

Design of a Biomimetic Mechanical Leg and Accompanying Sensor System for Terrain Detection

A Major Qualifying Project submitted to the faculty of Worcester Polytechnic
Institute in partial fulfillment of the requirements for the degree of
Bachelor of Science

Submitted by:

Molly Homchenko

Hannah Bond

Submitted to:

Prof. Yiming Rong

Prof. Ximing Huang

In cooperation with:

Prof. Wenchuan Jia, Shanghai University

December 26, 2014

Abstract

Autonomous robots are useful in a wide range of applications. However, finding a balance between speed and stability in an autonomous robot can be difficult. The goal of this project was to design a biomimetically-inspired robotic leg and accompanying sensor system for detecting terrain; the mechanical leg and sensor system designs in combination are intended to enable a quadruped robot to move quickly while maintaining its stability. In order to accomplish this goal, a leg was designed based on the leg of a cheetah and the team performed a variety of mechanical analyses on it. Additionally, the output from a force sensor landing on hard and muddy surfaces was collected and algorithms for determining which of the two surfaces the robot was walking on were developed.

Acknowledgements

At the culmination of our project, we would like to thank the many people who provided us with the resources and assistance necessary to make this project possible. Firstly, we would like to thank Shanghai University and Worcester Polytechnic Institute for creating the opportunity to be involved in this project. We would then like to thank Professor Wenchuan Jia from Shanghai University for creating the project proposal and working with us extensively in order to refine that proposal further. Lastly, we would like to thank Professor Yiming Rong and Professor Ximing Huang from Worcester Polytechnic Institute for the role they played in their advisory positions.

Executive Summary

As the field of autonomous robotic navigation expands, there are numerous problems yet to be solved – one of which is developing a robot that can quickly and safely navigate varying terrains. Traditionally, wheeled robots are capable of travelling at high speeds, while legged robots are capable of maintaining stability and traction while traversing uneven or slippery terrains, however, a robot that is capable of both quickly and reliably crossing difficult terrains would be of use in many applications, especially in the research community.

Our project involves designing a biomimetic robotic leg capable of quick locomotion. The leg is designed to be used on a four-legged robot and takes inspiration from the biological structure of a cheetah. The leg is supported by an accompanying sensor system used to detect the hardness of the surface the robot is walking on, which will allow the robot to better navigate uneven terrain while maintaining its stability. The leg was designed and analyzed using Siemens NX, Ansys, and Adams. The sensor system used a force sensing resistor and Arduino microprocessor for testing and preliminary algorithm design.

The mechanical design was completed and passed all tests, which were performed using computer models. The algorithm for the sensor system was designed on the microprocessor and was subsequently tested. The sensor system design began to be transferred over to an FPGA implementation – ideal for the on-leg computation due to speed and ability to perform simultaneous tasks. We make recommendations which involve completing a physical prototype of the leg and completing the transferring of the sensor system design over to the FPGA. Testing of the systems together could then be completed. Overall, the design of the mechanical design of the leg structure combined with the sensor system provide a good basis for future development.

Authorship

Chapter	Name
Executive Summary	Molly Homchenko
1.0 Introduction	Hannah Bond
2.0 Background	Hannah Bond
3.0 Methodology	Molly Homchenko
4.1 Biomimetic Leg Design	Molly Homchenko
4.2.1 Data Acquisition	Hannah Bond
4.2.2 Arduino Algorithm	Hannah Bond
4.2.3 FPGA Implementation	Molly Homchenko
4.3 System Analysis	Hannah Bond
5.1 Specially Designed Force Sensor	Molly Homchenko
5.2 Transfer Microprocessor Designed Algorithms to FPGA	Molly Homchenko
5.3 Improved Carrying Capacity or Applied Load of the Mechanical Design	Molly Homchenko
5.4 Create a Prototype	Hannah Bond
5.5 Integrate the Algorithm and the Controls	Hannah Bond
6.0 Conclusion	Hannah Bond

Table of Contents

Abstract.....	i
Acknowledgements.....	ii
Executive Summary.....	iii
Authorship	iv
Acronyms	x
1.0 Introduction	1
2.0 Background	3
2.1 Overview of Autonomous Robotics	3
2.1.1 History of Autonomous Robotics	3
2.1.2 Current Applications of Autonomous Robotics	4
2.2 Robotic Locomotion Systems.....	5
2.2.1 Non-legged Systems.....	5
2.2.2 Legged Systems.....	6
2.2.3 Biomimetics.....	6
2.3 Implementing Legged Locomotion	8
2.3.1 Robotic Gaits	8
2.3.2 Sensor Systems	9
3.0 Methodology.....	12
3.1 Mechanical System	12
3.1.1 Background Research.....	12
3.1.2 Design and Simulation of the Biomimetic Robotic Leg.....	13
3.2 Sensor system	15
3.2.1 Hardness Surface Attribute and Force Sensor Detection Method Decision Process	15
3.2.2 Theory of Force Sensor for Detecting Hardness of Surface.....	17
3.2.3 Force Sensor Testing.....	19
4.0 Findings and Discussion	21
4.1 Biomimetic leg design.....	21
4.2 Sensor System Design	26
4.2.1 Data Acquisition	26
4.2.2 Arduino Algorithm	30
4.2.3 FPGA Implementation.....	32
4.3 System Analysis.....	37

5.0 Recommendations	38
5.1 Specially Designed Force Sensor	38
5.2 Transfer Microprocessor Designed Algorithms to FPGA	38
5.3 Improved Carrying Capacity or Applied Load of the Mechanical Design.....	38
5.4 Create a Prototype.....	39
5.5 Integrate the Algorithm and the Controls	39
6.0 Conclusion.....	41
Works Cited.....	42

Table of Figures

Figure 1: Elsie, one of the Tortoise robots.....	4
Figure 2: Baby Elephant, a biomimetically inspired, load carrying robot.....	7
Figure 3: Demonstration of a trot gait.....	9
Figure 4: Preliminary leg designs (Cheetah 1, Cheetah 2, Dog).....	13
Figure 5: Exploded view of FSR.....	18
Figure 6: Force sensor theory (m - mass, g- gravity).....	18
Figure 7: Testing setup.....	19
Figure 8: FSR voltage divider resistor selection (Interlink Electronics (b)).....	20
Figure 9: Voltage divider circuit.....	20
Figure 10: Final mechanical leg design.....	21
Figure 11: Final motor selection HT05001.....	21
Figure 12: Foot design of the leg system.....	22
Figure 13: Joint torque calculations.....	22
Figure 14: Ansys rigidity analysis - link 1.....	23
Figure 15: Ansys rigidity analysis - link 2.....	23
Figure 16: Ansys rigidity analysis - link 3.....	24
Figure 17: Functional block diagram.....	25
Figure 18: Simulation constraints.....	25
Figure 19: Quadruped representation of leg design.....	26
Figure 20: Raw data acquired during testing on hard surface.....	27
Figure 21: Raw data collected from testing on a soft surface.....	28
Figure 22: Raw data collected from testing on a soft surface.....	28
Figure 23: Noisy data acquired while testing on a hard surface.....	29
Figure 24: Representative example of data acquired on hard surface.....	29
Figure 25: Representative example of data acquired on soft surface.....	29
Figure 26: Flowchart for the hardness vs softness detection algorithm.....	30
Figure 27: Left-hand side: Data before smoothing. Right-hand side: Data after smoothing.....	31
Figure 28: Demonstration of slope calculation portion of algorithm.....	32
Figure 29: High-level schematic of FPGA system.....	33
Figure 30: Block diagram of Verilog force sensor implementation.....	33
Figure 31: “writeByte” module block diagram.....	34

Figure 32: "readByte" module block diagram 35
Figure 33: "getADCreading" module block diagram 36

Table of Tables

Table 1: Animals researched.....	12
Table 2: Animals listed by number of joints in back legs.....	13
Table 3: Criteria for selection of leg.....	14
Table 4: Leg selection decision matrix.....	15
Table 5: Detection method decision matrix.....	17
Table 6: ADC register configurations.....	36

Acronyms

ADC – Analog to digital converter

AI – Artificial intelligence

FPGA – Field programmable gate array

FSR – Force sensitive resistor

LSB – Least significant bit

MSB – Most significant bit

1.0 Introduction

As the field of autonomous robotic navigation expands, there are numerous problems yet to be solved. One such problem is that of creating a robot that can quickly and safely navigate varying terrains. Traditionally, wheeled robots are capable of travelling at high speeds, while legged robots are capable of maintaining stability and traction while traversing uneven or slippery terrains (Gonzalez de Santos, Garcia, & Estremera, 2006). A robot that is capable of both quickly and reliably crossing difficult terrains would be of use in many applications, especially in the research community (Na, Choi, & Kong, 2015)

There are numerous ways to approach this challenge, one of which is to look for inspiration in the field of biomimetics. Biomimetics is the practice of drawing inspiration for human inventions from mechanisms found in nature. By using biomimetics as a basis for designing a robotic leg, we can take advantage of mechanisms that have naturally evolved in animals over thousands of years (Scobey-Thal, 2014).

For our project, we worked under a professor of mechatronics in order to design a biomimetic robotic leg capable of quick locomotion. The leg is designed to be used on a four-legged robot, also known as a quadruped. It is based off of a cheetah and supported by an accompanying sensor system used to detect the hardness of the surface the robot is walking on. This sensor system will allow the robot to better navigate uneven terrain while maintaining its stability.

In order to accomplish this project, we first researched a variety of animal legs to determine which animal would make a good model for the leg. Simultaneously, we researched existing sensor systems for detecting a variety of ground conditions. Next, we designed the chosen leg structure in Siemens NX, performed rigidity analysis on it in Ansys, and performed reachable workspace analysis on it with Adams. We also created an algorithm for determining the hardness of a surface on an Arduino board, and later began to transfer that design to a Field-Programmable Gate Array (FPGA). Lastly, we performed testing on these designs.

The results of this project will form a basis for further research into the field of autonomous robotic locomotion. By providing a mechanical leg design and a prototype sensor system, a physical implementation of the leg and its accompanying sensor system could be built and further tested and refined.

In Chapter 2 we discuss the background of our project, including the history of autonomous robotics and information on currently-used robotic locomotion systems. In Chapter 3, we discuss the methodology that we used to complete this project, showing why certain choices were made and detailing our research. In Chapter 4, we present and discuss our findings for both the leg design and sensor system. In Chapter 5, we provide our recommendations for future researchers continuing work on this project. Lastly, in Chapter 6 we summarize our project and offer a conclusion.

2.0 Background

Autonomous robots are used in fields ranging from military applications to the environmental monitoring. Capable of feats such as superhuman precision, strength, or endurance, and able to operate in unsafe areas without the concern of loss of life, these robots are becoming increasingly popular (Fahimi, 2008). As this occurs, engineering improved systems for robotic locomotion becomes a more relevant and important challenge. Not only must the physical locomotive system be designed, but the locomotive system must also be equipped to navigate on its own and adapt to any challenging terrains encountered along the way. In Chapter 2.1, we discuss the broad field of autonomous robotics, including its history and applications. In Chapter 2.2, we discuss the physical locomotive systems these robots use in further detail, with a particular emphasis on legged locomotion. In Chapter 2.3, we discuss robotic gaits and the state-of-the-art sensor systems currently being developed and used to aid autonomous robots in improving their gaits.

2.1 Overview of Autonomous Robotics

By definition, an autonomous robot is “a robot which can, by sensing information about its environment, work for a prolonged period of time, learn and adapt without human or other external intervention” (Atkins & Escudier, 2013). The field of autonomous robotics got its start in the mid-20th century, and it has been progressing at a rapid rate ever since.

2.1.1 History of Autonomous Robotics

Autonomous robotics’ history begins with the so-called tortoise robots, Elmer (ELECTroMEchanical Robot) and Elsie (Electro Light Sensitive with Internal and External stability). These robots, engineered by cybernetics researcher Willam Grey Walter in 1949, were three-wheeled robots similar in shape to a tortoise. The tortoise robots were capable of autonomously returning to their charging station when they detected that their battery was running low, a feat that was revolutionary at the time (Bladin, 2005). Later iterations of the robots also displayed the ability to learn and behavior that was seemingly random, similar to the behavior of actual animals (Levy, 2005).

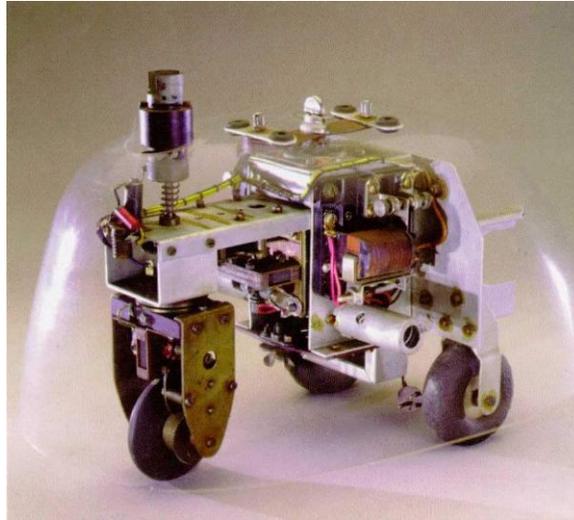


Figure 1: Elsie, one of the Tortoise robots.

The 1950s brought with them an academic interest in artificial intelligence (AI) after Alan Turing's proposal of the Turing Test, a test of how distinguishable a computer's behavior is from a human's (Levy, 2005). This increased interest in AI greatly helped to advance the field of autonomous robotics, as the ability to learn and adapt without human intervention is one of the core engineering challenges inherent in creating autonomous robots.

As time passed and the fields of electrical engineering, mechanical engineering, and computer science advanced, the field of autonomous robotics advanced along with them. The second half of the 20th century saw autonomous robots engineered to serve useful purposes in addition to the research purposes that were so common in the earlier part of the century.

2.1.2 Current Applications of Autonomous Robotics

Autonomous robots are currently used for a wide range of applications, including military, search and rescue, and entertainment purposes. In this section, we provide a brief overview of some of the state-of-the-art robots and research being used in each of these applications.

Autonomous robots are of particular interest to the military because they can perform many of the same functions as human soldiers, but without the need to put soldiers' lives in danger (Fahimi, 2009). Robots can also be used in order to perform tasks more cheaply than humans could perform them. One of the most commonly known autonomous military robots in the United States is Predator A, an Unmanned Aircraft System. Predator A is capable of reconnaissance and carrying anti-tank missiles. It

can also fly in autopilot, eliminating the need for a human being to put their lives at risk doing reconnaissance and eliminating the cost of employing a pilot (Hindle, 2013).

Autonomous robots are a popular topic in search and rescue for much of the same reasons that they are popular with the military: robots can enter spaces dangerous to humans and perform hazardous jobs without risking injury to a person. A great deal of research has been performed on how to best use robots to autonomously or semi-autonomously search out survivors of natural and manmade disasters. A typical example of such research was presented at the IEEE International Symposium on Safety, Security, and Rescue Robotics in 2013. The research focused on how to best visually seek out survivors by using camera and image processing. The research also discusses how to set up control of a group of autonomous robots to get the best results while minimizing cost (Tamassia, Signorini, et al, 2013).

Lastly, autonomous robots are used for entertainment purposes. Although this is more of a niche field than the previously mentioned applications, it has a large following. Robots are made to dance, play soccer, or even compete in Robot Olympics, all without the aid of a human controlling their actions (Levy, 2005).

2.2 Robotic Locomotion Systems

In order to perform their given tasks, autonomous robots must be able to move on their own. Broadly speaking, the physical structures used for robotic locomotion can be broken down into two types: non-legged and legged. There is a wide variety of locomotive systems within each type, each with their own benefits and drawbacks. Wheeled systems tend to be easier to implement and enable a faster robot, whereas legged systems tend to be more complicated but able to better navigate difficult terrain (Gonzalez de Santos et al., 2006).

2.2.1 Non-legged Systems

Non-legged locomotive systems generally utilize either a wheeled, treaded, or slithering mechanism in order to propel the robot, with wheeled and treaded systems being the most popular. While there was a lot of research into legged systems during the early days of modern robotics, the vast majority of robots that were actually built used a wheeled or treaded system. This is due to its ease of implementation – instead of having to coordinate the motion patterns and actuation of multiple joints

on each leg, engineers only had to concern themselves with driving a motor on each wheel (Gonzalez de Santos, et al., 2006).

Today, the majority of robots still use wheeled and treaded systems. When the robot will not need to navigate uneven or slippery terrain, implementing a wheeled or treaded system takes less time and resources, making it an attractive option for many purposes (Thueer & Siegwart, 2010). Additionally, wheeled robots tend to be able to move at higher velocities than legged robots (Sprowitz, Tuleu, Vespigani, Ajallooeian, Badri, & Ijspeert, 2013)

Slithering robots have an advantage over wheeled robots when it comes to slippery terrain. Since the entire length of the robot is in contact with the ground, it is able to generate more traction than a wheeled robot would. However, this advantage comes at a cost. Algorithms to make a robot slither are much more time consuming to create than algorithms for controlling wheeled robots (Ghosh & Majumder, 2011).

2.2.2 Legged Systems

Ever since the beginning of the field of robotics, there has been interest in creating robots that can walk. Much of this interest is due to the desire to create humanoid robots (Gonzalez de Santos et al., 2006). However, the reasons for wanting to create legged robots extend beyond the desire to create robots that mimic human beings. Legged robots are capable of feats that non-legged robots have difficulty with or simply cannot perform. For example, legged robots can navigate terrain that is uneven or muddy, something wheeled robots traditionally struggle with.

When designing a legged robot, the most important consideration is the number of legs the robot will have. Common choices include two (biped), four (quadruped), and six (hexapod). As the number of legs increases, so does the stability of the robot. This is because it has more points of contact with the ground and can distribute its weight more evenly. However, additional legs also lead to additional complexities in the robot's control algorithms.

2.2.3 Biomimetics

Despite its difficulties in implementation, there has been an increased focus in legged locomotion recently due to academic interest in the field of biomimetics. Biomimetics is the imitation of phenomena occurring in nature, adapted in order to be useful for humans. A famous example of

biomimetics is the invention of Velcro; plant burrs' mechanism for latching onto passing organisms inspired the creation of the popular fastening mechanism, which mimics the burrs' miniscule hook-and-latch structure (Scobey-Thal, 2014).

Biomimetics can easily be applied to robotic locomotion. Animals have evolved over thousands of years, resulting in bodies that are uniquely adapted to suit the animal's needs. In terms of animal locomotion, this has led to leg mechanisms that allow animals to run quickly, such as the cheetah's, or leg structures that allow a great deal of stability and maneuverability, such as the spider's (Liu & Sun, 2011). These pre-existing leg designs can be put to use when designing legged robots. Which animal the leg is mimicked after depends on the intended purpose of the leg.

There are many cases where biomimetics has been used in order to design a robot's locomotive system, however, these robots are typically quite complicated and only implemented in academic research. Baby Elephant is one such robot. As the name suggests, Baby Elephant was modeled after an elephant. This robot was developed at Shanghai Jiao Tong University, and was created in order to reliably carry heavy loads, even across rough and uneven terrain. Baby Elephant utilizes four hydraulically-actuated legs and has springs in its legs in order to store and release energy. This mechanism is useful in achieving the robot's given purpose of load carrying because it allows Baby Elephant to possess more power without the need for adding additional motors or actuators, which are often heavy and would reduce the amount of load Baby Elephant could carry (Chen, Gao, Qi, & Zhao, 2013).

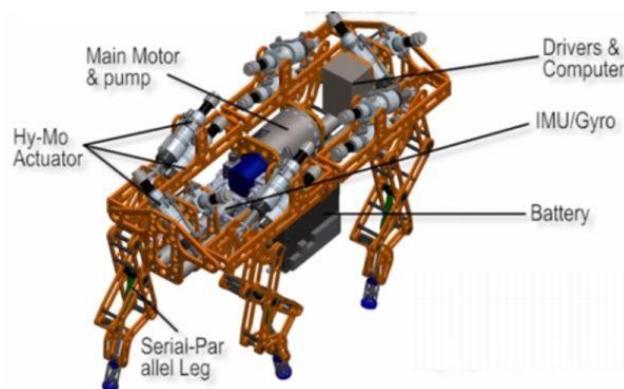


Figure 2: Baby Elephant, a biomimetically inspired, load carrying robot (Chen et al., 2013)

Another state of the art, biomimetically inspired robot is the Swiss Institute of Bioengineering's Cheetah-cub. Designed after a cheetah, but built to be about the size of a housecat, this robot serves a very different purpose than that of Baby Elephant. Cheetah-cub was designed for speed; it's the fastest four-legged robot of its weight and size. It was designed to move quickly over fairly flat surfaces, but does display some self-stabilizing behavior. However, despite the difference in purpose from Baby Elephant, some design elements remain the same for both robots, such as the use of springs to store and release energy (Sprowitz et al., 2013).

Although most biomimetically-inspired robots have only made it as far as prototypes, one example of a robot that has made it further is Big Dog. Created by Boston Dynamics for military purposes, the robot was inspired by the body of a dog. Similarly to Baby Elephant, Big Dog was designed as a load carrying robot that can navigate uneven terrain. However, in Big Dog's design, there is a much more significant emphasis on speed. Since the robot is designed to keep up with soldiers, it needs to be agile despite its strength and bulk (Barry, 2007). As this robot was designed for the military, there is not much information available about its design, unlike robots that are designed for research purposes.

2.3 Implementing Legged Locomotion

Even after the physical locomotion system for a robot is created, a challenge still lies ahead: how to use that system in order to move the robot. There are two distinct elements to this challenge. The first element is determining the robot's gait, which involves figuring out the pattern in which the robot's legs and feet should move. The second element is implementing a system so that the robot can adapt as it encounters varying conditions.

2.3.1 Robotic Gaits

There are a wide variety of gaits that a legged robot can implement, many of which are based on gaits found in various animals. The first consideration in regard to gait selection is the number of legs the robot possesses. As mentioned in Chapter 2.2.2, common legged robots include bipeds, quadrupeds, and hexapods. All these different types of robots have different gaits that they can implement, but we will focus specifically on the gaits employed by quadrupeds.

The creep is one of the slowest gaits a quadruped can perform. Commonly observed in animals attempting to maintain balance in precarious situations, the creep offers robots and animals alike a way to remain stable on highly irregular terrain. During the creep, a quadruped uses all four legs to propel

itself forward. It then lifts one leg and moves it forward. This process is repeated for each leg, and leads to the quadruped always having at least three legs on the ground at all times (Gonzalez de Santos et al., 2006).

A quicker gait that a quadruped can employ is the diagonal trot. Again, this gait is commonly seen in many different animals. During the diagonal trot, the quadruped lifts two feet that are diagonal from each other at a time and moves them forward. The quadruped then lifts the other two feet and moves them forward. This process repeats. During the trot, the quadruped always has two feet on the ground. This leads to less stability than the creep, but a greatly increased speed (Gonzalez de Santos et al., 2006).

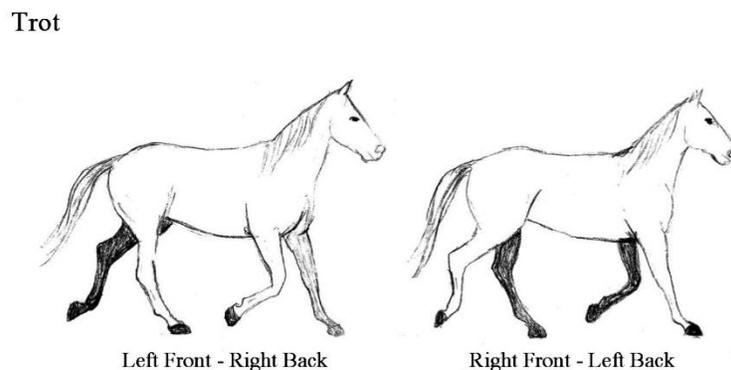


Figure 3: Demonstration of a trot gait

Lastly, one of the fastest gaits a quadruped can use is the bound. Many animals run using the bound gait, where the two front legs move forward in unison, then the two back legs move together in unison. During this gait, the hind legs are used for power and propulsion, while the front legs are used primarily for balancing. Like the trot, this leads to the robot always having two points of contact with the ground (Formal'sky, Chevallereau, & Perrin, 2000).

2.3.2 Sensor Systems

The use of sensor systems in autonomous robots allows the robot to correct its gait in such a manner that it remains stable and able to navigate. These systems can range from being relatively simple to extremely complicated, and are meant to sense everything from the approach of a cliff, to the roughness of the ground, to the permeability of the surface the robot is walking on.

An example of a sensor system with a very straight forward purpose is a cliff detector. This system is implemented by multiple autonomous robots, including iRobot's Roomba, a vacuuming robot

that automatically cleans the owner's entire floor (iRobot, 2004). In the Roomba, the cliff detection system consists of a transmitter and receiver for an infrared signal. Essentially, an infrared signal is transmitted towards the floor, and if it is received back, the Roomba knows it is not on a cliff edge. If the signal is not received back, the Roomba has detected that it is on the edge of a cliff and will not travel further in that direction (iRobot, 2014). Many other iRobot robots also possess the same cliff detection mechanism (Hall, 2009).

Sensing the roughness of the ground beneath the robot's feet involves a more complicated sensor system than cliff detection. This variety of sensor system is often implemented by detecting the traction between the robot's foot and the ground. Pre-developed tactile sensing systems can be bought and utilized in order to receive data about the traction between the robot foot and surface. Research at the University of Seoul worked on treating the data returned from such a system as the input to image processing to try to determine how much slippage there was as a robotic hand brushed across a surface. This information could then be used to see how much traction there was across a given surface (Ho, Nagatani, Noda, & Hirai, 2012). A study to try to find similar information using a different tactic was carried out at the University of Pennsylvania. In this study, researchers built a robotic foot with a compressible bottom, and dragged it across various surfaces. By seeing how much the foot compressed as it was dragged, the researchers were able to infer information about the traction between the surface and the foot, thereby giving them valuable data about the surface's roughness (Sinha, Xu, Bajcsy, & Paul, 1999). One final detection method for surface roughness uses infrared sensing. Smooth surfaces reflect more light directly back in the direction it propagated from than rough surfaces, which disperse the infrared signal on contact. By detecting how much infrared light is returned, researchers at Aveiro University were able to distinguish rough surfaces from smooth surfaces (Lomba, Valadas, & Duarte, 1998).

Lastly, sensor systems to determine the permeability and relative hardness of a surface have also been researched. Detecting these attributes is important so that the robot does not sink into a surface as it tries to walk over it. The study at the University of Pennsylvania that detected surface roughness using a compliant foot also used that same mechanism to detect the relative hardness of a surface and to detect if the robotic foot was sinking far enough into a surface to make the robot unstable (Sinha et al., 1999). Another method of detection was developed at the Massachusetts Institute of Technology. Researchers wished to know how far into sand the wheel of a robot meant for planetary exploration would sink. The resulting research involved marking the bottom of the wheel with

an easily-distinguishable color, then performing image processing to determine how much of the bottom of the wheel was visible versus how much was buried under sand (Iagnemma, Shinwoo, Shibly, & Dubowsky, 2004).

3.0 Methodology

The goal of this project was to advance the field of autonomous robotic locomotion through the use of biomimetics and smart sensing. This was accomplished by dividing the work into two main categories – one for the mechanical design and one for the sensor system. The objectives for the mechanical system were as follows: (1) Determine a biological model for the leg; (2) Design a mechanical leg; (3) Perform testing and analysis on leg performance. The objectives for the sensor system were as follows: (1) Determine a terrain attribute to detect and method for detection; (2) Acquire data to assist algorithm development; (3) Design algorithm. The methodology chapter is divided into these two categories.

3.1 Mechanical System

Initial research was conducted into different types of animal legs in order to help the team draw inspiration from nature. Ultimately, two mechanical prototypes of a cheetah leg and one prototype of a dog leg were designed; one of the cheetah legs was chosen as the final design and further development and testing were performed on that design.

3.1.1 Background Research

The structure of 12 animal legs were examined for the purposes of drawing inspiration from naturally evolved organisms. Quadrupeds of a wide range of different sizes and movement types were looked at. Table 1 below shows the animals researched by the team.

Squirrel	Cat	Horse	Rabbit
Mouse	Rhino	Bear	Wolf
Deer	Tiger	Dog	Cheetah

Table 1: Animals researched

Structurally, there were many similarities in the joints of these animal legs. It was found that all of the quadrupeds had the same number of joints in the front legs (one), but the animals differed in the number of joints in the back legs. They were evenly split between one and two joints, and the size of the animal did not seem to be a factor in the number of joints. Table 2 below shows the quadrupeds listed by number of joints in their hind legs.

One	Two
Rhino	Horse
Mouse	Cat
Rabbit	Deer
Bear	Tiger
Squirrel	Wolf
Dog	Cheetah

Table 2: Animals listed by number of joints in back legs

The team looked at the speed, agility, and stability of the animals as well. It was found that the cheetah was the fastest animal. Dogs were also found to have relatively high speed when compared to the cheetah given their smaller size; however the foot and lower leg bone have a small range of motion so the dog is considered to be immobile. The cheetah has one degree of freedom (DOF) more than the dog and this was thought to potentially aid in its agility and stability while moving at fast speeds.

3.1.2 Design and Simulation of the Biomimetic Robotic Leg

Three programs were used for designing and testing the mechanical structure of the leg. Siemens NX was used for modeling the leg. After the model was created, Adams was used for the motion simulation, and Ansys was used to perform rigidity analysis. Initially, three preliminary designs were created. Two took inspiration from the cheetah and one drew inspiration from the dog. These designs can be seen in Figure 4 below.



Figure 4: Preliminary leg designs (Cheetah 1, Cheetah 2, Dog)

Both cheetah designs used a motor drive while the dog leg design used hydraulics. The first cheetah-inspired leg weighs 1.35kg and includes shock absorber legs. The second weighs 0.85kg and creates a parallelogram structure for stability and strength. The hydraulic-driven dog-inspired leg weighs in at 2.56kg, the heaviest of the three designs.

Different characteristics of the legs were used to determine the best leg design of the three. These characteristics and the weights they were given are shown below in Table 3. Note that the maximum score is five and the minimum score is two. The decision matrix used to determine the leg design to develop further into the final design can be seen in Table 4 below.

Structure	0.25
Weight	0.15
Reliability	0.15
Costs	0.15
Manufacture	0.15
Installation	0.10
Accuracy	0.05

Table 3: Criteria for selection of leg

	Cheetah 1	Cheetah 2	Dog
Weight	5	4	2
Reliability	2	4	4
Installation	3	2	2
Structure	3	4	3
Accuracy	2	4	2
Manufacture	4	3	3
Costs	3	4	2
Total	3.25	3.65	2.7

Table 4: Leg selection decision matrix

From these preliminary designs, the second cheetah design was chosen for further development and testing. Reachable workspace and rigidity analysis were completed on this design, the results of which will be discussed in Chapter 4.

3.2 Sensor system

Initial research was conducted into current terrain detection systems for quadruped and other robotic systems. The hardness of the surface was chosen as the terrain attribute that the team would attempt to detect. Research was conducted to determine current methods of sensing the hardness of surfaces in robotic systems, and a force sensor detection method was chosen. A simple microcontroller and force sensor setup was used for the testing of the detection method.

3.2.1 Hardness Surface Attribute and Force Sensor Detection Method Decision Process

As mentioned, initial research was conducted into current terrain detection systems for quadruped and other robotic systems and hardness of the surface was chosen as the terrain attribute that the team would detect. The choice of a terrain attribute to detect was based on the prior existence of a minimal amount of systems for detection, the team's confidence in completing the chosen task, and the team's advisor's preference. The team then researched current methods of detecting the hardness of a surface and combined that with team brainstorming of unique force sensing ideas. Five different methods were considered for this project. The five methods are defined as follows:

1. Force: A force sensor placed on the bottom of a foot is used to detect the normal force provided from the movement of the foot through a soft surface. The force as the foot moves through a soft surface is greater than zero, but less than the normal force that is present when the leg is fully supported on the surface.
2. Speed: As the leg is moving, the system knows the speed that the leg is instructed to move at. The actual speed that the leg is moving is determined with use of sensors. If the leg is moving through a soft surface before reaching a stable standing position, the leg moves slower than instructed when encountering the resistance of a soft surface.

3. Image processing: The bottom of the leg is marked and a camera is mounted such that it can view the markings. Image processing is performed to see how many markings are visible. If fewer markings are visible, the bottom of the leg may have sunken into a soft surface.
4. Distance: Detection of the distance to a surface is accomplished by using sensors such as sonar. From this information, the leg is told to move downwards this distance. A sensor such as a force sensor is used to determine if the foot is on a hard surface supporting the weight of the robot or if the foot is still in the air or not yet fully supporting the weight. If the foot is not yet supporting the full weight of the robot after the initial distance is traversed, the leg may have moved into a soft surface. Distance is sensed and the distance between where the leg thought it would hit the hard surface and where it actually does could be used to determine the level of hardness of the surface.
5. Deformation: This method involves having the bottom of the foot designed to be deformable in a predictable manner such as with a spring system. Differences would be detectable between how such a system reacts to a hard versus a soft surface.

After the five potential methods were identified, criteria for evaluating each one were determined. Time involvement, accuracy, processing speed, the team's familiarity, the theory's plausibility, and simplicity were each weighted from one to five, with one being less important and five being very important. Time involvement was considered more important due to the short timeframe that the team had to complete this project. The team considered the accuracy of the sensors that would need to be used to accomplish each method, the signal processing, and algorithms needed to deliver the determination of level of hardness of the surface. The team's familiarity with the theory of the idea, sensors to be used, and implementation of each idea were considered into the decision to ensure that the project had the best chance of achieving complete results. The theory's plausibility was defined as, based upon current research and the team's knowledge, how likely the system was to be able to accurately detect the hardness of the surface. It is important to note that simplicity was not just considered for the sake of being simple, but to reduce sources of error. Complex sensor systems involving multiple sensors add more sources of error, especially considering the diverse environments that quadruped robots could be sent into. Simplicity tied into the idea of a robust system which could

sustain itself in diverse environments. Each of the above sensing methods were scored on each of these criteria; the decision matrix can be seen in Table 5 below.

		Criteria						
		Time involvement	Accuracy	Processing speed	Team's familiarity	Theory's plausibility	Simplicity	Score
	Weight	4	3	2	5	4	3	Max: 105
Detection Methods	Force	3	4	5	3	4	5	80
	Speed	3	2	5	4	2	4	68
	Image processing	1	4	1	1	3	3	44
	Distance	3	3	4	4	3	4	73
	Deformation	1	5	3	1	5	2	56

Table 5: Detection method decision matrix

Based on this decision matrix, it was decided that the force sensor method would be employed for this project.

3.2.2 Theory of Force Sensor for Detecting Hardness of Surface

A force sensor such as the one chosen for this application operates very simply as a variable resistor. The specific Interlink force sensor resistor (FSR) used for the testing in this project is a single zone force sensing resistor which can be implemented in a design in a simple voltage dividing circuit. The voltage measured across the FSR goes up as the force applied to the sensor increases.

The FSR is a polymer thick film device consisting of four layers. The four layers are the adhesive top layer, the top substrate, a spacer adhesive, and a bottom substrate. The spacer adhesive is designed to create an air gap between the two substrate layers. The upper substrate layer consists of “two sets of interdigitated fingers” that are separate electrical entities. Each set connects to one trace on a tail. The other membrane is coated with FSR ink that, when pressed on, shorts the two traces together (Interlink Electronics (a)). These layers can be seen in Figure 5 below.

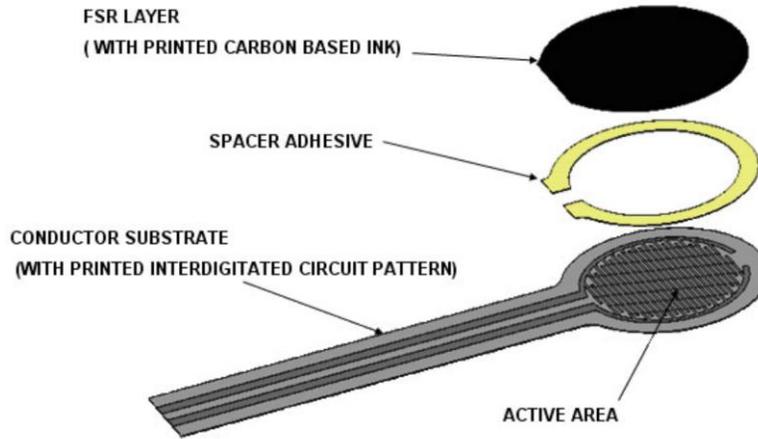


Figure 5: Exploded view of FSR (Interlink Electronics (a))

Due to the way these sensors are constructed as thin layered devices, there are certain issues which arise with the durability. If the layers separate at all, it can allow for outside contaminants to get within the layers preventing them from correctly making the contacts needed for an accurate reading. If a liquid such as water makes it in between the layers, it is possible for electrical contact to be made between the layers at all times preventing there from being any change in output of the device.

The theory behind the force sensor for detecting the hardness of a surface is simple. A force sensor placed on the bottom of the foot would read zero when the foot was not touching the ground. When the foot is down completely and the weight is distributed evenly, the normal force would be known. However, if for instance the foot was sinking into a soft surface, that surface would be supplying a normal force upwards which would not be the maximum force when the foot was all the way down but would be greater than zero. This is illustrated by the images below in Figure 6.

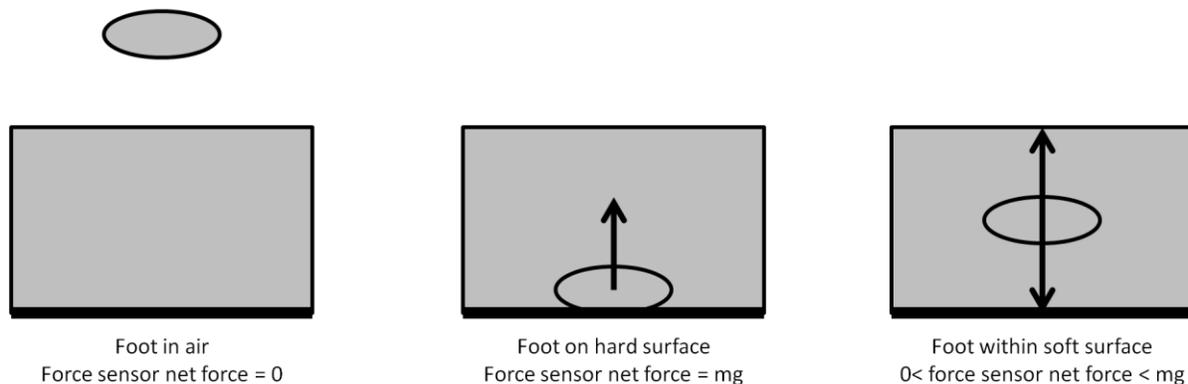


Figure 6: Force sensor theory (m - mass, g - gravity)

3.2.3 Force Sensor Testing

For the purposes of testing the force sensor in a timely manner, an Arduino microcontroller was used. The Arduino microcontroller allowed for quick implementation of the force sensor which could then be used to test the reaction of the sensor on different surfaces. The Arduino Mega microcontroller was programmed to read an analog input and output the data to the serial port. From this simple interface, data could then be read in from the force sensor and data could then be graphed and analyzed.

In order to test the force sensor on surfaces of various hardnesses, it was necessary to create a testing environment which would provide approximately equivalent “steps” in order to accurately compare the data from each surface. With limited resources, dropping a weight was determined to be the best way to test the force sensor with consistent “steps.” The test setup consisted of a weight attached to the force sensor. The weight contacted the entire surface of the force sensor, but did not overhang the surface of the sensor. The weight fit into a tube which would help guide the weight straight down as it was dropped onto various surfaces. The weight was dropped from a fixed height of 3.25 centimeters above the surface to be tested; this led to an impact velocity of 0.79 m/s. A schematic of the described test setup can be seen in Figure 7 below.

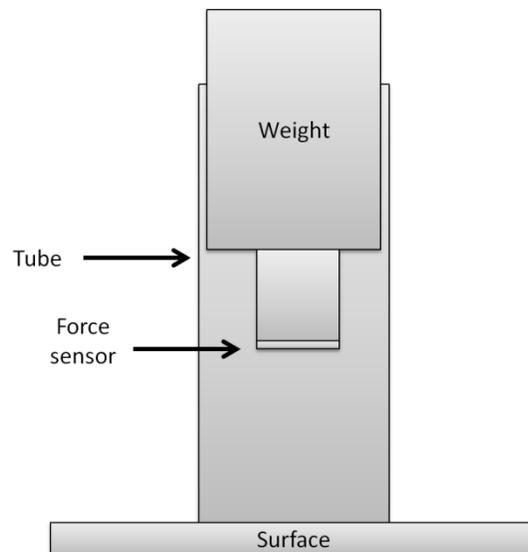


Figure 7: Testing setup

The force sensor was wired to the Arduino to capture the data. This was done through a simple voltage divider circuit. Per the force sensor data sheet, shown below in Figure 8, the resistor was chosen to be 3 kOhms in order to create the greatest resolution in the force sensor’s output while at the lowest end of the force that it could sense.

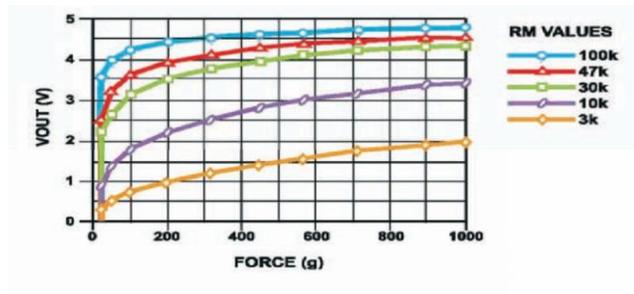


Figure 8: FSR voltage divider resistor selection (Interlink Electronics (b))

The voltage drop across the force sensor was read into the Arduino through an analog input pin. From the Arduino, data could be read on a computer through a serial port. This configuration can be seen in Figure 9 below.

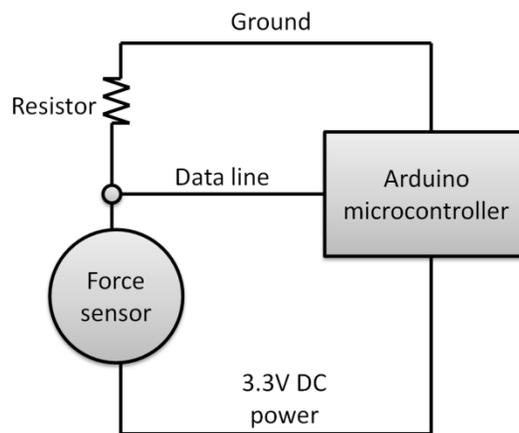


Figure 9: Voltage divider circuit

Two surfaces, hard tile and mud, were tested with this setup. The mud was a mixture of five parts dirt to one part water. The weight was dropped and data was recorded for each drop. The results of these tests will be discussed in Chapter 4.

4.0 Findings and Discussion

The mechanical leg design passed all tests and met the design criteria set forth by the team. The sensor system was tested with the microprocessor and the design began to be transferred to the FPGA implementation.

4.1 Biomimetic leg design

The final leg design can be seen in Figure 10 below. The leg creates the parallelogram structure which the team desired for structural stability of the system.

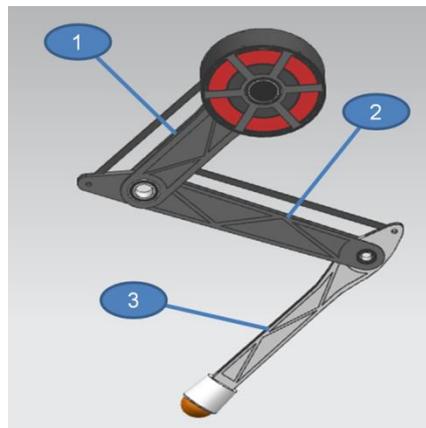


Figure 10: Final mechanical leg design

Reachable workspace analysis showed that joint one could move -71 to 63 degrees, joint two could move -20 to 83 degrees, and joint three could move -42 to 29 degrees. The final motor selection, which can be seen in Figure 10 above and Figure 11 below, is the HT05001. The HT05001 has a peak torque of 16.40Nm, weighs 1.30kg, has 12 poles, and a design voltage of 100 volts.

DIMENSIONS

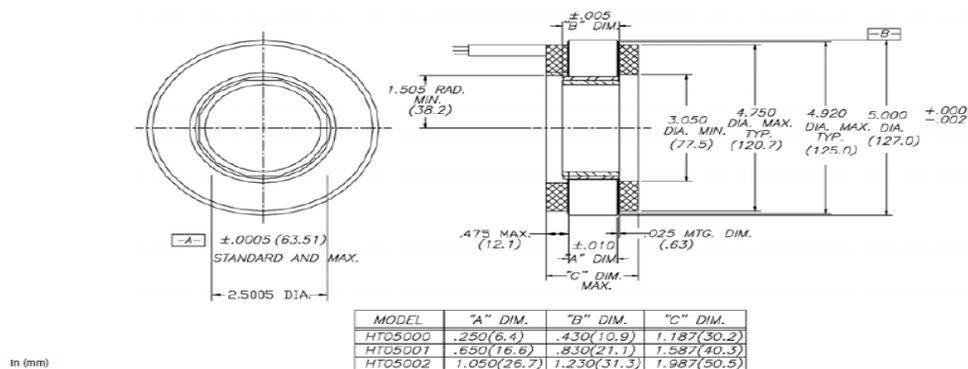


Figure 11: Final motor selection HT05001

The foot design, which is shown below in Figure 12, features a spring system for energy storage and release.

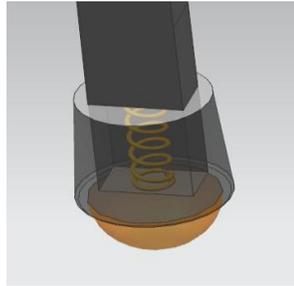
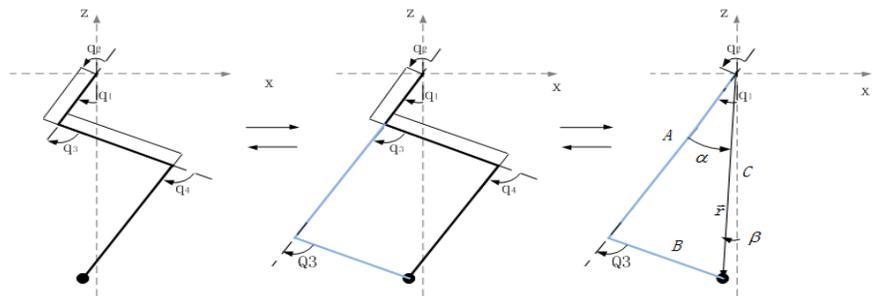


Figure 12: Foot design of the leg system

Joint torques were calculated for the joints of the designed cheetah-inspired leg. Reference images below in Figure 13 illustrate how the joint torques were calculated.



$$\alpha = \arcsin\left(\frac{B \sin(\pi - Q3)}{C}\right), \beta = -\arctan 2(r_x, -\text{sign}(r_x)r_z)$$

$$|q1 = \beta - \alpha$$

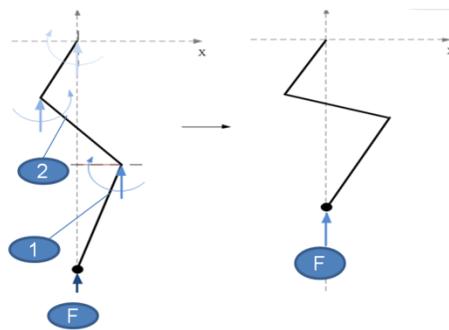


Figure 13: Joint torque calculations

Ridgidity analysis using Ansys was completed. Carbon fiber was used as the material and the material was said to be plastic in nature. First, link one was analyzed and the end force at the foot was found to be 72N. For all three links, the stiffness requirements were met using 72N as an applied force. Frictional force was neglected for the purpose of these approximations. The joint torque at joint one

was calculated to be 14.82Nm, and the joint torque at joint two was calculated to be 23.04Nm. Images produced by Ansys of all three links rigidity analyses can be seen in Figure 14, Figure 15, and Figure 16 below.

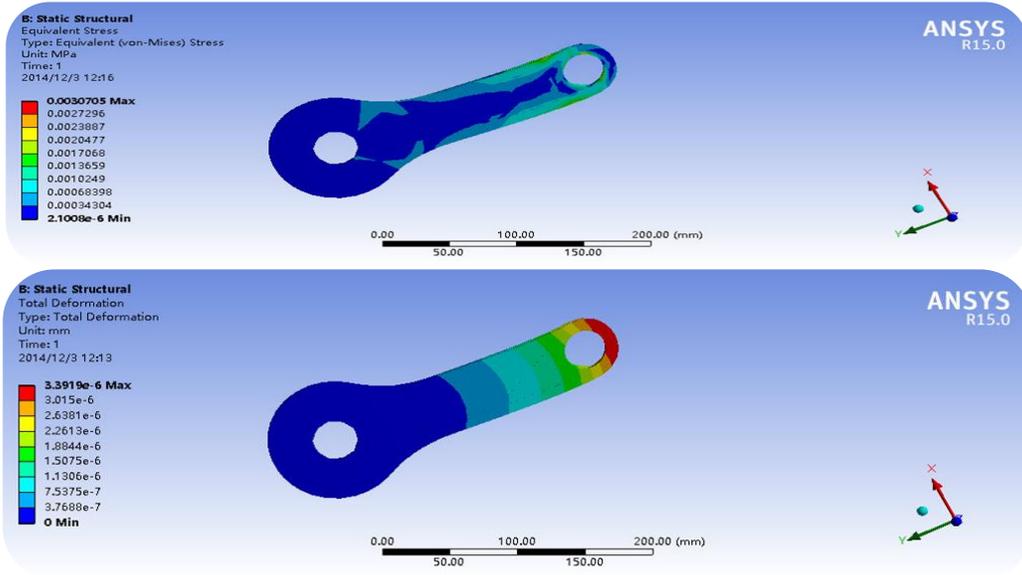


Figure 14: Ansys rigidity analysis - link 1

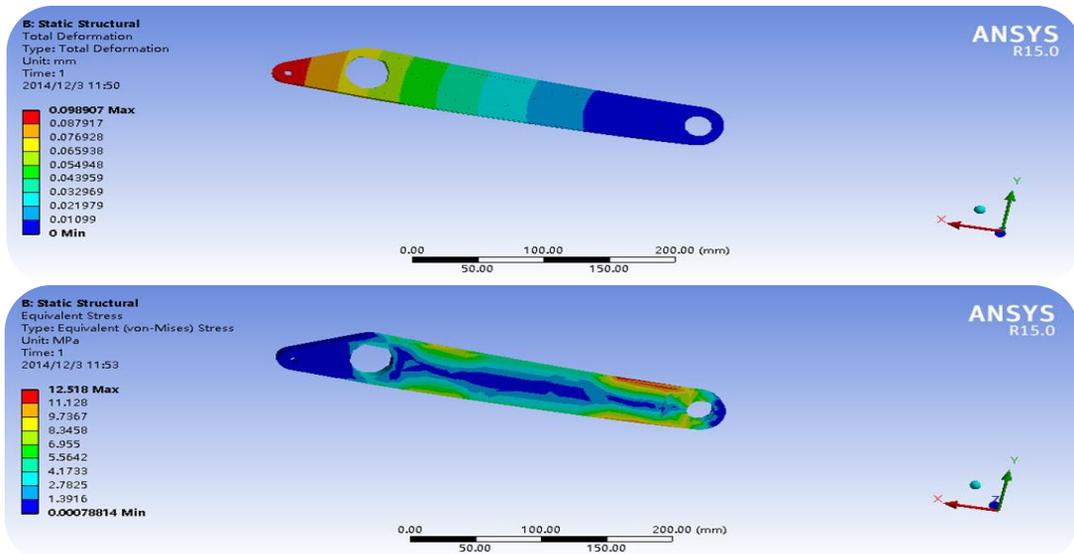


Figure 15: Ansys rigidity analysis - link 2

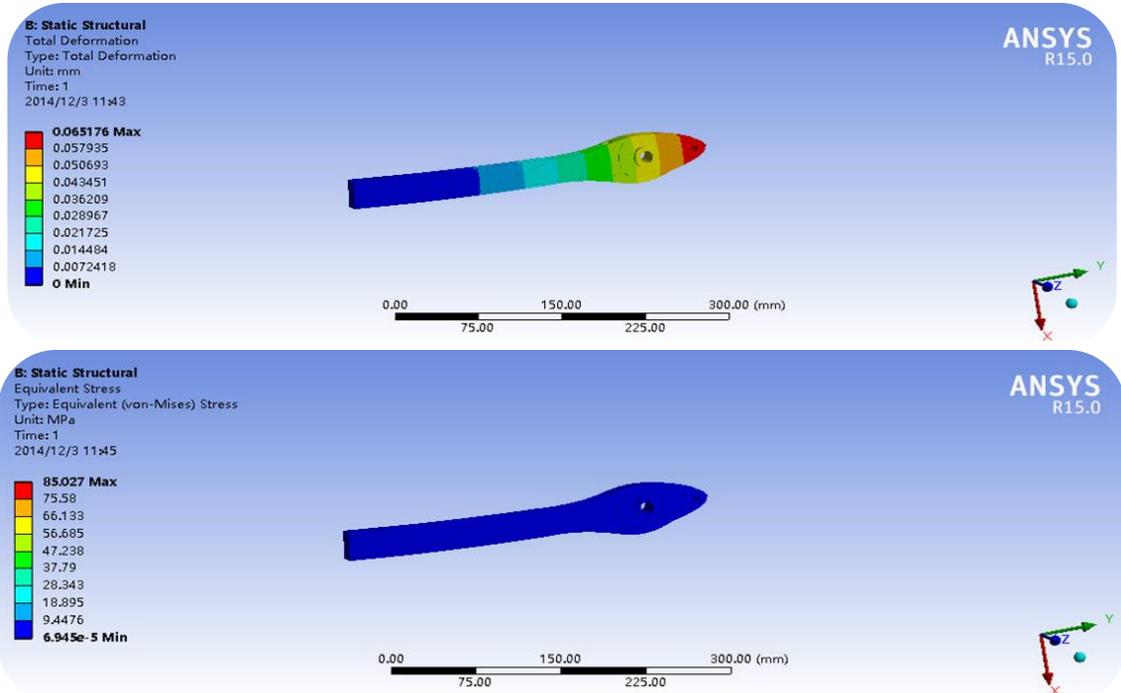


Figure 16: Ansys rigidity analysis - link 3

A computer motion simulation was also created. This was completed using the motion software Adams. A functional block diagram of the system is shown in Figure 17 below. Figure 18 shows the constraints imposed on the simulation. The motion consists of a constant speed, which then becomes a uniform deceleration, followed by a uniform acceleration.

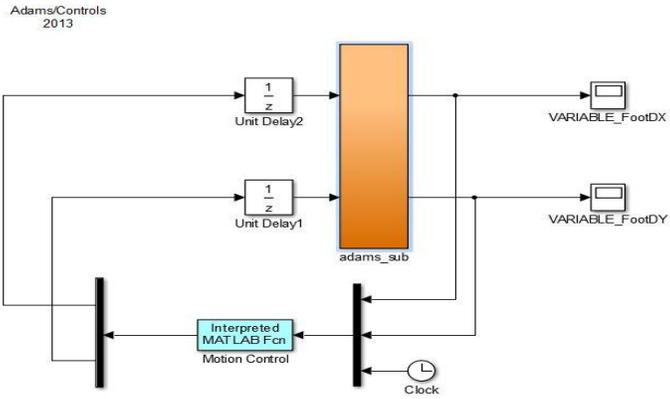


Figure 17: Functional block diagram

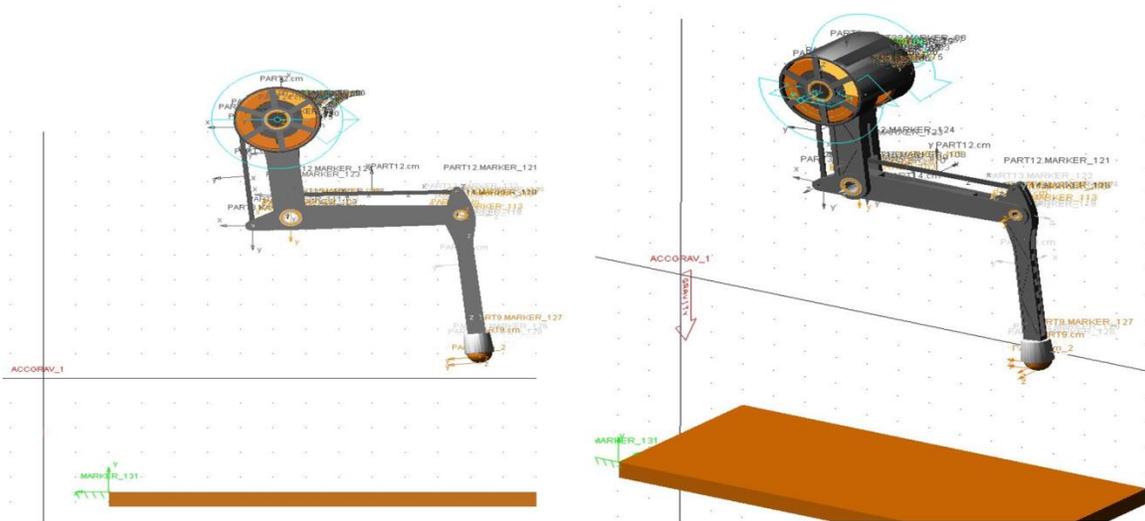


Figure 18: Simulation constraints

The legs were then combined into a model to show how the leg design may look as a full implementation as a quadruped robotic system. This image is shown below in Figure 19.

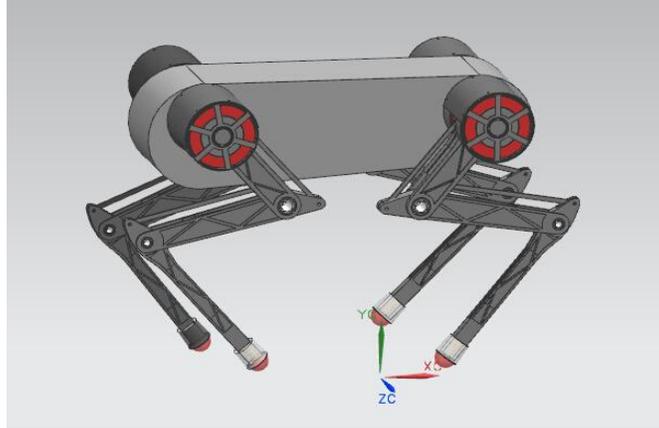


Figure 19: Quadruped representation of leg design

4.2 Sensor System Design

By performing the methodology outlined in Chapter 3, we were able to obtain the data necessary to design the sensor system used to complement the mechanical leg design. Further, once that data was collected, we were able to use it to design the algorithm needed for the sensor system to function.

4.2.1 Data Acquisition

After setting up a testing mechanism as described in Chapter 3, we tested the force sensor on a hard surface and a soft surface. We then analyzed this data to see what conclusions could be drawn. When plotted, the raw data acquired during testing roughly followed the pattern shown in the graph in Figure 20. The foot mechanism began at rest on the surface, hence the period of time where the force sensor's output is nearly 200. The foot mechanism was then lifted from the surface, explaining the force's drop to zero. Lastly, the foot mechanism was dropped, hitting the surface with an impact velocity of 0.79 m/s. This step is shown by the last portion of the graph, where the force increases from zero.

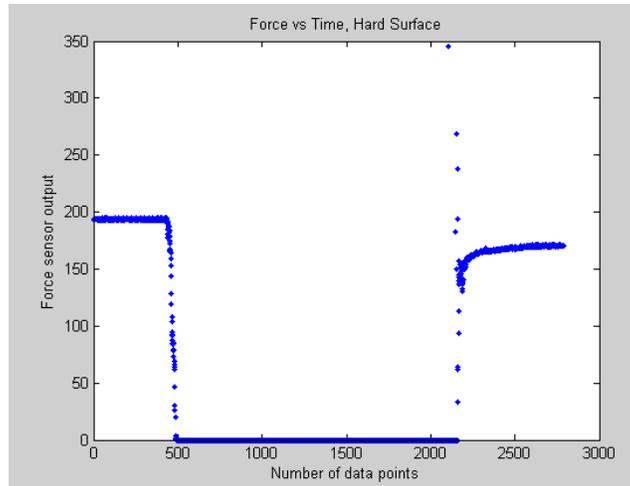


Figure 20: Raw data acquired during testing on hard surface

One unexpected result is that the normal force before and after the drop do not always match exactly. Of course, when the foot lands, it still weighs just as much as it did before it was dropped, so this indicates an issue with our set up. Despite our attempts to keep the foot perfectly straight as it landed, at times it would land on a slight angle, meaning that all of the foot's force was not being distributed onto the force sensor. Due to this error, the normal force before and after landing did not always match exactly.

Another unexpected result was that the data acquired while testing on a soft surface was not always consistent. As shown in Figure 21 and Figure 22, when plotted, the data follows very different shapes. There are two reasons for this deviation. The first is that the force sensor we used experiences significantly more error when it is used on soft surfaces (Tekscan). The second is that the soft surface we were dropping on was mud, which contained small rocky pieces in it. It is possible that on some drops, the force sensor hit a rock on the way down, while in other drops it hit only the dirt portion.

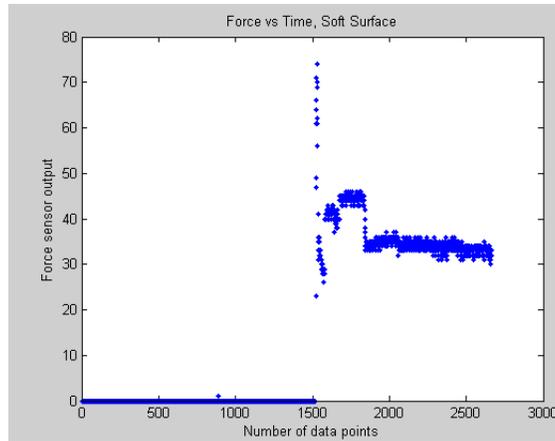


Figure 21: Raw data collected from testing on a soft surface

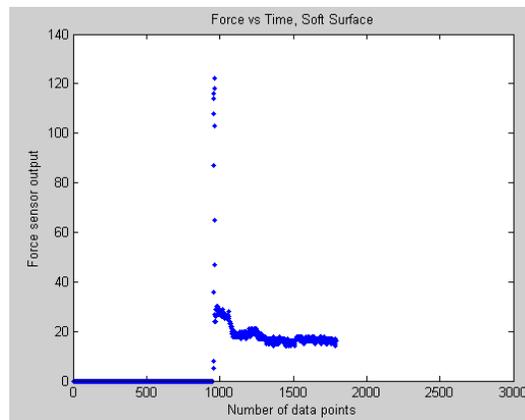


Figure 22: Raw data collected from testing on a soft surface

The last unexpected result was that the data was occasionally noisy even while dropping on a hard surface. Again, this is probably due to the particular force sensor that we were using. An example of noisy data while dropping on a hard surface is visible in Figure 23. Only the portion of the data where the foot is landing is shown. Data from additional tests is available in Appendix A.

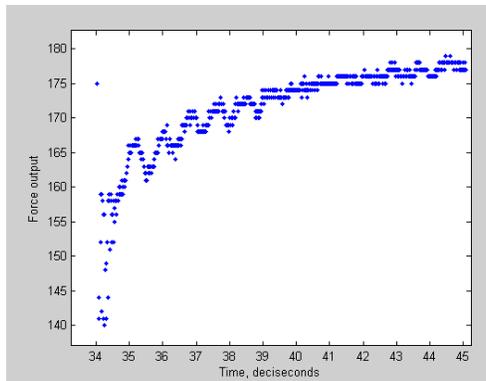


Figure 23: Noisy data acquired while testing on a hard surface

Despite the setup being inexact, there were clear differences between the data acquired by testing on a hard surface and the data acquired by testing on a soft surface. Figure 24 and Figure 25 below show representative examples of the data acquired when dropping on a hard surface and a soft surface, respectively.

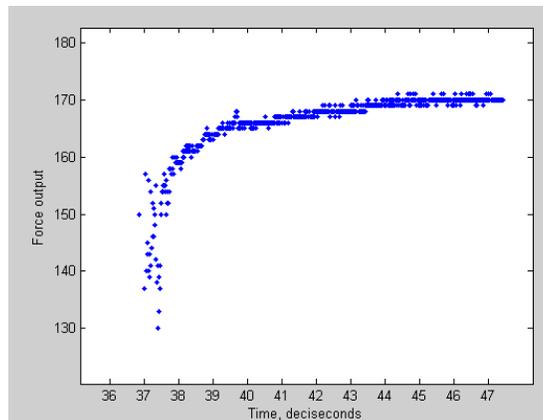


Figure 24: Representative example of data acquired on hard surface

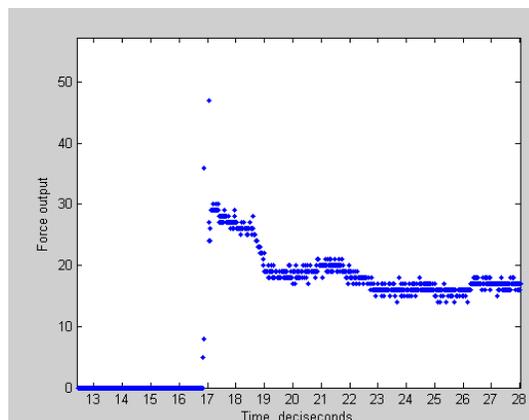


Figure 25: Representative example of data acquired on soft surface

As can be seen in Figure 24, the data acquired during testing on a hard surface follows a logarithmic curve. This is not true of the data acquired when testing on a soft surface, as can be seen in Figure 25. Instead, data acquired when testing on a soft surface has the distinct characteristic of having an initial spike in the force sensor output, then leveling out to the normal force afterwards. These trends in the data allowed us to design an algorithm to automatically detect whether the foot mechanism was on a hard surface or soft surface.

4.2.2 Arduino Algorithm

The algorithm was originally designed for use on an Arduino board due to the simplicity with which Arduino code can be implemented. However, despite the ease with which complicated functions can be programmed using Arduino, we strived to keep the algorithm simple so that it could be transferred to an FPGA later on. The code used to implement the algorithm is available in Appendix B.

As previously discussed, we discovered that while landing on a hard surface, the force output would increase from zero to normal in a logarithmic manner. While landing on a soft surface, the force output would increase from zero to a value greater than the normal force, then level back out to the normal force. Because of these findings, we determined that the algorithm should function by checking if the data followed a logarithmic curve, indicating a hard surface, or if it increased then decreased again, indicating a soft surface. The basic flow chart for the algorithm can be seen in Figure 26.

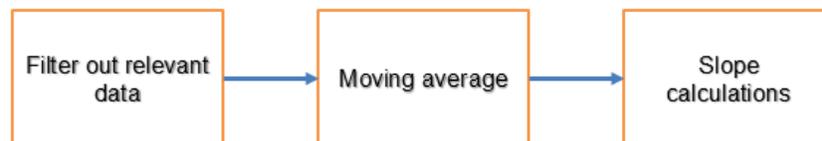


Figure 26: Flowchart for the hardness vs softness detection algorithm

The first step in the algorithm is to filter out only the data of interest, meaning the data where the foot is landing. This was accomplished by searching the force output for zeros. When zeros are found, it means that the foot is currently in the air. After zeros were found, the algorithm begins watching for non-zero data. Once this non-zero data is found, it means that the foot is in the process of landing, and the incoming data should be analyzed.

The next step of the algorithm is taking a moving average of the data points. As previously stated, the data acquired from the force sensor was somewhat noisy. In order to combat this noisiness and be able to view the overall trend in the data better, a moving average using 10 points at a time was

implemented in the algorithm. The number of points chosen per moving average, 10, was chosen through a process of trial and error. Figure 27 shows one set of testing data smoothed using a moving average.

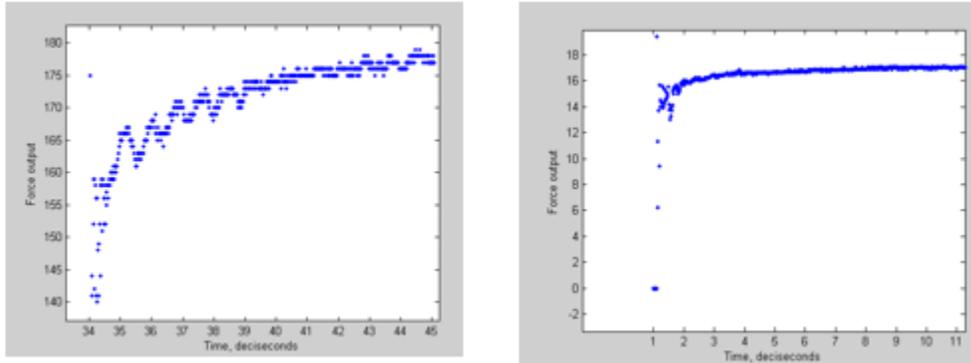


Figure 27: Left-hand side: Data before smoothing. Right-hand side: Data after smoothing.

The last step of the algorithm was the most complicated. Although attempting to fit the data to a logarithmic curve would be an ideal, precise solution, it would also prohibitively difficult to implement on an FPGA and be time and resource intensive. Since the algorithm has to run in real time and be fast enough to process data before the robot takes its next step, this option was ruled out. Instead, the algorithm uses periodic slope calculations to approximate whether or not the data is following a logarithmic pattern as well as determining if the data spikes and then lowers again.

If a curve is logarithmic, then towards the beginning of the curve, the slope will be very high. As you move towards the end of the curve, the slope will consistently be smaller at each point along the way, eventually becoming zero at the end of the curve. The slope will never be negative. If the data we acquired followed this pattern, the surface would be deemed hard. If, on the other hand, the data had a slope that dipped negative, as would happen if the data were to spike, then level out again, the surface would be deemed soft. A graphical representation of this concept is shown in Figure 28.

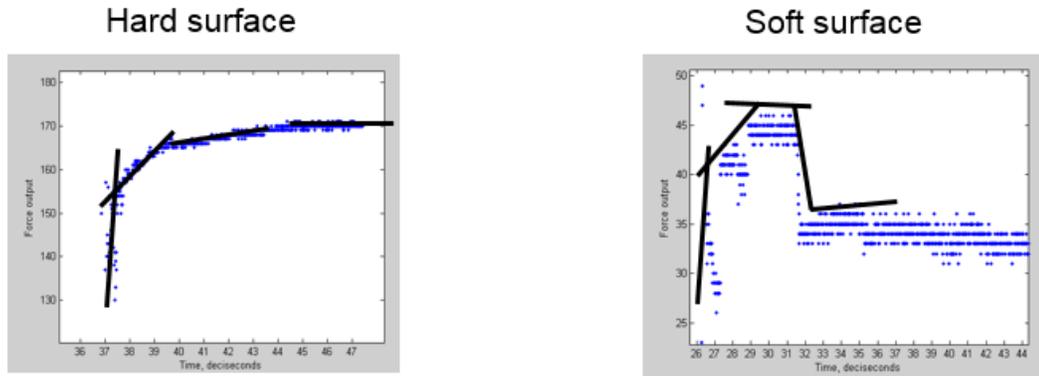


Figure 28: Demonstration of slope calculation portion of algorithm

4.2.3 FPGA Implementation

Though the testing of the force sensor was completed using a microprocessor, it was decided that for integration of the sensor system onto the mechanical leg design, an FPGA would be better suited for the task. A main consideration in this decision was the speed and simultaneous calculations allowed by the FPGA versus a microcontroller. Having the increased speed of calculation could potentially increase the accuracy of the system by being able to sample and process data faster in real time. In addition, the FPGA allows for simultaneous calculations. By choosing to use an FPGA for this application, it allows the project to be open to expansion by adding more sensors for sensing other criteria and being able to complete calculations at the same time.

In order to gather data from the force sensor using the Nexys 3 FPGA it was necessary to use an analog-to-digital converter (ADC) due to the nature of the force sensor being an analog device while the FPGA is a digital device. A diagram below (Figure 29) describes how the ADC, FPGA, and force sensor are wired together.

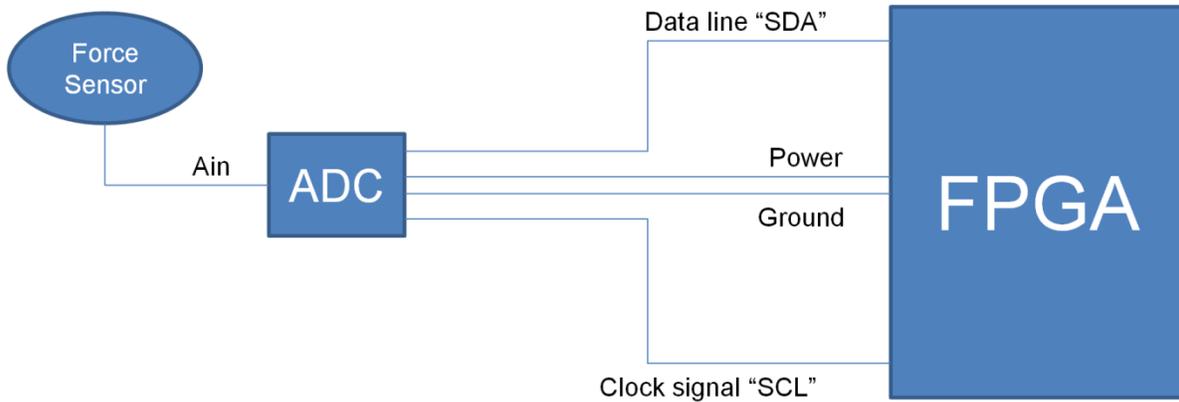


Figure 29: High-level schematic of FPGA system

The overall structure of the Verilog implementation of the force sensor is shown below in Figure

30.

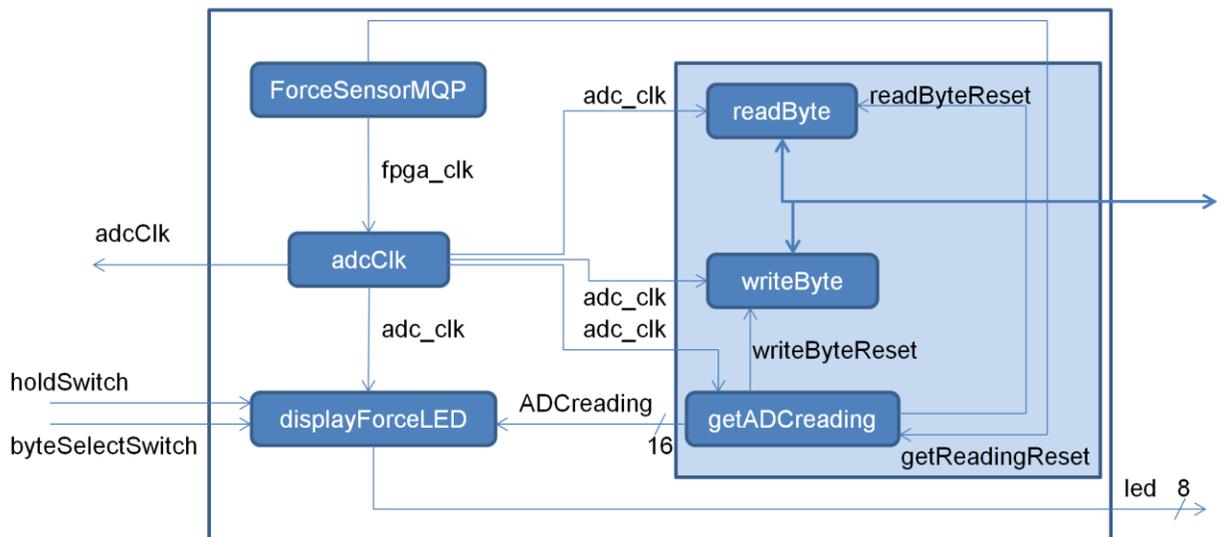


Figure 30: Block diagram of Verilog force sensor implementation

Using the ADS1115, it was necessary to create a clock signal for the program which would run at a rate less than 860 Hz due to the ADC's max sample rate of 860 samples per second. The FPGA clock operates at 100MHz and this signal was converted down to 800Hz for use by all modules of the program and the ADC itself. The module which handles converting the clock signal is called "adcClk". The "readByte" and "writeByte" modules fall under the "getADCreading" module and handle everything necessary to read and write bytes one at a time between the FPGA and the ADC. The "getADCreading" module handles getting a single reading from the ADC. In order to visually display the reading from the

ADC, the module “displayForceLED” uses switches on the FPGA Nexys board to hold a reading and select the most or least significant byte of the reading and outputs the 16 bit reading on the eight LEDs on the board.

The “writeByte” module operates sequentially within an always blocks which executes on the negative edge of the clock signal. A reset for the module is controlled by the higher module, “getADCreading”, which determines if the “writeByte” module should be active. A register holds the value of an index which is used to keep track of which bit of the byte is being written. On each clock cycle, the value of a bit is written to the data line – called SDA – and the index is incremented down by one. Once the index reaches a value of zero, the FPGA no longer writes values to the SDA line but instead reads in the value. If the value is low, this is due to the acknowledgement from the ADC which means that the byte has been received successfully. If this is the value on the SDA line, the index is kept at zero as the byte has been successfully written. However, if the SDA line is high, the ADC has not received the byte and the bit index is reset to eight in which case the module essentially restarts the process of writing the same byte. The flow of this module can be seen in the diagram below in Figure 31.

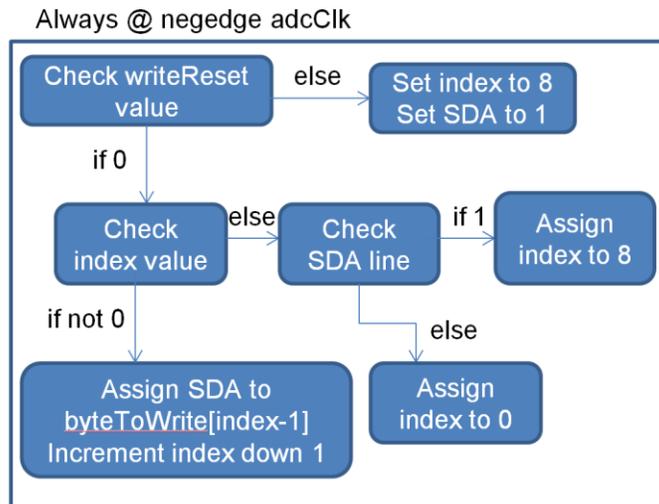


Figure 31: “writeByte” module block diagram

The “readByte” module operates in almost the exact same manner, except instead of writing to the SDA line, the SDA line is read from. Similarly, there is a reset controlled by the higher module “getADCreading” and an index. One difference is that instead of checking to see if the ADC acknowledged the FPGA, the FPGA must acknowledge the ADC by setting the SDA line low. This happens

when the bit index is one which means all eight bits of the byte have been read from the ADC. This can be seen in Figure 32 below.

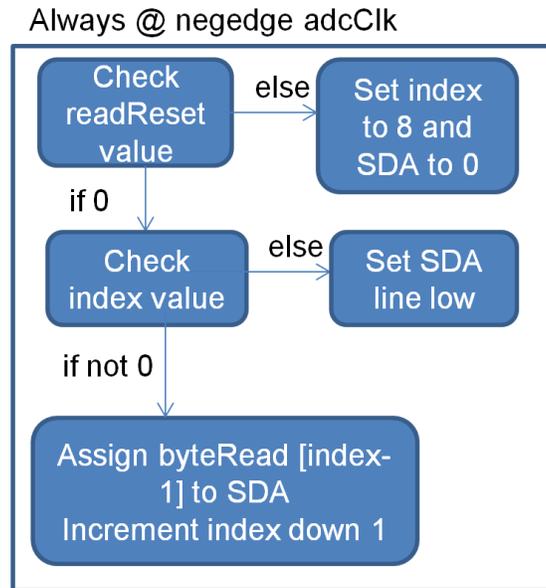


Figure 32: “readByte” module block diagram

The “getADCreading” module encompasses the “readByte” and “writeByte” modules as it uses them to set and read registers on the ADC. The “getADCreading” module consists of two simultaneous processes which control which state the state machine is in and what happens within that state. For each reading from the ADC, the config, pointer, and conversion registers must be set. Then the second and third byte of the conversion register which hold the 16 bit value can be read. The flow of the module and state machine can be seen in Figure 33 below and the actual values set to the registers and the meaning behind the values can be seen in Table 6 below.

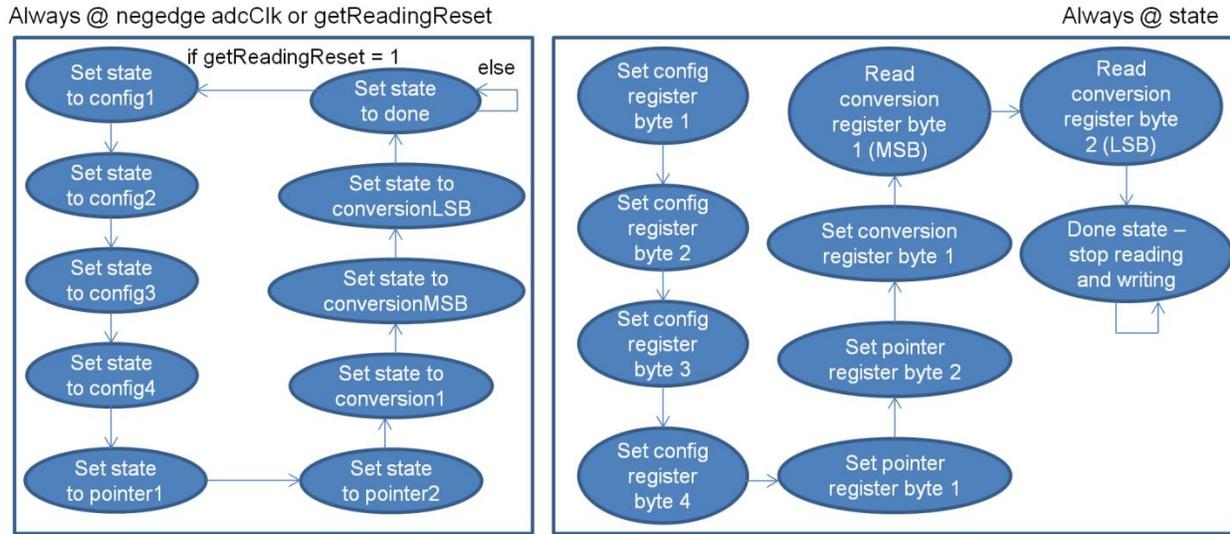


Figure 33: "getADCreeding" module block diagram

State	Byte Written	Meaning
Config1	1001 0000	7 bit ADC I ² C address
Config2	0000 0001	Points to Config register
Config3	1000 0100	MSB of Config register – Begin single conversion, use AIN1 from multiplexer, use default gain amplifier of 2.048V, continuous conversion mode.
Config4	1110 0011	LSB of Config register - set data rate to 860 samples per second, traditional comparator, active low comparator polarity, disable comparator.
Pointer1	1001 0000	7 bit ADC I ² C address
Pointer2	0000 0000	Access the conversion register
Conversion1	1001 0001	7 bit ADC I ² C address + read/write bit
ConversionMSB	Receive	Read back the MSB of the conversion
ConversionLSB	Receive	Read back the LSB of the conversion

Table 6: ADC register configurations

An additional module, "displayForceLED" was created for the purpose of visually displaying the 16-bit sensor value on the LEDs on the FPGA board. This output is not critical to the program, but informational for the purposes of testing. Two switches on the FPGA board control the output of the

eight LEDs. One switch holds the value from the ADC and the other toggles between the MSB and LSB of the two byte reading.

The FPGA implementation of the project was completed at the level presented without testing. In order to test the work that has been done, a tri-state buffer needs to be added to ensure that the SDA line is not being read and written to at the same time. After this functionality is added, the Verilog code could be synthesized and implemented on the physical hardware.

4.3 System Analysis

The sensor system and mechanical leg were designed to be incorporated in a manner where the results from the sensor system can determine the gait used by a quadruped robot employing the cheetah-inspired legs. Since the robot is intended to be fast, the default gait will be a trot. While traveling on a soft surface, the gait will shift to a creep in order to maintain stability.

The mechanical leg and sensor system are two parts of one whole. However, since the sensor system existed as a prototype on physical hardware while the mechanical design existed as a computer model, they were ultimately never combined. Despite this limitation, significant thought was given to how the two designs could be combined in the future. This is further discussed in the recommendations of Chapter 5.

5.0 Recommendations

5.1 Specially Designed Force Sensor

As predicted, there were issues which arose with the force sensor in the dirt and mud. Due to the nature of the force sensor, particles of dirt managed to spread apart and make their way into layers of the force sensor. This caused inconsistent readings at first and then eventually caused the force sensor to get “stuck” on one reading value. Because of the layers of the force sensor that are necessary for it to operate, as previously discussed, these force sensors can have issues in applications such as terrain detection. In order to have the layered force sensor work for an application such as this, it is recommended that a force sensor is specially designed for the application and a modification to the leg is created. This design would consist of the lower most layer being extended out and able to wrap up around the leg into a secured environment. Allowing the lowest layer which contacts with the ground to wrap up would prevent particles from working their way into the layers. The leg would need to be designed to have a way of securing this layer off, preferably at a height above which most terrain contact would occur.

5.2 Transfer Microprocessor Designed Algorithms to FPGA

The algorithms for taking in sensor data from the force sensor, calculating which type of terrain the robot is walking on, and altering the gait based upon that information was first designed on the Arduino microprocessor. This allowed for quick and easy implementation of the theory behind the operation of the algorithms. This allowed the team to test theories and easily make changes and adaptations; however for the actual implementation of on leg computation the FPGA is preferred. The parallel processing and increased speed benefits of the FPGA were previously discussed. Because of the benefits presented, it is recommended that the algorithms are fully transferred over to and implemented on the FPGA – a task that was unable to be completed by the team within the time constraints of the project.

5.3 Improved Carrying Capacity or Applied Load of the Mechanical Design

As presented, the purpose of the leg which the team designed was for speed. The leg passed all rigidity tests without an applied load. For further work, we recommend focusing on extending the carrying capacity of the robot as a whole while still maintaining the speed the leg was designed for.

Being able to carry a load at speed could significantly extend the opportunities for application of this robotic leg design.

5.4 Create a Prototype

An additional recommendation to future researchers is to create a physical prototype of the leg and sensor system. Although the physical leg design went through multiple simulations and analyses, computer programs are never 100% accurate to how designs will work when actually implemented. There are always additional sources of error that occur when a design is physically implemented, such as imperfections in the material.

Additionally, although the sensor system was physically tested, it was not tested in conjunction with the leg design. Since these two designs are elements of a cohesive whole, they will need to be tested together in order to ensure that the entire system works.

Once the leg and sensor system have been successfully prototyped, combined, and tested we recommend that the entire robot be prototyped. This will allow for the robot to be tested for its true purpose of autonomous locomotion.

5.5 Integrate the Algorithm and the Controls

Our last recommendation to future researchers is to integrate the algorithm from the sensor system with the controls system for the entire robot. This recommendation circles back to the purpose of the sensor system: by detecting whether a surface is hard or soft, the robot should be able to alter its gait. On a hard surface, the robot should continue onwards in a trot, but on soft surfaces it should implement a creep in order to maintain its stability.

Currently, the algorithm for detecting the hardness of a surface does not take into consideration factors like how quickly the robot moves and what gait it is currently using. It also does not tell the robot what gait to use. However, the algorithm was designed in such a way that it is easy to incorporate this information. When the algorithm detects a hard surface, that should signal to the control system that the robot should implement a trot. When the algorithm detects a soft surface, that should signal to the control system that the robot should implement a creep. Since the control system will always know whether the robot is in a creep or a trot, the robot's speed and the number of feet the robot has on the ground will always be known. This information could be incorporated into the algorithm so that instead

of filtering out relevant data through the process outlined in Chapter 4.2.2, data could be processed only when it is known that the robot is taking a step. This would save power and processing time.

6.0 Conclusion

Throughout the course of our project, our team worked closely with our advisors Dr. Wenchuan Jia and Dr. Yiming Rong in order to advance the field of autonomous robotic locomotion through biomimetics and smart sensing. By advancing the field of autonomous robotic locomotion in this manner, autonomous robots will better be able to navigate difficult terrains quickly while maintaining their stability. This will lead to an impact in a multitude of important fields, from military applications that help protect countries to search and rescue robots that save lives.

In order to accomplish our goal, we first conducted research on current biomimetic leg designs and sensor system designs intended to operate on legged robots. We then decided upon a biological prototype for the leg design and a basic method for retrieving relevant data from a sensor system. From there, we modeled the leg design and performed motion and stress analyses on it, along with prototyping and testing the sensor system on an Arduino board. Towards the end of the project, we optimized the mechanical leg design and began work to transfer the sensor system from an Arduino to an FPGA.

The end results of our project are a model of a biomimetic robot leg based on a cheetah's leg and an algorithm that can be used to detect the relative hardness versus softness of whatever surface the robotic leg is walking on. We recommend that the leg design be prototyped and that the sensor system be fully incorporated into the prototyped leg. It is our hope that our deliverables will serve as a basis for further research into the field of autonomous robotic locomotion.

Works Cited

- Atkins, A. G. & Escudier, M. P. (2013). Autonomous robot. In *A Dictionary of Mechanical Engineering*. (Ed. 1.). Oxford University Press.
- Barry, J. (2007). Meet 'Big Dog'. *Newsweek*, 149(12), 8.
- Bladin, P. F. (2006). W. Grey Walter, Pioneer in the Electroencephalogram, Robotics, Cybernetics, Artificial Intelligence. *Journal of Clinical Neuroscience*, 13(2), 170 – 177.
- Chen, X., Gao, F., Qi, C., & Zhao, X. (2013). Spring Parameters Design to Increase the Loading Capability of a Hydraulic Quadruped Robot. Proceedings from *2013 International Conference on Advanced Mechatronic Systems*, pp 535 – 540. Luoyong, China: IEEE.
- Chung, J., Lee, I., Cho, B., & Oh, J. (2013). Posture Stabilization Strategy for a Trotting Point-foot Quadruped Robot. *Journal of Intelligent & Robotic Systems*, 72(3), 325 – 341.
- Del Bue, A., Tamassia, M., Signorini, F., Murino, V., & Farinelli, A. (Eds.). (2013). Proceedings from SSSRR '13: *IEEE International Symposium on Safety, Security, and Rescue Robotics*. Linkoping: IEEE.
- Fahimi, F. (2008). *Autonomous Robots: Modeling, Path Planning, and Control*. New York, NY: Springer.
- Formal'sky, A., Chevallereau, C., & Perrin, B. (2000). On Ballistic Walking Locomotion of a Quadruped. *The International Journal of Robotics Research*, 19(8), 743 – 761.
- Gonzalez de Santos, P., Garcia, E., & Estremera, J. (2006). *Quadrupedal Locomotion: An Introduction to the Control of Four-legged Robots*. London, UK: Springer.
- Hall, K. (2009). *5 Homemade Robots that Offer Hope for DIY Geeks*. Retrieved from: <http://www.popularmechanics.com/technology/engineering/robots/4285232>
- Hindle, P. (2013). UAVs Unleashed. *Microwave Journal*, 56(8), 8.
- Ho, V., Nagatani, T., Noda, A., & Hirai, S. (2012). Proceedings from CASE '12: *2012 IEEE International Conference on Automation Science and Engineering*. Seoul: IEEE.
- Iagnemma, K., Shinwoo, K., Shibly, H., & Dubowsky, S. (2004). Online Terrain Parameter Estimation For Wheeled Mobile Robots With Application To Planetary Rovers. *IEEE Transactions on Robotics*, 20(5), 921 – 927.
- Interlink Electronics (a). (n.d.) FSR Integration Guide. Retrieved from: https://www.digikey.com/Web%20Export/Supplier%20Content/InterlinkElectronics_1027/PDF/Interlink_Electronics_Integration_Guide.pdf?redirected=1

Interlink Electronics (b). (n.d.) FSR 400 Series Data Sheet. Retrieved from:

<http://www.robofun.ro/docs/2010-10-26-DataSheet-FSR400-Layout2.pdf>

iRobot Corporation. (2014). *What Are Cliff Sensors?* Retrieved from:

http://homesupport.irobot.com/app/answers/detail/a_id/314/~what-are-cliff-sensors%3F

iRobot Corporation. (2004). *Roomba 800 Series User's Manual*.

Levy, D. (2005). *Robots Unlimited: Life in a Virtual Age*. Wellesley, MA: A K Peters.

Liu, D. & Sun, Y. (2011). *Biologically Inspired Robotics*. CRC Press.

Lomba, C. R., Valadas, R. T., & de Oliveira Duarte, A. M. (1999). Experimental Characterization And Modelling of the Reflection of Infrared Signals on Indoor Surfaces. *IEEE Proceedings – Optoelectronics*. 145(3), 191.

Majumder, S., Ghosh, S., & Maity, A. (2011). Serpentine Robot Locomotion: Implementation through Simulation. *International Journal of Innovative Technology and Creative Engineering*, 1, 60 – 65.

Na, B., Choi, H., & Kong, K. (2015). Design of a Direct-Driven Linear Actuator for a High-Speed Quadruped Robot, Cheetaroid-I. *IEEE/ASME Transactions on Mechatronics*, 20(2), 924 – 933.

Scobey-Thal, J. (2014). Biomimetics. *Foreign Policy*, 209, 20 - 21.

Sinha, P., Xu, Y., Bajcsy, R., & Paul, R. (1990). *Robotic Exploration of Surfaces with a Compliant Wrist Sensor*. Technical Reports (CIS). Paper 354.

Sprowitz, A. Tuleu, A., Vespigani, M., Ajallooeian, M., Badri, E., & Ijspeert, A. J. (2013). Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot. *The International Journal of Robotics Research*, 32(8), 932 – 950.

Tekscan. (n.d.). Force Sensors For Design. Retrieved from: www.tekscan.com/pdf/FLX-Force-Sensors-For-Design.pdf

Thueer, T. & Siegwart, R. (2010). Mobility Evaluation of Wheeled All-Terrain Robots. *Robotics and Autonomous Systems*, 58(5), 508 – 519.

[Elsie, one of the Tortoise robots]. Retrieved December 26, 2014, from:

http://www.theoldrobots.com/images48/Elmer_Elsie3.JPG

[Demonstration of a trot gait]. Retrieved December 26, 2014, from: <http://knowyourhorse.org/wp-content/uploads/2011/03/trot.jpg>