 dimensions. By using a summation of the forces and torques acting on the body of the ACM, Hirose developed “force density functions” along the parametrized length “ s ” of the robot, as functions of continuous torque, $T(s)$, and curvature, $\rho(s)$. Functions were developed in both the tangential direction (Equation 2.1) and the normal direction (Equation 2.2), and were integrated over the length of the robot to develop the propulsive force and the amount of lateral “pushing” done by the snake.

$$f_t(s) = \frac{dT(s)}{ds} \rho(s) \quad (2.1)$$

$$f_n(s) = \frac{d^2T(s)}{ds^2} \quad (2.2)$$

Using these equations and the expression for power, a power density function was also developed that expresses the power in terms of torque, curvature, and tangential velocity.

As for analysis of the kinematics, Hirose assumes that the body of the snake takes on a continuous curve where each segment follows the previous segment.

Hirose proposes a curve to describe snake locomotion as well as parameters that govern the specific locomotion. Observing that the curvature of a sine wave is irregular as a function of curve length, Hirose proposed two curves to describe snake locomotion. One is a composite curve based on the clothoid spiral by Umetani, with linear curvature with respect to length. The second curve, which Hirose named the serpentoid curve (Equation 2.3), has a curvature that varies sinusoidally with length. Hirose proposed that snakes moved with a serpentine curve because it has the “greatest amount of smoothness of contraction and relaxation of the motor muscles” [9].

$$\left. \begin{aligned} x(s) &= sJ_0(\alpha) + \frac{4l}{\pi} \sum_{m=1}^{\infty} \frac{(-1)^m}{2m} J_{2m}(\alpha) \sin\left(m\pi \frac{s}{l}\right) \\ y(s) &= \frac{4l}{\pi} \sum_{m=1}^{\infty} (-1)^{m-1} \frac{J_{2m-1}(\alpha)}{2m-1} \sin\left(\frac{2m-1}{2} \pi \frac{s}{l}\right) \end{aligned} \right\} \quad (2.3)$$

The parameter α is called the winding angle (Figure 2.15) and defines the angle in which the snakes body intersects with the line that indicates the direction of progress. J_m indicates the m^{th} order Bessel function. These two curves are compared with measured data from actual snake locomotion, along with a composite arc curve and a sinusoid, and are shown to closely agree with the measured data, with the serpentoid curve being the closest match.

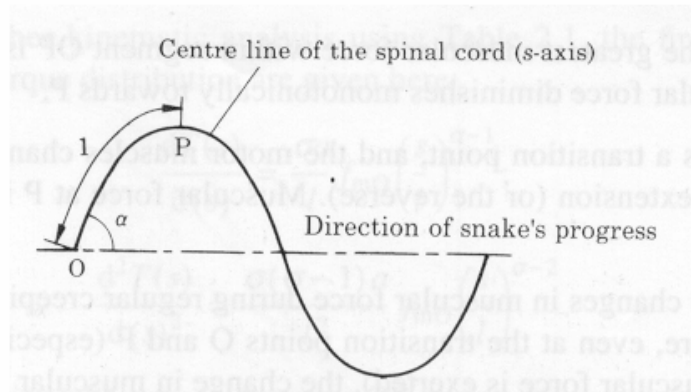


Figure 2.15. Nomenclature for Hirose's equations [9].

Upon developing functions to describe the shape of snake locomotion, Hirose proposed a function to describe the muscular force acting on each joint (Figure 2.16). Functions are proposed with the nature of the body shape and structure in mind, and are dependent on an independent parameter that was experimentally obtained from real snakes. With the functions for muscle force and gait shape developed, expressions for propulsive (tangential) force and normal force were developed. The experimentally-measured forces were shown to agree with the proposed model.

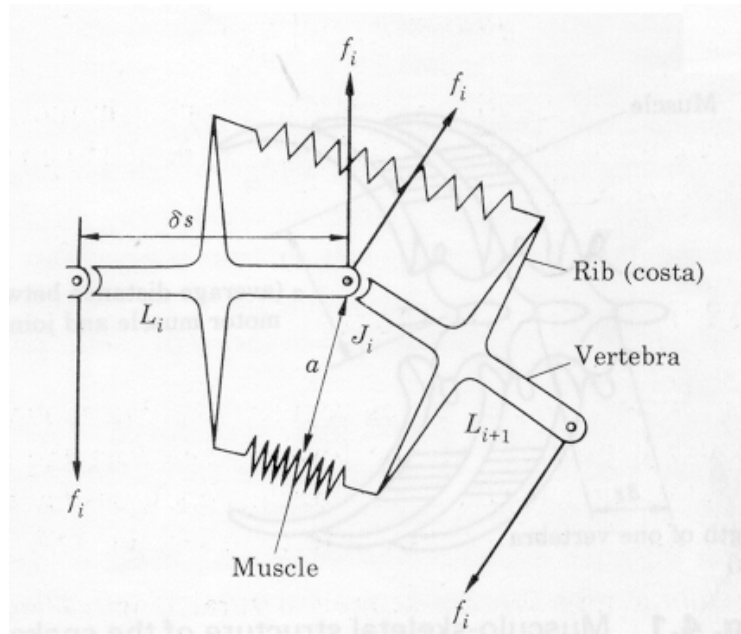


Figure 2.16. Model of muscle structure in snake joint [9].

The major contributions of Hirose's work can be summarized as the following. Hirose used a biologically inspired approach to describe and quantify the geometric and dynamic aspects of real snake locomotion. In his work, he developed a parameterized model of such locomotion, and both experimentally verified his conclusions on real snakes and a synthesized robot system. His work assumed that the geometry of the snake took on a continuous curve, but since he formulated his equations in terms of curvature and torque distribution, his calculations can be modified for a snake with discrete links.

Chirikjian and Burdick analyzed the kinematics of snake locomotion from a geometric standpoint [24]. They modeled the snake-inspired robot as a continuous backbone curve and analyzed the kinematics of gaits that used both "stationary" and "traveling" waves. Traveling waves would be those similar to rectilinear snake

locomotion, while stationary waves would be those more similar to inchworm locomotion. They used a two step process to compute the kinematics of snake locomotion where they first assume that the snake can be modeled by a continuous spline or “backbone curve”, and then use this backbone curve to specify actual joint displacements. The model does not consider the dynamics of the system, and assumes that there is sufficient friction to enact the gait.

In Chirikjian and Burdick’s work, the snake is assumed to take the shape of a spline, and a path of the motion is specified as well. The spline can be either extensible or inextensible. The spline is modeled by the following equation with “s” as the normalized arc length and “t” as the time:

$$\bar{x}(s,t) = \int_0^s l(\sigma,t) \bar{u}(\sigma,t) d\sigma \quad (2.4)$$

The component “l” denotes the length of the curve tangent, and “u” denotes the unit tangent vector of the curve, which is parameterized using Euler angles. In this formulation, the curve becomes a function of “shape functions” that specify the orientation and position of the backbone reference frame as a function of time and parameterized arc length.

A path that the robot takes is defined as a curve in R^3 , and the task becomes developing backbone curves that traverse the path curve, as shown in Figure 2.17. A “stride length” is specified, which defines how far the robot traveled during one cycle of the gait. The task is then to determine a curve with the specified stride length that matches the path curve at either end of the stride, but does not intersect the path

curve. Gaits can be developed using this framework and techniques such as variational calculus.

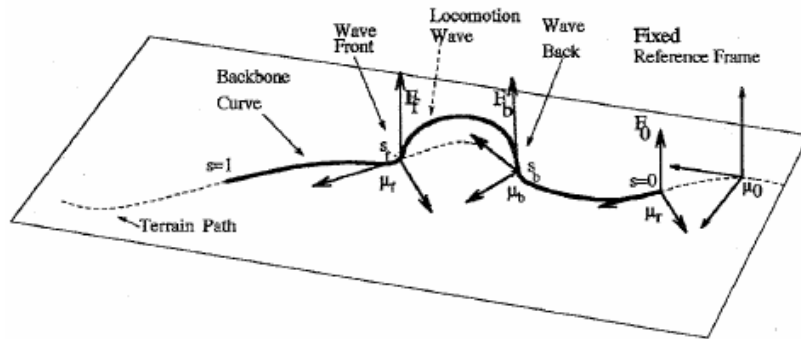


Figure 2.17. Parameters in Chirikjian and Burdick's approach [24].

Another approach to gait kinematics was developed by Burdick and Ostrowski [25]. This work models the snake-inspired robot as a discrete set of links with passive wheels on the bottom. The model is based on Hirose's first active cord mechanism (Figure 2.18). The variables of locomotion are divided into two sets: shape and position. The shape space is defined with a manifold with the order being the number of movable joints in the robot. The position variables are defined in the special Euclidean group $SE(2)$, because the robot is constrained to motion on a plane. The total configuration of the robot can be defined with these two sets of spaces.

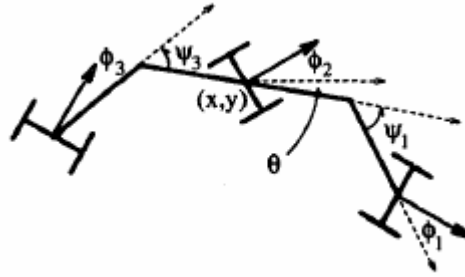


Figure 2.18. Burdick and Ostrowski model [25].

A framework is then developed using Lie algebra to determine how changes in the shape space affect changes in the configuration constraints through the nonholonomic constraint that the wheels place on the locomotion. The framework begins by defining a three segment snake, and the constraint is formulated as:

$$\left(-\sin \bar{\phi}_i, \cos \bar{\phi}_i\right) \bullet \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \end{pmatrix} = 0 \quad (2.5)$$

This equation means that locomotion of each segment must occur in the direction of the wheels. The relationship between segments is established using a connection, and the relationship between angles and the position-changing of the snake can be established. Three links are used to start, because this constitutes the “principal kinematic case”, the case where there are an equal number of constraints and equations. Adding a link adds two additional degrees of freedom, but only one constraint equation. When links are added, they are simply made to follow the first set of links.

Saito, et al. investigated the locomotion of a snake-like robot from a dynamic perspective [26]. They considered the case where a snake moves in a lateral serpentine gait due to anisotropic friction, but without wheels. Friction models were developed based on both viscous and Coulomb friction. The model found the frictional forces and torques acting on each link as a function of the shape and shape changing parameters. These equations were then assembled into a Newton-based formulation of the equations of motion. The behavior (velocity, acceleration) of the center of mass for the entire robot, as well as each link, can be computed from this set of equations. This framework resulted in a system of equations where the joint torque, joint angles, and inertial behavior of the entire robot are related. A similar framework was also developed by Ma et al [27].

Another dynamic framework for snake-inspired robot locomotion was developed by Cortes et al. [28]. In this work, a dynamic framework for the locomotion of a robotic eel was developed. The work used a Lie group formulation similar to [25], and the dynamics were addressed using a Lagrangian reduction process. Friction was modeled using a fluid friction model.

Discussion and Evaluation

While different kinematic and dynamic models for snake locomotion have been developed, none of them consider the dynamics of a snake using a vertical rectilinear gait. Work has been done by several different researchers to model the snake as both a continuous form, and a discrete set of links. Also, snakes with and without wheels have been investigated. Most of the models, however, deal with serpentine motion in

the lateral plane, with only [24] dealing with waves that travel in the vertical plane. This work only considered kinematics, and approximated the snake as a continuous curve. Gaits have been generated in the vertical plane, as will be seen in the next section, but minimal dynamic analysis exists for such gaits. The dynamic analysis does not exist because the changing points of ground contact apply reaction force loads intermittently, resulting in a complicated model.

2.5 Rectilinear Gait Generation

In this section, the design and generation of gaits in snake-inspired robots that resemble rectilinear locomotion of biological snakes is discussed. As described in section 2.2, rectilinear locomotion is achieved in biological snakes by slight lifting and compression of segments, in order to creep forward. In snake-inspired robots, this gait is achieved by propagating vertical waves from the tail to the front of the robot to achieve advancement. Much literature exists relevant to the generation of a variety of forms of gaits, from serpentine to even non-biologically-inspired gaits. In this section, discussion is limited to only gaits that are considered “rectilinear” and similar to the gait that is developed and analyzed in this thesis.

Gaits of a rectilinear nature, where the robot lifts up its body, have been proposed in many different works, including [13], [24], and [29]. They are desirable because they have minimal slip, and are less dependent on the specific friction characteristics of the surface on which they locomote, as opposed to other gaits which may require passive wheels. Another gait somewhat similar to rectilinear locomotion, meaning that advancement is achieved by lifting in the vertical plane, is

executed by a caterpillar robot developed by [30] and [31], as well as Nilsson's slip-free gait [32]. Robots utilizing rectilinear locomotion gaits have also extensively been developed at Carnegie Mellon's Biorobotics Lab [33].

Merino and Tosunlu [34] presented a rectilinear gait for a theoretical modular robot that performed like a snake-inspired robot. They posed the gait as a series of configurations that the robot would take (Figure 2.19). The end result of this is the propagation of a half-wave from the rear to the front of the robot. It can be seen that the amount that the gait advances during each cycle is a function of the link length, and the angle at which the two outside links of the half-wave are inclined from the ground. From this, and assuming a linear servo speed, it can be seen that the velocity of the robot increases linearly with the angle. The robot was then modeled with the software package Working Model to confirm that the gait would achieve a forward motion.

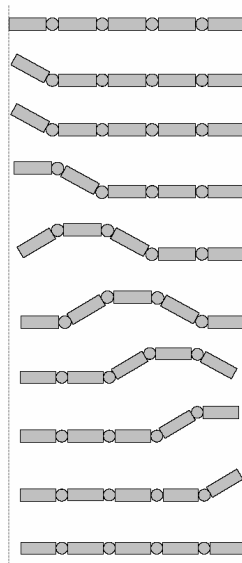


Figure 2.19. Merino and Tosunlu's Gait [34].

Chen et al. [35] presented the idea of a “traveling wave” gait similar to the work of [34] where the configuration of the vertical wave travels through a series of “phases” as it moves forward. They proposed moving the angles according to a serpentine curve to propagate waves from the rear to the front. Traveling wave locomotion is also addressed in [36].

Discussion and Evaluation

Rectilinear gaits have been shown to be successful in snake-inspired robots, and this thesis builds on the work presented in this section. This thesis builds on the idea presented by [34] that by using a sequence of configurations a wave can be propagated through the snake to achieve forward locomotion. The work of Chen et al. [35] similarly noticed that the snake’s motion could be broken into “phases” where different parts contact the ground. Both the modeling of these gaits and the drawbacks of the gaits that have been presented thus far are addressed in this thesis. The drawbacks of current gaits are that they have used either linear interpolations or sinusoids to specify their joint trajectories. In this work, a more controllable means of generating joint trajectories is used, resulting in trajectories that are more efficient.

2.6 Robot Trajectory Optimization

Developing gaits for snake-like robots includes the generation of joint trajectories in order to specify the time history of how a joint changes its position. These trajectories can be optimized using a mathematical framework. A considerable

amount of literature exists in the area of optimal trajectory generation in manipulator robotics, and similar techniques and principles can be applied to the snake gait problem. This section provides a brief review robot trajectory optimization.

In general, robot trajectory optimization can be classified into two approaches. The first approach uses an optimal control framework, and solves a two-point boundary value problem [37]. The second approach assumes that the joint trajectory follows a path that can be described using a set of parameters (often control points) and then seeks to vary the parameters until a local optimum is found. This approach has the advantage of being simpler; especially considering that robot manipulator equations are complex and highly non-linear. It also has advantages in constraint handling, because an initial path can be specified that satisfies constraints, and then an optimal path can be found by searching [38].

In this work, the performance of the joint trajectory is defined using a measure called “effort”. Early attempts to formulate optimal control paths for manipulator robots typically focused on the minimization of time [39]. However, a problem with approaches such as these is that they do not consider the wear and tear on the robot joints, or the amount of energy consumed. A better function to minimize would consider both the time that the robot takes to complete the trajectory and the amount of torque and energy required to complete the motion. More recent approaches to trajectory planning have considered these issues [40].

An example of a study where a parameter-based effort minimization scheme is used is the work of Martin and Bobrow [41]. This work uses B-spline curves to define the trajectory of the path (for a description of B-spline curves, consult [42] or

Chapter 3 of this thesis). The effort is defined as the integral of torque squared over time, incorporating the amount of torque required by the robot and the time that the motion takes to complete the motion. Martin and Bobrow present an iterative procedure to obtain analytic gradients of the effort function with respect to the control points, and use these gradients to search for locally optimal solutions. Quasi-Newton algorithms can be used to find optimal solutions. The authors stress the fact that the problems are often numerically ill-conditioned and thus finite difference gradients lead to poor convergence, hence the need for analytical gradients. A simple example of an optimal trajectory found for a two-link planar chain is shown in Figure 2.20 and Figure 2.21. This motion is occurring in a gravitational field.

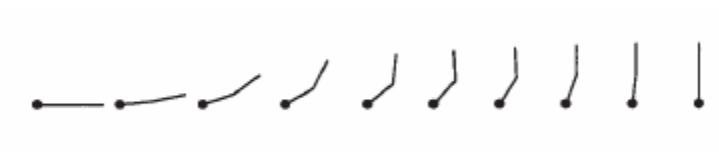


Figure 2.20. Initial path, 2-link manipulator [37]



Figure 2.21. Optimal solution, 2-link manipulator (working against gravity) [37].

The minimum effort optimization presented in [41] is only for open-loop serial manipulators; however, the work is expanded on to cover closed-loop manipulators in [43]. In order to compute minimum-effort joint trajectories for closed-loop mechanisms, a procedure is developed that reduces the closed-loop

system to an equivalent open-loop system and then maps the solution back to the closed-loop system. Exactly and redundantly actuated chains are considered.

Discussion and Evaluation

Effort-based optimization using B-spline curves has been presented by several researchers, and has been shown to be a good means of generating robot joint trajectories. In this work, effort-based optimization is used to evaluate and optimize the performance of a locomotion gait for a snake-like robot. The methods presented in this section are drawn on to develop a model for optimization of the rectilinear locomotion gait.

2.7 Summary

The prior work that is relevant to this thesis covers several different areas from biology to mechanical design, and such was divided into five different areas of discussion. Each is important to this thesis in separate ways. First, snake locomotion was discussed, as it is the inspiration for this work. Several snake-inspired robots that have been previously developed were discussed, and from an evaluation of them desirable and negative attributes of these robots were discussed. From an evaluation of prior designs, the lessons learned can be incorporated into the design developed in this thesis. The work in this thesis seeks to build on and improve the modularity of previous designs, as well as develop a design with lower manufacturing costs. The third and fourth portions of the related work discussed the state of the art in characterization of snake-inspired robot locomotion and gaits. Earlier

implementation of rectilinear gaits was reviewed, and a gait was identified that is further modified and enhanced in this work. A detailed kinematic and dynamic formulation will be developed in this thesis. Finally, robot trajectory optimization approaches were discussed because this work applies a similar approach to design a gait for snake-inspired robots. This will result in a highly-parameterizable formulation for optimized gaits.

Chapter 3 - Gait Development and Analysis

3.1 Introduction

In order to generate and analyze motion and performance for the snake-inspired robot, both a framework for analysis and a gait are required. The definition of a gait as it relates to a snake-inspired robot is a set of joint trajectories that are repeated to generate a forward motion in the robot. In order to generate a gait, the relevant parameters must first be defined, an approach that results in forward motion developed, and then the equations that can be used to describe the performance of the gait developed. Because of the heuristic-based and optimization-based approaches to gait design that are used in this thesis, performance information (fitness) must be generated in order to develop the gait in a feedback-like nature. A means of parameterizing trajectories so that that they may be optimized is also presented. In this thesis, a framework for optimizing joint trajectories is presented, and a separate, simpler, heuristic-based approach to obtain joint trajectories with improved performance is demonstrated.

A representation of the gait design process is shown in Figure 3.1. The relevant parameters of the gait and the physical design are identified as the gait angle, step time, length, and mass. Control points are used to determine the exact path of the trajectory, and all of this information is used in the dynamic model to obtain a torque signature. This torque signature can then be used to generate the effort.

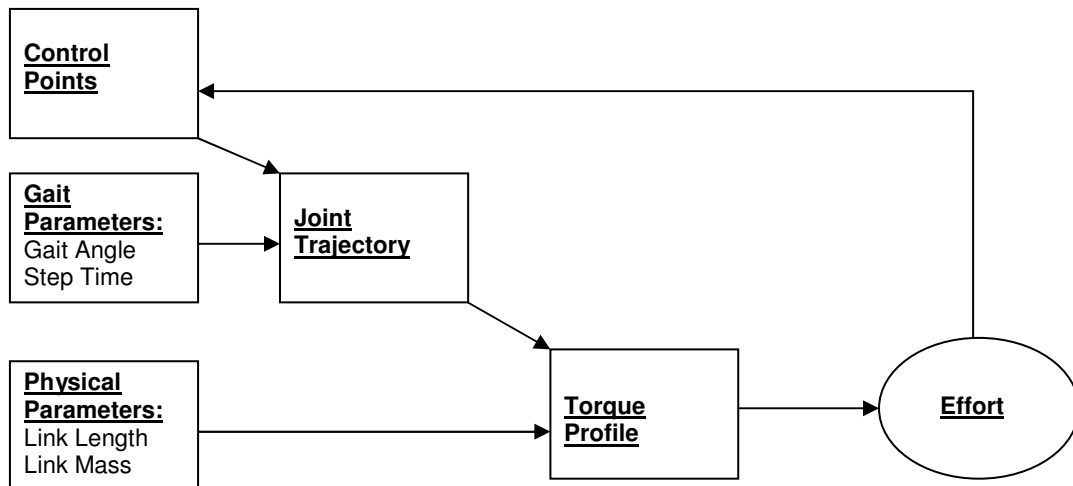


Figure 3.1. Representation of gait analysis and development procedure.

3.2 Background

Prior to developing the specifics of the snake-inspired robot gait, several elements of background material must be discussed. In order to describe and analyze the motion of a snake-inspired robot, basic robot representation must be discussed, and a basic description of robot link trajectories is presented. Furthermore, since B-spline curves are utilized to describe and specify the trajectories of joint motion, a description of B-spline curves is presented.

3.2.1 Basic Robotics

A basic description of robot kinematics and dynamics can be found in [44]. The trunk of the robot can be thought of as a robotic manipulator with a certain number of joints and links. Each link contains its own Euclidean coordinate system or frame, and the orientation between links can be described using a rotation matrix between the coordinate systems. Figure 3.2 shows link frames {A} and {B} relative to each

other. Since 2-D planar motion is only considered in the case of the rectilinear gait, the orientation of frame B can be represented with respect to frame A using the rotation matrix denoted as follows, with the X and Y directions labeled, and both coordinate systems are standard right-handed coordinate systems:

$${}^A R_B = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

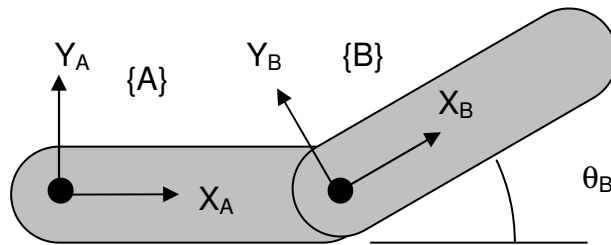


Figure 3.2. .Basic robot link representation.

A complete transformation, however, consists of both a rotation and a translation. In order to perform both simultaneously the homogenous transform is introduced. The homogeneous transform takes the form of:

$${}^A T_B = \left[\begin{array}{ccc|c} & & & \\ & {}^A R_B & & {}^A P_B \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3.2)$$

With ${}^A P_B$ being the vector from the origin of {A} to the origin of {B}.

Using the transformation and rotation matrices, the position of any robot link can be described with respect to any other robot link if the angles of the joints in between them are known. The kinematics and dynamics of the snake-inspired robot are developed using these transformations and several assumptions.

Any robot comprised of multiple linkages can be described as a kinematic chain. A kinematic chain is defined as any assembly of links that are connected by joints [45]. Kinematic chains can be divided into two major classes: Open-loop chains and closed-loop chains. Open-loop chains are kinematic chains whereby each link is connected to each other link by only one path. Closed-loop chains, on the other hand, are chains whereby links can be connected to each other by multiple paths. A snake-inspired robot with a vertical undulation is a unique case where the mechanism can take on the properties of either a closed-loop or an open-loop chain, depending on the topology of the mechanism, as shown in Figure 3.3. The multiple points of ground contact act as either a prismatic link (in the case of dynamic friction), or simply a rigid link (in the case of no slipping) in the closed loop case. Thus, both closed- and open-loop robot analyses must be addressed.

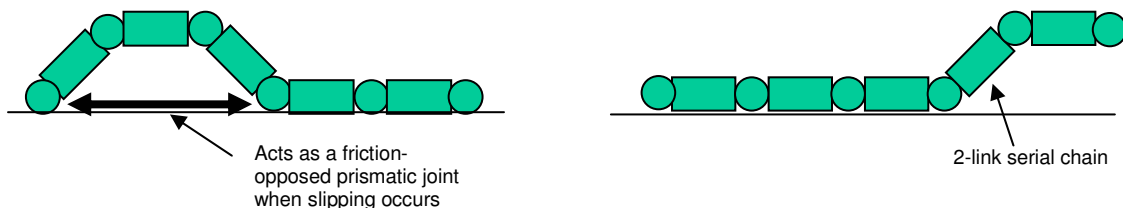


Figure 3.3. Closed-loop (left) and open-loop (right) configurations of a snake-inspired robot.

As others have mentioned in numerous works [25] [26], the configuration of the snake-inspired robot can be divided into two distinct sets of variables: shape and position (orientation). In the distinct case where the snake is only moving in a vertical 2-D plane, the orientation of the snake is determined by its shape, as there is typically one stable orientation for each shape (the assumption is that it remains in a stable orientation). Therefore, the motion of the snake-inspired robot can be described as a function of joint angles, provided that the friction is sufficient for locomotion.

3.2.2 Joint Trajectories

In order to specify how to physically execute the gait, joint trajectories must be developed that specify how the joints are moved between positions with respect to time. Because a snake-inspired robot does not contain wheels, it must rely on a net change in body shape to achieve a forward motion. In order to achieve this change in shape, the complete time histories of angular acceleration, velocity, and position of each joint will be specified to achieve the appropriate configurations for forward motion. These complete time histories are called the “joint trajectories”. Joint trajectories specify the motion parameters in what is known as the “joint space”. The mapping between Cartesian space and the joint space is discussed in [44].

Since the mechanisms considered in this work are closed mechanisms, there are more joints than free variables, thus, the trajectories can be freely specified for a number of joints, while the trajectories for the other joints must be calculated from the geometric constraints using inverse kinematics. There are bounding constraints,

however, on the free joints, so that the mechanism may remain in a feasible configuration. This will be elaborated on later.

3.2.3 Exact vs. Redundant Actuation

Since snake-inspired robots often take on closed-loop configurations, it must be acknowledged that there are multiple solutions for the actuation of closed-loop mechanisms. In the case of a closed kinematic chain, the mechanism will have less degrees-of-freedom than joints. In this situation, one can use an exact or a redundant actuation strategy. In an exact actuation strategy, a select number of the joints are allowed to be passive. The number of active joints is selected such that there are the same number of active joints as there are degrees-of-freedom. In this case, there is a unique solution to the problem of inverse dynamics. In a redundant actuation strategy, the number of actuated joints is greater than the degrees-of-freedom, thus there is not a unique solution to the inverse dynamic problem. Typically an optimization or other framework would be used to determine the inverse dynamics in this case.

In this framework, an exact actuation strategy is used. The rationale is that the gait design already requires several feedback loops for the heuristic-based search, and requiring another one to solve the redundant actuation problem would further complicate the gait design problem.

3.2.4 B-Spline Functions

For the generation of the free joint trajectories, it is assumed that the time history of the joint parameters follows that described by a B-spline curve. B-splines are a class of curve that originated in geometric modeling, and are popular in the geometric modeling community due to their versatility and their unique properties [42].

B-splines are composite curves composed of Bezier curves that are pieced together (For those not familiar with Bezier curves, consult [42]). B-spline curves are defined by a vector known as a knot vector and a series of control points. Changing the locations of the control points and the knot vectors changes the geometry of the curve. It is important to note, that unlike Bezier curves, the degree of a B-spline curve is not determined directly by the number of control points. This is a key advantage of B-spline curves, as complex curves can be modeled without the use of large polynomials. B-spline curves also have the advantage that the control points locally control the curve, but do not have large global effects on the curve. Thus, changing one control point does not have an effect on segments of the curve that are far away from it, which can occur with Bezier curves. Finally, B-spline curves have what is known as the convex hull property, meaning that the curve is fully contained within the convex hull of the control points. This is important in trajectory generation, as the control points defined in the trajectory generation will bound the values of the trajectory (i.e., $p(u) \leq \bar{p} \forall u \in [0, u_f]$).

The basic structure of a B-spline curve is represented as a compilation of a set of basis functions as follows:

$$p(u) = \sum_{i=0}^n p_i N_{i,k}(u) \quad (3.3)$$

p_i denotes the control points, and N denotes the basis functions. The basis functions are recursively defined as:

$$N_{i,1}(u) = 1 \quad \text{if } t_i \leq u \leq t_{i+1} \quad (3.4)$$

$$= 0 \quad \text{otherwise}$$

$$N_{i,k}(u) = \frac{(u - t_i)N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u)N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \quad (3.5)$$

The parameter k controls the degree of the curve, which is $(k-1)$. The parameter n designates the number of control points. The degree and number of control points are related to the knot vector by the following relationship, with T being the number of “knots” (or length of the knot vector):

$$n + k + 1 = T \quad (3.6)$$

B-spline curves do not necessarily interpolate the beginning end control points, but can be allowed to if non-uniform knots are used. In this case, repeated knots are desirable because the joint angles at the beginning and the end of the time step should be directly specified.

In selecting the B-spline parameters for trajectories, a cubic B-spline with 5 control points was chosen. In addition, the mechanisms should begin and end at rest,

thus the derivative of each curve endpoint should be zero. As mentioned before, the curve should interpolate the end points as well. For the interior knots, an even spacing is used. The parameterization of the curve was chosen to be from 0 to 1, meaning that each time step should take 1 second to complete. The formulation is as follows, note that there are 5 distinct control points, and the two exterior control points are repeated to achieve the desired boundary condition.

$$T = [0 \ 0 \ 0 \ 0 \ 0.25 \ 0.5 \ 0.75 \ 1 \ 1 \ 1 \ 1] \quad (3.7)$$

$$P = [\theta_i \ \theta_f \ C_1 \ C_2 \ C_3 \ \theta_f \ \theta_f] \quad (3.8)$$

The parameters θ_i and θ_f are the beginning and end angles of the interval, which are always 0, α , or $-\alpha$. The interior control points, designated as “C” are the free variables.

An important note on B-spline curves that is relevant to the problem of bio-inspired design is that recent studies in neuroscience have shown that trajectories of movements in humans closely resemble B-spline curves. A recent study has shown that when asked to move their hands in a circular motion, the actual achieved motion is best described using “bell-shaped” B-spline basis functions [41].

3.3 Gait Overview and Rationale

At this point, a description of the class of gait that will be used to achieve locomotion in the snake-inspired robot is presented. The framework will then be developed to

parametrically optimize this gait to improve performance. The class of gait is inspired by rectilinear locomotion in real snakes. This gait was selected because of its advantages over other classes of gaits, such as concertina and serpentine locomotion. The general strategy for achieving this gait is then broken down into a series of configurations that the robot must transition through in order to move its body mass forward.

In rectilinear motion in real snakes, sections of the snake expand and contract in order to propel the snake forward. This motion is achieved by muscles pulling and pushing the ribs closer and further apart. In order to achieve a similar effect in an artificial snake, which is more rigid and has less degrees-of-freedom, vertical motion is used. When the snake bends in the upward direction, the lateral component of the segment distance shrinks. The shorter segment is then propagated from the rear of the snake, moving the snake forward. Biological snakes use slight lifting and careful body positioning to shift weight and allow segments to slide. In this approach, the segments off the ground are simply lifted to allow forward motion. Work by [30], however, has shown that it is possible to shift weight (and thus frictional force) around a robot to allow sliding of desired segments, and this is something that could be considered for snake-inspired robot locomotion in the future.

3.3.1 Drawbacks of a Serpentine Gait

The majority of work to date on serpentine gaits in snake-inspired robots has been in the area of mathematical exercises and demonstration purposes. Since serpentine locomotion relies heavily on anisotropic friction, most of the realization of serpentine

locomotion has been achieved with passive wheels on the ventral surface of each robot link. These wheels allow for slipping in the tangential direction, but prohibit slipping in the normal direction. Other implementations, such as the work by Saito et al. [26], have used a similar idea, using large ridges on the ventral surfaces of the bottom of the robot to achieve anisotropic friction on a surface such as carpet.

Both concepts are feasible in the laboratory, but are heavily dependent on specific surface characteristics. In the field, there is a good possibility that a snake-inspired robot will encounter terrain that is too coarse for small wheels to have an effect. In addition, because serpentine gaits are reliant on dynamic frictional characteristics of the surface, it is difficult to predict or ensure satisfactory performance of the robot on surfaces that are not fully described a priori. Finally, further work needs to be done with the design of the ventral surface of snake-inspired robots to achieve frictional characteristics similar to biological snakes.

Furthermore, as mentioned by [3], serpentine locomotion requires careful control. Snakes actively control the trajectory of their curves to push against obstacles in their path. This requires sophisticated feedback control that is difficult to achieve in an artificial snake. In addition to using obstacles to push off, snakes using serpentine locomotion are constantly adjusting their weight and muscular force against the ground to control their frictional characteristics.

Finally, a primary application of snake-inspired robots is to be able to fit through tight spaces. As has already been discussed in the section about snake-inspired robot locomotion, biological snakes typically switch to concertina or rectilinear locomotion when they need to fit through tight spaces. This is because

serpentine locomotion typically requires a larger cross section (when looking forward) than other forms of locomotion. Biological snakes often make a conscious switch from rectilinear to (typically) concertina locomotion when presented with the challenge of fitting through a tight space. This would be another issue that would require sophisticated programming and control in artificial snake-inspired robots.

3.3.2 Advantages of a Rectilinear Gait

Gaits inspired by rectilinear locomotion have been successful in the laboratory, and have been demonstrated to traverse more difficult terrain and through tighter spaces (i.e. pipes) [33]. The primary benefits to a rectilinear gait are that it is easy to control and implement, and because it relies primarily on static friction, as long as the surface provides a reasonable degree of static friction with the ventral surface of the robot, the gait should be possible to implement.

Since the rectilinear gait relies on a regular pattern of muscular contraction, it should be easier to control and implement than a serpentine gait. Vertical waves of bending passed along the axis of the snake-inspired robot are all that would be required to achieve forward movement. This, of course, is only true provided that there is sufficient traction between the ventral surface and the ground. This, however, should not be much of a complication if the bottom surface of the robot is comprised of a high-friction material such as rubber. Conversely, with a gait such as a serpentine gait, this type of material would prohibit forward motion, because the rectilinear gait relies on dynamic friction to move itself forward. In a rectilinear gait, the snake exerts more pressure on the static points, slightly lifting the moving points

and allowing them to move forward. This would be a feasible strategy with most frictional surfaces in snake-inspired robots, provided that there is enough lifting of the moving points. Typically, the approach used by robot designers (and our approach) is to completely lift the portion of the snake that is moving forward. Using this strategy, forward motion can be achieved with a high coefficient of friction between the snake and the ground. This means that snake locomotion will be feasible on a wide range of surfaces.

Another possible advantage of the rectilinear gait is, because there is minimal slipping of the surface on the ground, there is not a lot of energy lost due to friction. On first glance, this would appear to be a major advantage over serpentine locomotion, however, the energy gains are largely offset due to the fact that the body of the snake does a significant amount of work to lift the trunk above the ground to avoid the friction. This is a tradeoff that has been slightly investigated in biology, but still must be investigated in snake-inspired robots.

3.3.3 Gait Sequence

The rectilinear gait studied here relies on lifting in the vertical direction to advance segments of the snake forward, using friction. Gaits that use this similar vertical motion have been implemented and shown to be successful by several groups [33]. The gait that has been chosen for this study is based on the gait presented by Merino and Tosunoglu [34]. The authors presented a sequence of joint configurations that should result in a forward motion if the robot is driven through them. This sequence is shown in Figure 3.4.

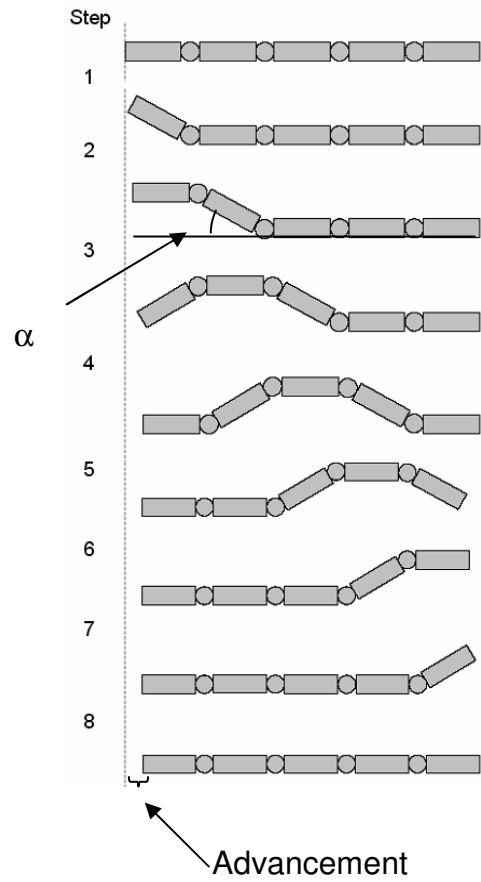


Figure 3.4. Gait presented by Merino and Tosunoglu [34].

This gait illustrates that if a vertical wave is passed through the robot, the result will be an advancement that is proportional to the gait angle α (labeled in Figure 3.4). This advancement is computed as:

$$x = 2l - 2\cos(\alpha) \quad (3.9)$$

The average velocity of the robot can then be determined by dividing the advancement by the cycle time. A gait inspired by this sequence has been chosen to propel the 6-link robot described in this thesis. Additionally, it can be seen that steps 1-3 in Figure 3.4 can be merged into one step to simplify the motion and prevent any unnecessary joint motions. For a full description of Merino and Tosunoglu's work, consult Section 2.5.

This rectilinear (or traveling wave) gait is difficult to analyze kinematically and dynamically because the topology of the mechanism changes during the course of the motion. This means that the ground contact is constantly changing, altering the points at which external reaction forces are applied. Figure 3.5 shows the different topologies of the mechanism during the course of the gait cycle. Mechanism 1, or M_1 , is the first part of the gait sequence and it is considered an open-loop mechanism, considering that the moving joints are free to move from the first position to the last position without a reaction force, provided that the joints remain in the feasible joint space (above the horizontal). Mechanisms M_2 and M_3 , however, have reaction forces that are applied on different ends of the moving joints. Thus, they needed to be treated differently as far as the kinematics and dynamics are concerned.

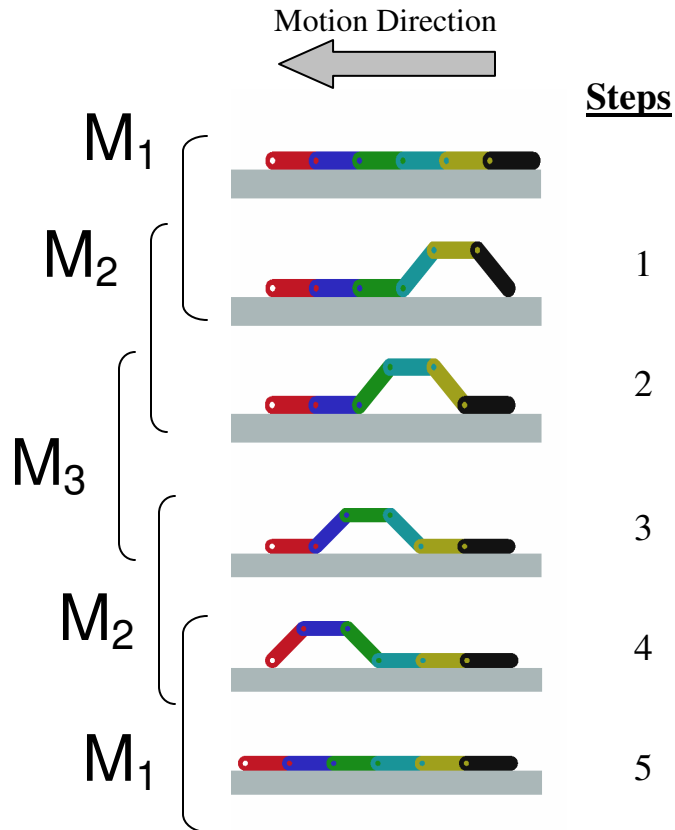


Figure 3.5. Modified traveling wave gait, steps and mechanisms are identified.

Locomotion from this gait will be achieved by driving the appropriate joints through desired trajectories. The trajectories will be specified using the B-spline curves that are presented in Section 3.2.4. The parameters (control points) of the B-spline curves can be selected heuristically and then tested to verify that the gait is feasible. However, a search-based approach is preferred, where a search is conducted to find control points that optimize the performance of the locomotion gait.

3.4 Kinematics and Dynamics Model

This section presents a formulation of both the kinematics and dynamics of the model. This is necessary because a dynamics model must be created in order to

determine the amount of torque required to actuate the robot. More importantly, the dynamics model can be used to evaluate metrics that determine the performance of the robot. The value in the kinematics model is that it is both necessary to develop the dynamics model, and also can be used to verify the feasibility of the gait.

In Section 3.3.3, the gait was divided into mechanisms with different topologies. In this section, the kinematics and dynamics model is constructed for each mechanism. The example of M_1 is used to show how the kinematics and dynamics models are developed, and then it is shown how this approach can be expanded and applied to M_2 . Finally, it is shown how the solutions to the steps and mechanisms can be mapped back to the overall mechanism to achieve one set of global solutions.

3.4.1 Gait Kinematics

Consider the first mechanism in the locomotion gait, M_1 , shown in Figure 3.6. The links of the snake-inspired robot must move from the flat shape shown in the first figure to the lifted shape shown in the second part of the figure. This motion can be described by three parameters in the joint space: the angular positions of joints 1, 2, and 3. It can also be seen that the feasible joint space is constrained to a region that amounts to a half-space due to the presence of the ground.

Since the constraint amounts to an inequality constraint, it may or may not be active depending on the joint space configuration of the mechanism. This complicates matters, because the dynamics are different for the case where the

constraint is active and the case where the constraint is not active. This issue is solved by posing the move as two different problems and then comparing the results.

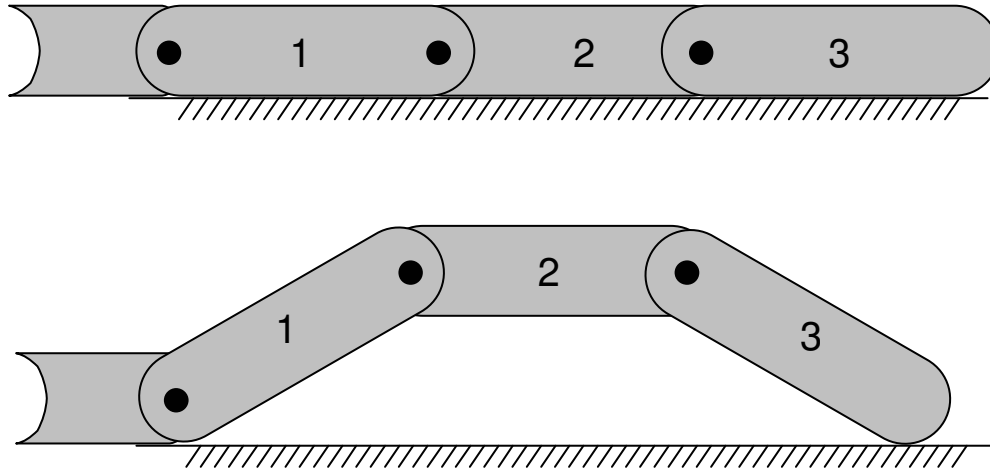


Figure 3.6. The first gait step.

The first way to model the sub-problem is to treat it as a free three-link system with three degrees of freedom. This system is shown in Figure 3.7. In this case, both the kinematic and dynamic parameters can be computed using the recursive Newton-Euler method that is presented in Craig [44] (Equations 3.10-3.18). In order to ensure that the configuration remains in the feasible joint space, the y-positions of each of the joints and the “tip” of the third link are calculated. If the position of either of the joints is equal to or less than zero, a penalty function is assessed in the fitness function (Section 3.5.3).

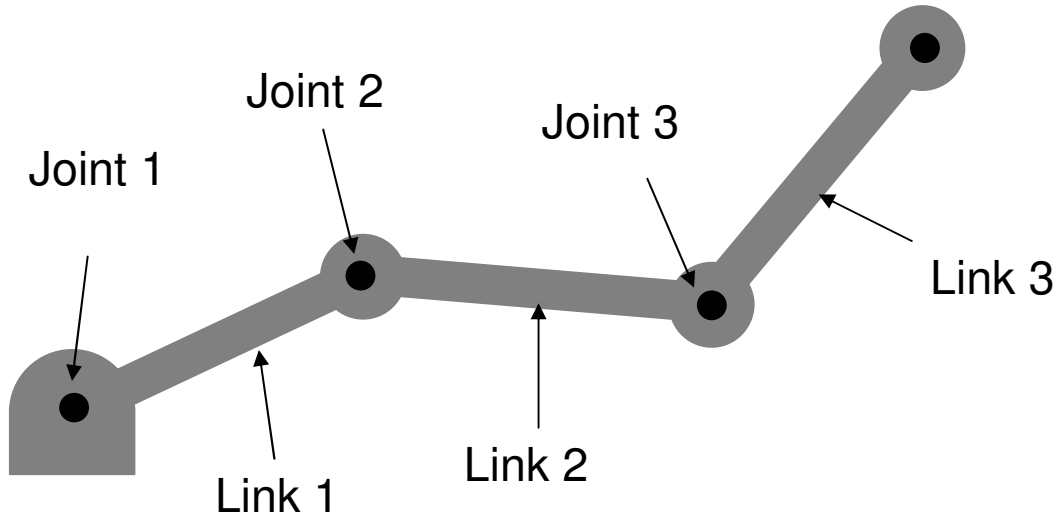


Figure 3.7. Representation of the first mechanism as a open-loop, planar, kinematic chain.

The iterative Newton-Euler scheme propagates the kinematics from the base of the manipulator to the tip, and then propagates the dynamics from the tip back to the base. The outward iterations compute the inertia of each link, and then it is assumed that the forces that achieve this inertia originate at the base. In order to account for gravity, a vertical acceleration is assigned to the base reference frame. In the notation used, the preceding superscript denotes the reference frame in which the value is computed with respect to, and the following subscript denotes the entity that the property relates to. In other words, ${}^1\omega_l$ denotes the angular velocity of link 1 with respect to reference frame 1. The dots indicate the derivatives of the values with respect to time, R denotes a rotation matrix, ${}^iP_{i+1}$ denotes the vector from the origin of coordinate system “ i ” to the coordinate system “ $i+1$ ”, and finally, τ denotes torque.

Newton Euler Kinematics and Dynamics Formulation [44]:

Outward Iterations $i: 0 \rightarrow 5$

$${}^{i+1}\boldsymbol{\omega}_{i+1} = {}^i\boldsymbol{\omega}_i + \dot{\boldsymbol{\theta}}_{i+1} {}^{i+1}\hat{\mathbf{Z}}_{i+1} \quad (3.10)$$

$${}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} = {}^i\dot{\boldsymbol{\omega}}_i + {}^i\boldsymbol{\omega}_i \times \dot{\boldsymbol{\theta}}_{i+1} {}^{i+1}\hat{\mathbf{Z}}_{i+1} + \ddot{\boldsymbol{\theta}}_{i+1} {}^{i+1}\hat{\mathbf{Z}}_{i+1} \quad (3.11)$$

$${}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}\mathbf{R} \left({}^i\dot{\boldsymbol{\omega}}_i \times {}^i\mathbf{P}_{i+1} + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{\omega}_i \times {}^i\mathbf{P}_{i+1}) + {}^i\dot{\mathbf{v}}_i \right) \quad (3.12)$$

$${}^{i+1}\dot{\mathbf{v}}_{C_{i+1}} = {}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{P}_{C_{i+1}} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times ({}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{P}_{C_{i+1}}) + {}^{i+1}\dot{\mathbf{v}}_{i+1} \quad (3.13)$$

$${}^{i+1}\mathbf{F}_{i+1} = m_{i+1} {}^{i+1}\dot{\mathbf{v}}_{C_{i+1}} \quad (3.14)$$

$${}^{i+1}\mathbf{N}_{i+1} = {}^{C_{i+1}}I_{i+1} {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{C_{i+1}}I_{i+1} {}^{i+1}\boldsymbol{\omega}_{i+1} \quad (3.15)$$

Inward Iterations $i: 6 \rightarrow 1$

$${}^i\mathbf{f}_i = {}^i\mathbf{R}^{i+1}\mathbf{f}_{i+1} + {}^i\mathbf{F}_i \quad (3.16)$$

$${}^i\mathbf{n}_i = {}^i\mathbf{N}_i + {}^{i+1}\mathbf{n}_{i+1} + {}^i\mathbf{P}_{C_{i+1}} \times {}^i\mathbf{F}_i + {}^i\mathbf{P}_{i+1} \times {}^{i+1}\mathbf{R}^{i+1}\mathbf{f}_{i+1} \quad (3.17)$$

$$\tau_i = {}^i n_i^T {}^i \hat{Z}_i \quad (3.18)$$

C_i denotes the centroid of the i^{th} link.

The second means of modeling the sub-problem is to assume that the “tip” of the third link remains in constant contact with the ground during the motion. In this instance, the problem can be modeled as a kinematic chain with four rotating joints and a slider mechanism that is shown in Figure 3.8. In this case, the geometric relations involve an equality constraint, as the y -position of the “tip” of link 3 always remains at $y=0$.

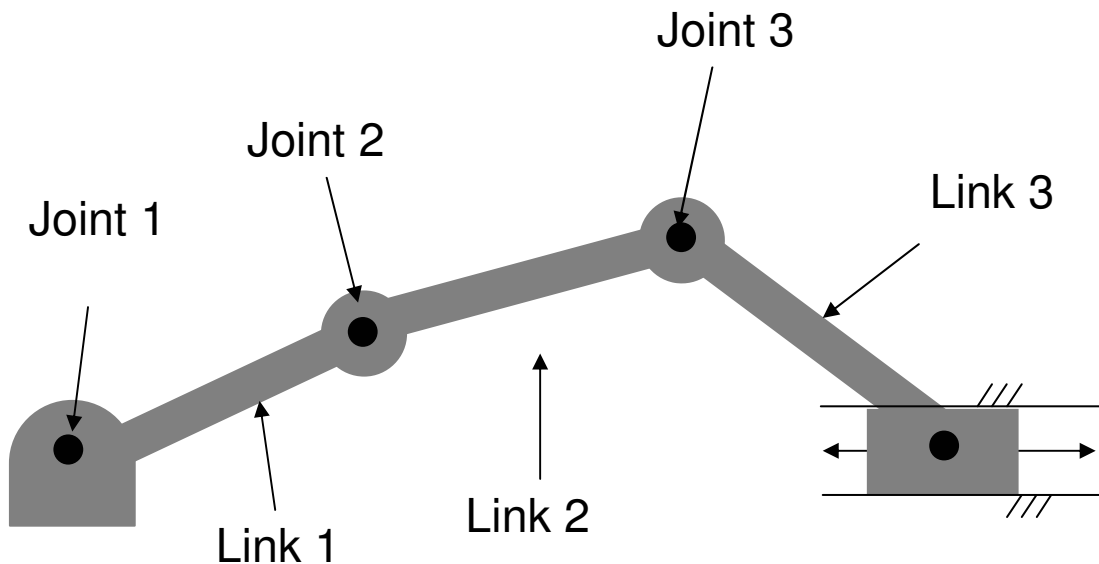


Figure 3.8. Representation of the first mechanism as a closed-loop, planar, kinematic chain.

In this case, the kinematic configuration of the mechanism is governed by a constraint relationship. The mechanism is converted from one with three-degrees of freedom to one with two degrees of freedom. In this system, the angles of joints 1 and 2 are treated as free variables, and the position of joint 3 is calculated using the following constraint relation:

$$\sin \theta_1 + \sin(\theta_1 + \theta_2) + \sin(\theta_1 + \theta_2 + \theta_3) = 0 \quad (3.19)$$

The parameter ϕ can be introduced to simplify the notation:

$$\phi_i = \sum_{j=1}^i \theta_j \quad (3.20)$$

The equation then reduces to:

$$\sin \theta_1 + \sin \phi_2 + \sin \phi_3 = 0 \quad (3.21)$$

The angle of link 3 can then be calculated by the solving the equation to yield:

$$\phi_3 = \sin^{-1}(-\sin \theta_1 - \sin \phi_2) \quad (3.22)$$

$$\text{with: } \theta_3 = \phi_3 - \theta_1 - \theta_2 \quad (3.23)$$

The only constraints on the angles of joints 1 and 2 are that they must satisfy the aforementioned criteria of not resulting in any Cartesian positions below $y=0$, and that:

$$\sin \theta_1 + \sin(\phi_2) \leq 1 \quad (3.24)$$

These constraints define the feasible design space of the problem. Like the previous constraint criteria, a penalty is assessed in the fitness formulation if this constraint is not met.

Both the angular velocity and acceleration of the third link can be computed using the same constraint relationship. The constraint equation can be implicitly differentiated once to yield the angular velocity, and twice to yield the angular acceleration as follows:

$$\dot{\theta}_1 \cos \theta_1 + \dot{\phi}_2 \cos \phi_2 + \dot{\phi}_3 \cos \phi_3 = 0 \quad (3.25)$$

and

$$\ddot{\theta}_1 \cos \theta_1 - \dot{\theta}_1^2 \sin \theta_1 + \ddot{\phi}_2 \cos \phi_2 - \dot{\phi}_2^2 \sin \phi_2 + \ddot{\phi}_3 \cos \phi_3 - \dot{\phi}_3^2 \sin \phi_3 = 0 \quad (3.26)$$

Solving for the angular velocity of link 3 yields:

$$\dot{\phi}_3 = \frac{-1}{\cos \phi_3} \left(\dot{\theta}_1 \cos \theta_1 + \dot{\phi}_2 \cos \phi_2 \right) \quad (3.27)$$

$$\text{with: } \dot{\theta}_3 = \dot{\phi}_3 - \dot{\theta}_1 - \dot{\theta}_2 \quad (3.28)$$

Solving for the angular acceleration of link 3 yields:

$$\ddot{\phi}_3 = \frac{-1}{\cos \phi_3} \left(\ddot{\theta}_1 \cos \theta_1 - \dot{\theta}_1^2 \sin \theta_1 + \ddot{\phi}_2 \cos \phi_2 - \dot{\phi}_2^2 \sin \phi_2 - \dot{\phi}_3^2 \sin \phi_3 \right) \quad (3.29)$$

$$\text{with: } \ddot{\theta}_3 = \ddot{\phi}_3 - \ddot{\theta}_1 - \ddot{\theta}_2 \quad (3.30)$$

It should be noted that this formulation of the jacobian can result in singularities at $\phi_3 = \pi/2$. However, the joint trajectories that are used in this work do not approach this singularity.

The complete kinematic description of each links parameters can then be developed by the outward iteration scheme presented as equations 3.10-3.15. It is important to note that the link velocity must also be calculated, it is given by:

$${}^{i+1}v_{i+1} = {}^i R^{i+1} ({}^i v_i + {}^i \omega_i \times {}^i P_{i+1}) \quad (3.31)$$

The accelerations must then be mapped back to the nonmoving base coordinate system by pre-multiplying by the appropriate transformation matrices, for example:

$${}^0 \dot{v}_3 = {}^0 R_1 {}^1 R_2 {}^2 R_3 {}^3 \dot{v}_3 \quad (3.32)$$

This is done in order to develop the equations for the inverse dynamics of the system.

3.4.2 Dynamics

The dynamic problem of the gait design involves computing which torque values are needed to achieve the desired joint motion. The dynamics have already been developed for the case of the open-loop configuration, as they only involve simple inward-outward propagation. This formulation depends on the manipulator moving in free space, and the reader can consult [44] for a full derivation of the algorithm.

For the case of the second representation, the dynamics must be treated differently than that of a serial manipulator due to the fact that reaction forces arise at the “tip” of the final link. The Newton-Euler equations are still used to develop the dynamics, but the simple iterative method cannot be applied. In order to apply the Newton-Euler equations to the closed-loop case, the forces and the moments are simply summed about each link.

Since there is a frictional force that acts on the “tip” of the third link, a Coulomb friction model represents this force. This means that the force can be calculated using the equation:

$$F_{fr} = -\mu F_R \text{sign}({}^0v_{3tip}) \quad (3.33)$$

The term ${}^0v_{3tip}$ denotes the velocity of the “tip” of the third link, and F_R is the vertical reaction force between the ground and the link. The “sign” denotes the signum function, meaning that the frictional force is always acting in a direction opposite to the velocity of the link. It is of note that an Eulerian framework is used to develop the dynamics for this problem because of the friction term. When forces are directly computed, frictional forces can be accounted for.

Figure 3.9 shows the free body diagrams for mechanism M_1 in the closed loop configuration. The mechanism consists of three bodies, both exerting forces on each other, and on the external environment. The entire mechanism is contained in a plane, so there are only three degrees of freedom for each body: Translation in the x and y directions, and a rotation about the z-axis. The Newton and Euler equations are applied as follows:

$$\sum F = m \dot{v} \quad (3.34)$$

and

$$\sum M = I \dot{\omega} \quad (3.35)$$

This results in three equations for each body. For example, the equations for link 1 are:

$$F_{01x} - F_{12x} = m \dot{v}_{1x} \quad (3.36)$$

$$F_{01y} - F_{12y} = m \dot{v}_{1y} \quad (3.37)$$

$$\tau_1 - \tau_2 + \frac{l}{2} F_{01x} \sin \theta_1 + \frac{l}{2} F_{12x} \sin \theta_1 - \frac{l}{2} F_{01y} \cos \theta_1 - \frac{l}{2} F_{12y} \cos \theta_1 = I \dot{\omega}_1 \quad (3.38)$$

With “ I ” being the moment of inertia of the link about the centroid, and “ l ” being the link length.

The equations can be generalized for any link (with the exception of the final link) as:

$$F_{(i-1)ix} - F_{i(i+1)x} = m \dot{v}_{ix} \quad (3.39)$$

$$F_{(i-1)iy} - F_{i(i+1)y} = m \dot{v}_{iy} \quad (3.40)$$

$$\tau_i - \tau_{i+1} + \frac{l}{2} F_{(i-1)ix} \sin \phi_i + \frac{l}{2} F_{i(i+1)x} \sin \phi_i - \frac{l}{2} F_{(i-1)iy} \cos \phi_i - \frac{l}{2} F_{i(i+1)y} \cos \phi_i = I \dot{\omega}_i \quad (3.41)$$

The generalized equations for the final link are (with $i=3$ in this case):

$$F_{(i-1)ix} + F_{fr} = m \dot{v}_{ix} \quad (3.42)$$

$$F_{(i-1)iy} + F_R = m \dot{v}_{iy} \quad (3.43)$$

$$-\frac{l}{2} F_{(i-1)iy} \cos \phi_i + \frac{l}{2} F_{(i-1)ix} \sin \phi_i + \frac{l}{2} F_R \cos \phi_i - \frac{l}{2} F_{fr} \sin \phi_i = I \dot{\omega}_i \quad (3.44)$$

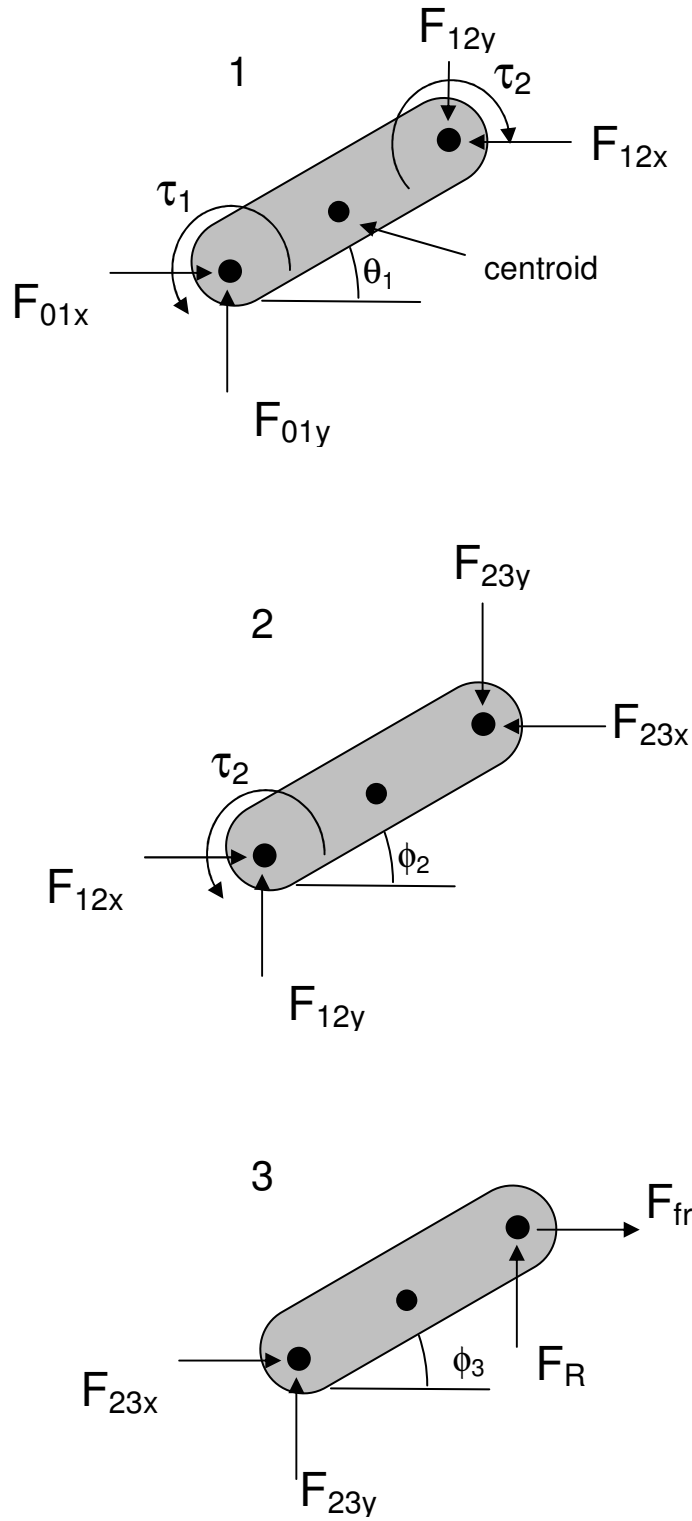


Figure 3.9. Free body diagrams for the links in M_1 .

In the case of M_1 there are exactly 9 unique terms and 9 equations. The entire system of equations can be reduced to one matrix equation, which can then be inverted to solve for each unknown term. The resultant matrices and vectors are:

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{l}{2} \sin \theta_1 & -\frac{l}{2} \cos \theta_1 & \frac{l}{2} \sin \theta_1 & -\frac{l}{2} \cos \theta_1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{l}{2} \sin \phi_2 & -\frac{l}{2} \cos \phi_2 & \frac{l}{2} \sin \phi_2 & -\frac{l}{2} \cos \phi_2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -\mu \text{sign}({}^0v_{ip}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{l}{2} \sin \phi_3 & -\frac{l}{2} \cos \phi_3 & \frac{l}{2} \cos \phi_3 + \frac{l}{2} \mu \text{sign}({}^0v_{ip}) \sin \phi_3 & 0 & 0 & 0 \end{bmatrix} \quad (3.45)$$

$$F_a = \begin{bmatrix} F_{01x} \\ F_{01y} \\ F_{12x} \\ F_{12y} \\ F_{23x} \\ F_{23y} \\ F_R \\ \tau_1 \\ \tau_2 \end{bmatrix} \quad (3.46) \quad \text{and} \quad F_i = \begin{bmatrix} \dot{m}v_{1x} \\ \dot{m}v_{1y} \\ I\dot{\omega}_1 \\ \dot{m}v_{2x} \\ \dot{m}v_{2y} \\ I\dot{\omega}_2 \\ \dot{m}v_{3x} \\ \dot{m}v_{3y} \\ I\dot{\omega}_3 \end{bmatrix} \quad (3.47)$$

The applied force vector F_a is solved for by inverting A as follows:

$$F_a = A^{-1}F_i \quad (3.48)$$

The values $F_a(8)$ and $F_a(9)$ give τ_1 and τ_2 respectively. The matrix problem is solved at each discrete time step of the sampling to obtain the needed values.

In the case of the first mechanism (M_1) both the open-loop and closed-loop solutions are possible. In the heuristic-based framework which is used in this thesis, each model will be tested, and the solution that gives the best results will be used as the actuation strategy.

3.4.3 Mechanism 2 (M_2)

The second step of the locomotion gait involves another configuration change. The Figure 3.10 shows the second step of the locomotion gait. The topology of this mechanism is different than that of the mechanism in the first step (M_1) as it involves four links that must change position instead of three. Further, it can be assumed that it is not possible to have an open-loop solution because of stability reasons. (In addition, testing the first mechanisms open-loop solution showed it to be inferior to the closed-loop solution, so it can be assumed that a solution that would involve lifting four links completely off of the ground would be even less desirable).

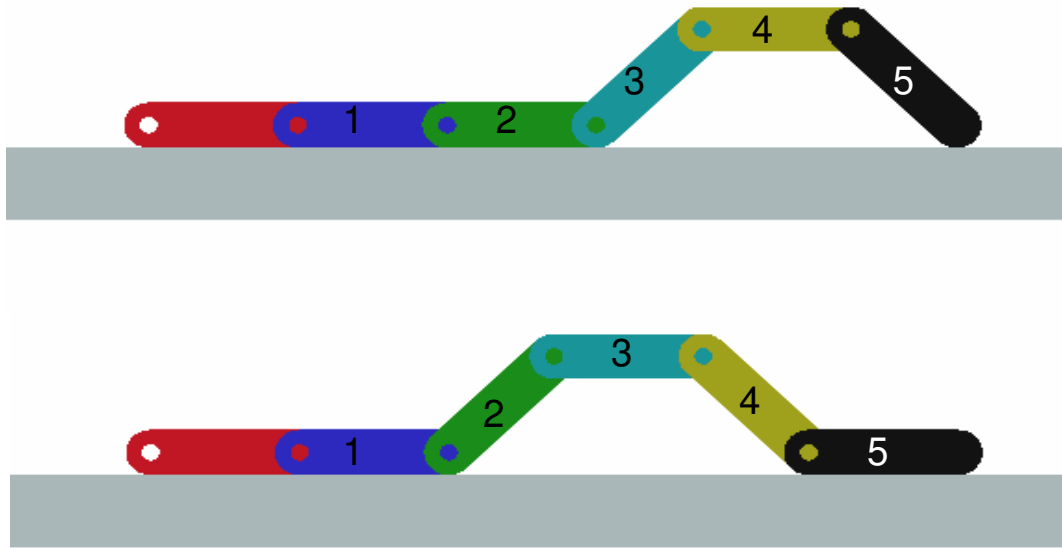


Figure 3.10. Gait step 2, which is modeled with mechanism 2 (M_2).

The mechanism M_2 is modeled similarly to the closed-loop model of M_1 with the difference being that M_2 has an extra link. This means that this mechanism has exactly three degrees-of-freedom. As seen in Figure 3.10, the link marked as “1” is the first non-moving link, and thus can be assumed to be the ground link. Like the analysis of M_1 , the assumption that the base link does not move can be tested once the total force vector is calculated. The rigid body model of the mechanism is shown in Figure 3.11, and the analysis is conducted in the same means as the closed-loop analysis of M_1 .

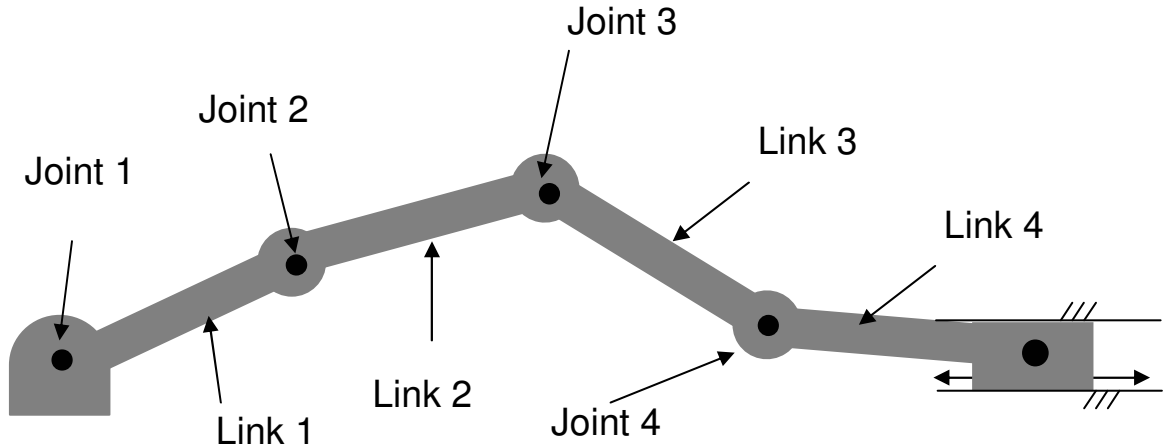


Figure 3.11. Model of M_2 .

The geometric constraint of this mechanism can be described using the following equation, with the same parameters as the previous sub-problem:

$$\sin \theta_1 + \sin \phi_2 + \sin \phi_3 + \sin \phi_4 = 0 \quad (3.49)$$

The kinematics of the mechanism can be developed in the same way that they were for M_1 by solving and differentiating the constraint relation. In this case θ_1 , θ_2 , and θ_3 are allowed to be free variables and θ_4 is the constrained variable. Again, in the case of the dynamics, the joints that are kinematically free are the driven joints and the fixed joint is free to rotate.

Similarly, the dynamics of M_2 can be developed using the exact same method as the dynamics were developed for M_1 . The complete formulation can be found in Appendix A. The solution for this mechanism involves inverting a 12 X 12 matrix, and the torque values are obtained.

3.4.4 Mechanism 3 (M_3)

The third step of the locomotion gait involves the configuration change shown in Figure 3.12. The topology of this mechanism is different than that of M_2 because the entire final link 6 in Figure 3.12 slides on the ground, as opposed to just the tip of a link. This means that there are actually five links that are assumed to be moving with respect to the base link instead of four. However, it can be seen that the degrees-of-freedom, the locomotion and the constraint equations are exactly the same as for M_2 . It can be seen that an additional angle parameter arises because of the 5th joint, however, this is related to the other parameters by $\theta_5 = -\phi_4$ (3.50) (Appendix A discusses this in more detail). It is also important to note that in this case, it cannot be intuitively concluded that the link that is assumed to be the ground link does not move relative to the ground.

The dynamics of M_3 are similar to those of M_2 . However, the difference is that because an entire link is sliding on the ground for M_3 , the slider mechanism shown in Figure 3.11 is assumed to have mass, and thus inertia. This results in a 14 X 14 matrix that must be inverted at each time step of the discrete sampling. The equations for the model of mechanism M_3 are presented in Appendix A.

A noteworthy difference between M_3 and the other mechanism should be made with concern to the base link. In the case of the other mechanisms, it can be seen how the two or three non-moving links act as an anchor for the rest of the motion. However, with M_3 , this is not the case. From Figure 3.12 it can be inferred that there are times when the moving links are bounded by one link that is on the

ground in both cases. For the formulation, it was assumed that one is moving and one is not. Using the exact actuation strategy presented in this section, simulation was conducted on M_3 , and the results show that the downward reaction forces on the side of the mechanism with the actuated links were larger than those on the non-actuated side. This would imply that the frictional force is greater on that link, and thus the assumption that the link does not slip would be correct. This simulation was only conducted with an exact actuation strategy, and with the gait found in Section 3.5.3. Future work should include a deeper analysis of this issue.

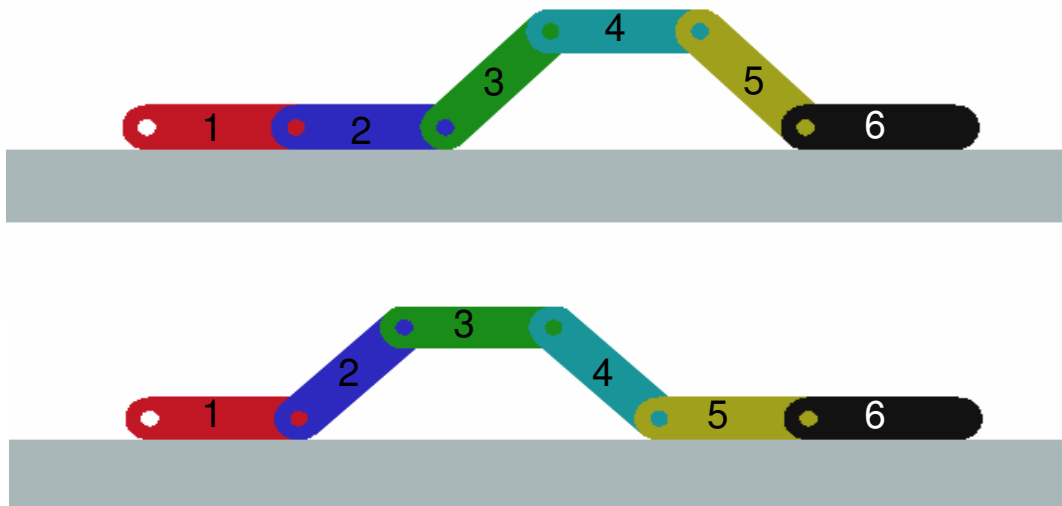


Figure 3.12. Gait step 3, which can be modeled with mechanism 3 (M_3).

3.4.5 Steps 4 and 5

Upon looking back at Figure 3.5, it can be seen that symmetry about M_3 can be exploited. Thus, steps 4 and 5 use mechanisms M_2 and M_1 , respectively, with reversed coordinate systems and opposite trajectory directions. In this case, the

equations do not need to be derived again. The solutions can be obtained using the framework of M_1 and M_2 .

3.4.6 Mapping Solutions

Now that the entire gait problem has been broken up into sub-problems and solutions are found, the problem needs to be re-compiled into a global solution. In other words, for each mechanism, the joints and links have been labeled for that step. The joints in the overall robot are labeled as well. The solutions for each sub-problem must be “mapped” back to the appropriate joints for the entire problem. In order to map the solutions, a simple matrix multiplier scheme is used. The solutions for the step sub-problem can be multiplied by a matrix to map them to the global coordinates of the joints using the matrix “ S_n ”, with “ n ” being the number of the step.

$$\theta_{global} = S_n \theta_{local} \quad (3.51)$$

$$\text{Step 1:} \quad \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \\ \bar{\theta}_3 \\ \bar{\theta}_4 \\ \bar{\theta}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad (3.52)$$

$$\text{Step 2: } \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \\ \bar{\theta}_3 \\ \bar{\theta}_4 \\ \bar{\theta}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \quad (3.53)$$

Step 3 is simply a 5 X 5 identity matrix.

$$\text{Step 4: } \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \\ \bar{\theta}_3 \\ \bar{\theta}_4 \\ \bar{\theta}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \quad (3.54)$$

$$\text{Step 5: } \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \\ \bar{\theta}_3 \\ \bar{\theta}_4 \\ \bar{\theta}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad (3.55)$$

Similarly, the torque solutions can be mapped using the same matrix:

$$\tau_{global} = S_n \tau_{local} \quad (3.56)$$

After the solutions for the sub-problem are mapped to their global coordinates, the solutions can be pieced together in the order of the steps to obtain the time-histories of the joint angles and joint torques for the entire step of the gait.

Note about singularities and the joint space

It should be noted that the developed model assumes rigid body kinematics and dynamics only. This means that the model does not allow for the link lengths to vary, or for the links to deform. This is important to note, because in problems of complex manipulator kinematics and dynamics such as this one, there can be a limited feasible workspace and singularities can arise. If the joints are allowed to slightly bend or extend, many of the configurations that are infeasible or singular in the rigid-body model can actually be physically realizable. This is important to note in a case like this, because it means that there may be gaits that are feasible in practice but are not feasible in the rigid body model.

3.5 Trajectory Generation

With the kinematic and dynamic model completed, this information can now be used to generate trajectories with desired characteristics. In this section, the kinematics and dynamics model is used to generate a metric called “effort”. With the effort metric, a method to generate optimal joint trajectories that minimize effort can then be constructed. In this section, the problem is first formulated and a strategy to arrive at an optimal solution is proposed. A simpler heuristic-based framework that was used to generate trajectories is then presented and several of those trajectories are presented.

3.5.1 B-Spline Trajectory Formulation

As was mentioned in Section 3.2.4, the core of the solution to the trajectory generation problem lies in the representation of the joint trajectories. In this work, the joint trajectories are represented using B-spline curves that are parameterized by a set number of control points. The B-spline representation consists of basis functions that are weighted using control points. The order of the curve and the location of the control points define the curve. By using B-splines to represent the joint trajectories, the joint trajectory can be represented using a collection of parameters. This converts the problem of trajectory optimization from an optimal control problem to a standard parametric optimization problem that can be solved using a search-based approach.

3.5.2 Trajectory Optimization

In this work, a search-based optimization strategy is developed. This entails using an initial guess of the parameters that define the trajectories, and modifying them to find a better solution. The aim of this approach is to find a locally-optimal solution. A local search strategy involves determining which direction to change the parameters to achieve a decrease in the objective function, in our case, the effort. This movement must remain within the constraints of the problem.

The objective function that is minimized in the development of this gait is known as effort. Effort is a standard metric that is used in robot optimization because it incorporates both the time that the action takes, and the amount of torque required by the joints. It is important to note that it is a better metric to use than mechanical work, because mechanical work only measures the amount of work that is done on

the object. Mechanical work does not consider the fact that when a static torque is applied using a motor, the motor consumes energy just to resist the torque, even if it is not moving the load. The effort metric is formulated as:

$$J(\tau) = \frac{1}{2} \int_0^{t_f} \|\tau\|^2 dt \quad (3.57)$$

The overall optimization problem for a minimum effort trajectory-generation problem can be formulated as the following [41]:

$$\text{minimize} \quad J(\tau) = f(\tau(\cdot)) \quad (3.58)$$

$$\text{subject to} \quad M(q)\ddot{q} + h(q, \dot{q}) = \tau \quad (3.59)$$

$$\underline{q} \leq q(t) \leq \bar{q} \quad (3.60)$$

$$q(0) = q_0, \dot{q}(0) = 0 \quad (3.61)$$

$$q(t_f) = q_f, \dot{q}(t_f) = 0 \quad (3.62)$$

The variable “ q ” is used to denote the vector of joint positions in this case, with the Equations 3.60-3.62 specifying bounds and boundary conditions on this variable. These equations are satisfied in the construction of the B-splines, because the boundary control points are fixed. In this case, “ q ” is replaced with “ θ ”. Equation 3.59 is a general equation known as the “state-space” equation; it expresses the torque as a function of shape and dynamic values. In this situation, it is used to

designate the sub-problem of computing the joint torques which has already been developed.

In order to generate optimal joint trajectories at each step of the locomotion gait, a gradient-based search algorithm, similar to that shown in [41] and [43] is proposed. A gradient-based search is possible, because the joint trajectories are specified as B-spline functions, as described in Section 3.2.4. This means that the trajectory of each joint angle is only a function of a set of control points and time. A similar strategy to that in [41] can be used to search for locally-optimal trajectories from initial guesses. Gradient-based searches begin with an initial guess, and work by iteratively updating the parameter values until a local optima is found, as determined using a convergence criterion. Updating the parameter value vector “ x ” is accomplished using the following equation:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} s(x^{(k)}) \quad (3.63)$$

The value $x^{(k)}$ is the current guess of the optima in the parameter space, and $x^{(k+1)}$ is the updated value. In order to move from the current value to the next value, the step size “ α ” and the search direction “ s ” are required. The search direction is constructed using the gradient of the objective function. The step size is typically obtained by finding a value of “ α ” that minimizes the objective function. This is accomplished using a “line search”, meaning that a single variable optimization problem must be solved. A pre-determined step size can also be used.

The major task remaining in order to compute the updated parameter values is the generation of the search direction. The majority of the differences between different search methods lie in the means that the search direction is constructed. Quasi-Newton methods are a class of methods that use the gradient to determine the search direction. An example of such a method would be the Davidson-Fletcher Powell method [46]. Such methods use the gradient as well as an approximation of the Hessian to generate the search direction. The methods vary in the procedures that are used to obtain the approximation of the Hessian. Quasi-Newton methods are desirable as compared to Newton's method, because they only require first-order derivative information as opposed to the second order information that is required to compute the Hessian. This is desirable for more computationally-intensive problems such as robot dynamics.

With the search direction and step size determined, the new value in the parameter space can be obtained, and convergence criteria can be evaluated. If convergence is not reached, the process is repeated with the current $x^{(k+1)}$ value updated to $x^{(k)}$, and a new $x^{(k+1)}$ is found. To summarize, a gradient-based search algorithm would be implemented using the following procedure:

- 1) Assign initial guess values to the variables.
- 2) Compute the gradient of the objective function.
- 3) Construct the search direction vector based on the gradient.
- 4) "Move" in the direction of the search direction.

- 5) Check the termination condition, if satisfied, then terminate. If termination condition is not satisfied, go back to step 2.

The major challenge in the implementation of such an algorithm is the computation of the gradient of the objective function, Equation 3.57. A common strategy that is used when gradient computation becomes intensive is to use finite-difference gradients. To compute a finite-difference gradient, the gradient of the objective function is estimated by perturbing the variable values by a small amount. Unfortunately, it has been reported that such methods give poor results for robot trajectory optimization problems [41].

In order to conduct a gradient-based optimization for this trajectory generation problem, the gradient of the objective function, Equation 3.57, must be computed with respect to the control points, p , in the B-spline formulation (Equation 3.3). The gradient of the objective function can be formulated as [41]:

$$\nabla_p J = \int_0^{t_f} \tau^T \cdot (\nabla_p \tau) dt \quad (3.64)$$

Like the work in [41], this function is integrated using a trapezoidal rule, thus the exact gradient of the approximated integral is calculated.

In computing the gradient of the objective function, the most significant challenge is to compute the gradient of the joint torque function with respect to the control points that define the path. This is, however, possible, because of the nature of the equations. By using a B-spline formulation, this means that:

$$\tau = \tau(p, t) \quad (3.65)$$

because:

$$\theta_i = \theta_i(p, t) \quad (3.66)$$

$$\dot{\theta}_i = \dot{\theta}_i(p, t) \quad (3.67)$$

$$\ddot{\theta}_i = \ddot{\theta}_i(p, t) \quad (3.68)$$

and

$$\tau = \tau(\theta_1, \theta_2, \dots, \theta_N, \dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_N, \ddot{\theta}_1, \ddot{\theta}_2, \dots, \ddot{\theta}_N, t) \quad (3.69)$$

for a given t , and with $i=[1:N]$, with N being the number of joints in the mechanism.

The given formulation means that the gradient of the torque function can be generated by:

$$\frac{\partial \tau_i}{\partial p_j} = \frac{\partial \tau_i}{\partial \theta_k} \frac{\partial \theta_k}{\partial p_j} + \frac{\partial \tau_i}{\partial \dot{\theta}_k} \frac{\partial \dot{\theta}_k}{\partial p_j} + \frac{\partial \tau_i}{\partial \ddot{\theta}_k} \frac{\partial \ddot{\theta}_k}{\partial p_j} \quad (3.70)$$

The calculation of the partial derivative of the joint angle with respect to the control points is trivial, and can be ascertained from the construction of the functions in Equation 3.3.

For the open loop mechanism, computation of Equation 3.69 can be directly performed by differentiating the equations for torque that are produced by the iterative Newton-Euler scheme. An example of such equations is presented in Chapter 4, with Equations 4.2 and 4.3. This computation is relatively straightforward.

In the case of the closed-loop mechanism, the computation of the gradient is more complex. Recall that a system of linear equations was used to generate the torque values for the closed-loop mechanism, thus the solutions for torque were computed by inverting a matrix. Since it would impractical to invert an entire 9 X 9 or 12 X 12 matrix symbolically to generate expressions for torque, implicit differentiation must be used to compute the derivatives of torque with respect to the control points. The general formulation of the implicitly differentiated Equations 3.42-3.44 is shown as the following:

$$\frac{\partial F_{(i-1)ix}}{\partial p_j} - \frac{\partial F_{(i+1)ix}}{\partial p_j} = m \frac{\partial \dot{v}_{ix}}{\partial p_j} \quad (3.71)$$

$$\frac{\partial F_{(i-1)iy}}{\partial p_j} - \frac{\partial F_{(i+1)iy}}{\partial p_j} = m \frac{\partial \dot{v}_{iy}}{\partial p_j} \quad (3.72)$$

$$\begin{aligned} \frac{\partial \tau_i}{\partial p_j} - \frac{\partial \tau_{i-1}}{\partial p_j} + \frac{l}{2} \left[\frac{\partial F_{(i-1)ix}}{\partial p_j} \sin \phi_i + \frac{\partial \phi_i}{\partial p_j} F_{(i-1)ix} \cos \phi_i \right] + \frac{l}{2} \left[\frac{\partial F_{(i+1)ix}}{\partial p_j} \sin \phi_i + \frac{\partial \phi_i}{\partial p_j} F_{(i+1)ix} \cos \phi_i \right] \\ - \frac{l}{2} \left[\frac{\partial F_{(i-1)iy}}{\partial p_j} \cos \phi_i - \frac{\partial \phi_i}{\partial p_j} F_{(i-1)iy} \sin \phi_i \right] - \frac{l}{2} \left[\frac{\partial F_{(i+1)iy}}{\partial p_j} \cos \phi_i - \frac{\partial \phi_i}{\partial p_j} F_{(i+1)iy} \sin \phi_i \right] = I \frac{\partial \dot{\omega}_i}{\partial p_j} \quad (3.73) \end{aligned}$$

Since the values of the forces and torques are already computed, and the derivatives of the kinematics can be computed with respect to p using the same procedure that was used to compute the derivatives of the open-loop system, this system can be re-organized into a new matrix equation:

$$\frac{\partial F_i}{\partial p_j} = A'^{-1} \frac{\partial f_i}{\partial p_i} \quad (3.74)$$

Where the term F_i denotes a vector of the torques and applied forces, as in Equation 3.48, A' is the new coefficient matrix, and f_i is a new matrix that is created by combining the inertial derivatives and the terms on the right hand side of Equations 3.70 through 3.72 that do not contain derivatives of F_i . As in 3.48, the derivatives of torque are components of the applied force gradient. The values of Equation 3.73 at the discrete time steps can then be inserted into Equation 3.63, and the gradient computed using a trapezoidal integration.

In order to perform this computation, a means of generating Equations 3.70-3.72 from Equations 3.42-3.44 must be generated. This operation should be performed by an automated means because it would be cumbersome to execute manually. All that would be required to automate the generation of such equations would be a simple string processing algorithm to complete the implicit symbolic differentiation. This would be simple because the terms are restricted to functions of ϕ and products of these terms with sine and cosine terms. A substitution scheme

could be written to complete this operation. This program has not yet been written, and should be included in future work (see Section 6.3.3).

3.5.3. A Heuristic-Based Trajectory Generation Approach

In view of the fact that complications arise when generating optimal gaits, a heuristic-based trajectory generation approach was used to demonstrate how trajectories could be generated using the kinematics and dynamics model. This approach should also be able to lend some insight to what trajectories would be better than others in the trajectory generation problem. This approach does not, however, find optimal solutions, and has the following three drawbacks:

- 1) It is based on a binary approximation of the gradient.
- 2) It is based on a discrete approximation of the effort metric.
- 3) It can only search discrete points in the variable space.

In general, this approach can be described as a simple perturbation-based search. The approach begins with an initial set of parameters, and the effort metric is approximated. The parameters are then individually perturbed to see if they result in a decrease in effort, and the results are recorded. A new set of parameter values are then generated based on which perturbations resulted in decreased effort. The search continues until a specified number of iterations is reached.

To use this approach, the effort metric must first be approximated. The effort metric can be approximated by:

$$J(\tau) = \frac{1}{2} \sum_{j=1}^N \sum_{m=1}^{M-1} \left(\frac{\tau_{(m+1)j} + \tau_{mj}}{2} \right)^2 \Delta t \quad (3.75)$$

This function estimates the average torque over an interval using the endpoints. There are $M-1$ intervals, with M boundary points of the intervals. The value of Δt is calculated by taking the number of intervals and dividing by the total time to complete the step. N is the number of joints in the mechanism.

The heuristic-based search approach works by having an algorithm that takes the non-boundary control points of the trajectory and generates the effort required by that trajectory as an output. The program is called COMPUTE_DYN and takes a $N \times M$ matrix as an input, with N being the number of free control points that define each joint trajectory, and M being the number of free joints in the particular sub-problem. The output of the program is the effort required to achieve robot motion.

With the control point data entered, and the fixed boundary control points stored in the program, the complete trajectories of the free (driven) joints are then calculated, along with the associated time derivatives. The results are discretized and stored as discrete point sets. This is done because of the simplicity associated with working with discrete point sets as opposed to continuous functions, as explained earlier. The second step of the algorithm is to calculate the inverse kinematics of the fixed joints. This is accomplished using the method presented previously. At this point, the complete trajectories for each angle in the snake have all been generated.

Thus, the shape of the snake, and the rate of change of shape are fully defined and stored as point sets.

Upon knowing the inverse kinematics, the direct kinematics of the system must then be computed. This must be done because in order to apply the Newton equation to the system, the rectilinear accelerations of each link must be known. After the direct kinematics are calculated, the matrix equation is then constructed in order to calculate the dynamics of the system, and the torque values of each driven joint are extracted at each step of the time sampling.

At this point, constraints on the system must be addressed. The constraints, which have been previously mentioned in the kinematics section, are that each joint of the mechanism must not lie below the position $y=0$. This is due to the fact that if the joints are configured such that this was the case, then the assumption about the ground contact would be invalid. The second constraint is that the equation that was used to solve the position of the fixed joints must have a real solution. The algorithm tests each of these constraints and determines if both of them are satisfied.

The final step of the algorithm is calculating the effort metric. The metric is shown as equation 3.63. A simple penalty scheme is used to address the constraints. The way that the penalty is assessed is that if the constraint criteria are violated, the effort value is set at 100 (arbitrary large number). If the constraint criteria are satisfied, the calculated effort value is the output.

With the previous program functioning as the objective function, a separate program was developed to run the search routine. This program uses COMPUTE_DYN and its output to search for a good solution. The program begins

with an initial value for the matrix of control points, C_0 , a number specifying the number of iterations the search process should be attempted, and a value specifying the step size for the search, Δ . The program then runs COMPUTE_DYN with the initial control points C_0 , and obtains the effort value of the gait step that is described by the initial control points. After this value is obtained, an approximated gradient of the effort value with respect to the free control points is calculated. This is accomplished by individually perturbing the control points by both Δ and $-\Delta$. After each perturbation, the new effort value is calculated. If the new effort value is lower than the initial effort value, then a 1 is stored in the direction (“Dir” in Figure 3.14) matrix. If not, a zero is stored. This matrix can then be used to construct an approximated gradient that stores the complete $N \times M$ direction of decreasing effort value. It is important to note that this is a coarse solution because it only considers each parameter (control point) individually, and it does not store information related to the amount of effort change achieved by each perturbation. Nonetheless, it results in a direction of decreased effort value in most cases. A search direction is then constructed by multiplying this “gradient” by Δ , and the C matrix is updated by $C=C+search$. This process is repeated and the search finds trajectories with lower effort values until it reaches the specified number of iterations. A flow chart for this routine is shown in Figure 3.14.

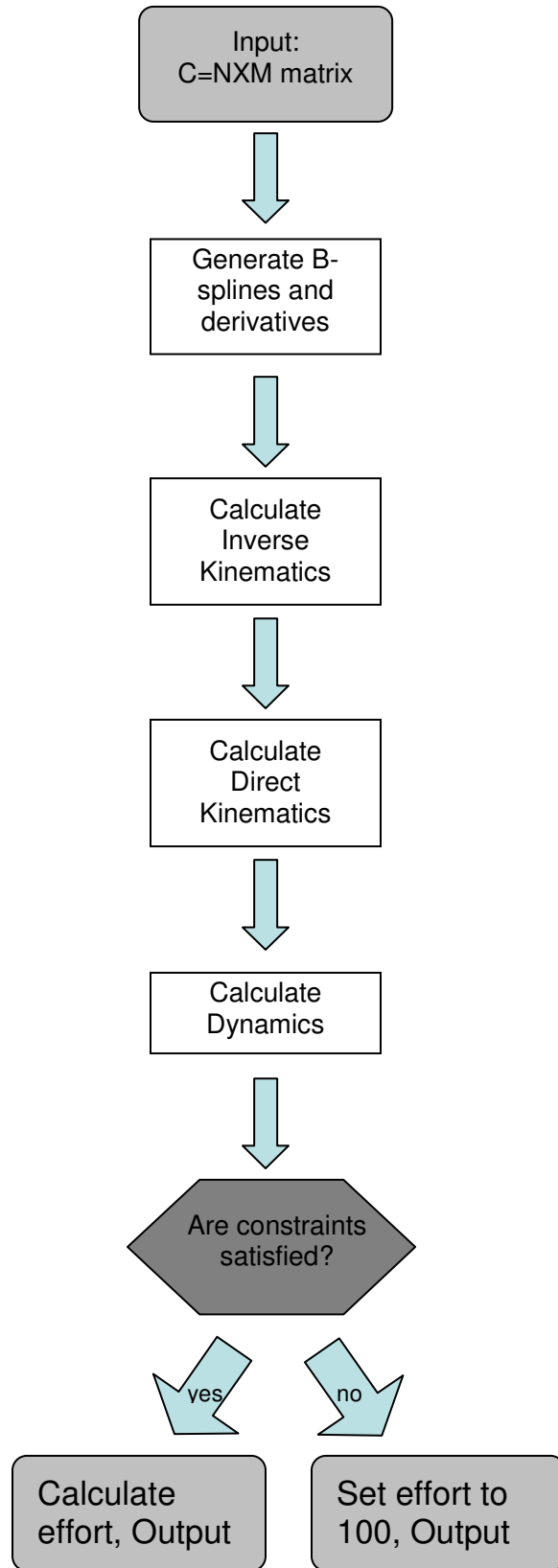


Figure 3.13. Architecture of fitness evaluation function – `COMPUTE_DYN(C)`.

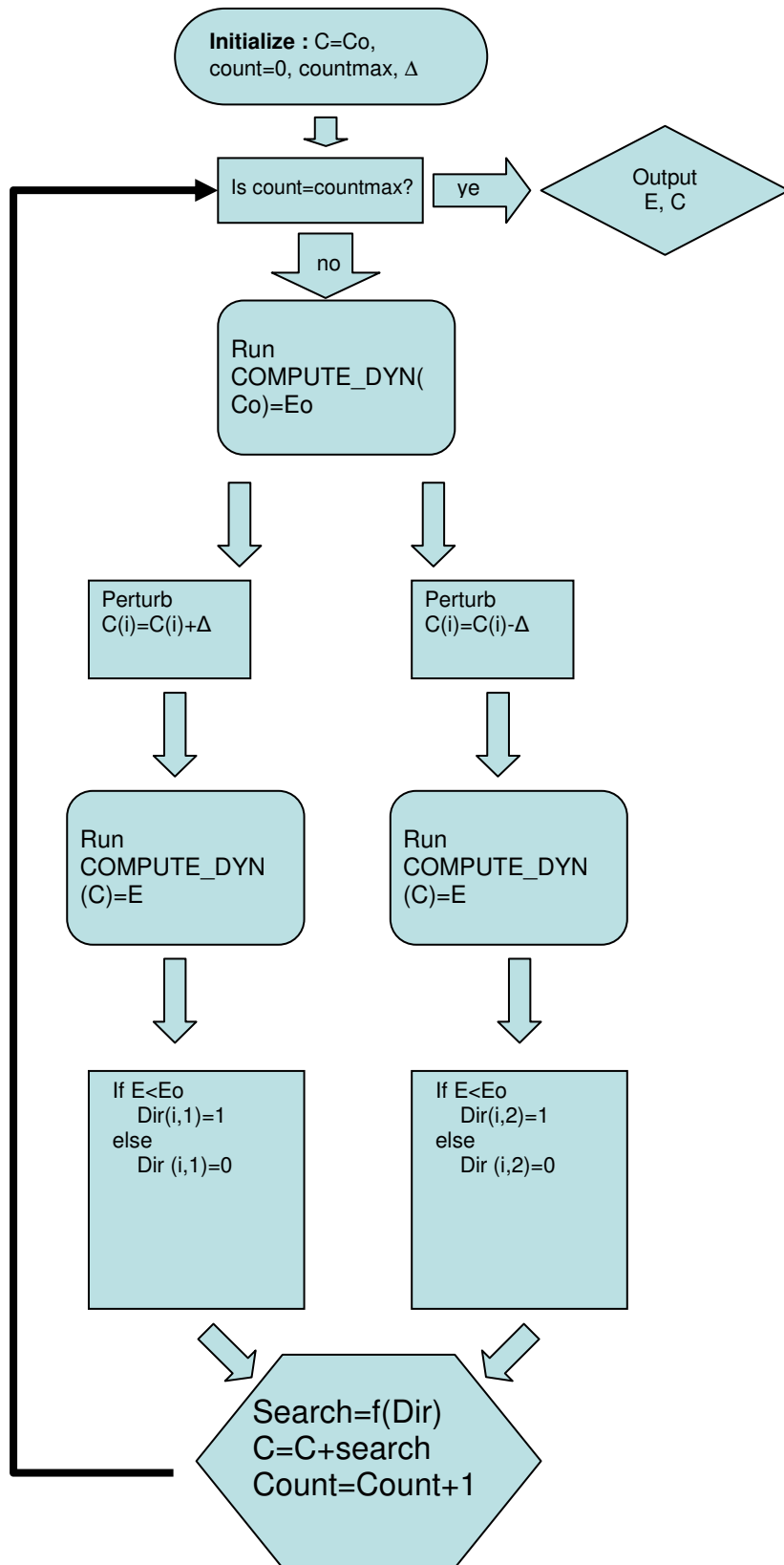


Figure 3.14. Organization of search algorithm.

3.6 Results

The heuristic-based search routine was conducted for each of the gait steps. Each step was analyzed individually, and effort was minimized for each sub-problem. For the first search run, $\pi/4$ is used as the gait angle, 1 second as the gait step time, and a friction coefficient of $\mu=0.4$. This would be a typical value of the coefficient of kinetic friction for a material such as plastic on a smooth concrete or metal. In addition, the length of the link is $l=0.14\text{m}$ and the mass of the link is $m=0.015\text{kg}$. These parameters are chosen because they are the actual link parameters of the realized robot (Chapter 5). The moment of inertia is calculated by treating the link as a thin rod, this equation was chosen because in most snake-inspired robot implementations, the links take the form of rods, where the length is significantly greater than the link. Subsequent refinement of the study could easily plug in a different equation for the moment of inertia to better fit the specific snake that one would wish to analyze. The discrete sampling was conducted at a rate of 20 times/interval, and the value of Δ was set at $\alpha/100$. These parameters were generated heuristically after testing the algorithm.

The results are shown here for 50 iterations of the algorithm. This number of iterations was shown to give good convergence. However, there were several cases where the convergence oscillated. In these cases, the control points that produced the lowest effort value were determined to be the appropriate solution.

For step 1, both the open and closed-loop models were used to generate a solution. It was determined that the closed-loop solution required significantly less effort than the open loop solution (effort values on the order of 6.0×10^{-3} for open-loop

and 1.7×10^{-3} for closed loop for a 45 degree gait angle), thus the closed-loop solution was used. This should be somewhat intuitive, because although friction affects the solution, the joints do not have to support all of the weight of the links in the closed loop solution.

Figure 3.15 shows the results of the heuristic-based search for gait step 1. Each data point represents a solution found during the algorithm. The effort values are shown to decrease, and then converge to a solution after 27 iterations.

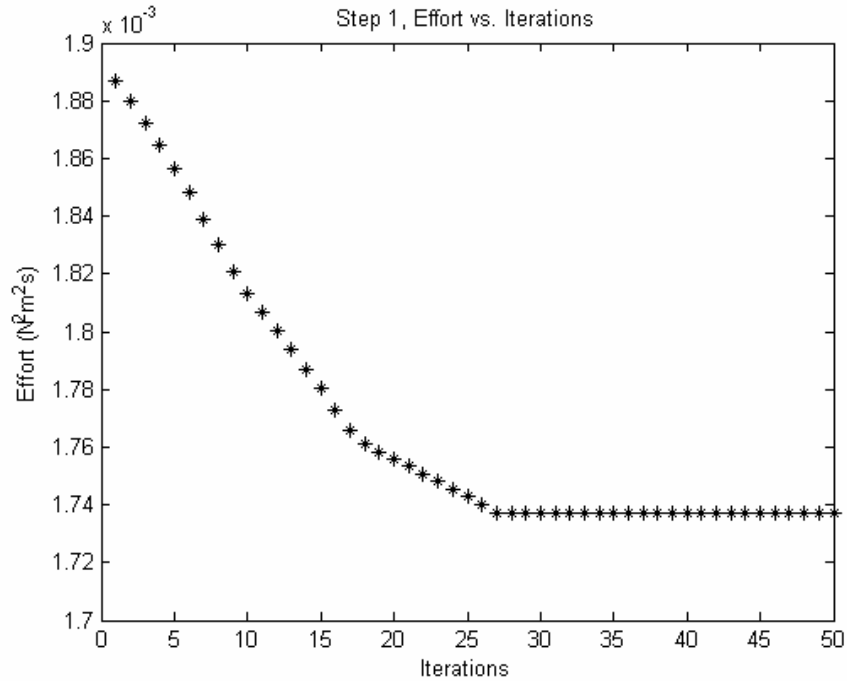


Figure 3.15. Effort vs. iterations for step 1 search.

For this problem, the solution to the trajectory problem was assumed to follow an approximately linear path, with zero-velocity endpoints. This meant that the

control points were assumed to fall on a straight line between θ_o and θ_f . The initial C matrix is then:

$$C_o = \begin{bmatrix} 0 & 0 & \frac{\alpha}{4} & \frac{\alpha}{2} & \frac{3\alpha}{4} & \alpha & \alpha \\ 0 & 0 & -\frac{\alpha}{4} & -\frac{\alpha}{2} & -\frac{3\alpha}{4} & -\alpha & -\alpha \end{bmatrix}$$

With $\alpha=\pi/4$,

$$C_o = \begin{bmatrix} 0 & 0 & 0.1963 & 0.3927 & 0.5890 & 0.7854 & 0.7854 \\ 0 & 0 & -0.1963 & -0.3927 & -0.5890 & -0.7854 & -0.7854 \end{bmatrix}$$

The final value of the C matrix for this solution was found to be:

$$C = \begin{bmatrix} 0 & 0 & 0.1335 & 0.2670 & 0.3848 & 0.7854 & 0.7854 \\ 0 & 0 & -0.2670 & -0.5262 & -0.7933 & -0.7854 & -0.7854 \end{bmatrix}$$

The plots of the B-splines created from the resultant C matrix are shown in Figure 3.16 and Figure 3.17. These show the time history of the angles of joints 1 and 2 during the step of the gait.

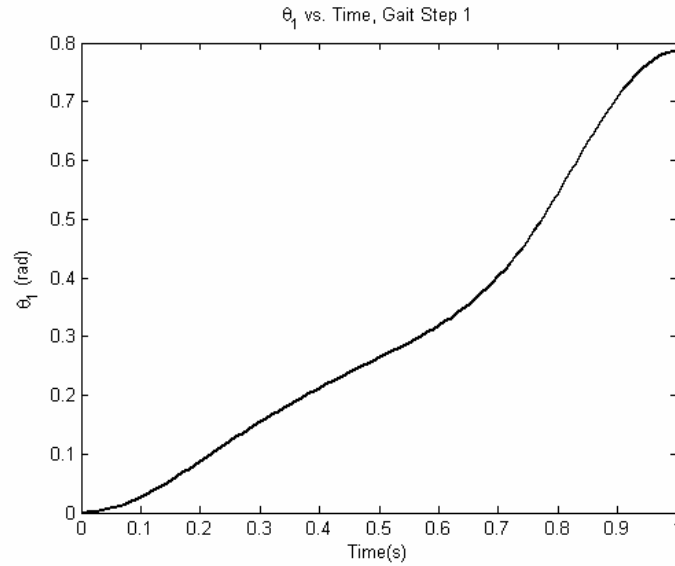


Figure 3.16. Resultant trajectory function for joint 1.

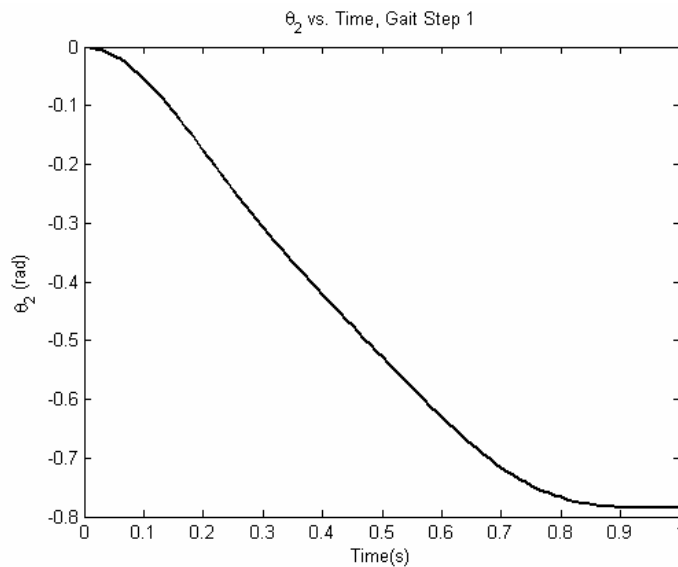


Figure 3.17. Resultant trajectory function for joint 2.

Snapshots of the motion of the mechanism are shown in Figure 3.18. These snapshots are sampled ten times over the course of the gait step. The leftmost link is link 1, with the next link being link 2, and the rightmost link being link 3.

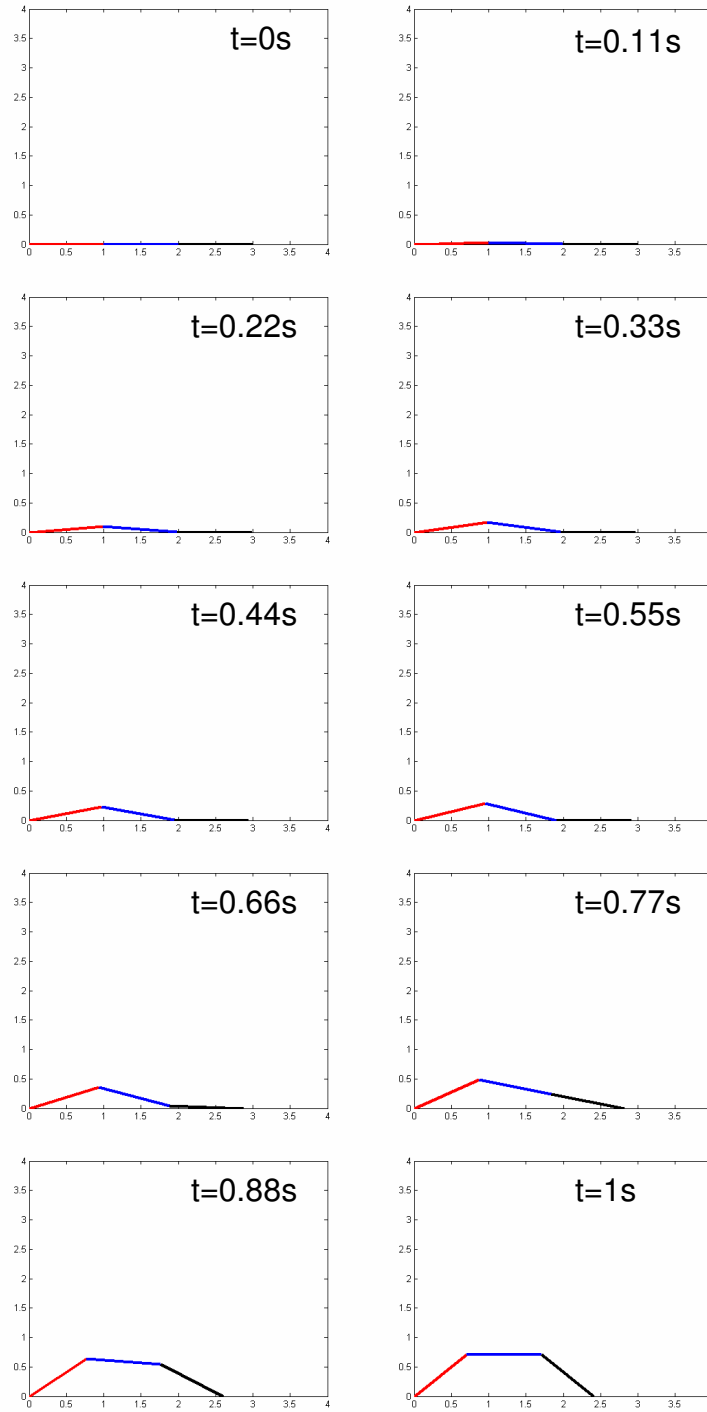


Figure 3.18. Snapshots of gait step 1, shown at 0.1s intervals.

With the complete joint trajectories calculated, the actuation torque required for each joint for the entire time period of the gait step can then be calculated. The torque calculation is already completed during the formulation of the dynamics, and can be extracted. The torque vs. time profiles for joints 1 and 2 are shown in Figure 3.19 and Figure 3.20.

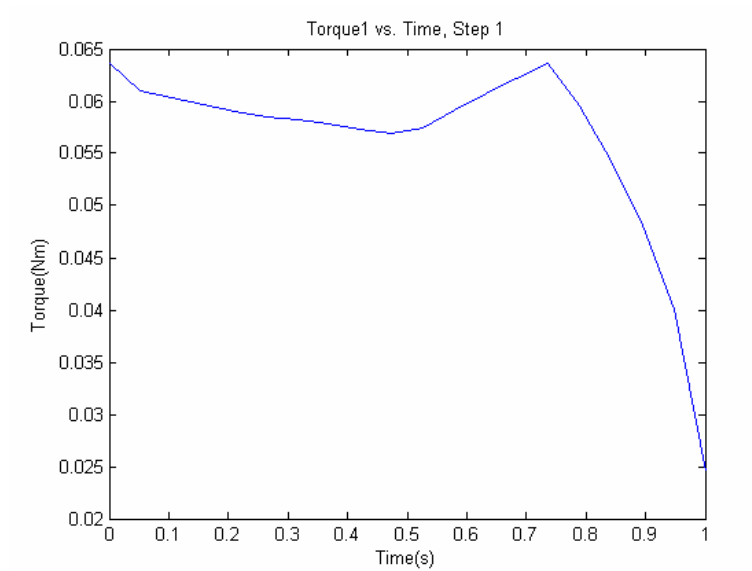


Figure 3.19. Torque versus time for joint 1.

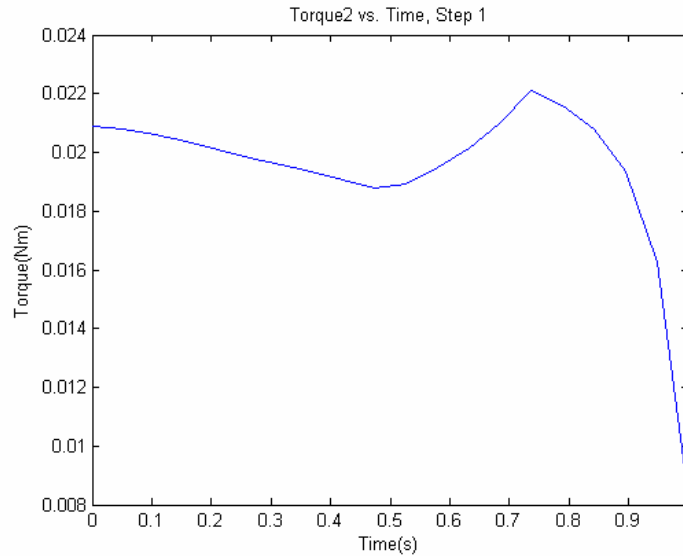


Figure 3.20. Torque vs. time for step 2.

For step 2, the same approach is used. However, in this case, control points following a linear path are not in the feasible joint space of the mechanism. In this case, several point configurations were sampled in order to find one that is in the feasible joint space to use as C_o . The initial point used:

$$C_o = \begin{bmatrix} 0 & 0 & 0.2 & 0.4 & 0.5 & 0.7854 & 0.7854 \\ 0.7854 & 0.7854 & 0 & -0.1 & -0.2 & -0.7854 & -0.7854 \\ -0.7854 & -0.7854 & -0.2 & -0.7 & -0.7 & -0.7854 & -0.7854 \end{bmatrix}$$

The solutions found by the routine are shown in Figure 3.21. Since the solution converged to one with a slightly higher effort value than the solution with the lowest effort value, the solution with the lowest effort is used. This occurs because the gradients are discretely sampled and not directly computed.

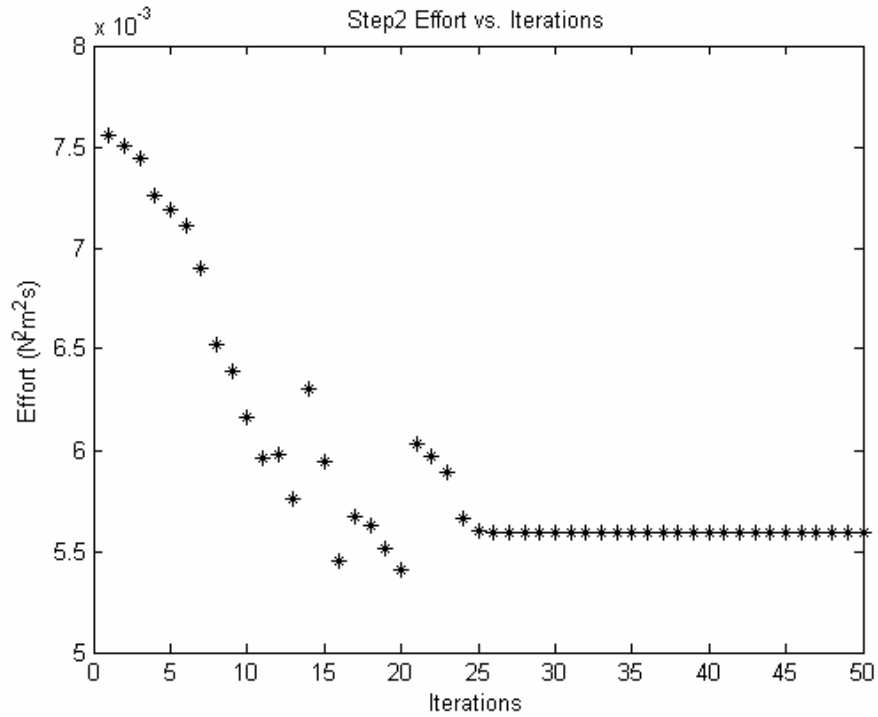


Figure 3.21. Effort vs. iterations for step 2.

The best value of C is found to be:

$$C = \begin{bmatrix} 0 & 0 & 0.2628 & 0.5021 & 0.5314 & 0.7854 & 0.7854 \\ 0.7854 & 0.7854 & 0.0707 & -0.0372 & -0.1843 & -0.7854 & -0.7854 \\ -0.7854 & -0.7854 & -0.0979 & -0.6764 & -0.7393 & -0.7854 & -0.7854 \end{bmatrix}$$

The trajectories for joints 1-3 are shown in Figure 3.22 - Figure 3.24. In addition, the snapshots of the movement are shown in Figure 3.25. As in the snapshots for step 1, the joints are in order from left to right.

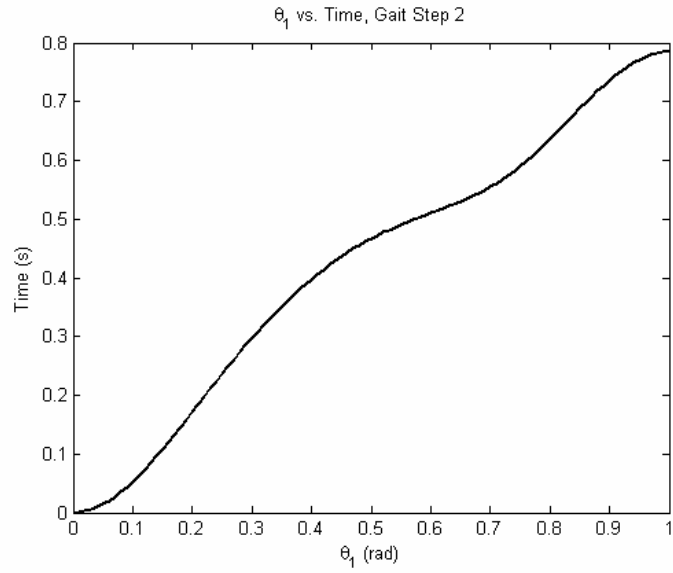


Figure 3.22. Resultant trajectory for joint 1.

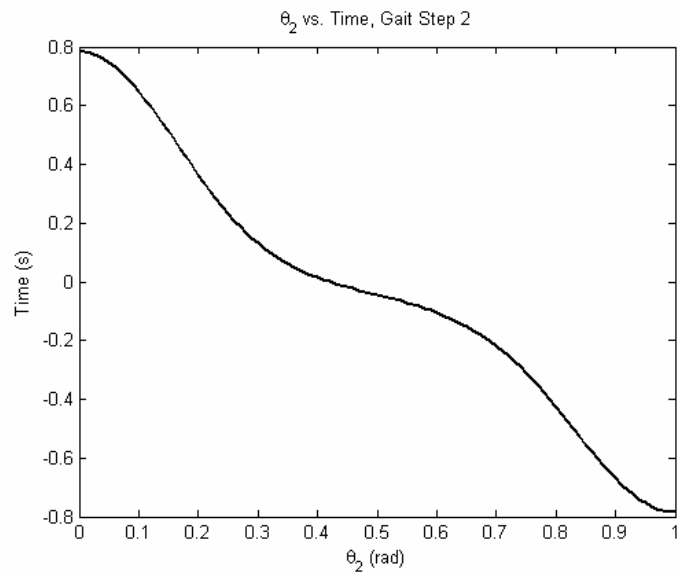


Figure 3.23. Resultant trajectory for joint 2.

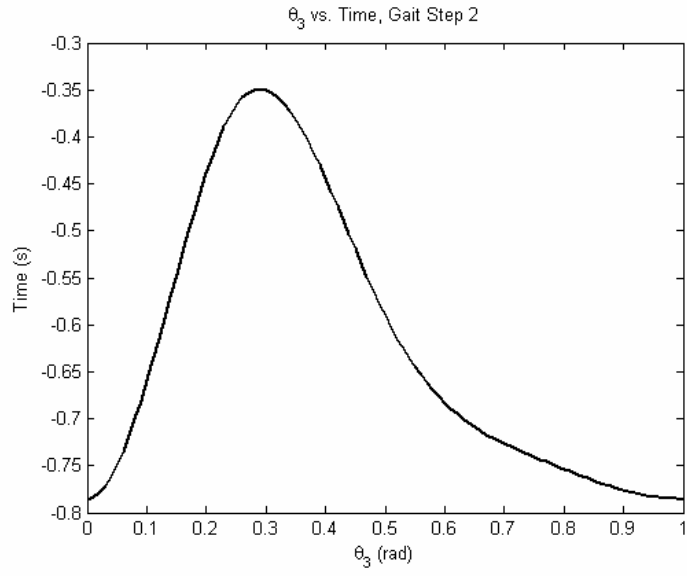


Figure 3.24. Resultant trajectory for joint 3.

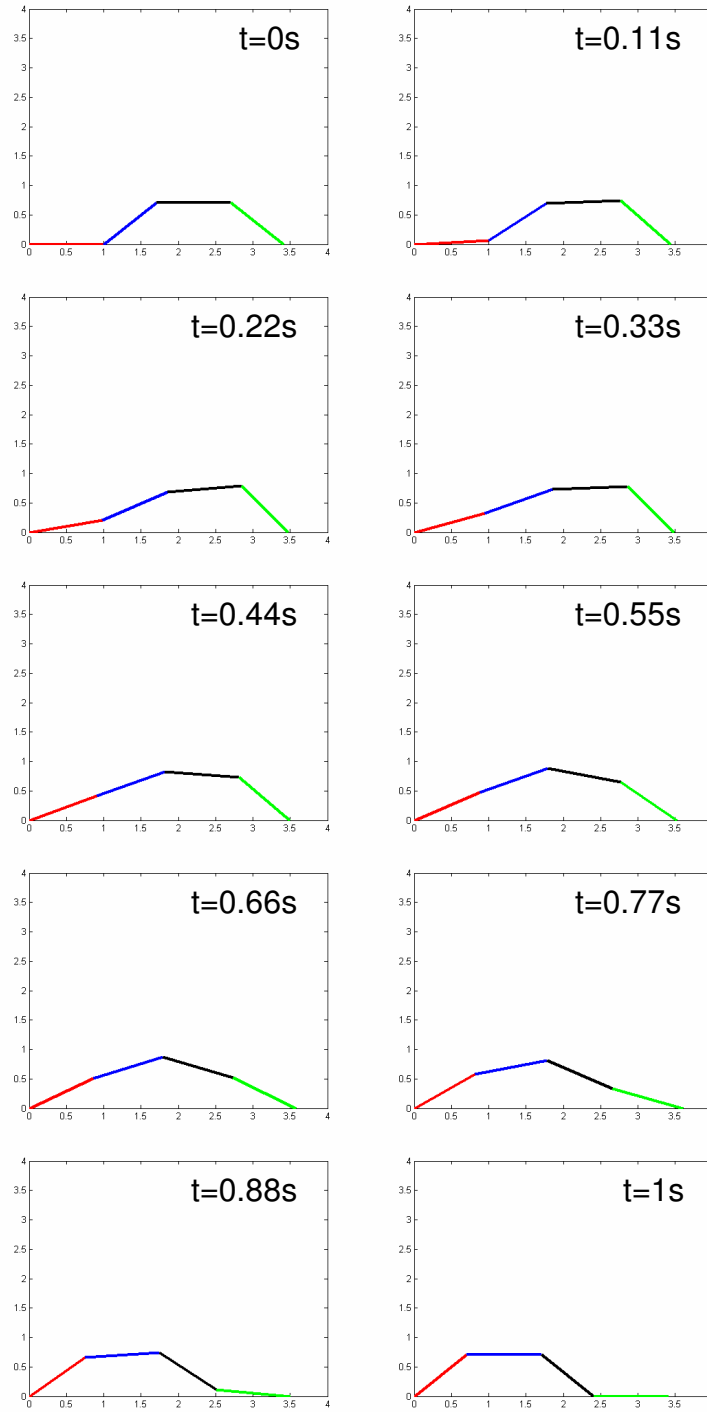


Figure 3.25. Snapshots of gait step 2, shown at 0.1s intervals.

The torque as a function of time for this gait step can also be computed like it was computed for step 1. The torque versus time for the three actuated joints in this gait step is shown in Figure 3.26 - Figure 3.28.

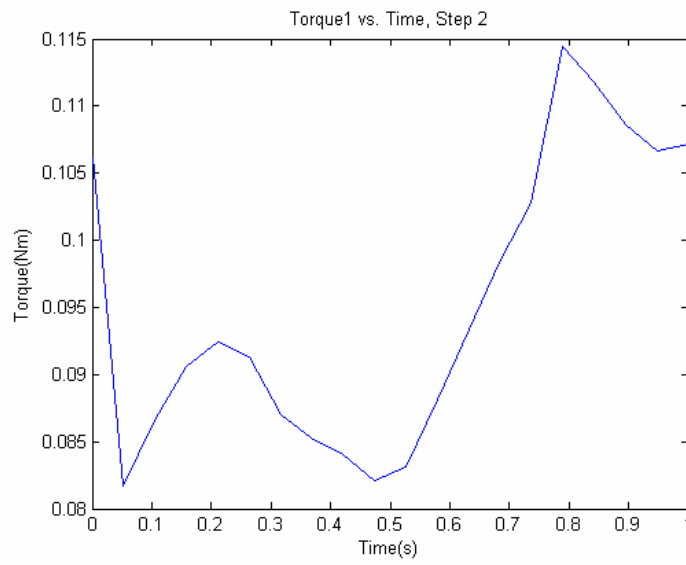


Figure 3.26. Torque vs. time for joint 1.

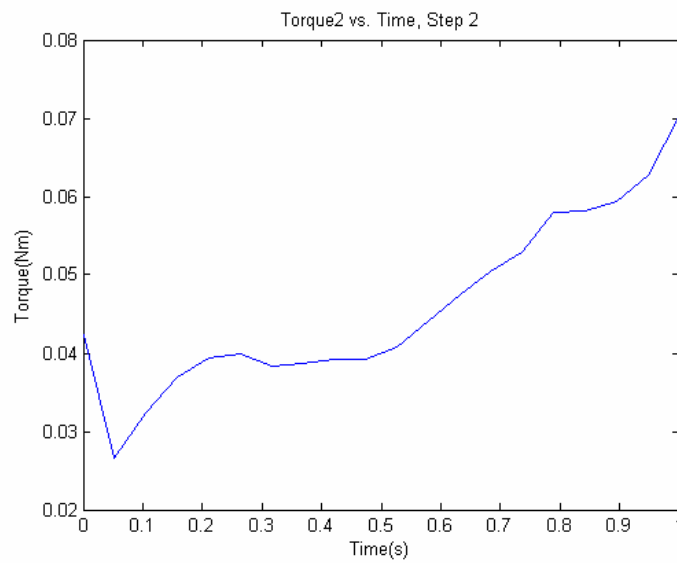


Figure 3.27. Torque vs. time for joint 2.

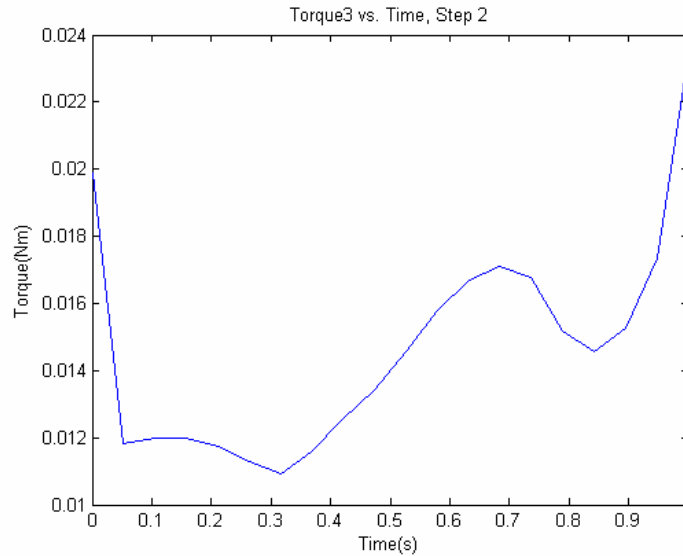


Figure 3.28. Torque vs. time for joint 3.

3.7 Validation

In order to provide experimental validation for the gait proposed in this chapter, a simple snake-inspired robot was developed and the gait was demonstrated. This robot was constructed using Lexan plastic links and servomotors, and was controlled using a microchip. Unlike the robot that is presented in Chapter 5, this robot is not modular and does not operate on its own power supply; it was constructed as a simple an efficient means to demonstrate the rectilinear gait.

The major assumption that was made in the development of the gait was that during the motion, the stationary links would prevent backward slipping as the wave was propagated up the robot. If this assumption is correct, the robot should advance by the distance calculated in Equation 3.9 during each gait cycle, and the forward velocity at which the robot would move could be calculated by dividing that

advancement value by the time required by each gait cycle. This assumption was verified by demonstrating a snake-inspired robot moving on a carpet with the generated rectilinear gait, and measuring if slip occurred.

A 12-point linear approximation of the gait generated in this chapter was used in this experiment, and the gait cycle required three seconds to complete. Each link used was 8.9 cm long. Because the servomotors used in this experiment only allowed 40 degrees of motion, the gait was slightly modified to accommodate this, meaning that all angles above 40 degrees in the computed gait were reduced to 40 degrees in the implemented gait. Snapshots of the moving robot are shown in Figure 3.29.

The robot gait was demonstrated for several gait cycles, and the distance that was traveled was measured with a ruler and compared to the predicted distance. The distance per gait cycle that was predicted for the robot from Equation 3.9 was calculated as 4.14 cm, resulting in a forward velocity of 1.38 cm/s. The average measured value of advancement per cycle from experiments was 3.8 cm, resulting in an average velocity of 1.27cm/s, or an 8% error from the estimated value. This would indicate that the error due to slippage not accounted for in the model is no more than 8%. The error could, however, be due to factors in addition to slippage such as the time response of the servomotors. Observation would indicate that in several instances the servomotors do not seem to be able to attain the specified angle in the amount of time required.

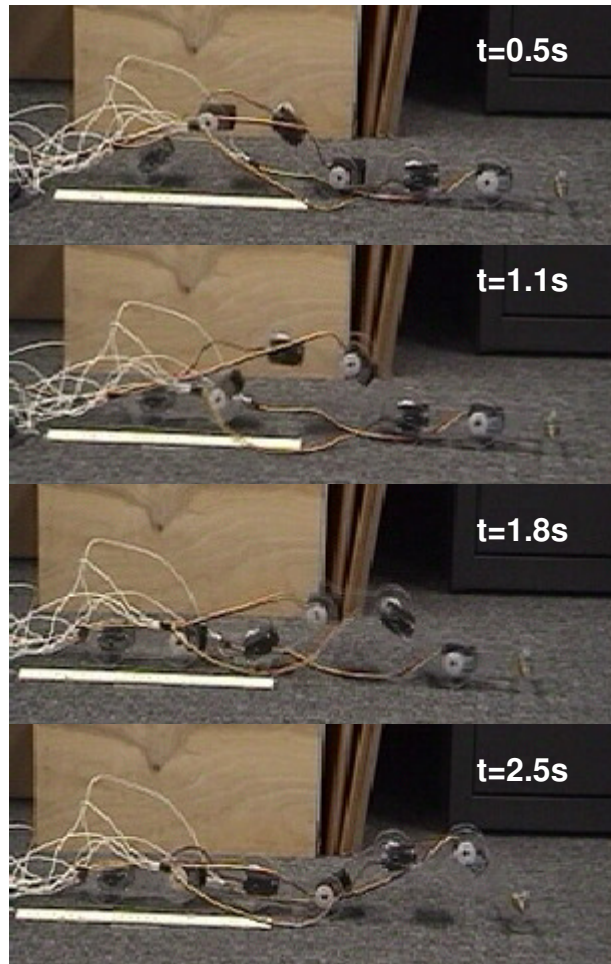


Figure 3.29. Snapshots of gait.

3.8 Summary

To summarize, in this section a means of computing the kinematics and dynamics of a snake-inspired robot moving with a vertical, rectilinear gait has been developed. This allows for direct computation of joint torques during this motion. This model was developed by breaking the motion up into a series of steps with their respective topological mechanisms, and analyzing each individually. This information was then used to develop an algorithm for gait synthesis, based on a parameterization of joint trajectories using B-spline curves. A means of directly computing the gradient of the

torque function with respect to the curve functions was also proposed. This allows for Quasi-Newton based search to be applied to this trajectory generation problem. Results were presented for a simplified, heuristic-based search algorithm. The work in this chapter allows computation of all of the relevant dynamic information (especially joint torques) at any instant in time for a particular class of gait, and generates joint trajectories for these gaits. Finally, the generated gait was demonstrated on a snake-inspired robot, and the assumptions about slipping were shown to be valid in the demonstrated case.

Chapter 4 - Parametric Study

4.1. Introduction

In this chapter, the kinematic and dynamic model and the discrete search algorithm presented in Chapter 3 are used in order to conduct a parametric study on snake-inspired robot design. Parameters that affect the performance of a snake-inspired robot are identified and studied. The goal of this study is to determine how variation in bulk parameters such as mass and length affect the performance of snake-inspired robots. Additionally, the effects of parameters such as gait step time on the gaits are discussed to determine how gaits can be modified to be improved when the parameters of the robot are changed. An understanding of how changes in parameters of snake-inspired robots affect their performance would provide information needed to better design them with respect to specific mission requirements and constraints.

4.2. Mission Requirements

Snake-inspired robots can be built to a variety of specifications, and their designs can be subject to a variety of constraints. Constraints and requirements on snake-inspired robots can be placed on the robot either by the mission, the manufacturing process, or other reasons. An example of a mission requirement would be a payload that a snake-inspired robot would need to carry. If there is a significant payload (for example: sensors) that the robot would need to carry, then the robot would have to be designed to accommodate such a payload. Other important mission requirements may consist

of parameters such as maximum velocity, range, and the cross-section of the smallest passageway that the robot could fit through. In order to develop functional snake-inspired robots, they must be designed to meet such requirements.

In order to design with respect to mission requirements, there are many trade-offs that must be made, and trends that must be considered during the design process. For example, a strong robot would often be heavier and bulkier. Similarly, a faster robot may be less efficient and thus not able to withstand a longer distance mission. Designers would have to make decisions as to how to select parameters in view of such trends and trade-offs.

Consider the following task. A robot designer needs to design a robot for a search and rescue mission. Previous experience has indicated that in typical missions where a robot has to travel into such an environment to locate survivors, the robot must have a maximum cross sectional area of 50 cm^2 and must be able to travel at least 300 meters. Both the volume and mass of the payload is known, and the designer has a group of servomotors and batteries that they can select from. The design should be developed in order to minimize the amount of time that the robot takes in order to travel the specified distance, meaning that the average velocity should be maximized subject to the aforementioned constraints.

In order to properly select the design to complete such a task, the designer would need a model that determines the effects that certain design parameters have on the overall mission performance. These design parameters would be those affected by design decisions, such as length and mass of the links, as well as the speed of the

motors. These correlations can be conceptualized as a “mapping” between the design parameters and performance parameters (Figure 4.1).

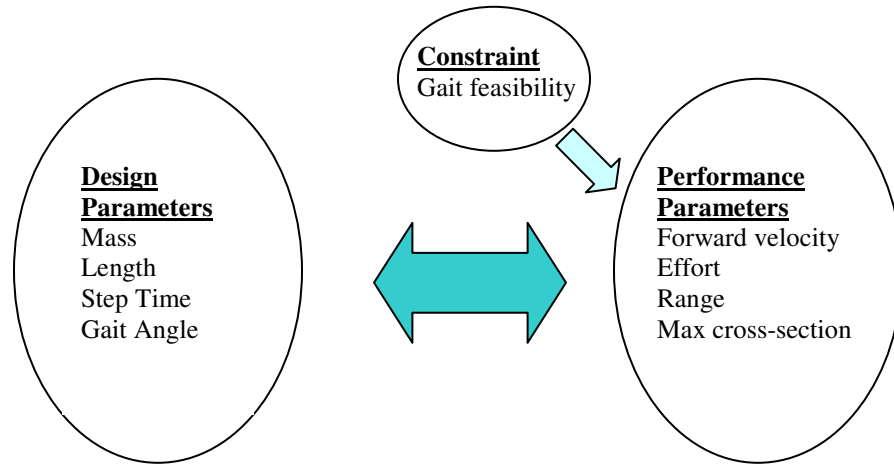


Figure 4.1. Mapping between design parameters and performance.

4.3. Discussion of Varied Parameters

Several design parameters were chosen for study. These parameters describe the gait and the physical design of the snake-inspired robot. The gait angle and gait step times are both parameters of the gait and not necessarily the physical design of the robot. However, the gait angle is constrained by the physical design because of the joint design, and the gait time is constrained by the properties of the selected servomotor. The other parameters selected for analysis were link length and link mass. These are physical parameters important to the overall design. Outside of the generated trajectories, the time step and gait angle are essentially the only means to which the gait can be changed. The mass and length were selected because of their

importance to the overall design. Several of the varied parameters are shown in Figure 4.2.

Gait Angle

The maximum angle that a joint can rotate is one of the most important issues in the area of snake-inspired robot design. Numerous works have discussed the implications that the maximum rotation angle has on both the maneuverability and manipulability of these robots. Additionally, at least one work has shown that there is an inverse correlation between rotation angle and the amount of torque required for climbing applications [47]. Finally, by looking at Equation 4.1, it can be seen that there is a positive correlation between the forward velocity of a snake-inspired robot and the gait angle.

At this point, the definition of the word “gait angle” is re-introduced. The gait angle is the standard angle of the configuration that the robot assumes at the end of each gait step. Although the gait angle and the maximum rotation angle are not necessarily the same thing, there is a correlation between the two. The maximum rotation angle can be constrained to be the same as the gait angle.

The maximum rotation angle of a snake-inspired robot is an important issue because of design and manufacturing constraints. It is often difficult to design a joint that allows a large range of rotation, while still allowing wires to transmit through it to the next module. Similarly, there are also geometric difficulties associated in the design of joints that allow a wide range of motion, particularly joints that need to be strong, and thus can be somewhat bulky.

Step Time

When trajectories were generated using the B-spline curve formulation, the duration of the trajectory was specified. The simplest means of speeding up the robot would be to run the servos faster, in other words, decrease the amount of time specified for each gait step. There is an inverse correlation between the forward velocity of the snake-inspired robot and the gait step time, since the forward velocity of the robot is simply the amount of advancement completed in each cycle, divided by the cycle time. The step time should affect the amount of effort required to complete each gait step because acceleration is increased by decreasing the gait step time. When acceleration increases, the dynamic torque required by the servomotors is increased.

Link Length

Link length is another parameter that will affect the performance of the robot. Increasing the length of each link will increase the forward velocity of the snake-inspired robot, according to the following equation:

$$v = \frac{2 \cdot l - 2 \cdot l \cdot \cos(\alpha)}{5 \cdot t} \quad (4.1)$$

Where “ l ” is the link length, “ α ” is the gait angle, and “ t ” is the step time. However, an increase in link length will increase the moment of inertia of the individual links, resulting in a greater amount of torque required to move the robot. Additionally,

increasing the link length may also increase the frontal cross-sectional height of the gait, which could be detrimental to movement through tight spaces.

Link Mass

A final parameter for study is the mass of each link. The mass is an important parameter for study because the mass may vary due to many different requirements and constraints placed on the robot. Material properties, payload, and battery size are among the factors that could result in an increase or a decrease in design weight.

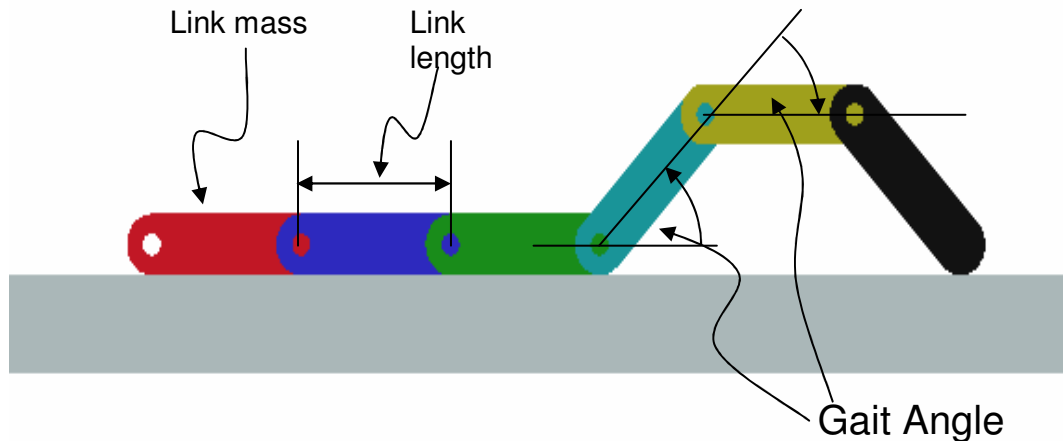


Figure 4.2. Varied parameters. The gait angles occur at the transition points, or “steps.”

4.4 Discussion of Results

The study was conducted by two means. Gaits were synthesized using the heuristic-based algorithm in the case of the variation of gait angle, step time, and link length. For the case of the variation of mass, the gait that had been found previously was used, and the new effort and torque values were calculated.

The obtained trajectories and thus gaits should change as the angle, step times, and link length are changed. This is because there are two classes of forces that contribute to the torque values, and thus the effort. These two classes of forces are both the dynamic (inertial) forces and the static forces. The dynamic forces are those that are needed to impart acceleration onto each link. The static forces are those that are needed to oppose the effects of the gravitational field that is acting on the links. The static forces are so-called, because they are in existence even if the links are not accelerating.

Changing the time step, gait angle, and link length parameters varies the effects of each of the two different forces, and switches which regime becomes more dominant. It is easy to see that decreasing the time of the gait step means that the links must accelerate faster, requiring more dynamic torque. In addition, because time is a factor in the effort metric, the longer that the static forces act on the mechanism, the more effort is required to remain in the configuration. Shortening the time of the gait step both increases the dynamic effects and decreases the static effects at the same time. The gait angle parameter similarly changes the effects of the different contributors to torque. Having a larger gait angle means that the links spend more time in a vertical position. This decreases the cross product between the gravitational force acting on the link, and the axis where the torque is applied. Figure 4.4 explains this effect. This would explain why a larger gait angle actually requires less effort at the specified time step, as shown in Figure 4.8. This is also the reason that the obtained trajectories for longer time steps are those where the joints remain at a high angle for longer, while gaits at shorter step times are those where the joints use

a smoother function to minimize large accelerations. This can be seen in Figure 4.3, where the trajectory for the longer step time moves to a larger angle comparatively faster than the trajectory with the shorter step time.

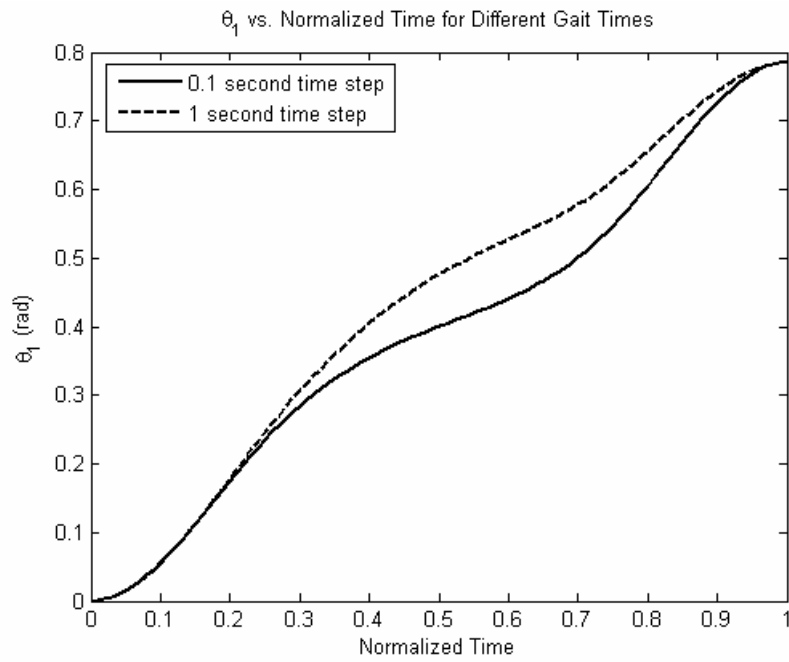


Figure 4.3. Trajectories generated for step 2 at different step times.

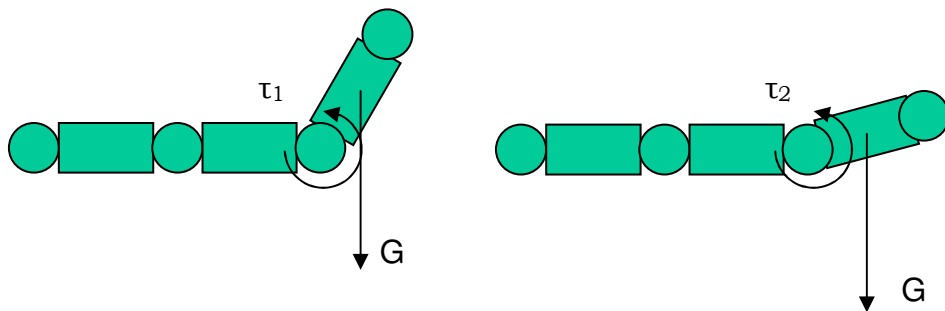


Figure 4.4. Torque 2 would be greater than Torque 1.

In order to understand the structure of the equations, consider the example of a simple manipulator. The effects of the length parameter can be better understood by consulting Equations 4.2 and 4.3. These are closed-form solutions for torque in a simple, two-link manipulator presented by Craig (Figure 4.5) [44]. Observing the equations, it can be seen that the terms related to the inertia of the links are proportional to the square of the length, while the gravitational terms (static terms) vary linearly with the length. This means that as the link length increases, the inertia terms begin to have more of an effect over the torque than the static terms.

The other portion of the parametric study was accomplished using the joint trajectories previously found, and conducting an analysis on how varying the mass of the links changes the effort value. The trajectories are not generated again, because the link masses do not affect the trajectories. There is no effect on the final trajectories because the mass should affect both the static and dynamic torque components equally, as can be seen in Equations 4.2 and 4.3.

Equations for sample manipulator (Figure 4.5):

$$\tau_1 = \underbrace{ml^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2 \right) + ml^2 \cos \theta_2 \left(2\ddot{\theta}_1 + \ddot{\theta}_2 \right) + 2ml^2 \ddot{\theta}_1 - ml^2 \sin \theta_2 \dot{\theta}_2^2 - 2ml^2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2}_{\text{Inertial Terms}} + \underbrace{mgl \cos(\theta_1 + \theta_2) + 2mgl \cos \theta_1}_{\text{Gravitational Terms}} \quad (4.2)$$

$$\tau_2 = \underbrace{ml^2 \cos \theta_2 \ddot{\theta}_1 + ml^2 \sin \theta_2 \dot{\theta}_1^2 + ml^2 (\ddot{\theta}_1 + \ddot{\theta}_2)}_{\text{Inertial Terms}} + \underbrace{mgl \cos(\theta_1 + \theta_2)}_{\text{Gravitational Term}} \quad (4.3)$$

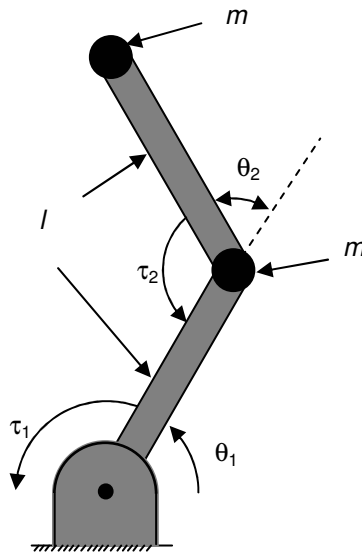


Figure 4.5. Sample two-link manipulator [44].

The effect of gait step time on the effort is shown in Figure 4.6. It can be seen that there is an inverse relationship between the time and the effort, because there is an inverse relationship between the accelerations and velocities of the joints and the time. This dominates the direct effect of time on the effort value. The relationship between the forward velocity of the snake and the effort required is shown in Figure 4.7. It can be seen that there exists a trade-off between these two parameters.

This relationship could be readily used to design a snake-inspired robot and an appropriate gait. For example, if a robot designer wanted to design a robot that moved at specified velocity for a maximum distance, they could use this relationship to select the gait step time accordingly. If the battery size is known, the designer could use the computed effort and the advancement term presented in Equation 3.9 to determine the range of the robot. Consider the case where a designer wants to select the appropriate servo speed for the robot in order to maintain a minimum velocity of 10 cm/s. The designer could consult the plot of velocity versus effort for the design to bound the feasible design space, as shown in Figure 4.7, and determine the minimum effort that would be required to enact this gait, $0.08 \text{ N}^2\text{m}^2\text{s}^2$. The designer could then consult the plot of effort vs. step time in order to determine the maximum allowable step time to achieve the desired performance (as shown in Figure 4.6), and use this step time to select the servo speed. This procedure could then be repeated with the other relationships found in this section to appropriately select battery sizes (via changing the mass), gait angles, and link lengths in an iterative manner in order to select the appropriate design for the task.

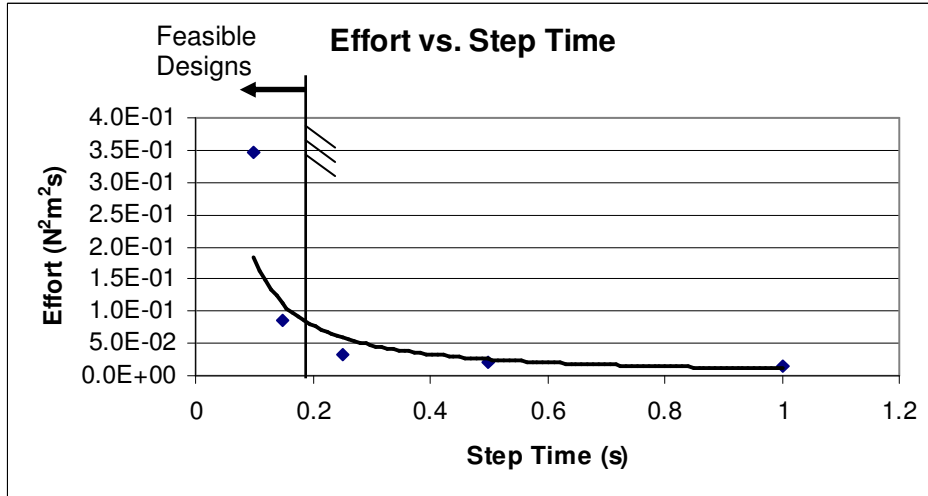


Figure 4.6. Effort versus step time. Feasible design space for sample exercise shown.

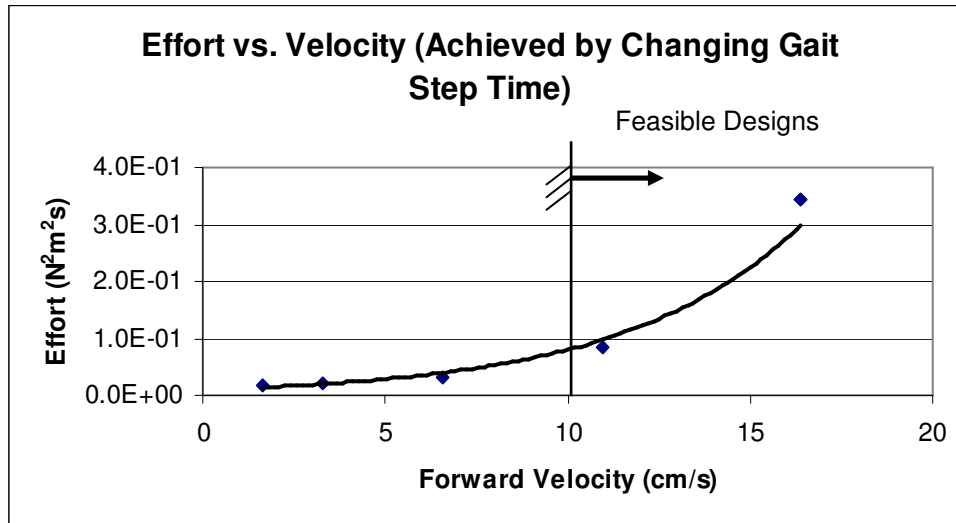


Figure 4.7. Effort versus velocity, changing step times. Feasible design space for sample exercise shown.

The effects of the gait angle on the effort and velocities are shown in Figure 4.8 - Figure 4.11. Gait angles of 15, 25, 30, 35, and 45 degrees were sampled, and trajectories were found for each solution. The standard parameters that were

presented in Section 3.6 were used for the first analysis (Figure 4.8-Figure 4.9), meaning a length of 0.14m, a mass of 0.015kg and a time step of 1 second. The effort values plotted are those given by the final solutions. From the graph, it can be seen that there is a general negative correlation between the gait angle and the effort when the servos are run at this slower speed. This can be explained due to the fact that the static torque is higher when the manipulator links are at shallower angles. A larger gait angle does, however, increase the amount of acceleration needed to move the link through its range of motion, so the general effect is a combination of these two factors. This is further illustrated in the results shown in Figure 4.10 and Figure 4.11. At this speed, there is a positive correlation between gait angle and effort, meaning that a larger gait angle requires more effort. This occurs because a larger angle means that the servos must move faster to traverse the entire angle. In this case, the dynamic effects dominate the static effects.

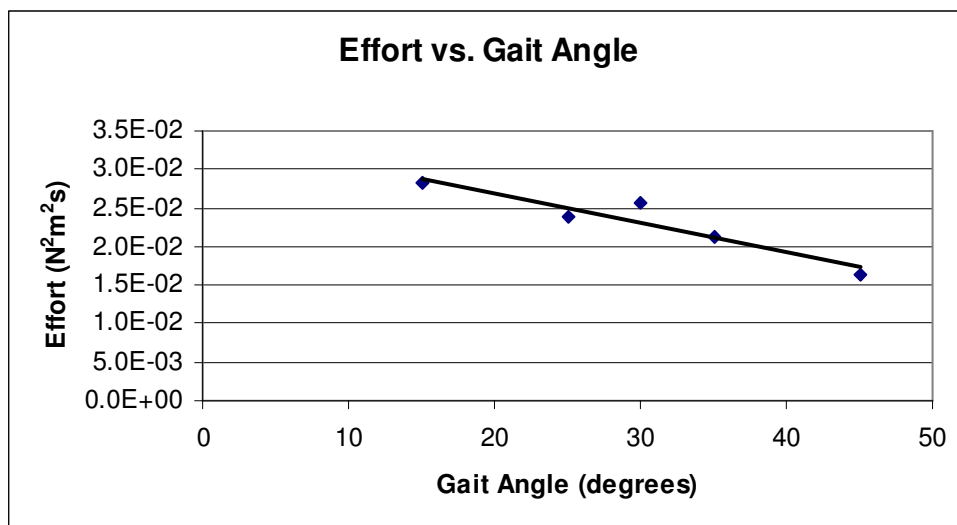


Figure 4.8. Gait angle versus effort, gait step time = 1s.

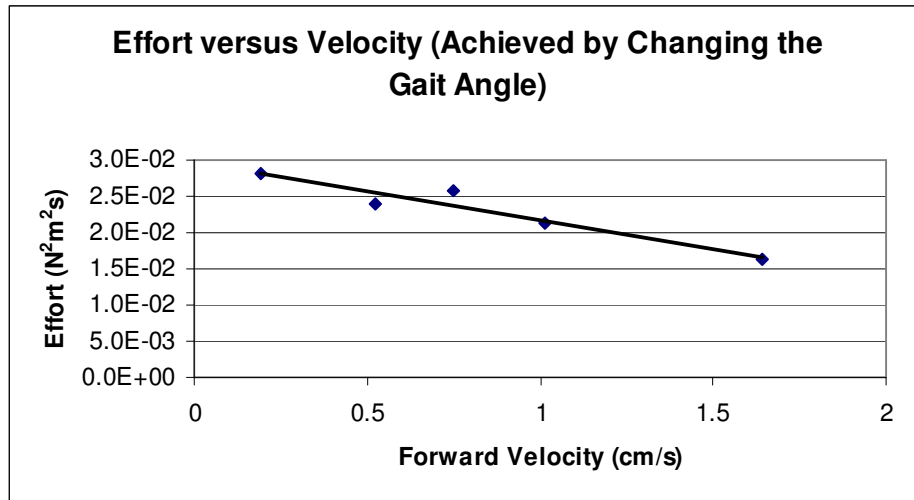


Figure 4.9. Effort versus velocity, achieved by changing the gait angle. Gait step time is 1s.

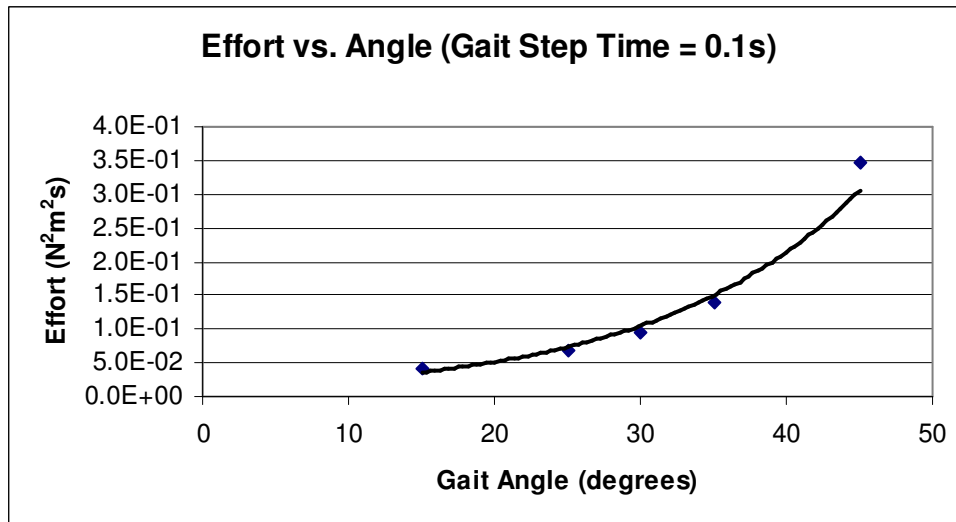


Figure 4.10. Effort vs. angle when gait step time is 0.1s.

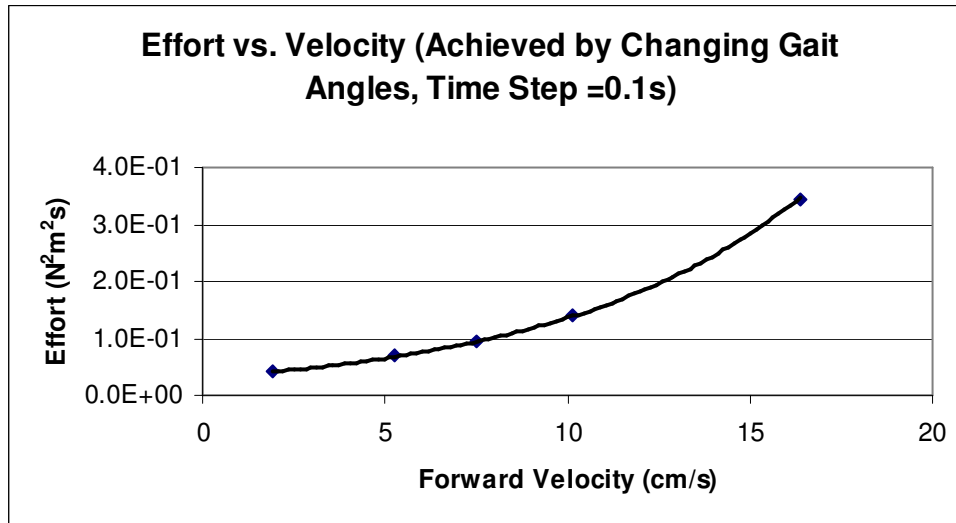


Figure 4.11. Effort vs. velocity (achieved by changing gait angle) when gait step time is 0.1s.

With link mass and length, the relationship between the selected parameters and the efforts and velocities is relatively straightforward. The effect of link length on the effort should be that of a fourth-order polynomial, as the effort is proportional to the square of the torque and the torque is a function of length and length squared. Because of this combined effect, different trajectories would be desirable for different lengths, as the effects of inertial and static terms vary accordingly. A plot of effort versus length can be seen in Figure 4.12. There is a direct relationship between length and velocity, so the correlation between velocity and effort achieved by changing the link length is of one order less than the relationship between length and effort. This can be seen in Figure 4.13. Effort simply varies with the square of link mass, as the torque varies linearly. A plot of effort versus link mass can be seen in Figure 4.14.

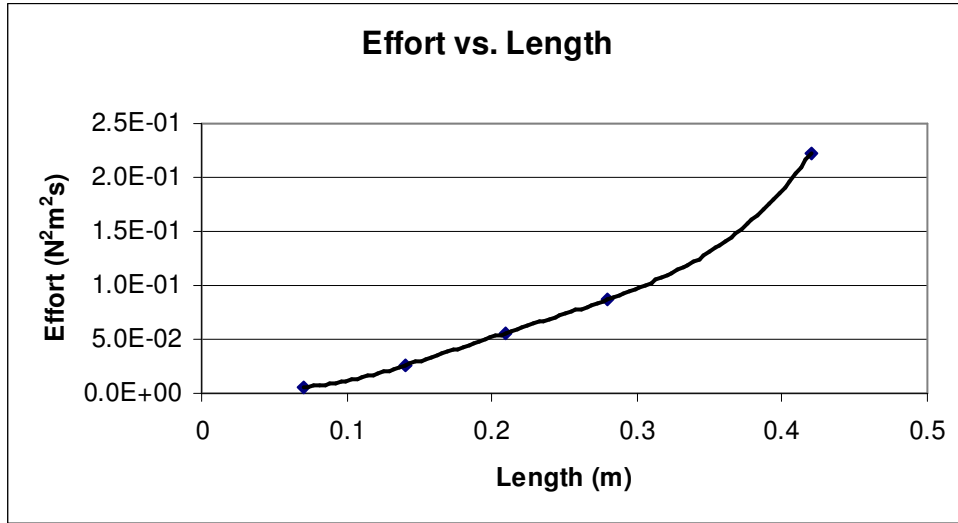


Figure 4.12. Effort versus length.

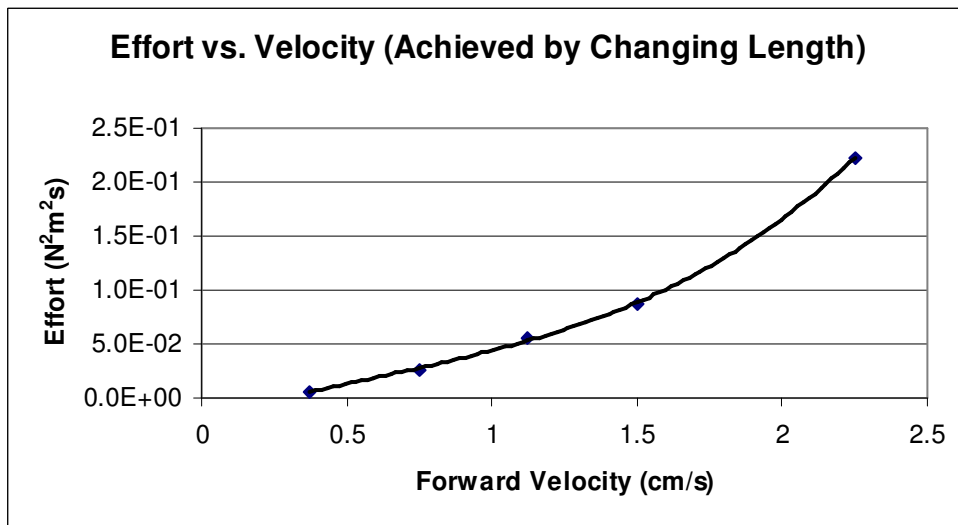


Figure 4.13. Effort versus velocity, varying length.

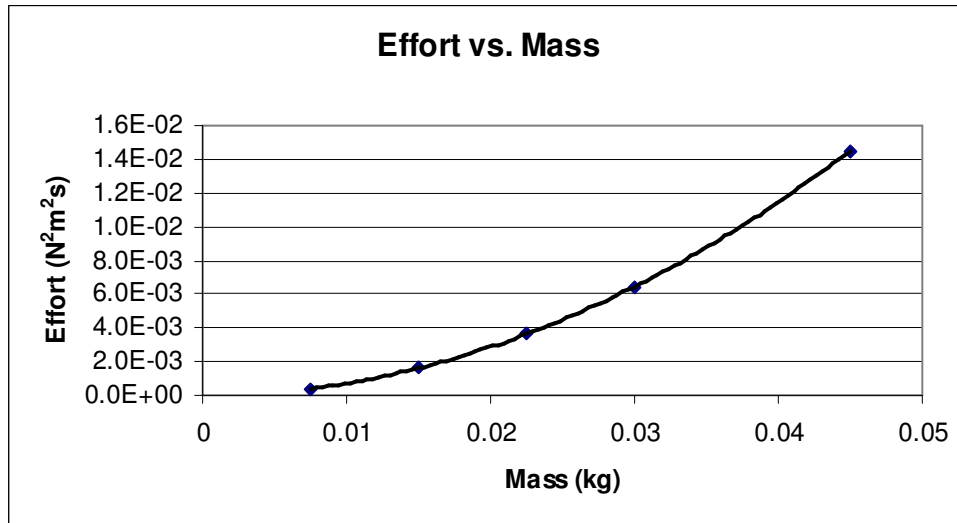


Figure 4.14. Effort versus mass.

4.5 Summary

In this chapter, results of a parametric study for the motion of a snake-inspired robot were presented. The trends and data presented in this section provide information that can be used as design guidelines for designers of snake-inspired robots, providing an understanding of trade-offs and trends that govern performance. The tradeoffs between effort, velocities, servo speeds (using the time parameter), and length are discussed. In addition, the fact that less effort is used with larger gait angles at slow velocities is shown, as well as the relationship between mass and effort. In general, this information can be used to allow decision-makers to design snake-inspired robots in view of certain mission requirements and manufacturing constraints.

It should be noted that the effort values presented in this section are those that were found using the heuristic-based search. Thus, there is some variation in the data

that results from the inherent limitations of the algorithm. The general trends are shown in the plots, but each specific data point does not reflect an exact solution.

Chapter 5 - Snake-Inspired Robot Realization

5.1 Introduction

In this section, the design requirements and desirable attributes of a snake-inspired robot are revisited. The design, fabrication, and demonstration of the snake-inspired robot are fully described.

5.2 Design Goals and Objectives

In order to discuss the realization of a snake-inspired robot design, the task description and the design characteristics of such a robot must first be revisited. The design task is to build a small search, rescue, and reconnaissance robot that can be used by firefighters and other USAR personnel. The design should be able to locomote in a rectilinear snake-like fashion, as has been described in the earlier chapters of this work, and it should include a second degree-of-freedom to allow steering on the 2-D ground surface that could be implemented in subsequent gait designs. The design should incorporate the desirable characteristics of a snake-inspired robot that were discussed in the introduction such as redundancy, modularity, and a small cross section. The robot should be developed considering the following desirable attributes, as were mentioned in Chapter 1:

- Low Cost
 - Components, manufacturing, and assembly costs
- Rapidly Deployable
 - Rapid, on-site assembly
- Robust
 - Ability to operate with failed segments
- Customizable
 - Assembled to desired length for application

As was discussed in Chapter 2, the basic design of a snake-inspired robot consists of many links connected by many revolute joints to allow twisting in the appropriate direction to achieve motion. In the case where the robot moves with a rectilinear gait generated by vertical waves, the joints of the robot must allow for this vertical bending. In addition to the vertically propagated waves that propel the robot, the robot must also be able to steer itself. This means that each link must have 2-degrees-of-freedom, including rotation about the z-axis (out of page) as well as the vertical y-axis.

Each link must be actuated, and actuation must be accurately controlled so that a desired trajectory can be specified for each joint. The control should be easily accomplished. The actuation for each joint should be accomplished with a conventional form of actuation, and the actuators should be standardized for the entire robot for simplification of the design.

The entire robot should be based on one fundamental unit that is repeated, much like a real snake with a repeated vertebral and skeletal structure. For many reasons, a completely modular architecture based on many repeated modules is preferred. The modules should be manufactured and assembled separately, and then linked together to form the entire robot. A modular architecture can provide advantages in areas such as manufacturability and cost because individual modules could be mass produced. A modular design would also make the design more customizable and rapidly deployable because modules could be rapidly assembled to the desired specifications on-site. Finally, a modular robot would be more robust because it could operate with failed segments. An example modular snake-inspired robot is shown in Figure 5.1.

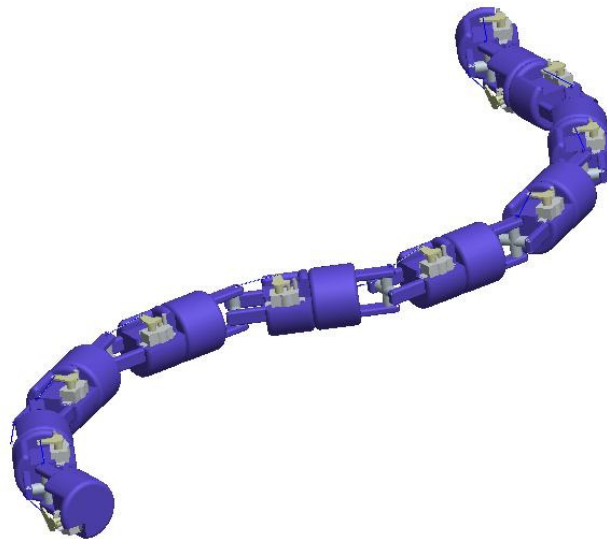


Figure 5.1. Example of a fully-modular snake-inspired robot.

Summary of the Design Problem

In summary, the task at hand is to design and build a snake-inspired robot for use in a laboratory setting. The robot should be able to demonstrate the desired actuation gait to provide some validation of the previous sections. Specifically, it will be used to demonstrate that the class of gait studied in this work will result in forward motion of the snake-inspired robot. In addition, the robot should demonstrate some of the principles of modularity, and that it is possible to design and build a fully-modular snake-inspired robot, based on a completely repeated design architecture. The robot should demonstrate the value of constructing a robot that is constructed using multi-stage multi-material molding as well. The design task is not to construct a robot that is fully-functional in search and rescue requirements. Subsequent design iterations will need to address the issues of the use environment (robustness, durability, etc.), intelligent control, and payload capability (i.e. sensor equipment) in order for such a robot to be functional as a search and rescue robot.

5.3 Structural Design

The first important sub-system of the robot is the structure. The structure would be defined as the portion of the robot that contains, supports, and to an extent protects the power, actuation, and control components of the robot. The structure of the robot should successfully contain all of the components in the proper place, and protect them from impact with the ground. The structure will also provide the contact surface between the robot and the ground, so it should be designed so that the robot will remain stable. Finally, the structure should be articulated to allow for the

bending of the individual modules to achieve the proper motion of the robot. The structure should also be designed with the proper design characteristics (durability, low cost) considered.

5.3.1 Molded Structure

It was decided that the robot would be made using a molded plastic structure. This offers several advantages over alternatives such as machined structural components (either plastic or metal), forged or cast metal components, or new emerging technologies such as additive manufacturing processes. Molding technology is also an ongoing research topic at the University of Maryland's Manufacturing Automation Laboratory. Thus, equipment and expertise to rapidly and efficiently manufacture a robot using molding technology was readily available, and this product provided a platform to test and demonstrate molding technologies such as multi-material molding and the molding of articulated joints.

Injection-molding is a manufacturing process whereby molten plastic is injected at pressure into a mold cavity that is a negative of the part. The plastic occupies all of the empty space and then solidifies as the manufactured part. The part is then ejected from the mold. Typically, in injection molding the material used is a thermoplast polymer, and the polymer is heated to allow it to flow at pressure. Thermoset materials can also be used, as well as multi-component reacting mixes. The latter case is known as reaction injection molding.

There are numerous advantages of injection molding over other manufacturing processes that makes molding desirable for a product such as a snake-

inspired robot. Molding is a high throughput process, as opposed to other processes such as machining and sheet metal bending. This means that injection molded products can be produced in short time and with minimal labor costs, once the machining set up is complete. Injection molding can be largely (even fully) automated, significantly reducing labor costs. Another important advantage of a process such as injection molding is that it is extremely versatile, allowing a wide variety of geometries. This gives the designer of the robot a wide freedom of forms for components, allowing a better fit for the desired function.

Injection molding is a process that is specific to polymers, which also offers several advantages. Polymers are typically very low cost and low weight as compared to other engineering materials, such as engineering metals and ceramics. Molding also has the advantage of not requiring hazardous solvents and high temperatures that are often needed to manufacture using metals and ceramics. This has implications in both energy costs and environmental issues. Finally, molding is cost effective because there is minimal waste. Because it is a forming process, as opposed to a material removing process, minimal material is wasted. If the molding is performed with thermoplastics, then any excess material can be grinded down and re-used, resulting in almost zero waste.

Many of the properties of the injection molding process are also the reason that experimental roboticists and developers of search and rescue robots have avoided using molding to manufacture robots in the past. Injection molding is a process that can achieve a very high throughput and a very low price per unit. However, molding requires a significant overhead investment in order to manufacture the molds and

purchase the molding equipment. For small-batch items such as search and rescue robots, it is often more economical for manufacturers to use processes that are more suited for those kinds of quantities, processes such as machining and sheet metal bending. Another drawback to molding is that polymers are often weaker than metals, which are typically used for such applications. This drawback can be alleviated by using improved polymer and polymer composite materials, in addition to functionally-graded and multi-material designs, which will be discussed later.

By developing a robot that is fully modular and disposable, a high throughput process such as molding becomes more desirable. Costs can be reduced by mass-producing the individual modules that are then quickly assembled to form a full snake. Mass production also results from producing many inexpensive robots that can be considered to be expendable, as opposed to a few expensive robots. Gains can be significantly increased if multi-material molding is used, further reducing assembly operations.

5.3.2 Multi-Material Molding

Recently, there has been interest in a technology known as multi-material molding [48-52]. Multi-material molding is a process in which multiple shots of plastic material are combined to produce an object that is heterogeneous. This heterogeneous object can consist of two or more injection-molded plastic materials. Multi-material molding is typically accomplished in a sequence of stages or shots where molten material is injected into different mold sections to produce the heterogeneous component.

Two common means of multi-material molding are overmolding and multi-shot molding. Overmolding is a process where a part is fabricated during one molding shot, and then is removed and assembled into a different mold. The second shot of material is shot into the second mold to form a product that is made out of two materials (Figure 5.2). Multi-shot molding relies on a toggling of mold pieces, and the piece made from the original shot is not removed (Figure 5.3). Several different means of multi-shot molding exist, including rotary platen molding and core-toggle molding. For a general overview of the techniques and technology used in Multi-material molding, consult [52].

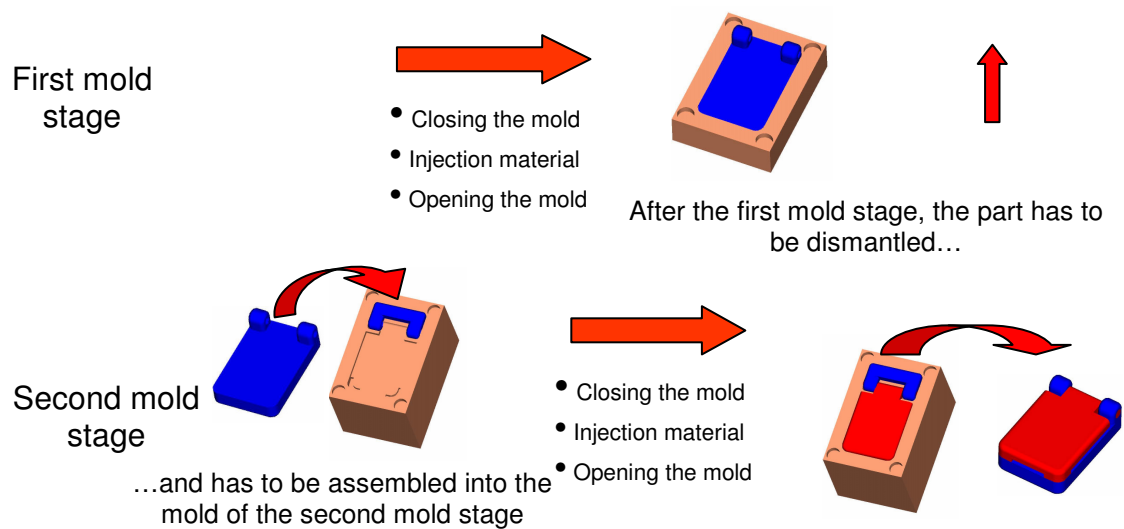


Figure 5.2. Overmolding [52].

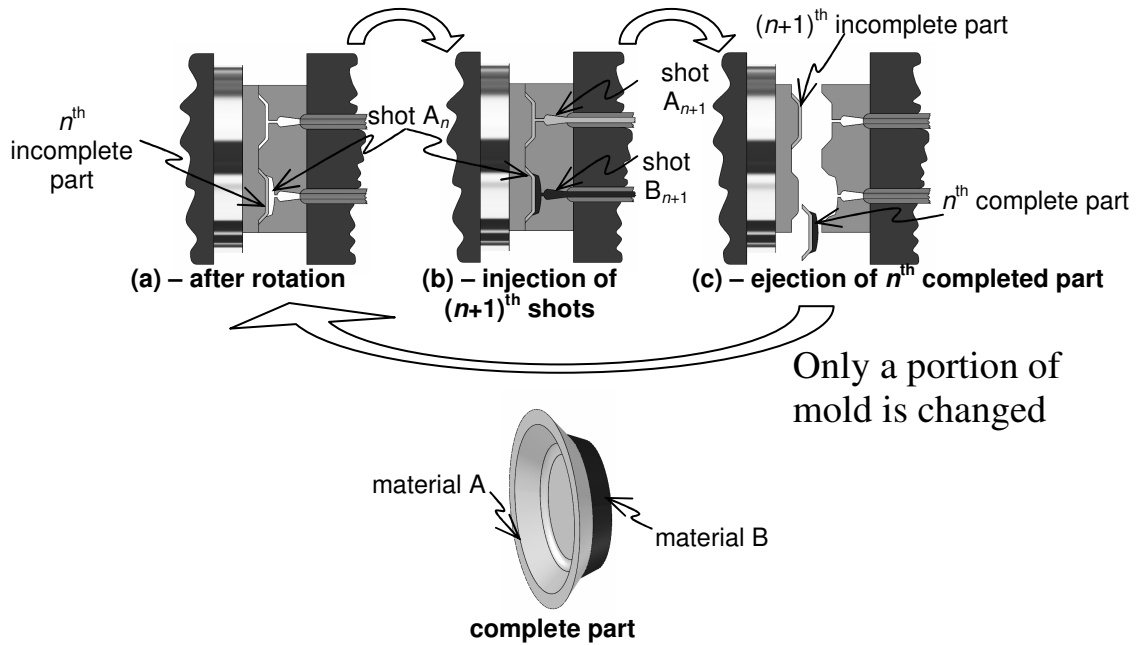


Figure 5.3. Multi-Shot molding [52].

Multi-material molding offers several advantages over traditional single material molding that are especially of interest for snake-inspired robots. For an understanding of how value can be added by multi-material molding, one must compare the traditional manufacturing paradigm with the multi-material molding paradigm.

Traditional manufacturing involves the assembly of many individual parts that are fabricated separately. Parts are fabricated using a process such as machining or injection molding, and then assembled either manually or by a form of automation, such as robots. Many difficulties and disadvantages arise from this form of manufacturing. Assembling many parts consumes both time and labor, which can be costly. Also, as components and assemblies continue to get smaller, manual assembly can become more difficult, consuming even more time, and requiring

specialized skills and tools. Defects could also be increased. Assembly of components also increases part counts because of all of the components in the assembly that exist purely to facilitate assembly. This includes components such as fasteners.

The interest in biologically-inspired products such as robots is increasingly pushing the demand for more complex geometries and interfaces. The more geometrically complex components are, the more difficult they can become to assemble.

The multi-material molding paradigm offers an attractive alternative to the traditional manufacturing paradigm for heterogeneous components and assemblies that can be manufactured using polymer molding. In multi-material molding, assembly steps are removed by performing several molding stages on the object while changing the cavity. This means that assembly and fabrication can be completed concurrently. Concurrent fabrication and assembly eliminates manual assembly operations. Using multi-material molding for concurrent assembly and manufacturing means that assembly can be accomplished by geometric interlocking instead of fasteners, reducing weight and part count. Multi-material molding also allows for easier manufacturing of small assemblies because the product size is not limited by the constraints of manual assembly. The only limiting factors in the size of a multi-material molded component are the physical limits of the molding process. This opens the door to meso-scale assembled components.

Multi-material molding can be used to create both free and chemically bonded interfaces, allowing for the manufacturing of articulated structures using both rigid

joints and compliant joints. Spherical, revolute, and prismatic joints can be developed using multi-material molding. Shown in Figure 5.4 is a 2-DOF gimbal mechanism that was produced using multi-material molding. The gimbal mechanism was removed from the mold fully assembled.



Figure 5.4. Multi-material molded gimbal mechanism [51].

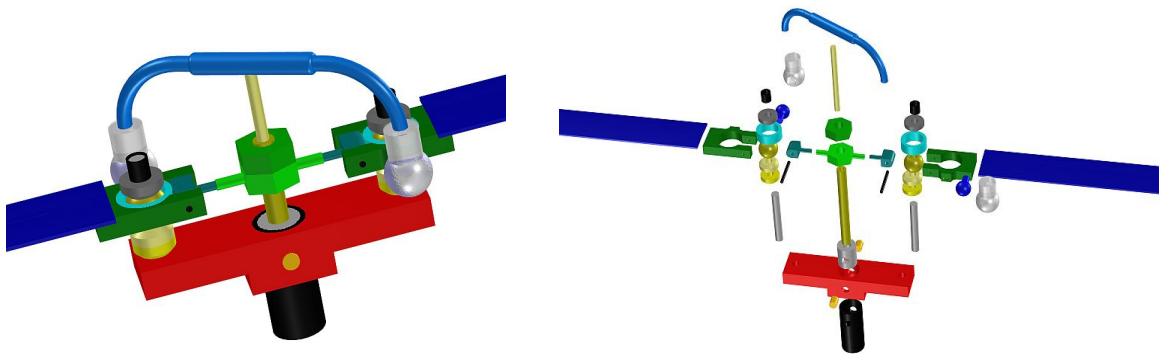


Figure 5.5. Original rotor assembly [51].

To illustrate the advantages that can be realized with multi-material molding, consider a case study that was performed on an unmanned aerial vehicle rotor

assembly [51]. The original assembly consisted of 33 different parts that required manual assembly (Figure 5.5), for a total of 32 assembly operations. A redesigned version of the rotor assembly was developed that was primarily manufactured and assembled concurrently using multi-material molding. The redesign of the assembly reduced the part count to 8 parts (Figure 5.6).

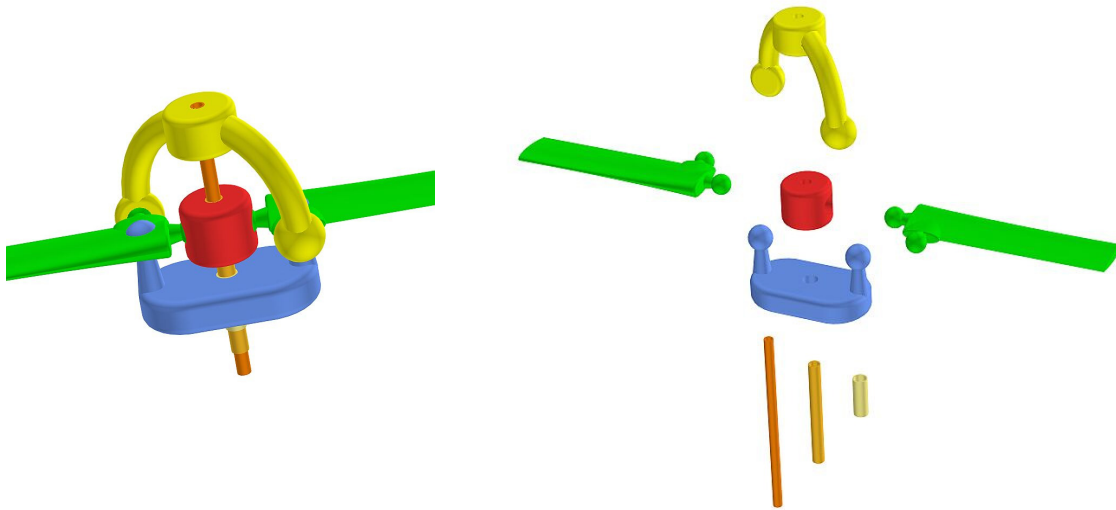


Figure 5.6. Redesigned rotor structure: Only 8 parts [51].

Multi-material molding is an attractive option for snake-inspired robots because they typically consist of many small parts and moving joints. Fabricating and assembling many small parts into a product with many degrees of freedom is costly and time consuming. Additionally, in mold assembly can also be expanded to include the embedding of electronic, actuation, and power components inside the material structure, further reducing the part count and assembly operations [53]. This is an aim of future work, but embedded components were not used in this work,

because the ability to easily replace failed circuits, servos, or batteries in the working prototype was desired.

5.4 Component Selection

The components for this robot module were selected based on several criteria. Cost has been shown to be an important issue in the development of a USAR robot, and each component was selected with cost constraints in mind. In addition, components that were simple to obtain, program, and implement were desired. Other constraints such as weight, size, and runtime were considered. The weight was not used as a direct constraint, because similar components typically do not come in a wide range of weights. The size of the modules was to be minimized, in order to have a robot that could travel into tight and hard-to-reach areas. However, a major constraint on the size was that it could not be too small, prohibiting assembly by hand in the lab. There were also other geometric limits placed on the structure that will be discussed later. Runtime was another issue that was important to component selection. An urban search and rescue robot must have enough runtime to complete a mission. In this task, the target runtime was one hour, meaning that the robot should be able to run continuously for one hour straight before the energy supply would need to be replaced or recharged.

5.4.1 Actuators

For the actuation of this snake-inspired robot, servomotors were selected. Servomotors were selected because of their size, cost, and simplicity. Servomotors

are a conventional and proven technology, and many control platforms currently exist for their implementation. Servomotors are readily available and come in many different shapes and sizes.

Servomotors were chosen as opposed to newer, experimental technologies, and other conventional motors. Servomotors can be contrasted with new, novel actuation technologies such as artificial muscles and shape memory alloys; technologies that may be ideal for snake-inspired robot actuation in the future, but currently they would require too much study in order to incorporate them into this design.

The servomotor chosen for this application was the HS-55 made by HiTec. This motor was selected because of both its size and its cost. In order to make the robot as small as possible, it must be designed around a small motor. The HS-55 is considered to be a “sub-micro” servomotor, which is the smallest class of motors. The dimensions of the HS-55 are 22.8 x 11.6 x 24mm, and it is the most affordable motor in its size range. Motors such as these are typically used to control the flight surfaces of small RC aircraft. The motor and its dimensions are shown in Figure 5.7.

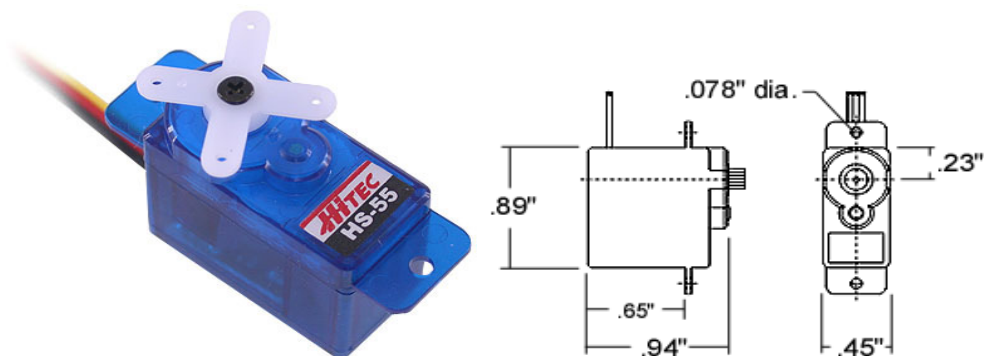


Figure 5.7. Servomotor.

Source: www.servocity.com

HS-55 Servomotor Specifications:

Control System: +Pulse Width Control 1500usec Neutral
Required Pulse: 3-5 Volt Peak to Peak Square Wave
Operating Voltage: 4.8-6.0 Volts
Operating Temperature Range: -20 to +60 Degree C
Operating Speed (4.8V): 0.17sec/60 degrees at no load
Operating Speed (6.0V): 0.14sec/60 degrees at no load
Stall Torque (4.8V): 15.27 oz/in. (1.1kg.cm)
Stall Torque (6.0V): 18.05 oz/in. (1.3kg.cm)
Operating Angle: 40 Deg. one side pulse traveling 400usec
360 Modifiable: Yes
Direction: Clockwise/Pulse Traveling 1500 to 1900usec
Current Drain (4.8V): 5.4mA/idle and 150mA no load operating
Current Drain (6.0V): 5.5mA/idle and 180mA no load operating
Dead Band Width: 8usec
Motor Type: 3 Pole Ferrite
Potentiometer Drive: Direct Drive
Bearing Type: None, outer case serves as bearing
Gear Type: All Nylon
Connector Wire Length: 6.29" (160mm)
Dimensions: 0.89" x 0.45"x 0.94" (22.8 x 11.6 x 24mm)
Weight: 0.28oz (8g)

5.4.2 Power

Providing power for a robot is a very significant issue that is being addressed by many different communities from the space community to the UAV community. It is a significant issue because in order to perform meaningful missions, robots must be able to carry a power supply on-board that will allow them to run for an acceptable amount of runtime. The power systems also need to be sized appropriately to fit the mission requirement, which is a significant issue considering that efficiency often decreases with size of power systems.

Since the actuation of this device is to be performed by electronic servomotors, it makes sense to use batteries to provide the power. Batteries are well-suited for smaller systems such as small robots. They are also a proven technology and are simple to implement. While experimental technologies such as fuel cells and

microturbines may prove useful for applications such as small robots in the future, the current state of technology is not ready for implementation in product design.

In selecting a battery, the objective is to maximize power density, while keeping constraints such as size and cost into consideration. When choosing a small, high-performance battery, the standard choices are NiCad, NiMh, and LiPo (Lithium Polymer). As can be seen in Figure 5.8, Lithium Polymer batteries are a new technology that give significantly better performance (energy stored per weight) than the other small batteries that are commercially available. They are also used in the RC hobby industry, and are readily available.

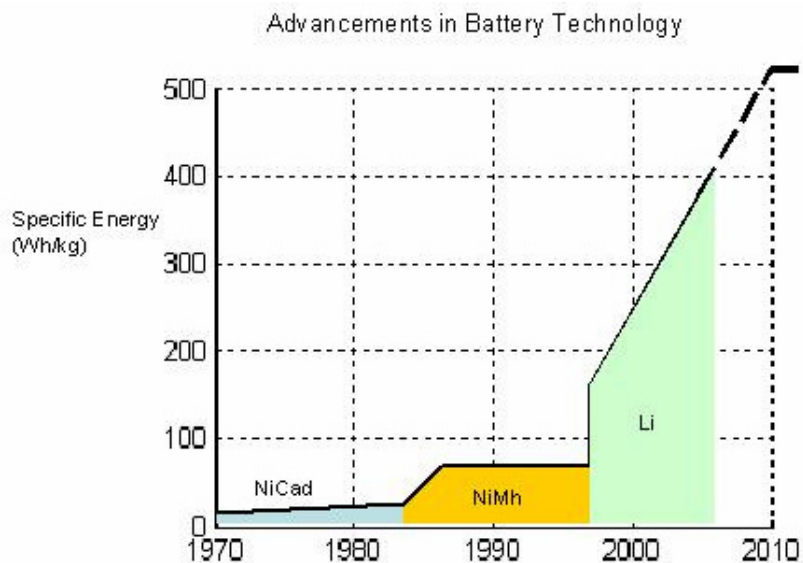


Figure 5.8. Battery technology trends (adopted from [54]).

Lithium Polymer batteries are commonly available in the RC and MAV industries, and thus are affordable and readily available. The industry standard for a LiPo cell is 3.7 volts, and both 1 and 2 cell batteries are commonly available.

Because the selected servomotor requires between 4.8 and 6.0 volts to operate, a 2 cell LiPo battery was selected. The battery that was selected is a 145 mAh, 2 cell battery made by FMA direct. This battery was selected because it was the most compact battery available. A rough calculation also yields that this battery should be able to drive the selected servomotor at no-load for roughly 48 minutes. The selected battery is shown in Figure 5.9, with its specifications shown.

Lithium polymer batteries also have the added advantage of being rechargeable. This may or may not come into use in the search and rescue environment, because if the robots are used in a disposable manner, then they will not need to be recharged. However, this is an added advantage for researchers in that the batteries do not need to be removed and replaced, they can simply be recharged.



Figure 5.9. Lithium polymer battery.

Source: (www.FMAdirect.com)

Battery Specifications:

145 mAh, 2 cell series (7.4V), heatshrunken with 125mm connector

Size: 30mmH x 19mmW x 9mmT

Weight: 9 grams

Ratings: 15C , 11C/140°F

Outputs: 7.4V Nominal, 145 mAh

Applications: Indoor/park electric aircraft

5.4.3 Control System

In order to successfully implement the gait, the robot must successfully complete an actuation pattern that is repeated. A control system architecture must be developed to automate the process of controlling the servomotors. This means that electrical signals must be generated, and supplied to each servomotor, dictating to the motor which its position should be at a given time. This should be preprogrammed into the system so that the positions do not need to be manually given at the relevant time.

It is important that this control system remain in the modular architecture of the overall system. This means that the control system should be located locally on each module, and each module's system should be identical. However, it is important to note that in order to achieve a locomotion gait, all of the actuation sequences of each individual gait must be synchronized with each other, in order to produce a global action that is moving the snake forward. The design of the system should address these challenges.

5.4.3.1 Micro Controller

Servomotors rely on pulse-width modulation (PWM) signals to obtain their position values. The PWM signal dictates the position in which the servos arm should be, according to the width of the signal pulse. The servo arm then moves to that position accordingly, using an embedded feedback control system. This means that a device that can be programmed to output varying PWM signals at different times to drive the servomotors is required.

A simple and cost effective means to achieve this is by the use of a PIC microchip. A PIC microchip can be simply programmed to output the desired sequence of PWM signals. The program is stored on the chip, and the chip begins to output the desired signals in the desired sequence when it is supplied to power. Microchips are available in a variety of memory sizes, and output options.

The chip selected for this application was the PIC12F629 by Microchip. It was selected due to its low cost, flash memory, and two internal timers. The memory in the chip was also deemed to be sufficient for this application.

The chip was programmed using assembly language by Lawrence S. Gyger. A series of via points were programmed into the chip in order to specify the trajectory, and the pattern was specified to repeat in order to have the servomotors move to the desired position. The internal timers were used to output the PWM signals that designated the via points, by using a dwell on the output voltage until the proper width of the PWM signal was attained, and then voltage is dropped, and the next portion of the pulse begins.

It is important to note that the microchip and servomotor both run on 5 volts, and the battery output is rated at 7.4 volts. Additionally, in the lab, battery outputs as high as 8 volts have been measured. In order to run the circuit off of the battery, a voltage regulator circuit needs to be used to reduce the output of the battery to 5 volts. This is a simple circuit involving a semiconductor called a voltage regulator, and several capacitors.

5.4.3.2 Overall System Architecture

A complete control system architecture system was designed as follows. The PIC control circuit is contained on each module. In addition, each module was designed with a local power control, meaning an on/off switch. Furthermore, it is desirable to have an operating mode where the battery could be charged while still inside of the module. Finally, there should be one master control that starts the entire sequence so that each of the modules actuation sequence is synchronized. The overall system architecture is shown in Figure 5.10.

The system was designed so that there is one bus wire that runs the length of the snake and, when switched on, indicates to each control circuit that it should begin its sequence. This was accomplished by adding a bus that runs the entire length of the robot. The bus acts as an input into each individual PIC. In order to begin the actuation of the snake-inspired robot, the bus is switched on, inputting +5v to each PIC microchip. Each circuit on each module is then switched on. Each microchip is programmed such that it is instructed only to begin its actuation sequence when it is not reading an input voltage in the pin that is connected to the bus. When each

module is switched on, the voltage on the bus is switched off, and all circuits being their sequence at the same time.

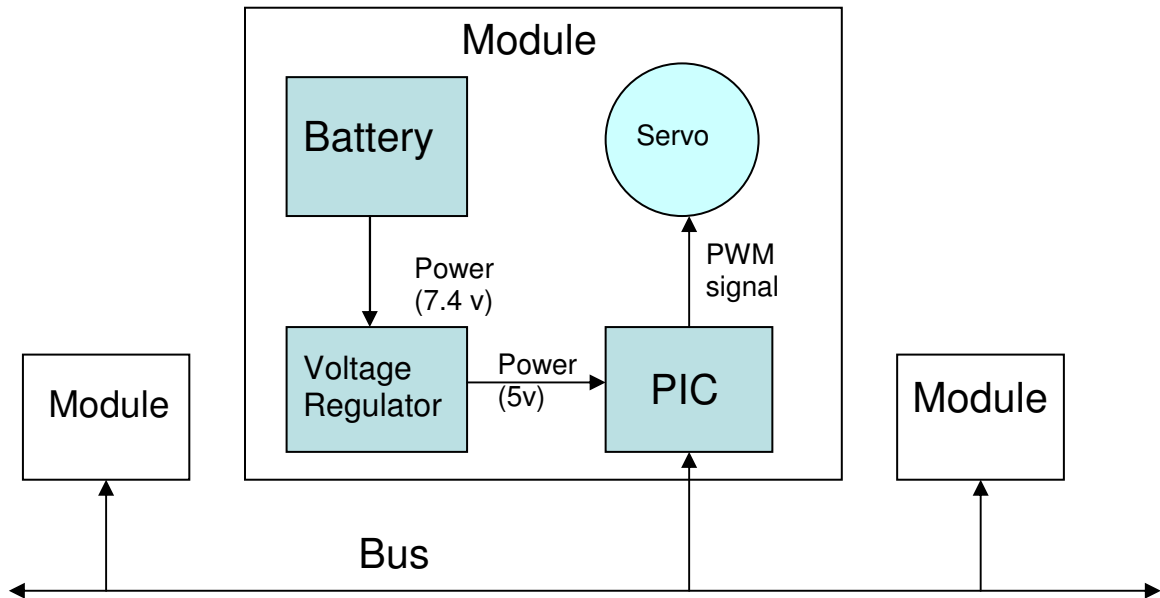


Figure 5.10. Modular circuit architecture.

5.4.3.3 Detailed Circuit Design

The modular control circuit was designed by Lawrence S. Gyger [53]. The circuit layout can be seen in Figure 5.11. The PIC chip is programmed to input the joint trajectory points to the servomotors by using PWM signals, and the pattern is repeated for the duration that the circuit is powered. The chip is programmed externally, and then assembled onto the board. The ground bus input to the circuit connects to the terminals marked PAD8 and PAD6, while the PWM signal is transmitted through the terminal marked PAD4. The positive and ground signals to the servomotor are connected through PAD5 and PAD9. The voltage regulator chip

and accompanying capacitors form the circuit that drops the voltage from 7.4 volts to 5 volts.

The battery charging circuit is built in to the integrated circuit. The circuit is powered through the terminals PAD1 and PAD3. The terminals PAD2 and PAD7 can be used to connect to the positive and negative sides of the battery charger, respectively. The 3-way toggle switch is moved to the appropriate position to allow this to occur. This is done so that the charging wires are not energized when the circuit is either running or off. The other positions on the switch turn the circuit on and off.

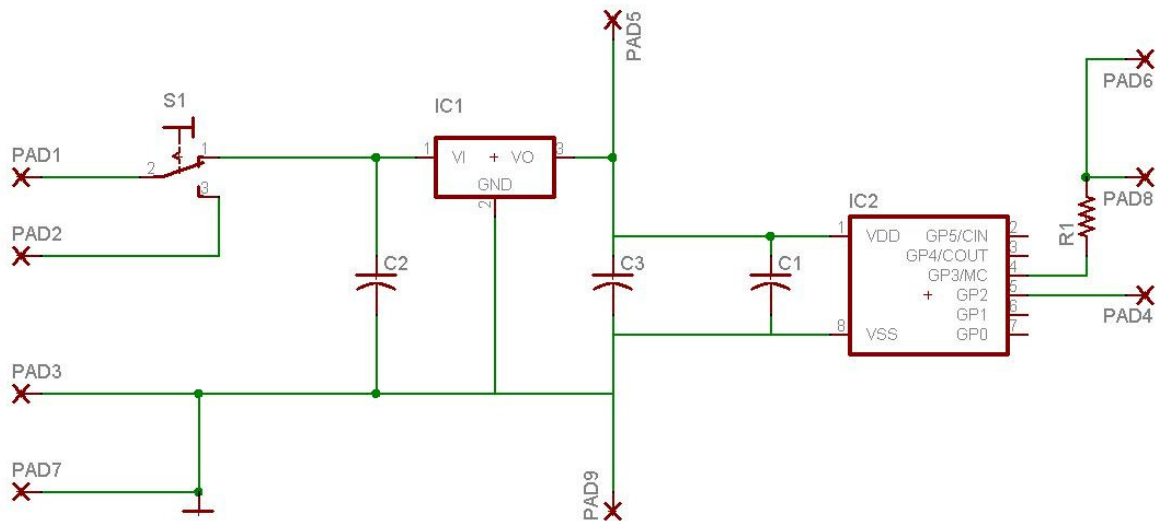


Figure 5.11. Circuit layout. Components IC1 and IC2 are the voltage regulator and the PIC, respectively. C1, C2, and C3 are .1, .33, and .1 μF capacitors, respectively.

A physical layout of the circuit is shown in Figure 5.12. The components were arranged to minimize space, and allow easy access of the switch and the PIC

chip. The circuit board was custom made by Advanced Circuits, Inc. It was fabricated out of FR-4, a fiberglass composite and the circuits are copper, coated with tin for corrosion resistance.

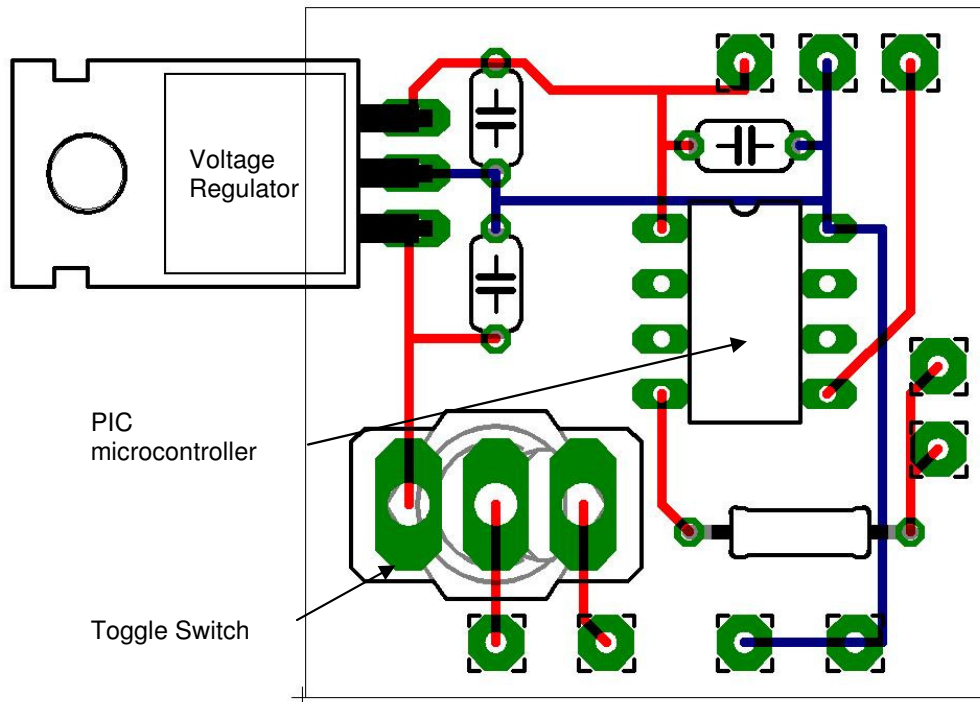


Figure 5.12. Circuit board layout, key components are labeled.

5.5 Overall Module Mechanical Design and Manufacturing

The module was designed as two portions connected by a universal joint. The rationale for this design was that it allows the snake-inspired robot to have a degree-of-freedom in the vertical direction for rectilinear locomotion, and a degree of freedom in the horizontal direction that allows for steering. The current control

architecture and gait does not yet steer the robot, but the mechanical design was developed so that this feature could easily be included. A universal joint was the simplest and most compact means of achieving the desired 2 degrees-of-freedom, and similar designs based on universal joints have been already demonstrated by Wolf et al. [18].

The general structure was designed to contain the selected components, allow for actuation of the universal joint, and provide the contact surface between the robot and the ground. The design of the modules was subject to the constraints of the manufacturing process, which included constraints due to the molding process and the manufacturing of the molds. The structure needed to be rigid enough to not bend during use in order to properly execute the gait. The structure also should not break during “normal” laboratory operation; however, rigid durability targets were not designed for or tested.

The final module design is shown in Figure 5.13 and Figure 5.14. It consists of an outer casing that supports the circuit, the two servomotors, and the battery. The switch protrudes from the module for easy access and the conservation of space, and the components are distributed on each side of the universal joint to conserve volume and distribute weight. Male and female connection features are included on each side of the module to provide alignment references when assembling the modules. In subsequent iterations these will be replaced by functional snap-fit devices. The actuation is accomplished by attaching the servo horn to the adjacent part of the module on the other side of the universal joint using a tie-rod system with small ball

joints designed for hobby aircraft. The joint has a range of approximately 45 degrees in each direction.

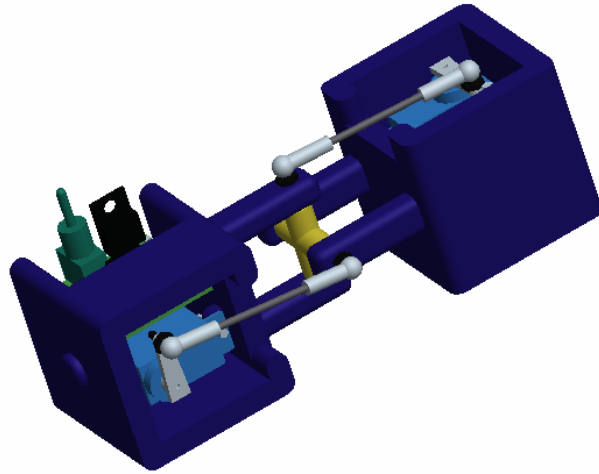


Figure 5.13. Completed design module.

The module structure was constructed of industrial polyurethane from Innovative Polymers, Inc. The specific grade, IE-72 DC was chosen because of its high strength and hardness. This material system was selected because it is a two-part resin and hardener mix that is ideal for small prototyping projects. It can be simply mixed in a standard laboratory, and injected with a syringe, as opposed to an expensive machine. The process can be more accurately described as polymer casting, as the material is injected into the top of the mold and the mold is filled by gravity. Designing molds in this fashion alleviates geometric constraints that exist with injection molds due to clamping and runner systems. In addition, the process

requires no additional heat. This process and material system has been demonstrated to be able to produce rigid joints in the past [51].

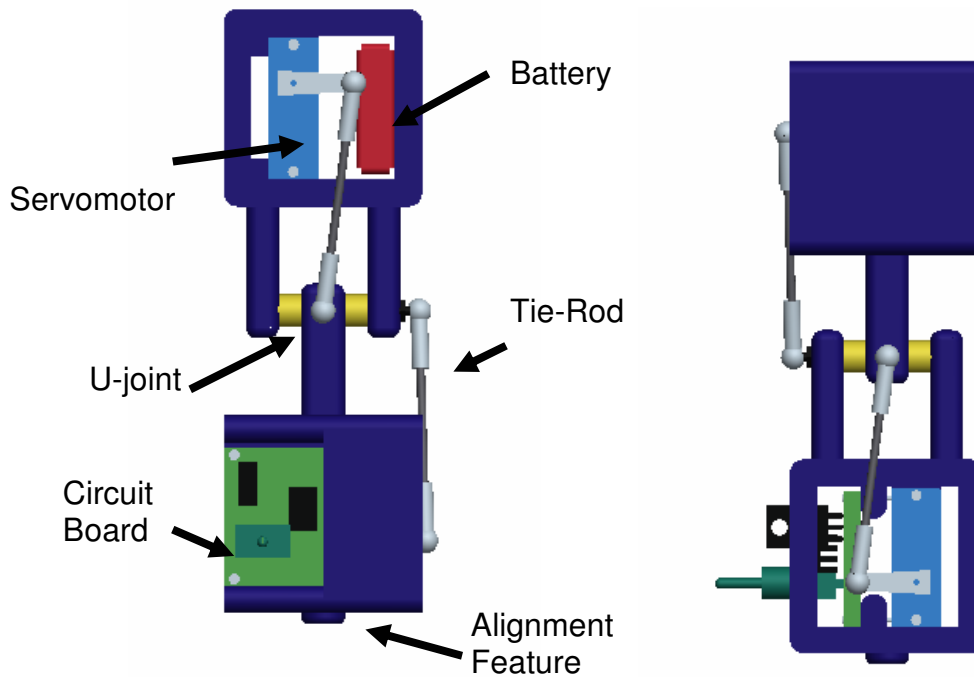


Figure 5.14. Side views of modules.

The molds for the design were produced using a 3-axis computer numerically controlled milling machine. Using a CNC milling machine allowed for good geometric control of tolerances, and a relatively wide range of possible features, including 2D and even 3D curved surfaces. The molds were fabricated out of a medium-density polyurethane tooling board. The tooling board was selected because it causes significantly less wear on the machine tools than a metal, but is tough enough to be used for many molding cycles.

Molds were designed using Pro/Engineer Wildfire and the Pro/Manufacturing kernel. The parts were designed manually with parting directions and the multi-material molding process in mind. Constraints on the part design included wall thickness limitations due to the machine tool parameters. The thinnest shell in the direction perpendicular to the parting direction that was possible to create was 1/8", due to the fact that the thinnest machine tool in the machine tool library with a sufficient length was 1/8". In addition, there are constraints arising from the length of the machine tools. Pro/Mold Design was used to generate the molds from the solid models of the parts, with modifications made to the resultant molds to accommodate the multi-material process. The final solid models of the molds were then used to generate G-code using Pro/Manufacturing, and the G-code was uploaded onto the CNC milling machine.

A two step overmolding process was used to fabricate the module structures. The two main pieces (blue in Figure 5.15) were molded using two piece molds with one parting direction. A split core technique was used to make the interior features that contain the pins of the universal joint. Additionally, a side core was used to make the female connection (alignment) feature. The first molding stage is shown in Figure 5.16 and Figure 5.17. The second step of the molding process involves the removal of part #2 from the first stage of mold, and placing it in the designated slot in the mold half from part #1, with the second stage mold piece already in place. A second shot of material is inserted into the second stage mold, and the universal joint piece is formed. The part is removed during the curing process, and the joint is flexed

to allow the interface to freely move. The second molding stage is shown in Figure 5.17 and Figure 5.18.

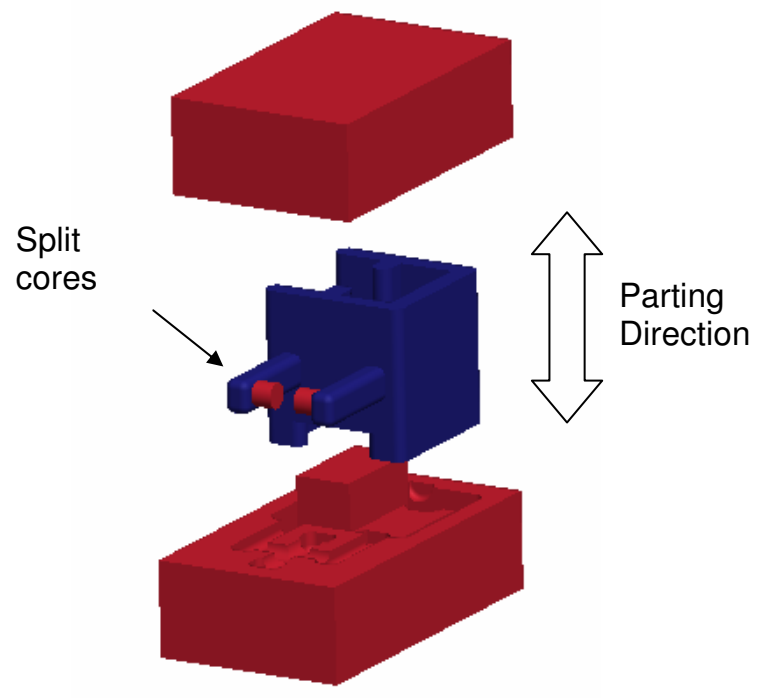


Figure 5.15. Stage 1 of molding for part 1.

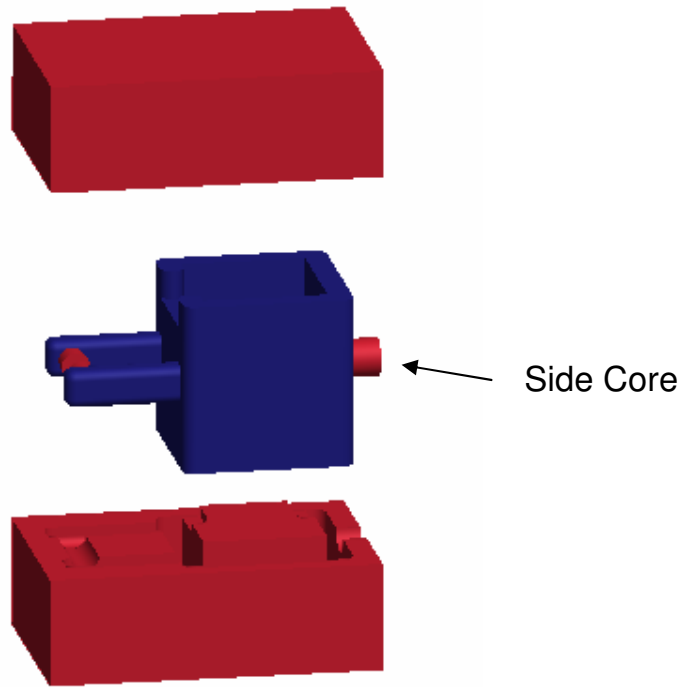


Figure 5.16. Stage 1 molding part 2.

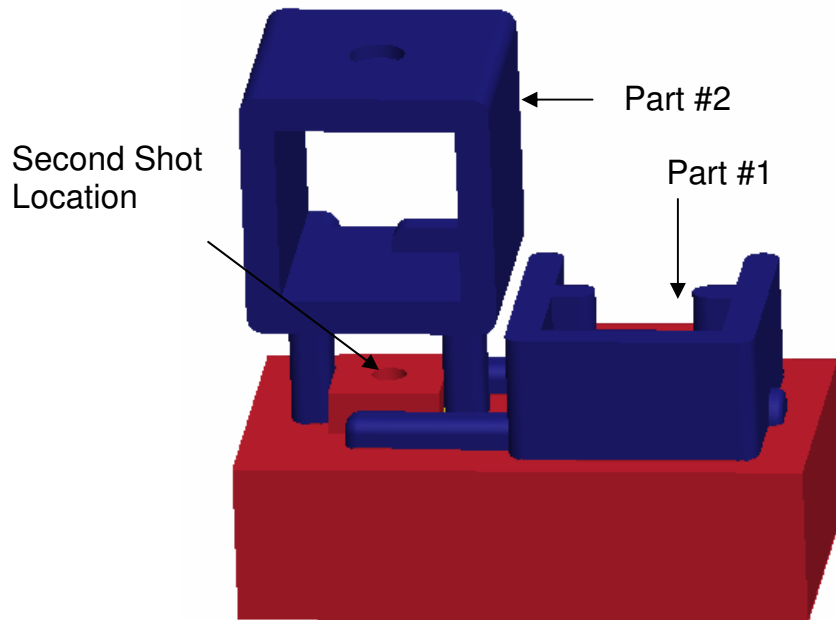


Figure 5.17. Stage 2 molding.

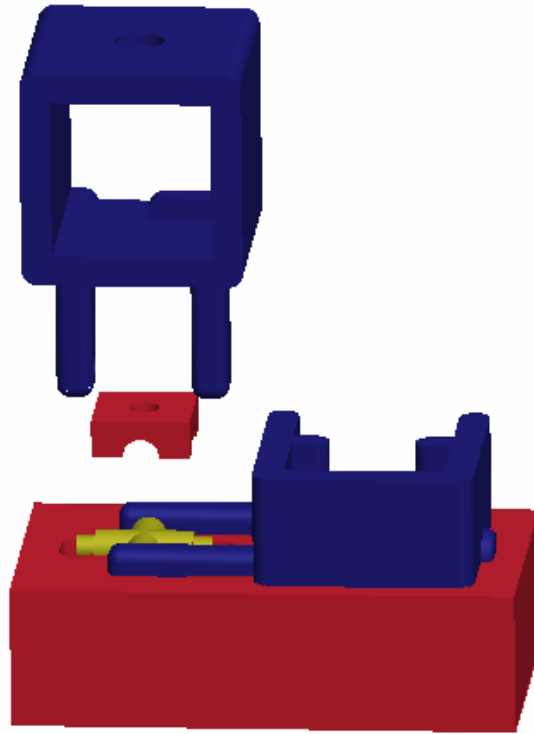


Figure 5.18. Second Stage exploded view.

The molded, three piece, articulated structure comes out of the mold fully-assembled and the components are then assembled to the part. The only additional operations that need to be performed on the structure are the drilling of mounting holes for the components, and flash removal. The servomotors and circuit are then mounted to the structure using 0-80 screws and hex nuts. The batteries are fastened to the side of the structure with plastic tie wraps. Wiring is performed manually and wires are strain-relieved with tie wraps. Finally, the push-rods for actuation are fastened to both the servo horn and the adjacent part using the threaded portions of the ball joints. A completed module is shown in Figure 5.19. The entire width of the module, including the push-rods is 5.2 cm. The switch protrudes and additional

1.7cm, however, this will be changed in subsequent iterations. Drawings of the module as well as a list of components are contained in Appendix B.

The total cost of the components for each module was approximately \$70 for this phase of prototyping. However, these it can be anticipated that many of these costs can be reduced as the scale of production is increased. The major contributors to the costs in the prototype were the battery and the servomotors. These items were almost \$50 combined. Additional costs were the electronics (estimated at \$10) and the hardware and material combined (estimated at \$10). These costs should decrease as the components are purchased in larger quantities. Hardware costs could be further decreased by using additional shots of material to take the place of hardware components.

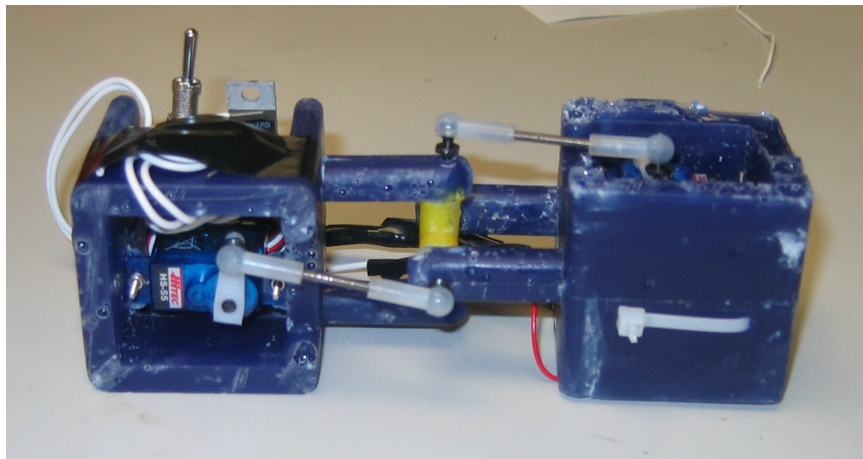


Figure 5.19. Robot module.

Upon completion of all six modules, the entire snake was assembled. The assembly was completed using plastic ties that are firmly tightened. The alignment features were used to place the modules in the proper place with respect to each other,

and two plastic ties were applied. After the entire snake was assembled, the ground bus was then added to synchronize the servo actions by wiring between adjacent modules. A switch was added to the last module to switch the ground bus on and off. The assembled robot is shown in Figure 5.20.

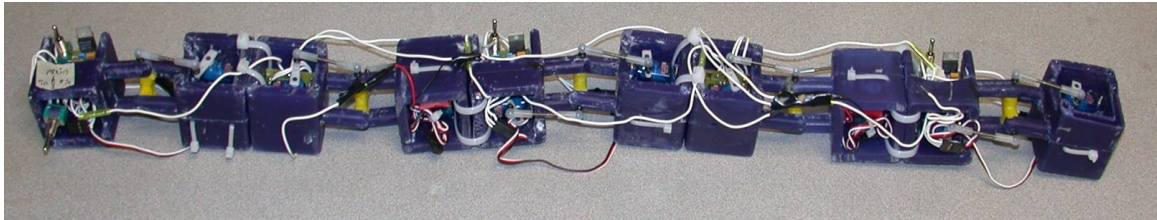


Figure 5.20. Fully assembled modular snake-inspired robot.

It is important to note that there is a slight mismatch in the physically realized snake-inspired robot, and the modeled robot. In order to develop a simple model of a snake-inspired robot, the basic structure of a snake-inspired robot was assumed to consist of uniform links connected by joints. As can be seen in any of the figures in Chapter 3, the robot consists of N links, and $N-1$ joints. However, in order to develop a fully-modular snake-inspired robot, it was necessary to make the number of joints and number of links the same. This design means that there is essentially one-half of a link on each side of the joint, leaving half links on either end of the entire robot. In order to rectify this, “dummy” modules are placed on either end of the snake-inspired robot to make all of the links the same size. This is shown in Figure 5.21. The addition of these parts makes the entire snake 79.4 cm long.

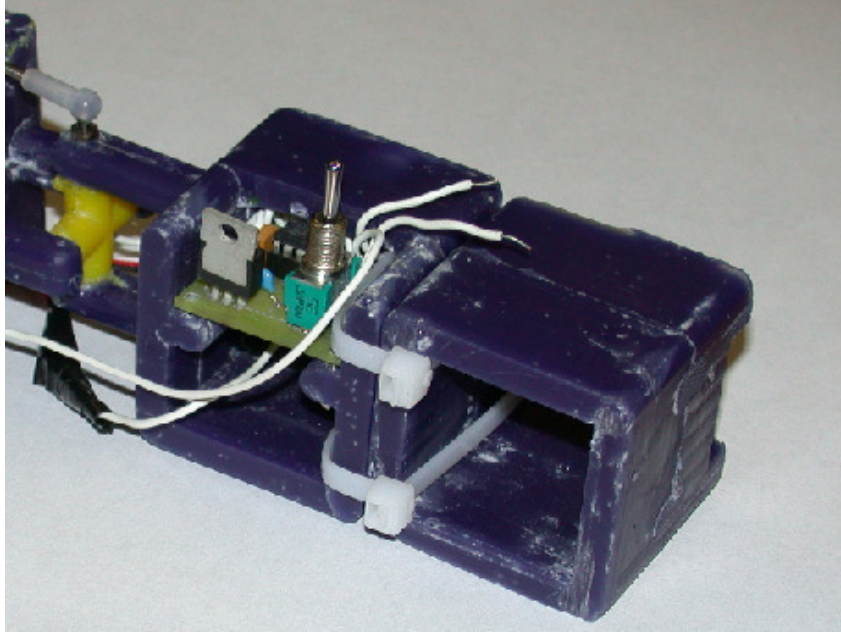


Figure 5.21. End module.

5.6 Locomotion Results

This robot was used to demonstrate a gait similar to that presented in [34] and shown in Figure 3.4. The major difference between the two gaits is that the wave that was propagated through the snake once-at-a-time in the proposed gait is successively propagated through in the implemented gait, achieving continuous motion. Consulting Figure 3.4, it can be seen that it is not practical for the snake to lie flat at any point in the motion; it is more efficient to successively propagate the wave through. The servos were moved to the successive positions using a linear interpolation.

Snapshots of the locomotion are shown in Figure 5.22. In these snapshots, the snake is moving from right to left, and successive snapshots show that a wave is

propagating from the right to left, moving the snake in that direction. From this study, it can be concluded that this class of gait with the vertical rectilinear locomotion will result in forward propulsion of a snake-inspired robot.

Testing of the robot demonstrated that in 10 gait cycles, the snake progressed for 16.3 cm. This took 8.7 seconds, resulting in a forward speed of approximately 1.9 cm/s. The servo speeds in this experiment were set such that the completion of one “gait step” as defined by the formulation in Section 3.3, would take approximately 0.3 seconds. This speed is sufficient to demonstrate forward motion, but significantly less than the speeds predicted in Chapter 4. A part of the explanation for this is inefficiency of this gait as opposed to the gait proposed in Chapter 3. However, a majority of the discrepancy can be explained by the fact that the servos seemed slightly underpowered for the task, and consequently became out of synchronization at times, this eventually led to the slowing, and even once the robot completely stopped. They also could not lift up to the appropriate angle. This is an issue that will be addressed in future work.

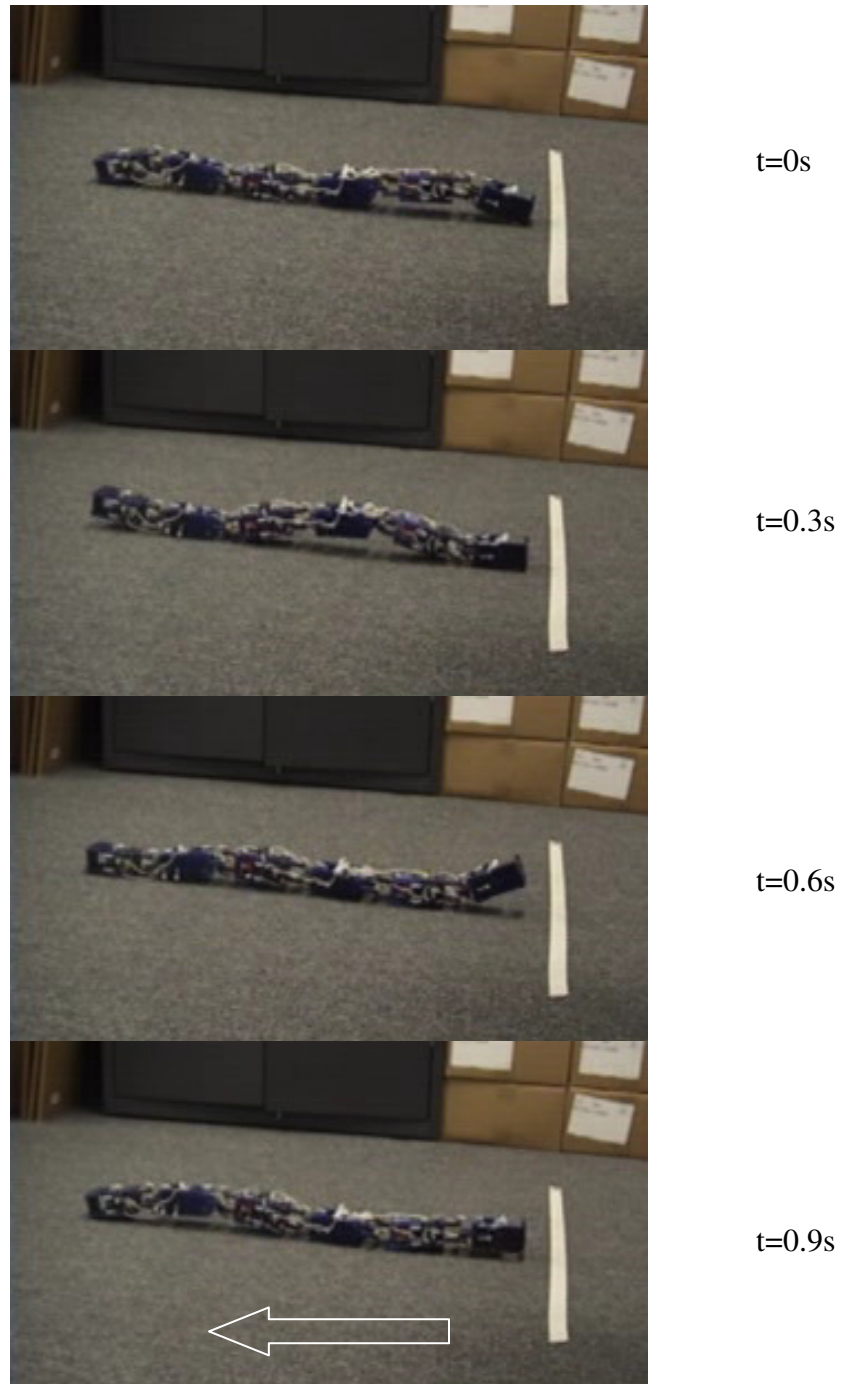


Figure 5.22. Snapshots of locomotion.

5.7 Next Generation Snake: Embedded Components

As mentioned before, a further goal of this work is to develop a module utilizing a fully-embedded design. That is, all of the components of the module would be pre-

assembled in the mold, and the module would be ejected from the mold fully-assembled. The aim of this design is to significantly reduce the amount of assembly operations that arise from fastening the numerous components such as motors and batteries to the basic mechanical structure. In addition, many parts such as fasteners can be eliminated from the design, because the molded structure dually functions as the outer structure and the fastening material. Realizing such a module could significantly reduce part and labor costs. Furthermore, by embedding components, designs could give more protection to components from thermal and mechanical shock, in addition to other hazards.

Preliminary work has been conducted in the development of a module with embedded components. A design was developed using the same batteries previously mentioned, a microchip, and a pager motor to demonstrate the technology. In this design, the chip was preprogrammed, and the entire circuit was soldered together, eliminating the need for a circuit board, and maximizing usage of the three-dimensional space of the module. The circuit was pre-assembled into the mold, with the wires exposed to connect to the motor and battery in the adjacent module. In the opposite module, the battery was placed, with the wires exposed as well. The molding process is conducted in the same fashion as shown in Figure 5.15 through Figure 5.18, and the module was removed from the mold with the circuit and battery fully assembled. The completed embedded module is shown in Figure 5.23 and Figure 5.24. The wires were connected after the component is removed from the mold, and the motors and tie-rods were assembled after de-molding. The ability to successfully embed a motor has been recently demonstrated, however, allowing the

actuation arm to be fully exposed. In this case, the mold and shut off surfaces were designed such that it is possible to have the actuation portion of the motor protruding from the molded structure. The embedded motor is shown in Figure 5.25. Furthermore, while the molded module was developed using a polymer casting process, separate experiments that have been conducted have shown that it is possible to embed a microchip in injection-molded polyethylene at 170 degrees C with no ill effects.

The module was shown to run for more than thirty minutes successfully. It was programmed with a simple actuation sequence, and demonstrated the actuation of one of the two degrees-of-freedom of the universal joint. After this demonstration, the battery was re-charged while still embedded in the module, and the module was shown to function after the charging process.

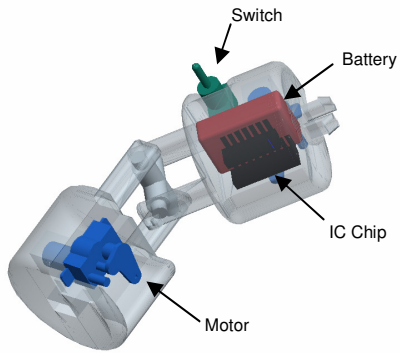


Figure 5.23. Solid Model of embedded module.



Figure 5.24. Embedded module.

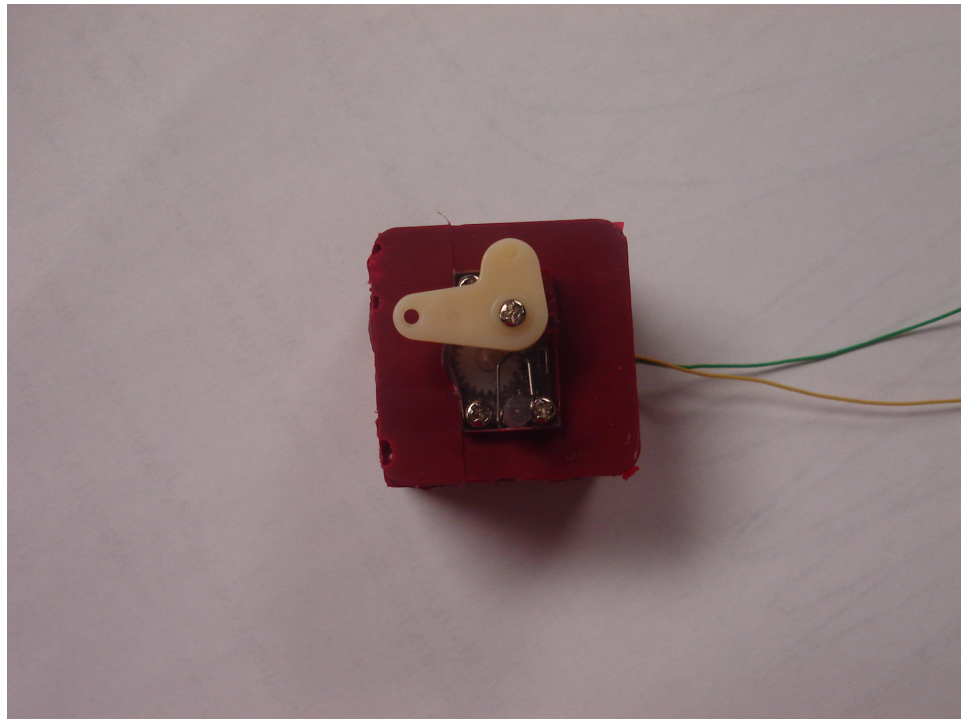


Figure 5.25. A pager motor that has been partially embedded in polyurethane.

5.8 Summary

This chapter overviewed the design and development of a snake-inspired robot. The robot was based on a fully-modular design, and the structure was fabricated using a multi-material molding technique aimed at reducing assembly operations and part count. The design goals and objectives were first outlined, and then the entire design was described, including the manufacturing process. The snake has a frontal cross section of only 36 cm^2 , and testing demonstrated that it could move at an average of 1.9 cm/s . Finally, results of experimentation and prototyping with embedded components was discussed, with the aim that an entire snake-inspired robot could be fabricated in such a manner.

Chapter 6 - Conclusions

This work aimed at facilitating the design of snake-inspired robots in several areas. A specific class of gaits was analyzed, and a kinematic and dynamic model was developed in order to better understand the mechanics of the locomotion as it relates to the mechanical design, as well as the gait design. A means of generating gaits was presented based on this model, as well as a new formulation for joint trajectories. Finally, a snake-inspired robot was designed and built using a fully modular architecture and an innovative manufacturing process aimed at reducing costs.

6.1 Contributions

The contributions of this work draw from several different aspects of snake-inspired robot design, and thus can be separated into 4 different contributions:

- 1) A detailed dynamics analysis of a rectilinear gait was developed that uses an Eulerian framework, and accounts for the morphing topology of the mechanism. The model incorporates Coulomb friction. To our knowledge, this is the first formulation of expressions for joint torques over the entire course of motion for a snake moving with a vertical rectilinear gait.
- 2) A means of generating gaits was developed based on an effort-minimization scheme. The joint trajectories were represented using B-splines, allowing for improved parameterization of the motion. Direct gradient computation of the

- Newton-Euler equations was performed using an implicit scheme that directly addresses the matrix formulation.
- 3) The effects of changing the length and mass of the robot links, as well as the gait angles and times, on the performance of the robot were determined using the effort metric.
 - 4) A modular snake-inspired robot based on an articulated universal joint fabricated using multi-material molding was designed and developed. To our knowledge, this is the first snake-inspired robot that has been produced using multi-material molding. The embedded module is also the first snake-inspired robot module that uses mechanically embedded electronics.

6.2 Anticipated Benefits

The work conducted and reported on in this thesis has the potential to benefit the area of snake-inspired robot design in several different ways. The development and demonstration of a snake-inspired robot based on a fully-modular and multi-material molded design has the potential to make snake-inspired robots rapidly and mass producible, decreasing costs. The embedded module that was developed would decrease assembly operations even more. This would allow the robots to be expendable and thus more functional for applications where budgets are tight, and damage to such a robot is likely. Additionally, lowering costs would also allow users to be able to afford more robots. Finally, the work done towards generating a fully-embedded actuated and powered module may have other implications in the design and manufacturing of small, actuated systems such as robots and MAVs.

There are also numerous anticipated benefits to the modeling and gait design that was presented in this thesis. By having a model that predicts the joint torques and forces during the motion, designers can better design snake-inspired robots. Additionally, by having performance information about gaits, this information can be used to design better gaits, as has been presented in this work. The development of gaits that use less effort will make snake-inspired robots more efficient in their locomotion, and thus able to travel greater distances using the same amount of energy. Finally, in this thesis a study on the variation of bulk parameters of the snake-inspired robot was presented. This study will allow designers to understand trade-offs in the performance of robots designed with different parameters, and will allow designers to better design snake-inspired robots for certain mission requirements.

6.3 Future Work

While considerable work has been done in the direction of the development of a modular, multi-material molded snake-inspired robot, in addition to the tools for mechanical design, gait design, and trade off analysis, much future work remains to realize the full potential of the project.

6.3.1 Model Improvement and Validation

The model that was developed in this thesis must be improved to better match with the real snake-inspired robot, and the model needs to be compared against the actual

robot for further validation. In order to do this, several areas must be addressed. First, the model needs to be modified to accommodate a redundant actuation scheme, since the actual robot has all of its joints actuated during the course of the motion, instead of merely the ones that are needed to give the desired link movement. In addition, a more detailed analysis of the actual links should be conducted, such that the real moments of inertia and other properties could be used in the model, instead of a general approximation. Similarly, the model developed in this work assumed that the actuation was directly applied at the joint, as opposed to using a lever arm, like the actual robot that was developed in this thesis; this also needs to be accounted for in the torque equations. A better friction model also needs to be developed that accounts for the transition between static and dynamic friction, and friction characteristics between the robot and surfaces should be measured. Joint frictions and deformations in the module should also be accounted for. Finally, the obtained joint trajectories from Section 3.6 should be implemented on the prototype snake-inspired robot to compare performance.

6.3.2 Improved Prototype Design

There are two main areas that should be addressed with regard to the prototype snake-inspired robot. A better design needs to be developed that allows for better and more rapid testing of gaits, and also addresses some of the weaknesses identified with the current prototype. Additionally, a snake-inspired robot utilizing fully-embedded power, control, and actuation components should be developed that is sufficiently

rugged to use in desired applications. The plans for future work in this project involve pursuing both of these goals in parallel.

An improved snake-inspired robot should be developed for the design and testing of gaits. While the fully-modular architecture has several advantages that have been highlighted, it is inconvenient for the rapid testing of gaits. The controller from each module would have to be programmed separately with its specific role. Additionally, issues were encountered with regard to keeping the servos synchronized in the current prototype snake. The next step towards developing and testing gaits should be the development of a snake-inspired robot with a centralized processor that allow rapid programming and re-programming to better facilitate the generating and testing of gaits in the laboratory setting.

Simultaneously, work should be done to advance the development of the fully-modular and multi-material molded design. A new, improved robot is currently under development with more robust and protected wiring. This robot will contain electrical connectors so that each robot may be snapped together, as opposed to hard-wired, which was done in the previous design. Additionally, stronger servomotors should be used. This design should move the project towards the fully-embedded snake-inspired robot. Upon completion of the new module, which still requires assembly of components, a new embedded module will be developed that has all components embedded. This should improve on the work discussed in Section 5.7 because it would have servomotors and servo control embedded into it, and would be able to be re-programmed.

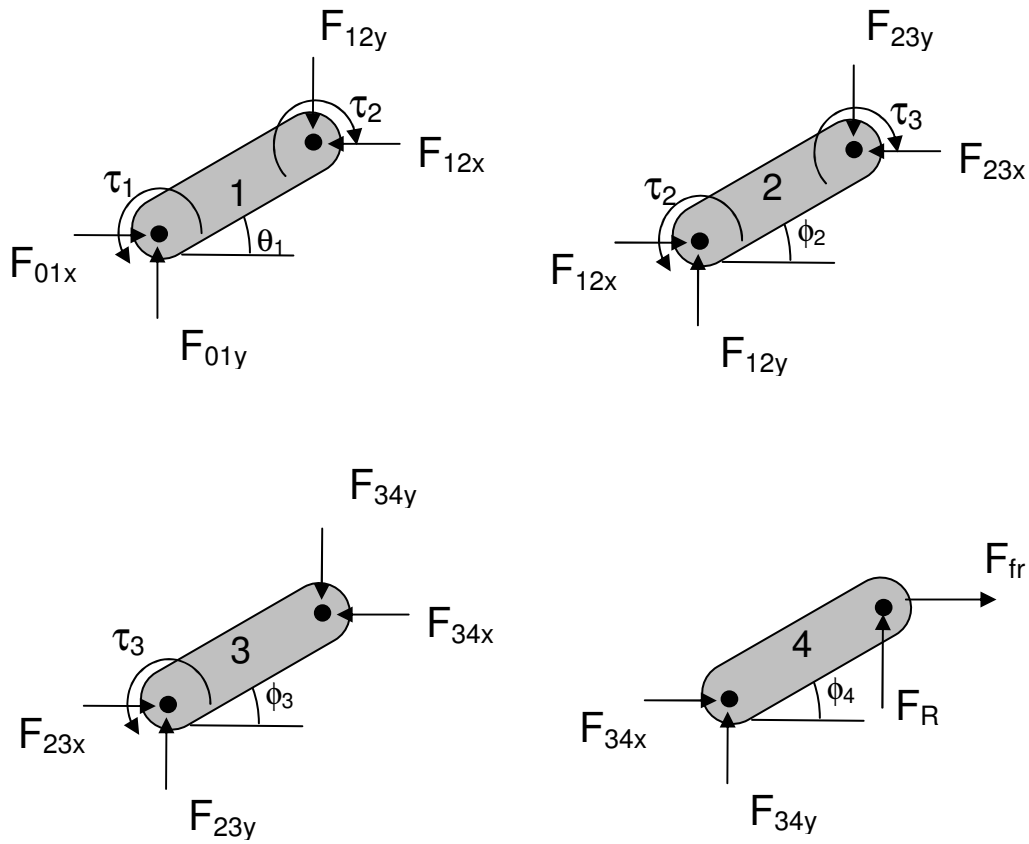
6.3.3. Generation of Optimal Gaits

The work presented in Section 3.5.2 should also be elaborated on to develop joint trajectories for the rectilinear locomotion gait. In order to do this, the framework presented should be used to compute the gradient of the torque functions for each of the joints in each of the mechanisms, with respect to the B-spline control points, in order to find minimum-effort solutions. This should be accomplished by the creation of a simple string-processing algorithm that would automatically conduct an implicit differentiation of the functions in order to generate the new equations. Then, a gradient-based algorithm should be developed and tested to ensure that the generated trajectories are locally-optimal solutions.

Appendix A: Equations

Formulation of the matrix equation for mechanism M_2

Free body diagrams:



Equations follow the form:

$$F_a = A^{-1}F_i$$

“A” matrix on following page

With “vtip” indicating the x-component of the velocity of the tip of the final (4th) link, in the base reference frame.

$$\text{And } h = \frac{l}{2} \cos \phi_4 + \frac{l}{2} \text{sign}(v_{tip}) \mu \sin(\phi_4)$$

$$\underline{F}_i = \begin{bmatrix} \dot{m} v_{1x} \\ m v_{1y} \\ \dot{I} \omega_1 \\ \dot{m} v_{2x} \\ m v_{2y} \\ \dot{I} \omega_2 \\ \dot{m} v_{3x} \\ m v_{3y} \\ \dot{I} \omega_3 \\ \dot{m} v_{4x} \\ m v_{4y} \\ \dot{I} \omega_4 \end{bmatrix} \quad \underline{F}_a = \begin{bmatrix} F_{01x} \\ F_{01y} \\ F_{12x} \\ F_{12y} \\ F_{23x} \\ F_{23y} \\ F_{34x} \\ F_{34y} \\ F_R \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

The values for the torque on joints 1, 2, and 3 (τ_1 , τ_2 , τ_3 , respectively) are found from:

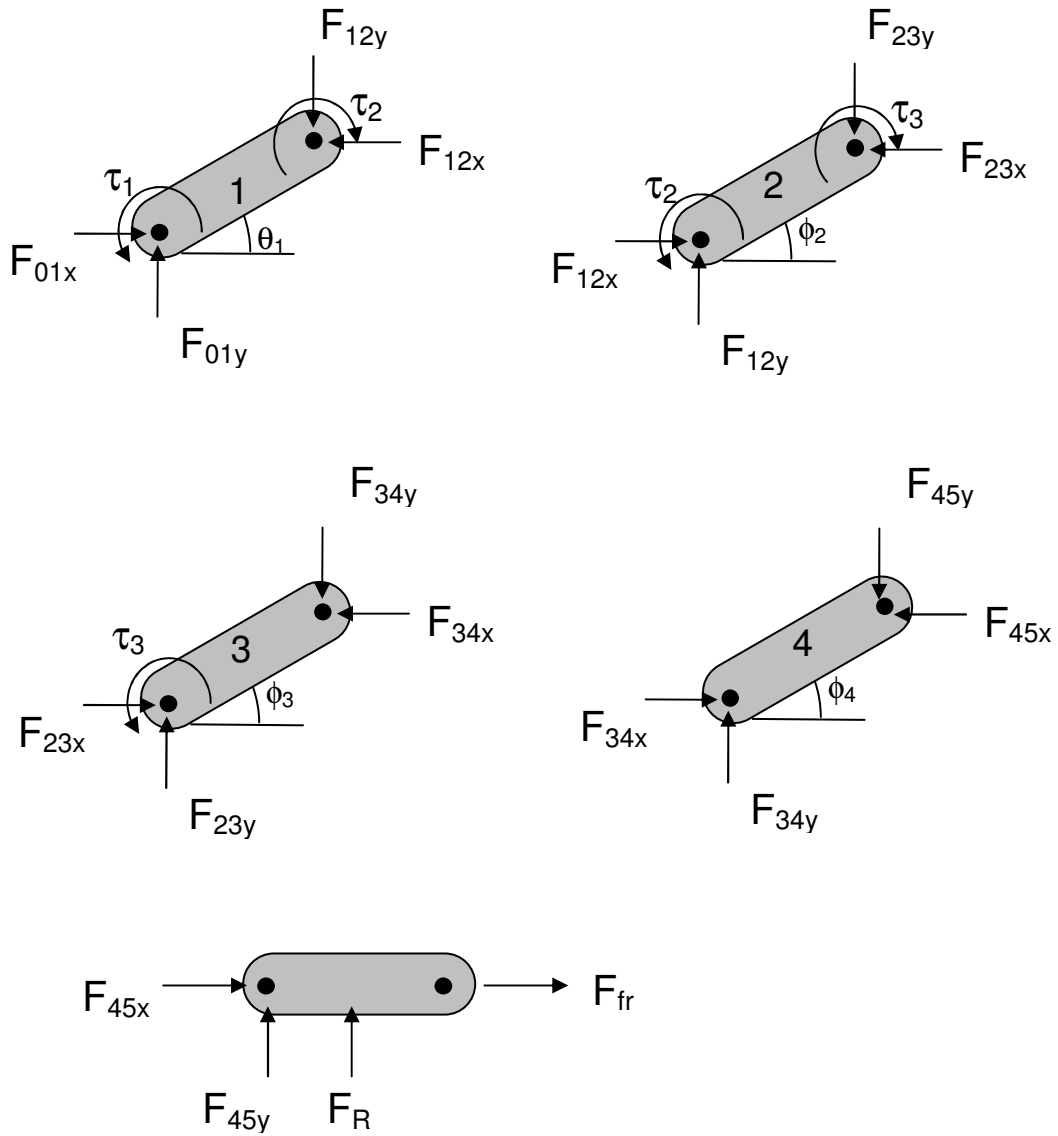
$$F_a(10) = \tau_1$$

$$F_a(11) = \tau_2$$

$$F_a(12) = \tau_3$$

Formulation of the matrix equation for mechanism M_3

Free body diagrams:



Equations follow the form:

$$F_a = A^{-1}F_i$$

“A” matrix on following page

With “v5” indicating the x-component of the velocity of the 5th link in the base reference frame.

$$\underline{F}_i = \begin{bmatrix} \dot{m}v_{1x} \\ \dot{m}v_{1y} \\ I\dot{\omega}_1 \\ \dot{m}v_{2x} \\ \dot{m}v_{2y} \\ I\dot{\omega}_2 \\ \dot{m}v_{3x} \\ \dot{m}v_{3y} \\ I\dot{\omega}_3 \\ \dot{m}v_{4x} \\ \dot{m}v_{4y} \\ I\dot{\omega}_4 \\ \dot{m}v_{5x} \\ \dot{m}v_{5y} \end{bmatrix} \quad \underline{F}_a = \begin{bmatrix} F_{01x} \\ F_{01y} \\ F_{12x} \\ F_{12y} \\ F_{23x} \\ F_{23y} \\ F_{34x} \\ F_{34y} \\ F_{45x} \\ F_{45y} \\ F_R \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

The values for the torque on joints 1, 2, and 3 (τ_1 , τ_2 , τ_3 , respectively) are found from:

$$Fa(12) = \tau_1$$

$$Fa(13) = \tau_2$$

$$Fa(14) = \tau_3$$

Development of Kinematics for Mechanism M₂

Constraint relation:

$$\sin \theta_1 + \sin \phi_2 + \sin \phi_3 + \sin \phi_4 = 0$$

$$\dot{\phi}_4 = \frac{-1}{\cos \phi_4} \left(\dot{\theta}_1 \cos \theta_1 + \dot{\phi}_2 \cos \phi_2 + \dot{\phi}_3 \cos \phi_3 \right)$$

$$\ddot{\phi}_4 = \frac{-1}{\cos \phi_4} \left(\ddot{\theta}_1 \cos \theta_1 - \dot{\theta}_1^2 \sin \theta_1 + \ddot{\phi}_2 \cos \phi_2 - \dot{\phi}_2^2 \sin \phi_2 + \ddot{\phi}_3 \cos \phi_3 - \dot{\phi}_3^2 \sin \phi_3 - \dot{\phi}_4^2 \sin \phi_4 \right)$$

Development of Kinematics for Mechanism M₃

Constraint equation is the same as M₂.

Additional term arises due to the extra link (link 5) that is always flat on the ground. The angle of link 5 relates to the angle of link 4 by the equation:

$$\theta_5 = -\phi_4$$

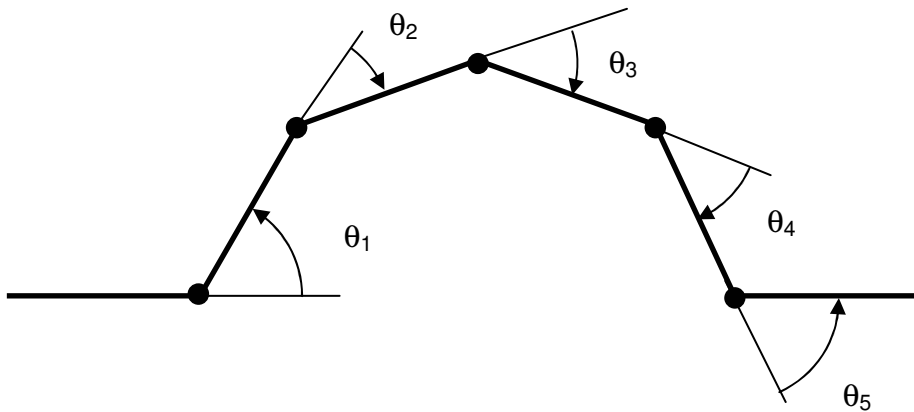


Figure A.1. Angles in M₃.

Appendix B. Design Information

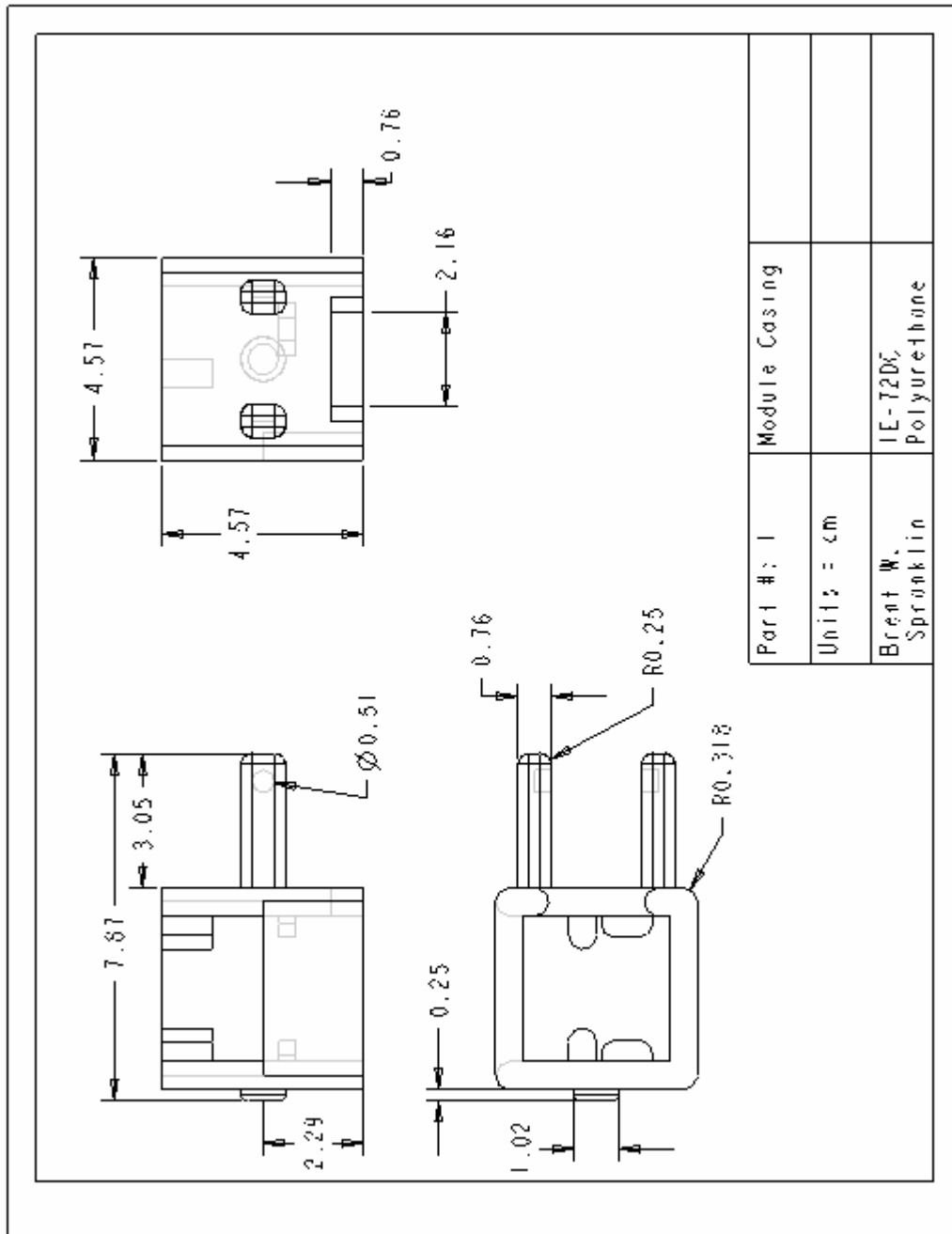


Figure B.1. Engineering drawing of part 1.

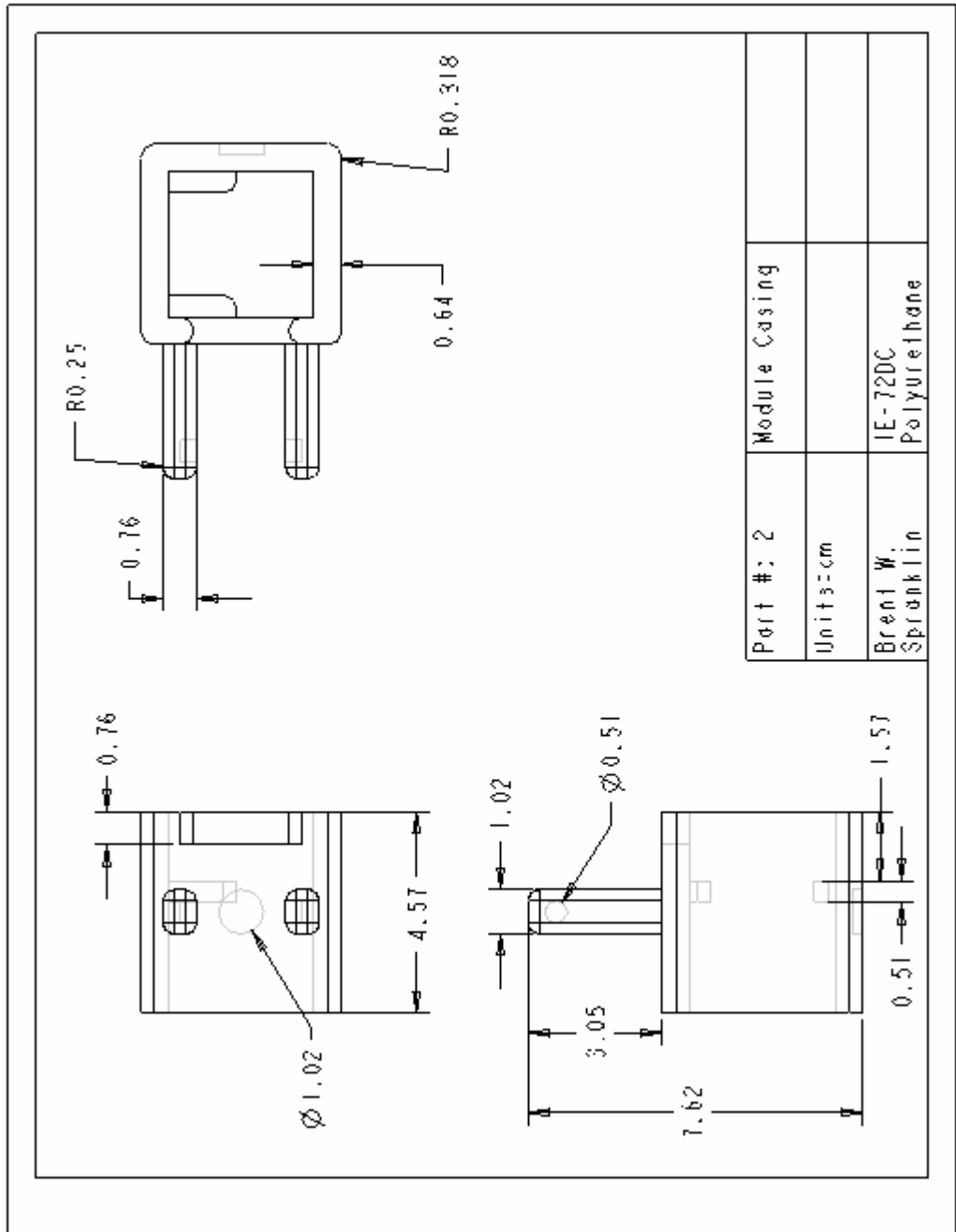


Figure B.2. Engineering drawing of part 2.

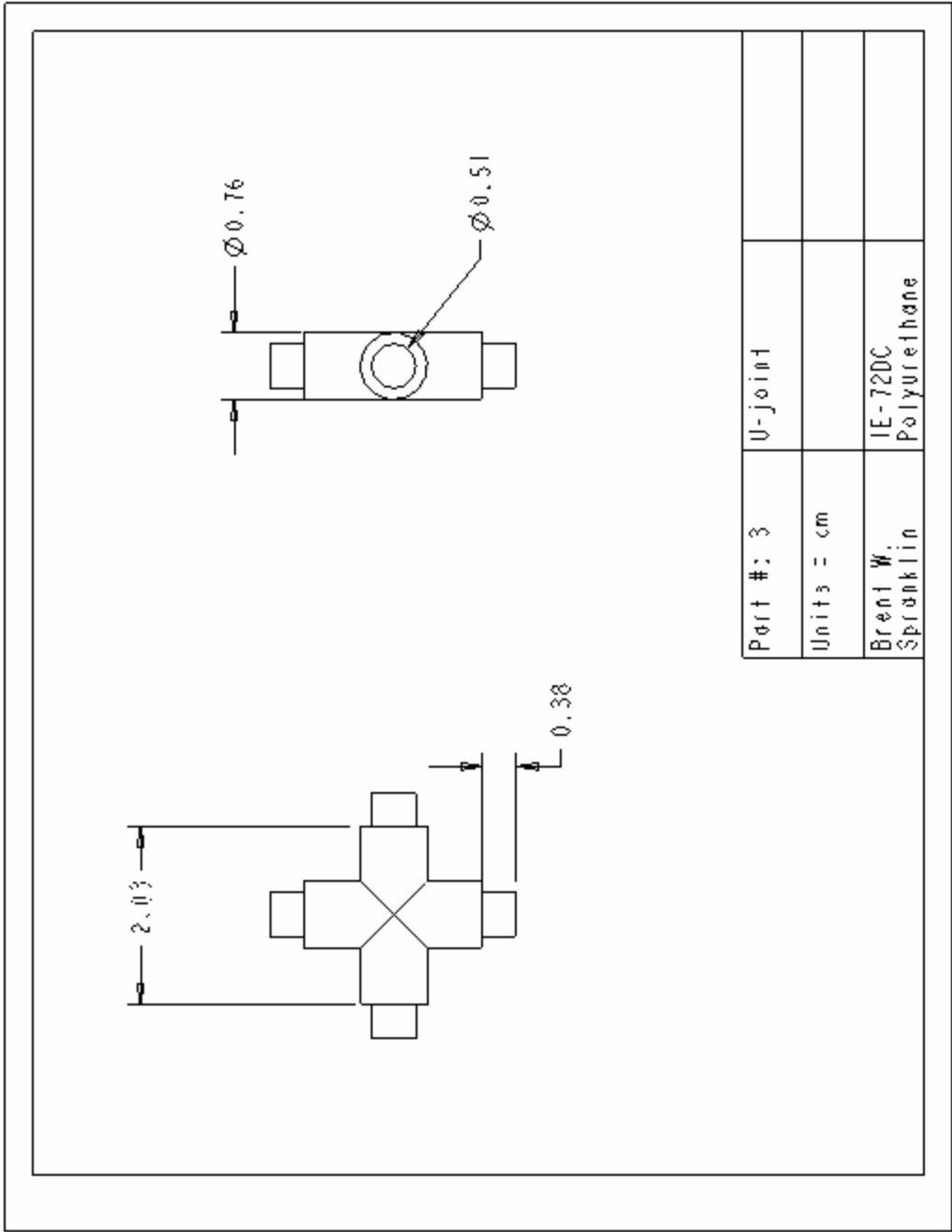


Figure B.3. Engineering drawing of part 3.

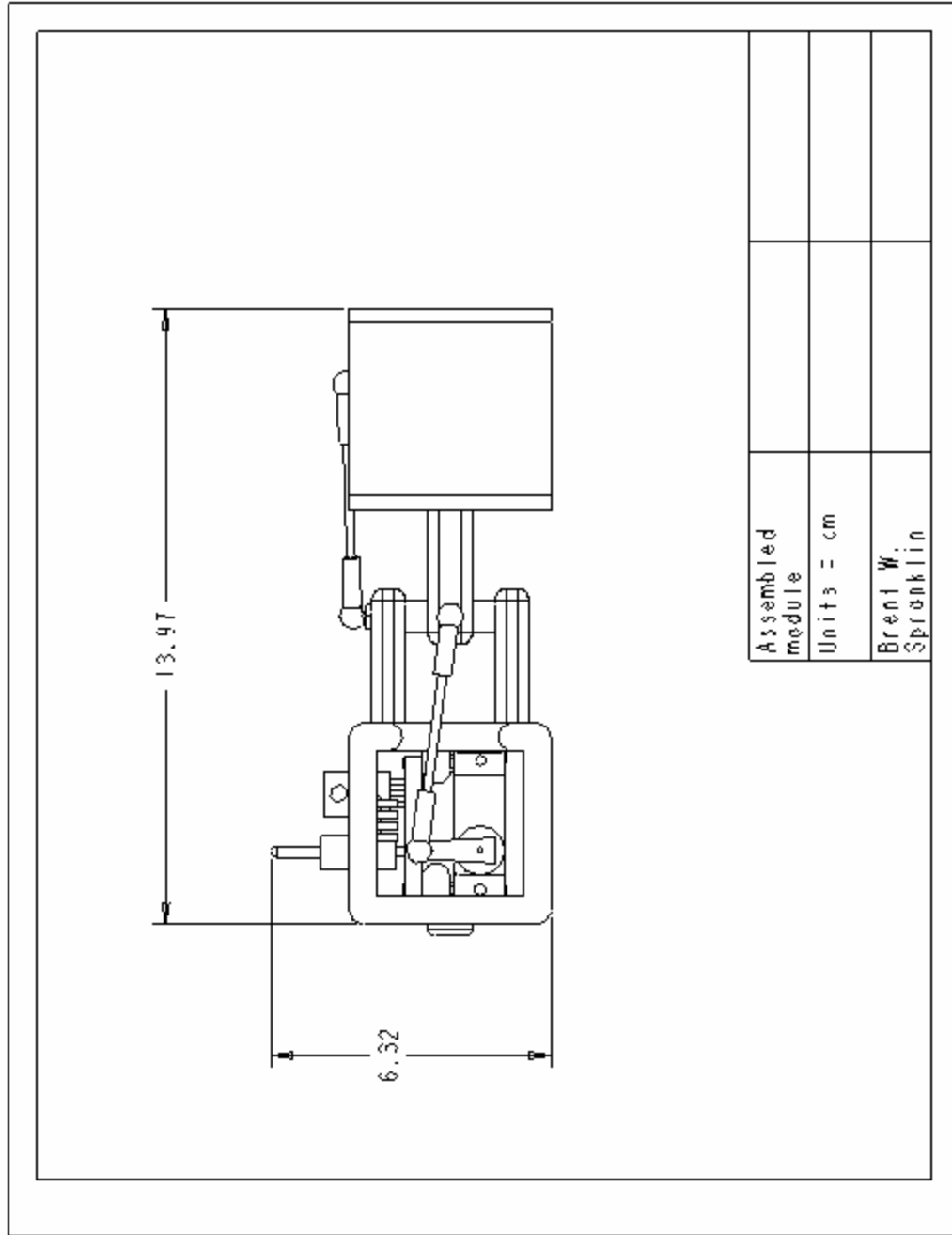


Figure B.4. Engineering drawing of entire module.

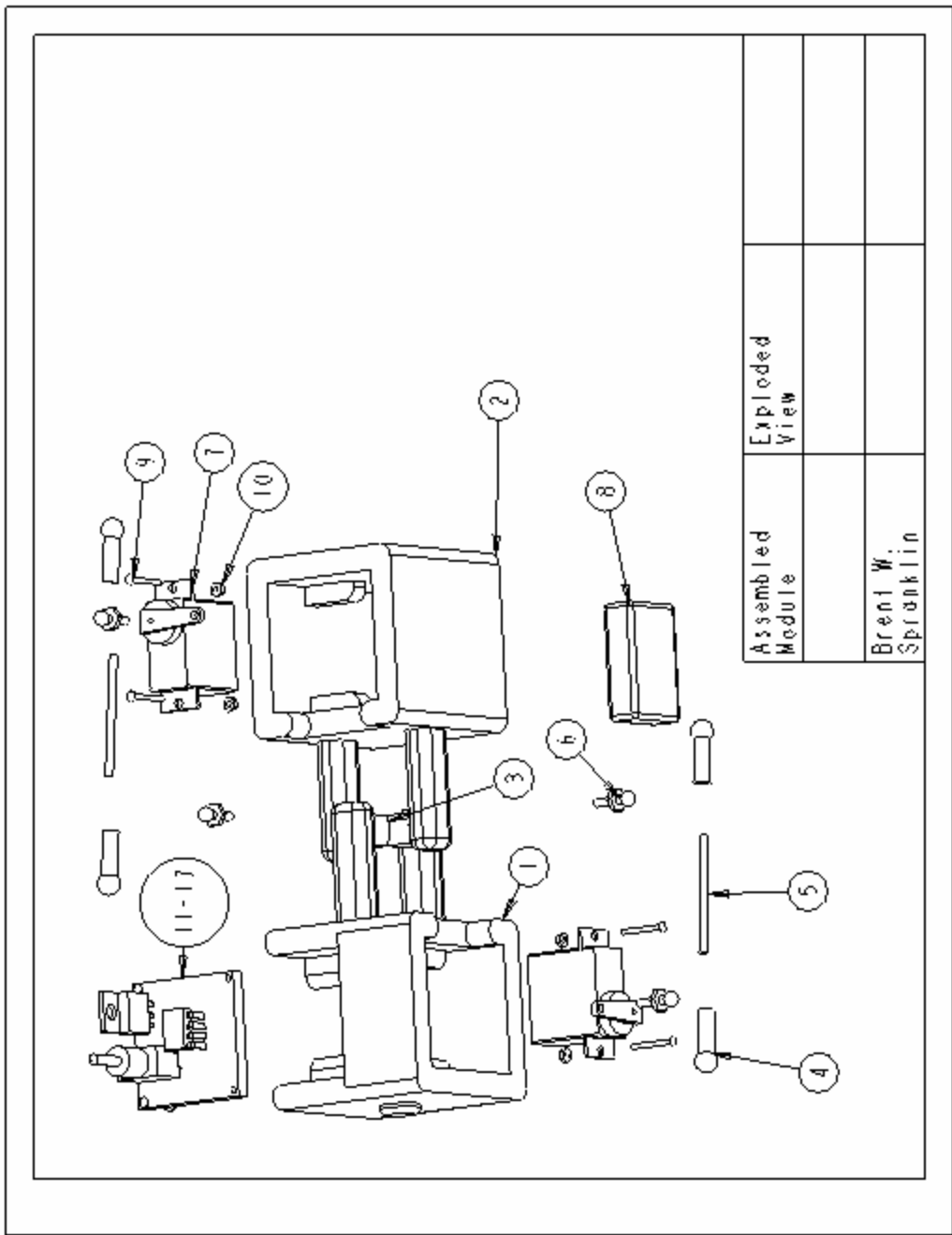


Figure B.5. Exploded view of entire module, numbered items are in Table 1.1.

Table B.1. – Components List (Per Module)

Part #	Name	Material	Manufacturer	Quantity	Note
1	Casing Part #1	IE-72DC Polyurethane	In house	1	Molded using polymer casting
2	Casing Part #2	IE-72DC Polyurethane	In house	1	Molded using polymer casting
3	U-joint piece	IE-72DC Polyurethane	In house	1	Molded using polymer casting
4	Ball-joint socket			4	
5	Threaded Rod	Stainless Steel		2	
6	Ball-joint ball/screw			4	
7	Servomotor		Hi-Tec	2	0.7" servo horn included
8	Battery	Lithium Polymer cell	Kokam	1	
9	0-80 screw	Stainless Steel		6	
10	0-80 hex nut	Stainless Steel		6	
11	Switch		Alco	1*	3 way toggle switch
12	Microcontroller		MicroChip	1	
13	Voltage Regulator			1	
14	Printed Circuit Board	FR-4, tin, copper		1	Designed by Lawrence S. Gyger
15	.33 μ F Capacitor			1	
16	.1 μ F Capacitor			2	
17	1 k Ω Resistor			1	

* - An extra switch is required on the end module.

References

- [1] M. Yim, D.G. Duff, and K. Roufas. Modular Reconfigurable Robots, An Approach To Urban Search and Rescue. In Proceedings of the *1st Intl. Workshop on Human-friendly Welfare Robotics Systems*, Taejon, Korea, Jan. 2000.
- [2] R.S. Desai, C.J. Rosenberg, and J.L. Jones. Kaa : An Autonomous Serpentine Robot Utilizes Behavior Control. In *Proceedings of IROS'95*, 3: 250-255, 1995.
- [3] J.S. Gray. *Animal Locomotion*. Norton, 1968.
- [4] H.W. Lissman. Rectilinear motion in a snake (*Boa Occidentalis*). *J. Exp. Biol.*, 26: 368-379, 1950.
- [5] M. Walton, B.C. Jayne, and A.F. Bennett. The energetic cost of limbless locomotion. *Science*, 29: 524-527, 1990.
- [6] S. Hirose and E.F. Fukushima. Snakes and Strings: New Robotic Components for Rescue Operations. *The International Journal of Robotics Research*, 23 (4-5): 341-349, 2004.
- [7] G.S.P. Miller. Snake robots for search and rescue. In Ayers J., J. L. Davis, and A. Rudolph (Eds.). *Neurotechnology for Biomimetic Robots*, MIT press, 2002.
- [8] W.-M. Shen, B. Salemi, and P. Will. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Trans. Robot. Automat.*: 700–712, 2002.
- [9] S. Hirose. *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford University Press, 1993.

- [10] M. Mori and S. Hirose, Development of active cord mechanism ACM-R3 with agile 3D mobility. In Proceedings of the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, 1552–1557, 2001.
- [11] Expo AICHI Japan 2005. <http://www.expo2005.or.jp/en/index.html>
- [12] HiBot Company Website.
http://www.hibot.co.jp/html/html_eng/products/available.html#ACM-R5
- [13] K. Dowling. *Limbless locomotion: learning to crawl with a snake robot*. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1997.
- [14] G.S. Chirikjian and J.W. Burdick. A hyper-redundant manipulator. *IEEE Robotics & Automation Magazine*, 1(4): 22-29, Dec. 1994.
- [15] Y. Shan and Y. Koren. Design and Motion Planning of a Mechanical Snake. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4): 1091-1100, 1993.
- [16] H. Ikeda and N. Takanashi. *Joint Assembly Moveable Like a Human Arm*, US Patent 4683406, July 28, 1987. Assignee: NEC Corporation.
- [17] E. Paljug, T. Ohm, and S. Hayati. The JPL Serpentine Robot: a 12-DOF system for inspection In the *Proc. of the 1995 IEEE/ICRA*, 3: 3143-3148, 1995.
- [18] A. Wolf, H.H. Choset, B.H. Brown, and R.W. Casciola. Design and control of a mobile hyper-redundant urban search and rescue robot. *Advanced Robotics*, 19(3): 221 – 248, 2005.

- [19] G.S.P. Miller. www.snakerobots.com
- [20] A. Crespi, A. Badertscher, A. Guignard, and A.J. Ijspeert. AmphiBot I: an amphibious snake-like robot. *Robotics and Autonomous Systems*, 50(4), 163-175, 2005.
- [21] J. Borenstein, G. Granosik, and M. Hansen. The OmniTread Serpentine Robot – Design and Field Performance. In the *Proc. of the SPIE Defense and Security Conference, Unmanned Ground Vehicle Technology VII*, Orlando, FL, March 28th to April 1st, 2005.
- [22] J. Borenstein, M. Hansen, and H. Nguyen. The OmniTread OT-4 Serpentine Robot for Emergencies and Hazardous Environments. *2006 International Joint Topical Meeting: “Sharing Solutions for Emergencies and Hazardous Environments,”*, Salt Lake City, Utah, USA. February 12-15, 2006.
- [23] B. Klaassen, and K.L. Paap. GMD-SNAKE2: A Snake-Like Robot Driven by Wheels and a Method for Motion Control. *IEEE International Conference on Robotics and Automation*, 4: 3014-3019, 1999.
- [24] G.S. Chirikjian and J.W. Burdick. The Kinematics of Hyper-Redundant Robot Locomotion. *IEEE Transactions on Robotics and Automation*, 11(6): 781-793, 1995.
- [25] J. Ostrowski and J. Burdick. Gait Kinematics for a Serpentine Robot. *IEEE International Conference on Robotics and Automation*, 2: 1294-1299, 1996.
- [26] M. Saito, M. Fukaya, and T. Iwasaki. Modeling, analysis, and synthesis of serpentine locomotion with a multilink robotic snake. *IEEE Contr. Syst. Mag.*, 22: 64-81, 2002.

- [27] S. Ma, W. J. Li, and Y. Wang. A simulator to analyze creeping locomotion of a snake-like robot. *IEEE International Conference on Robotics and Automation*, 4: 3656-3661, 2001.
- [28] J. Cortes, S. Martinez, J.P. Ostrowski, and K.A. McIsaac. Optimal gaits for dynamic robotic locomotion. *International Journal of Robotics Research.*, 20(9): 707-728, 2001.
- [29] G.M. Kulali, M. Gevher, A.M. Erkmen, and I. Erkmen. Intelligent gait synthesizer for serpentine robots. *IEEE International Conference On Robotics And Automation.*,2: 1513-1518, 2002.
- [30] D.M. Rincon and J. Sotelo. Ver-vite: dynamic and experimental analysis for inchwormlike biomimetic robots. *IEEE Robotics & Automation Magazine.*, 10(4): 53- 57, 2003.
- [31] K.D. Kotay and D.L. Rus. Navigating 3D steel web structures with an inchworm robot. In the *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems.*, 1: 368-375, 1996.
- [32] M. Nilsson, Cranking along: slip-free locomotion by snake robots without torsion-free joints. In the *Proc. of the 1998 IEEE Intl. Symposium on Intelligent Control (ISIC)*:239-242, 1998.
- [33] Carnegie Mellon Biorobotics Laboratory. www.snakerobot.com
- [34] C.S. Merino and S. Tosunoglu. Design of a Crawling Gait for a Modular Robot. In the *Proc. of The 17th Florida Conf. on Recent Advances in Robotics, FCRAR 2004*, University of Central Florida, Orlando, Florida, May 6-7, 2004.

- [35] L. Chen, Y. Wang, S. Ma, and B. Li. Analysis of traveling wave locomotion of snake robot. In the *Proc. of IEEE Intl. Conf. on Robotics, Intelligent Systems and Signal Processing*, 1: 365- 369, 2003.
- [36] G. Poi, C. Scarabeo, and B. Allotta. Traveling wave locomotion hyper-redundant mobile robot. In the *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1: 418-423, 1998.
- [37] M. Zefran, V. Kumar, and X. Yun. Optimal trajectories and force distribution for cooperating arms. In the *Proc. of the IEEE International Conference on Robotics and Automation*, 1: 874, 1994.
- [38] E. Gilbert and D. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Journal of Robotics and Automation*, 1(1): 21- 30. 1985.
- [39] Sahar, G. and J. Hollerbach. Planning a minimum-time trajectories for robot arms. In the *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 2: 751-758, 1985.
- [40] S.R. Malladi, M.C. Mulder, K.P. Valavanis, and Y. Zhang. Behavior of a minimum-effort control algorithm for a multi-jointed robotic arm. In the *Proc. of the IEEE Intl. Symposium on Intelligent Control*:34-41, 1992.
- [41] J.B. Martin and J.E. Bobrow. Minimum-effort motions for open-chain manipulators with task-dependant end-effector constraints. *Intl. Journal of Robotics Research*, 18(2): 213-224, 1999.
- [42] M.E. Mortenson. *Geometric Modeling*, 2nd ed. John Wiley & Sons, 1997.

- [43] S-H.L Lee, J. Kim, F.C. Park, M. Kim, and J.E. Bobrow. Newton-Type Algorithms for Dynamics-Based Robot Movement Optimization. *IEEE Transactions on Robotics*, 21(4): 657-667, 2005.
- [44] J.J. Craig. *Introduction to Robotics: Mechanics and Control*, 3rd Ed. Pearson Education, 2005.
- [45] L.-W. Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, 1999.
- [46] G.V. Reklaitis, A. Ravindran, K.M. Ragsdell. *Engineering Optimization: Methods and Applications*. John Wiley and Sons, 1983.
- [47] Nilsson, M. Snake robot-free climbing. *IEEE Control Systems Magazine*, 18(1): 21-26, 1998.
- [48] H.A. Bruck, G. Fowler, S.K. Gupta, and T.M. Valentine. Using Geometric Complexity to Enhance the Interfacial Strength of Heterogeneous Structures Fabricated in a Multi-stage, Multi-piece Molding Process. *Experimental Mechanics*, 44(3): 261-271, 2004.
- [49] X. Li and S.K. Gupta. Geometric algorithms for automated design of rotary-platen multi-shot molds. *Computer Aided Design*, 36(12):1171-1187, 2004.
- [50] R.M. Gouker, S.K. Gupta, H.A. Bruck, and T. Holzschuh. Manufacturing of multi-material compliant mechanisms using multi-material molding. Accepted for publication in the *International Journal of Advanced Manufacturing Technology*.
- [51] A.K. Priyadarshi, S.K. Gupta, R. Gouker, F. Krebs, M. Shroeder, and S. Warth. Manufacturing Multi-Material Articulated Plastic Products Using

In-Mold Assembly. Accepted for publication in the *International Journal of Advanced Manufacturing Technology*.

- [52] G. T. Fowler, *Cost and Performance Evaluation Models for Comparing Multi-Shot and Traditional Injection Molding*. M.S. Thesis, University of Maryland, College Park, MD, 2004.
- [53] L.S. Gyger Jr., B.W. Spranklin, S.K. Gupta, and H.A. Bruck. Bio-inspired, Modular, and Multifunctional Thermal and Impact Protected (TIPed) Embedded Sensing Controls Actuation Power Element (ESCAPE) Structures. In the Proc. of the *SEM Annual Conference & Exposition*. St. Louis, Missouri USA. June 4 - 7, 2006.
- [54] M. Keennon and J. Grasmeyer. Development of the Black Widow and Microbat MAVs and a vision of the future of MAV Design. In the *Proc. of the AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Years*; Dayton, OH; July 14-17, 2003.