April 2011

Aeacus: The Design and Realization of an Ant-Like Robotic Platform

Colin Francis Roddy
Worcester Polytechnic Institute

Daniel Michael Praetorius
Worcester Polytechnic Institute

Neal David Anderson
Worcester Polytechnic Institute

Development of a Bio-inspired Robotic Platform for Swarm-based Multi-object Transport over Uneven Terrain

Neal Anderson
Daniel Praetorius
Colin Roddy

Professor Stephen S. Nestinger
Professor William R. Michalson

Aeacus: Development of a Bio-inspired Robotic Platform for Swarm-based Multi-object Transport Over Uneven Terrain

A Major Qualifying Project
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfilment of the requirements for the
degree of Bachelor of Science

by

Neal Anderson

_____

Daniel Praetorius

_____

Colin Roddy

_____

Date:

May 2, 2011

Report Submitted to:
Professor Stephen S. Nestinger
Professor William R. Michalson

Abstract

A large volume of recyclable material is inescapably placed in landfills, despite modern recycling efforts. In order to create an effective way of recovering such material, we have designed and manufactured an ant-like robot with the potential to do so. The hexapod platform developed can lift two times its own body weight and travel at two times its body length per second. With the intent of future development, the system was designed to be versatile, inexpensive and easy to modify. The platform has the final goal of becoming part of swarm in order to improve productivity and increase efficiency of the recycling process.

Acknowledgements

Our team would first like to thank Professor Stephen S. Nestinger and Professor William R. Michalson. Without their guidance and direction throughout the entire process, the project would not have been possible.

We also extend our thanks to the WPI Robotics Engineering Program, for without the generous funding we would be unable to see this research project come to fruition.

Our team would also like to acknowledge Blake Alberts for his assistance and support throughout our lengthy and iterative design period. His expertise and skills provided our team with a resource that we would otherwise be without, and for that, we are grateful.

We would also like to thank the other teams sharing the workspace with us: FSAE, Bomb Disposal, MOSAIC and especially Sabertooth Robotics. Their peer reviews and continuous support made our project what it is today.

The team would also like to extend our gratitude and thanks to Erica Stults. She is the grad student in charge of the rapid prototype machine at WPI and without her, our project would not have been possible. Even though we set her back several days (on several occasions) with our orders, she was always cheerful and would rush our parts if needed.

Executive Summary

Many recyclables like plastic bottles, aluminum cans and precious metals from electronics will be placed in a landfill despite modern recycling efforts. To address this, this project aimed to develop a bio inspired robotic platform based on the Carpenter Ant - *Camponotus pennsylvanicus*. This robotic platform is able to traverse the difficult terrain of a landfill and lift two times its own body weight, could then be later improved upon to identify and sort those recyclables that were placed in the landfill.

The team aimed to meet the following design specifications:

1. Ant-like Hexapod
2. Weight of 5 pounds
3. Maximum length of 16 inches
4. Minimum payload of 2 times the body weight (10 pounds)
5. Minimum walking speed of 2 times the body length (36 inches per second)
6. Operating life of at least 30 minutes
7. Able to navigate over obstacles 2 inches in height

These design specification were then couple with additional design considerations that would improve the quality of the platform and allow the project to succeed in the future. These considerations included the ability to move fluidly, the ability to be expanded upon in the future and cost effective manufacturability.

The project was separated into three systems: mechanics, controls, and electronics. The mechanics system entailed the design and construction of the physical robot. The controls system included the simulation of the robot to predict how the robot would work and the development of software. The electronics system included the development and fabrication of the devices needed to drive the physical robot.

The robot was divided into four discrete systems: the main body, six legs, tail, and head, each acting as independent subassemblies. These were designed to be modular to facilitate quick and simple assembly, repair, and modification. The main body acts as a connecting hub for the other subassemblies, as well as housing the main electronics. The legs are designed to be identical, and are completely self-contained,

requiring only three electrical connections for power and signal. The tail houses the batteries and power management unit, serving a dual role as a counterbalance as well. Currently, the head is responsible for object manipulation, and is intended to be used as housing for vision and other sensors in the future.

The control system consists of a 32-bit ARM Cortex-A8-based main controller, responsible for the heavy computation and inverse kinematics and dynamics required for gait synthesis, networked with a group of less powerful 8-bit slave controllers, one on each limb, responsible for the raw I/O and positioning control loops. The master and slave controllers communicate via a custom protocol over an RS-485 bus.

While the robot has yet to walk on its own due to complications with software, mechanically and electrically the robot meets most of the design specifications.

The mechanical system met and superseded many initial expectations at the cost of exceeding the initial planned weight. While the initial design specification limited the robots weight to 5 pounds, alterations to the final design caused it to be overweight by a pound. Strength testing also showed that many components are stronger than initially expected, increasing the total payload the frame can withstand.

With the results found could still be improved upon further. The focus of the improvements would be to make the robot lighter without degrading structural integrity, improving maximum payload and increasing power efficiency. Some potential improvements to the mechanical system include using molded plastic parts, higher quality motors and improvements optimizing the joint angles by adjusting the shape of the leg.

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Problem Statement

Despite modern efforts many recyclable items are still trapped in landfills. Plastic bottles, aluminum cans, as well as, precious metals used in electronics and common hazardous wastes like lead-acid batteries. Current methods to sort these materials include hand sorting and plasma gasification, both of which are costly or to labor intensive to pursue. This project aimed to find an alternative solution to sort these recyclables.

1.2 Biologically Inspired

In the never-ending search to advance the technologies and methods which govern the production of robots, engineers have looked to increasingly diverse sources of inspiration. Many of the robotics solutions which exist in the world today are characterized by limited flexibility, and a lack of robustness (both problems of which limit the effectiveness of the system and cost companies money and time). A trend which has risen to popularity within the field of robotics engineering is that of biologically-inspired robotics. Biologically inspired is a trend which can be generalized as a view that robotic systems can be improved by emulating the biological characteristics and behaviors of animals, as animals demonstrate far greater flexibility and robustness in the face of environmental challenges.

It was decided that within the scope of this enterprise we desired to investigate the potential of taking inspiration from, and implementing, robots whose design is focused around the biology of ants. Ants can be found on every continent in the world, except Antarctica, which serves testament to their admirable ability to adapt and flourish in virtually any environment the world can present. Individually, ants have notable physical capabilities, taking their size into account. But it is the interaction of ants in colonies which drives the true inspiration behind this project. Few people can deny the awe-inspiring ability of ant colonies to perform large and complex tasks which far exceed the ability of any individual member of the colony.

1.3 Long Term Goals

The goal of the Aeacus project is to develop a system of, inexpensive, ant-like hexapod robots that can navigate the difficult terrain of a landfill. These ants will search out and sort the recyclable materials

mixed with the general waste. Being inexpensive, many units could be purchased, forming a colony. A single ant robot will not sort a substantial amount of material; however, as a larger system, the colony of ant robots will be able reduce landfills to only materials that cannot be recycled.

Looking farther into the future, the robots, modified to fit the various situations, could perform task outside of landfills. The ants could be used to clean other environments such as beaches, national park and city streets. Other uses for the system, outside of cleanup, include search and rescue after a national disaster and various military applications. The colony, equipped with the appropriate sensory systems could look for people trapped in rubble where the terrain is too dangerous for human search teams. The robots, being inexpensive, could be lost rather than a human life. Military applications include using the robot for improvised explosive device disposal and the search for insurgents in dangerous areas. The large number of units that could be sent out, working as a larger system, could be more efficient than current solutions.

1.4 Project Goal

The goal of this project was to begin the development and prototyping of the robotic platform. This platform was to be able to walk autonomously on level ground as well as lift and manipulate objects in its environment. The platform was also to be designed for the further development of swarm behavior in future projects.

2 Background

2.1 Ant Physiology

Ants, as do all insects, have three pairs of legs and three primary body segments (head, thorax, and abdomen). All legs stem from the thorax, within which are three sub-segments that have one pair of legs each.

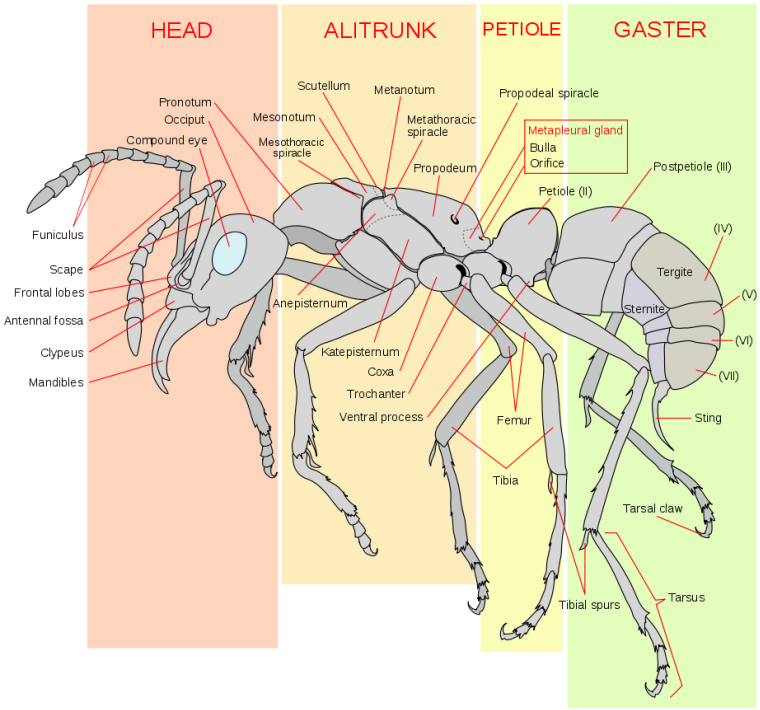


Figure 1: Anatomy of a *Pachycondyla verenae* worker ant (Ruiz, 2006)

The legs themselves consist of five primary segments: the coxa, trochanter, femur, tibia, and tarsus, in that order. The coxa, which means hip in Latin, is, as its name would suggest, equivalent to the human hip, and is what connects the leg and body. In practice, the coxa is primarily responsible for the swinging motion of legs, allowing the ant to move forwards and backwards. The trochanter is the smallest of the segments and in some cases is fused to the coxa. The femur and tibia also perform similar functions to their human counterparts, serving as the thigh and shin, respectively. The tarsus is actually comprised of several segments, but these are typically fused.

2.2 Ant Locomotion

In legged animals, the term gait refers to the stepping patterns used by a given animal in order to move. Though gaits are unique to each animal, they can be categorized into multiple different gaits, each with its own capabilities. For example, humans, being bipedal, are capable of both walking and running gaits. Ants, being hexapods, rely on the tripod gait for locomotion. This is a two-beat gait, where each step consists of the front and rear legs on one side and the middle leg on the opposite side making contact. As compared to the bipedal walking gate, which has one point of contact per step, and the quadruped trot gate, which has two points of contact per step, the hexapod tripod gate offers static stability in each step thanks to its three points of contact.

Ants, as well as essentially all insects, generate step patterns through the same mechanism, a type of neural network called a central pattern generator. On its own, a central pattern generator, or CPG, is capable of producing rhythmic neural signals which form the basis of a typical step. Each leg is paired with its own CPG, which is why an insect leg will continue to twitch in a regular pattern if separated from the insect's body. Insect legs are also home to a surprising number of biological sensors, including organs analogous to strain gauges. The CPG is able to use this feedback to adapt itself to varying terrain, making each leg an almost completely self-contained system.

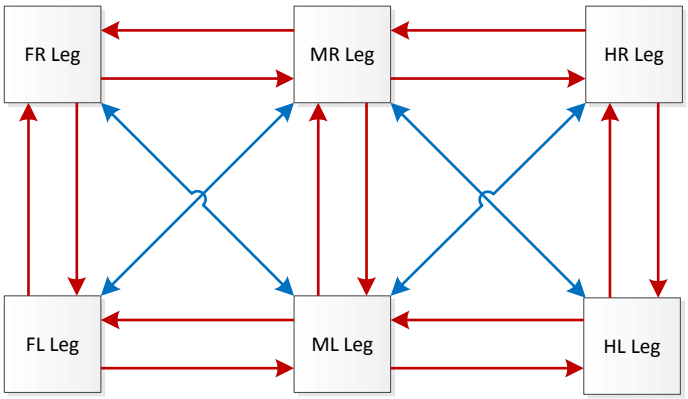


Figure 2: Simplified diagram of six-leg CPG network

To enable the insect to walk in a coordinated fashion, however, the legs must be able to communicate with one another. This is accomplished by several neural connections used to signal other CPGs. Any legs that are currently in the air, known as the "swing" phase, send a signal to stop legs directly adjacent to it from lifting off of the ground, as they are in the "stance" phase, responsible for supporting the insect. These can be seen as the red connections between CPGs in Figure 2. Only when the swing legs have

secure ground contact do the swing legs stop this inhibitory signal, allowing the swing legs to change phases. In addition, legs that are in the same phase can send a signal to one another to synchronize with one another, represented by the blue arrows.

With basic stepping coordination handled by the CPGs themselves, the central nervous system of the insect is essentially only responsible for providing signals to control speed and direction. These stepping phases are the defining characteristic of a gait. In the insect world, there are two predominant periodic gaits, known as the tripod and wave gaits.

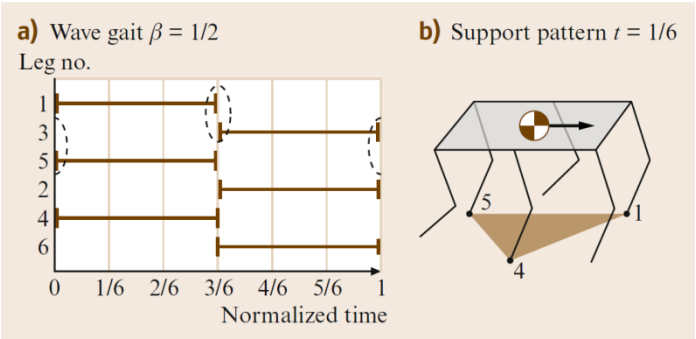


Figure 3: Phase diagram of a tripod gait (SpringerHandbookofRobotics)

As one might be able to extract after a quick glance at its name, the tripod gait always has at least three legs in contact with the ground at a time. This gait is primarily seen on six-legged (hexapod) insects. The legs are separated into two groups, each of which are comprised of the front and rear legs on one side of the body as well as the center leg on the opposite side of the body. This leaves a roughly equilateral triangle-shaped contact polygon. As half of the insect's legs are being repositioned per step, the tripod gait is an inherently faster gait.

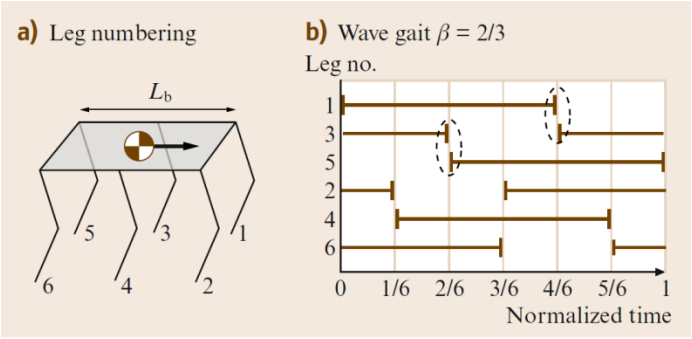


Figure 4: Phase diagram of a wave gait (SpringerHandbookofRobotics)

The wave gait, on the other hand, is what is used by insects such as the millipede, where the legs take steps in a wave that travels forward along one or both sides of the body. The increased number of feet in contact with the ground is beneficial for stability, but does have an adverse effect on top speed due to the much longer time period needed to make a full step cycle.

2.3 Gait Synthesis

Emulating and replicating the system used by insects, if not all animals, to determine where to step is currently one of the biggest challenges in the field of legged robotics. Generating the synchronized, periodic motion of each leg is trivial, but the ability to control the position and orientation of body parts within the robot frame of reference poses a much greater challenge. As such, it has been the subject of extensive research over the past few decades, and a wealth of experimental data and proposed solutions is readily available.

One solution of note is the virtual model system, where a model of the robot --- (Pratt, 1995). Using a combination of virtual forces acting on different parts of the robot, it becomes possible to finely control the position and orientation of individual parts of the robot, such as ground clearance and strafing or side-stepping.

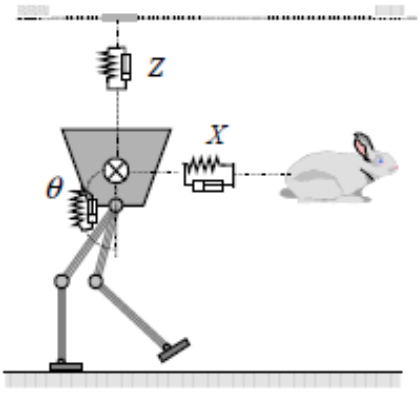


Figure 5: Virtual model control representation of biped robot (Hu, Pratt, Chew, Herr, & Pratt, 1998)

These virtual forces are simulated through the use of virtual springs and dampers, as illustrated in Figure 5. Here, X controls forward velocity, Z controls height, and θ controls body orientation. For example, X connects the model of the robot to a virtual rabbit moving at a desired speed, analogous to the rabbit

used on a dog racetrack. In all cases, springs are used to generate the virtual forces needed to move in a desired direction, while the damper ensures that it will do so at a certain velocity. In the case of Z and θ, these will typically have no velocity component. Therefore, the damper has no effect on the motion of the system, and can be removed from the model.

In the case of a hexapod robot using the tripod gait, the solution can be broken up into two discrete systems: one representing the body and stance legs, and the other being the swing legs.

The first system, representing the stance legs, is what determines the actual motion of the robot. A set of virtual forces is applied to the body of the robot to control any desired parameters. The system must then be solved to determine what forces will be seen on each leg, at which point necessary joint motion can be determined. This step is the most difficult of the process, but can be solved using a combination of sensor feedback and careful model simplification and reconfiguring.

With the body and stance legs' motion determined, the motion profile of the swing legs can now be found. The net motion of the swing legs should be opposite that of the rest of the body, so that after the phase transition, the new stance legs are able to maximize their range of motion.

However, careful attention must be paid to the position of the center of gravity (CG) relative to the contact polygon, not only of the stance legs, but also the future contact polygon of the current swing legs. The model ensures that the contact polygons overlap at the transition between phases, with the CG located at some point within the intersection. By doing so, the robot should remain statically stable once the swing legs lift during the transition to swing.

2.4 Ant Navigation

When searching for food or other needed materials ants tend to travel in random paths with many turns and overlaps. This method of searching for food may seem inefficient; however, with different numbers of ants searching for food, the method of searching will change slightly to optimize its efficiency. The randomness of these paths is determined by the number of ant searching for food. As the number of ants searching for food decrease, their search paths become straighter and increase in distance away from the nest. With the fewer ants searching, straighter paths will reduce the probability of searching an area already searched by another ant. With few numbers of ants, searching already searched areas is a waste of limited manpower. As the number of ants increase, the paths overlap more because the small wastes of searching an area already searched can be afforded. Once an ant has found a source of food it

will return to that source until the scent of the food is gone. If no source is found the path the ant takes will rotate about the ant hill. (Gordon, 1997)

When returning back to the hill, ants travel in a nearly straight line no matter how complex their search path was. As ants travel, a "home vector" is calculated, which will lead them back to their hill. The home vector can be determined in flat planes, with no landmarks as shown in the figure below.

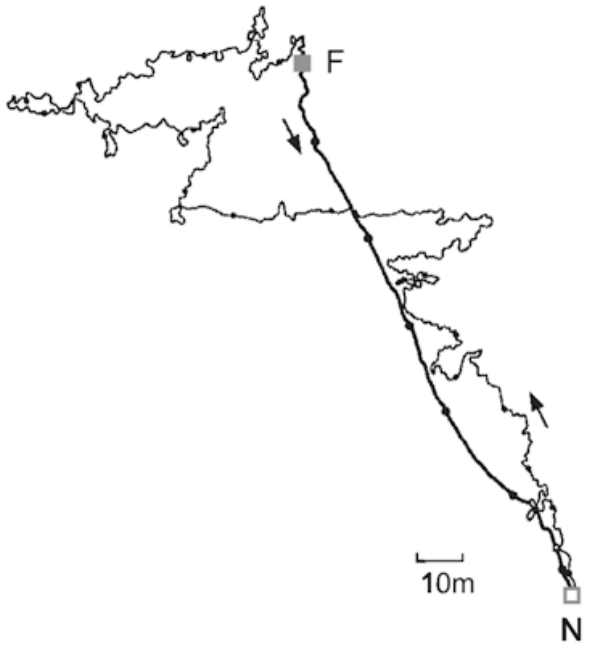


Figure 6: Search path and return path determined by a home vector (bold line) (Wehner, Desert ant navigation, 2003)

The orientation of the home vector is believed to be determined in two ways, dead reckoning and magnetic particles in the ant's body. Dead reckoning uses path integration and the sun as a reference to determine the path back. Magnetic particles have been seen in some ants and can be used as an internal compass (Acosta-Avalos, Wajnberg, Oliveira, Leal, Farina, & Esquivel, 1999). To calculate the distance away from the hill, ants count steps along its search path. By knowing the length of each step along its search path, the ant is able to calculate the distance needed to travel along its home vector. If an ant is displaced while traveling along its home vector, the ant will continue to follow its previous path as if it was not displaced. If an ant reaches the end of its home vector and the hill cannot be found, the ant will search randomly until it is found. In the figure below lines in blue show ant's paths as they continue towards the hill, represented by N, along their home vector after being displaced, point R, from a food

source represented by F. Lines in red shows paths taken by a “zero vector” ant, or an ant that is not foraging when displaced thus having no home vector.

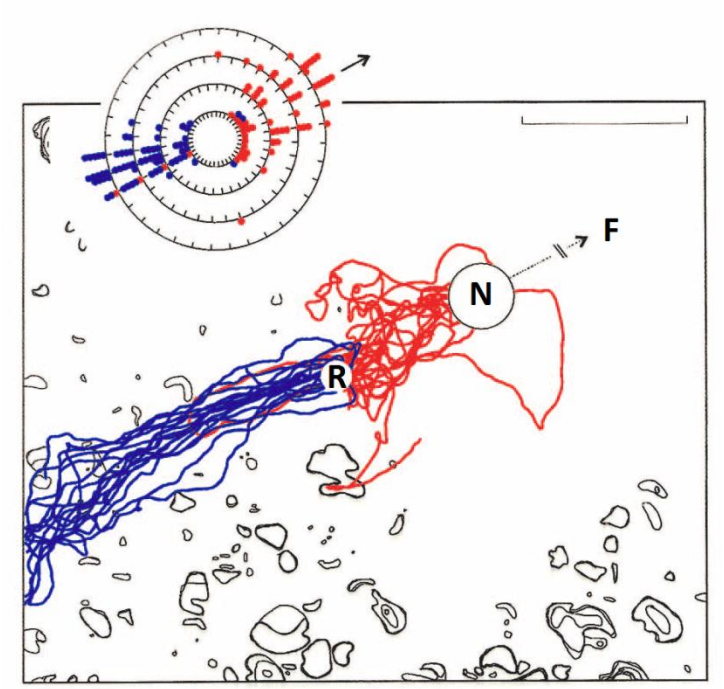


Figure 7: Ants continuing along their home vector after being displaced (Wehner, Barbara, & Antonsen, Visual Navigation in Insects, 1996)

Despite being able to navigate flat, object-less planes, ants can use landmarks, such as rocks, to make navigation easier. Landmarks are used for both locating the hill and relocated food supplies. When a zero vector ant is placed away from the hill, it will randomly search for the hill until it is found. However, if landmarks are placed around the hill, the ant will locate the hill using the landmarks. This was shown in an experiment done by R. Wehner, B Michel and P. Antonsen. In this experiment zero vector ants where displaced from ant hills with and without landmarks. Ants, displaced from an ant hill, without landmarks, into an open plane, will search for the hill in arbitrary paths. If ants are displaced from a hill with landmarks around it, into a plane with similar landmarks, it will first search the area where the ant hill would be in respect to the landmarks before search arbitrarily (Wehner, Barbara, & Antonsen, Visual Navigation in Insects, 1996).

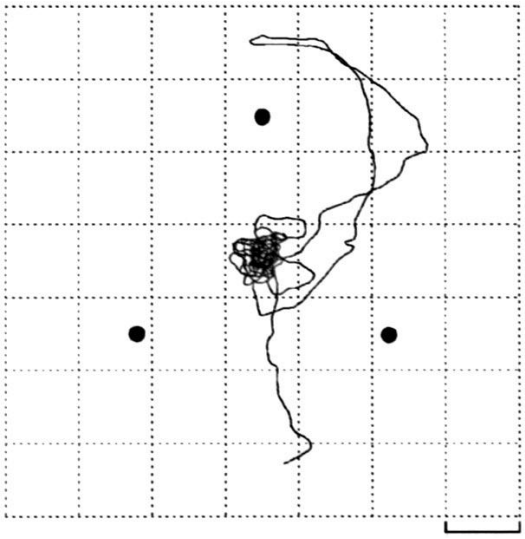


Figure 8: Search path for hill of an ant trained to look for the hill in the center of three cylinders (Wehner, Barbara, & Antonsen, Visual Navigation in Insects, 1996)

This ability to use landmarks as reference points is also used when returning to sources of food. In an another experiment performed by R. Wehner, as the number of landmarks increase along the path to the source of food, the faster the ant can replicate that path (Wehner, Barbara, & Antonsen, Visual Navigation in Insects, 1996).

2.5 Ant Hierarchy

The hierarchy of ant colonies was briefly researched to determine how ants are assigned there individual roles. This was done in regard to how the robots would get their communications and how certain tasks would be carried out. In general ants are born into their respective roles based on different genetic attributes. For example, the protectors of some colonies are not only bigger in size but also have larger poison glands. (Holldobler p.255) It's genetic differences like this that distinguish the roles of each ant. Another example is the difference between major and minor workers; in some colonies, the minor workers lay the scent trail whereas, the major workers just follow it. The minor workers in this scenario have glands particular to laying these paths but the major workers do not. (Holldobler p.255) Despite the different roles, there is a tendency to have underlying actions that are the same; the attack or defend reaction for example.

2.6 Communication

The communication of ants was researched to get a better understanding of how exactly ants communicate with one another. The robots would then be modeled after this communication as closely as possible. The main communication technique of ants is chemically through pheromones, or as stated in *The Ants*, "If any single generalization applies to all of these categories [of communication], it is that chemical signals pervade them." (Holldobler p. 227) This type of communication is the most difficult to emulate electronically, therefore other means of communication were noted. The other forms of communication utilized by ants are acoustic, tactile, and visual. Acoustic communication between ants is primarily transmitted through the ground via vibrations. For example, when danger is in the nest, workers may stop moving and the defenders will increase their pace and move toward the vibration, wherein attacking anything that moves. (Holldobler p. 256) Another form of acoustic communication is "verbal" where the ants make some sort of noise, such as a small chirp. The noises can even be as loud "as 75 decibels at 0.5 centimeters from the major worker" (Holldobler p.257). The tactile communication is simply differing touching portrays different commands depending on ant species; exchange of food between ants from one ant touching in a certain area for example. Visual communication is said to be at best very minor, but there have been instances where vision was the only explainable solution to communication. For example "when lone workers of the Formica lugubris foraging in a field encounter an insect, they dash in erratic circles around it and attract others workers that happen to be in the vicinity."(Holldobler p. 259)

3 Methodology

3.1 Design Specifications

With the goals given and with preliminary research the following design specifications were developed:

- Ant-like Hexapod
- Weight of 5 pounds
- Maximum length of 16 inches
- Minimum payload of 2 times the body weight (10 pounds)
- Minimum walking speed of 2 times the body length (36 inches per second)
- Operating life of at least 30 minutes
- Able to navigate over obstacles 2 inches in height

3.2 Design Objectives

In order to develop the robot in an efficient manner the project was separated into three systems: mechanics, controls, and electronics. The mechanics system entailed the design and construction of the physical robot. The controls system included the simulation of the robot to predict how the robot would work and the development of software. The electronics system included the development and fabrication of the devices needed to drive the physical robot.

The project was divided in this manner so that each system could be developed simultaneously. Each system was independent enough so that it was not reliant on the day to day progress of the other systems. Additionally, each team member could be placed as a "system lead." The duty of this role was to ensure that the assigned system was on-track for completion, not the sole team member responsible for the work needed to be done.

An iterative process was used to better the design of the robot. The iterative process would allow the robot to be designed in major stages. The stages would lower the overall cost of producing the robot and simplify fabrication.

3.3 Design Considerations

To further define the performance of the robot beyond the design specifications multiple design considerations were developed. The first of which is to develop a control system that could move the robot fluidly and accurately. This included the development programing that could control the robots center of gravity and react to changes in the walking environment.

With the consideration in mind that the robot would be further developed in future projects; all systems were designed so that they could be worked with and improved upon by people unfamiliar with the initial design. The final iteration of the systems would need to be developed in way that made the simple to understand and modify. Furthermore, each system needed to have the ability to be built upon with new or alternative components.

Finally, the robot would need to be design so that it can be manufactured in a cost effective manner. To build a large swarm of robots, individual units must be inexpensive. This would allow to be replaced when damaged or not a financial burden if placed in a search and rescue or bomb disposal situation.

4 Design and Fabrication

4.1 Mechanical Design

The mechanical design of the robot was a large part of this project. There were many aspects that needed to be considered because of where the robot would be. The first of which was materials.

4.1.1 Materials

Due to the size and weight limitations and expected payloads, the selection of materials was limited. The three primary choices of materials were ABS plastic, aluminum, and Acrylic, each with their own design considerations.

Material	Density (lbs./in^3)	Modulus of Elasticity (ksi)	Yield Strength (psi)	Ultimate Strength (psi)
ABS Plastic	0.044	320	5300	8200
Acrylic	0.043	406	11700	7830
Aluminum	0.0975	10000	40000	45000
Lexan	0.043	1160	10000	14000
Titanium	0.163	15000	40000	50000

*500,000,000 cycles completely reversed stress

Table 1: Material properties

4.1.1.1 ABS Plastic

The Dimension SST 1200es fused deposition modeling (FDM) rapid prototype uses ABS plastic as its work material. This machine "prints" .01 inches layers of the ABS plastic into any shape or form with a resolution of .006 inches. Each layer is a tightly bound matrix of a light weight, low density plastic with relatively low strength. Each layer of light weight plastic is bordered by a denser and stronger form of the same material. The combination is approximately 0.044 lbs./in^3 and has an ultimate tensile strength of 8200 psi. In addition, because the plastic is printed, complicated shapes and internal structures without labor intensive manufacturing.

When designing with ABS plastic specific things need to be considered to ensure that the part does not fracture because of the way it's made. First, the Dimension rapid prototype machine can only properly

print the plastic matrix in circular path along its vertical. When circular features are printed along the horizontal axis, the circle cannot be printed perfectly because of the printing process. The boarder of the circle is drawn in steps of 0.1" because of the layers rather than an actual circle. The denser boarder material can also be easily pulled off the inner material because of the stepped edges.

Tolerance is another important consideration when designing parts made of rapid prototyped ABS plastic. The ABS plastic can expand up to approximately .005 inches. This small amount needs to be especially considered when incorporating screw or shaft holes and when designing features that interface with other parts made from ABS plastic. Holes need to be increased a minimum of .01 inches and interfacing features need to be decreased in size .005 inches.

Finally, minimum thicknesses need to be taken into account when using ABS plastic. The thickness of the border plastic is .05 inches resulting in a minimum printing wall thickness of .1 inches. However, no inner material will be added in between the two boarder walls and this lack of material will promote shearing. To compensate for this, the recommended minimum wall thickness is .125 inches.

4.1.1.2 Aluminum

Aluminum has a density of 0.0975 lbs./in^3, 2.2 times denser than ABS plastic making aluminum much heavier than the plastics. Aluminum is much stronger than acrylic, Lexan, or ABS, but due to weight limitations, aluminum needed to be used only in circumstances requiring much greater strength. Additionally, all parts of aluminum need to be machined by the group, requiring more labor hours than parts made of plastic.

4.1.1.3 Acrylic

Acrylic is a stiff, but brittle plastic that can be cut with the VSL4 Laser Cutter; making it an ideal candidate for quick prototyping. It has a density of 0.043 lbs./in^3 but the decrease density comes at a cost with dramatically reduced yield strength at 11700 psi. In addition, Acrylic does not deform plastically well and breaks easily under impact loading. To avoid cracks due to impacts acrylic should only be used in areas where forces are distributed through multiple contact points.

As previously stated, to manufacture parts made from acrylic sheets a laser cutter can be used up to a thickness of .375 inches using the XX laser cutter. The laser cutter can cut various profiles with a tolerance of -.0025 inches. This is because the laser cutter follows the line making up the profile. The

laser makes the shape by removing material at a thickness of .005 inches on both side of the profile boarder.

4.1.1.4 Lexan

Lexan is a form of polycarbonate with great tensile strength and flexibility with a density of 0.043 lbs./in^3. Due to its flexible properties, Lexan would be used where there would be high impact loading and stiffness was not a necessity. Lexan must be machined, i.e. not laser cut or printed, and therefore would take longer to produce parts from. The material's properties, however, allow it to be used in high impact situations, but with large deflection.

4.1.1.5 Titanium

Titanium is a very stiff and very strong metal with a density of 0.163 lbs./in^3. This is the heaviest of the materials considered for the robot, but also the strongest. Due to the very high strength, less material could be used, reducing the weight of the overall part. Machining titanium is extremely difficult and time-consuming; therefore, it would only be used when very high strength was necessary.

4.1.2 Degrees of Freedom

One design consideration that needed to be approached was the number of degrees of freedom (DOF) that each leg would have. This decision would weigh heavily on the complexity of the leg system and the design time. An ant has four degrees of freedom per leg but one of them is passive. Seeing as the last joint is passive and serves to give the ant more grip on surfaces, the group decided to negate the functionality of the last joint.; the choices were narrowed to two and three degrees of freedom. When considering the two options, the main arguments were maneuverability versus robustness; the more DOF the leg has, the more agile it is, but it also allows for more places of failure.

4.1.2.1 2 DOF

A leg with two degrees of freedom would only have forward and backward shoulder movement and vertical upper leg movement. This would mean the leg-span of the robot would be fixed at all times. Using a leg with only two DOF would certainly make the design and build of this project easier and faster; there are fewer parts involved as well as fewer systems to account for. Along with being easier to design, this simplified approach would also be more robust; having more parts and more points of motion only adds to the probability of something breaking. Having less mass and fewer objects to move reduces the load on the motors as well as the computational load on the processor. Having few degrees

of freedom would also mean fewer motors which would require less power; overall making the lifetime of the ant longer. With all these benefits of a two DOF system, the robot would be able to walk in a short period of time. However, leg motion would be severely limited and hinder many necessary actions of the ant.

The first large issue that comes from a simplified leg is lack of balancing; the robot cannot widen its stance to accommodate an awkward or heavy load. The robot would be severely limited to the terrain it could traverse. If the ant were to pick up a load and attempt to walk across uneven terrain, it would not be able to account for the changes in pitch as it walked. The robot would have the ability to move the point of contact with the ground closer to or further away from the body, but that would awkwardly weight the robot and cause some motors to work much harder than others. There is also the possibility of making the body unstable and moving the center of gravity (CG) of the robot into a bad location, thus causing the robot to tip over.

The robot would be limited in the number of tasks it could accomplish via the two DOF leg. Any sort of clinging or holding would be determined by the shape of the leg; it would be like trying to grasp an object with a hand but only being able to move the first set of knuckles. Another example would be trying to push something with its front two legs, the robot would have to walk while holding the legs out in front of it in order to get any significant travel. The robot would not be able to right itself even if the shoulder had 180° of rotation due to the orientation of the leg. The objects that the ant would be able to get over would be limited to the range of vertical motion.

The simplified design would be easier to make walk but the motion would be not be ideal. When walking with a tripod gate, three legs are in contact with the ground at all times, so when the robot moves forward, it moves on three supports. Thinking of the body as the stationary object and the legs and ground as moving, the legs will make three swinging arcs as they move backward to propel the body forward; this is assuming no friction. When friction is involved, the legs will not want to make arcs, but straight lines because the foot will not want to move outward. Trying to force the legs to move forward in an arc will cause something to give, either the foot placement or the legs themselves if the legs cannot physically move outward. Moving the legs in a straight line would require moving the body up and down, which is not the ideal case when trying to move with a load over uneven terrain. This would also cause issues when trying to raise or lower the body of the ant for similar reasons. When trying to stand, the legs would have to pull themselves closer to the robot, again depending on lack of friction.

This would put a very large torque on the motor and possibly cause it to stall. To get a better idea, it would be similar to a person do the splits and trying to drag his/her feet inward to stand up rather than stepping in and standing up. Overall, with a two DOF leg, the robot's maneuverability would be severely limited and would only make it easier to get walking sooner.

3 DOF

A leg with three DOF is overall more complicated than a two DOF leg, but much more maneuverable and agile. Having the added degree of freedom also better replicates the leg of an ant. When referring to the third joint, it will be called the elbow joint of the leg. The elbow would add another vertical motion and separate the leg into and upper and lower leg.

A leg with three DOF is overall much more capable than a two DOF leg. The problem of balancing the robot when carrying a load becomes much more achievable with more articulation. For instance, the leg-span can be increased to better stabilize the robot, but does not have to lower the body at the same time. When traversing uneven terrain, having more motion in the leg allows for better placement of the leg and proper articulation for movement.

The walking issue that arises when using a two DOF leg is solved with a three DOF system. When moving the leg backward, the elbow joint can remove the force created by the legs trying to push outward. This will allow for smoother walking cause fewer problems when moving. The added joint will add another motor to the leg and make the leg itself heavier. The additional motor means more initial power is required during movement, but in the end, the amount of power required to walk may be equal. When walking with three DOF, the most torque the leg would see would be accelerating the leg backward to move the body forward, whereas, with a two DOF system, the most torque the robot would see would be when it repeatedly lifted the robot and payload to prevent buckling. More joints means more motors, thus having to move more weight when walking will cause the motors to have to work harder and therefore draw more power. Relieving that strain and distributing the movement between three motors would even out the power draw and make the use of three motors rather than two, for power consumption reasons, obsolete.

Adding another joint to the leg will also increase the lifting capability of the robot. When lifting an object with its legs (i.e. getting underneath an object to lift) having the elbow joint will allow the ant to position its legs so that the load exerted on each leg is evenly distributed between the two motors. If there were

only two DOF, there would be no way to control how much torque is placed on the vertical motion motor. Only having one joint for vertical motion also means the feet must be dragged across the ground to lift the body. The added friction would severely reduce the lifting capacity of the legs and would give the 3DOF leg an advantage of more than 2 times the lifting capacity than the 2DOF. The added assistance of the placement also means the motors don't have to work as hard and would therefore not use as much power.

Another benefit to using a three DOF leg is that it would have the ability to right the robot if it were to end up on its back. The elbow joint would allow for the necessary articulation to roll the robot over and stand back up.

Due to the nature of this project and the kind of terrain that the robot will be traversing, a three DOF leg would be more suitable. This will make the design of the robot a little more complicated, but it will also immensely improve the ant's ability to navigate a trash heap.

4.1.2.2 Modularity

Due to the harsh environment of a landfill, the group anticipated the robot needing repairs after some time of use. Therefore, the robot was built with modularity in mind so that the parts of the robot could be easily replaced. To help make this more achievable, the robot was split into four sub-assemblies; the central body, legs, tail and head. The central body was designed to be two identical plates to which all other sub-assemblies would attach to. The head and neck assemblies were designed for mounting with minimal hardware. The legs, which were the most complex sub-assemblies, were designed to be identical irrelevant of position on the body. This way only one leg had to be constructed and replacement would be much easier.

The sub-assemblies, were designed so that each were independent of one another. This means that each could be built, repaired or modified without requiring the disassembly or removal to other sub-assemblies.

4.1.3 Robot Design

The robot was separated into four major sub-assemblies; the body, legs, head and the tail. Each system was designed separately and went through multiple design iterations. The early stages served as a base for the team to build upon and adjust future work accordingly.

4.1.3.1 Legs

Most hexapods drive their legs with a direct drive system. This is by far the easiest solution because there is no system required to transfer power. What makes this system easier to build and control is also what makes it worse for the hexapod needed for this project; the motors are on the legs. This exposes the motors to harsher conditions and makes the legs heavier.

One large consideration when choosing which leg system to go with was size and weight. The less space taken up by the system, the smaller the entire robot can be made; direct drive is the smaller of the two systems. Direct drive removes the actuator from the body of the robot and puts them out on the legs themselves. This is also lighter because there is no mechanism necessary to transfer the force from the source to the respective joint, although it moves all the weight of the actuator to the mechanisms being moved. This significantly increases the inertia and momentum of the leg which will therefore require a larger actuator. This is where performance becomes a very important part in the design process.

One of the goals of this project is to build a fast and agile ant-like hexapod, therefore, only considering weight and size would not be sufficient. As stated before, moving the actuators out onto the legs makes them heavier, which in turn makes them harder to move. Since the legs of the cable driven system are not moving the weight of the actuators, they are able to achieve much higher accelerations than the direct drive. This acceleration affects the stride frequency which will determine how fast and how quick the robot moves; the cable driven system can therefore move faster than the direct drive. Having more weight on the legs also makes it harder to hold the legs in a wider stance and the wider the stance is, the longer the stride. With a longer stride, the robot can reach higher speeds, therefore allowing the cable driven system to travel faster. The acceleration of the cable driven system is also more consistent over a larger range of speeds, giving it the ability to make agile moves at higher speeds. The direct drive system is, however, able to push off, or launch, faster from a stand still. This gives the direct drive a better launch from a stand still. (Bowling 1-5) A part of being agile also involves the ability to balance and maneuver obstacles and the legs play a very large part in balance.

Most often when something loses balance, the initial reaction is to readjust footing. The secondary reaction is to use limbs to counteract the imbalance. With lighter legs and the mass of the actuators being in the body, adjusting the center of gravity would be less effected as the robot moved. This means that if the robot was trying to adjust for an imbalance, the movement of the legs would not cause the COG to move much from where it is. This would make balancing the robot easier because it would not

have to react to changes to COG while balancing. The risk of moving a leg in such a way that would cause the robot to tip is also significantly reduced.

By moving the actuators into the body, there is also more room for expansion; larger motors can be used for instance. The inside of the body is also a safer place for the motors to be. With the possible application of these robots, the environment will be anything but friendly to electronics; therefore, the fewer exposed electronics the better.

There were also more cons that were brought up about using the cable driven system over the direct drive; the largest being efficiency. With a cable driven system, the force exerted by the motor is transferred to the respective joint via a cable and pulley system, which causes the efficiency of power transfer much less efficient than direct drive. The cable driven system is also a much more complicated system, which allows for more points of failure.

Weighing the pros and cons of the two systems, the cable driven system fit the needs of the robot better than the direct drive. Therefore, the rest of the leg design was driven to accommodate a pulley driven system.

Shoulder

The shoulder of the robot was intended to hold the motors for each leg. This way, the attachment for the legs and motors would be held in the same assembly. Having the two attachment points in one piece adheres to the modular design goal.

The initial design of the shoulder, which can be seen in Figure 9, combined the use of pulleys and a direct drive. With this design, the forward and backward movements of the leg are directly driven while the other joints are controlled by a pulley driven system. Driving the outer joints with pulleys meant that the motors could be held close to the point of forward/backward rotation. Having the forward and backward motion of the leg directly driven means that the pulleys for the other two joints are always aligned. With the motors close to the forward/backward rotation, the center of mass of the leg is moved significantly closer and therefore reduces the mass moment of inertia for moving the leg. With this configuration, the ant can move the legs quicker to achieve a higher top speed. (Bowling 1-5)

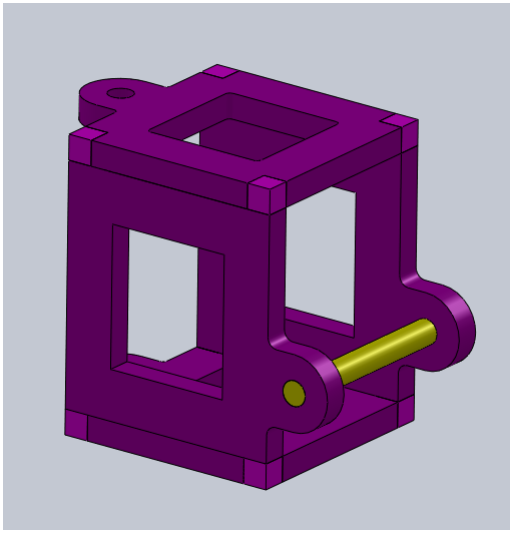


Figure 9: Initial Shoulder Design

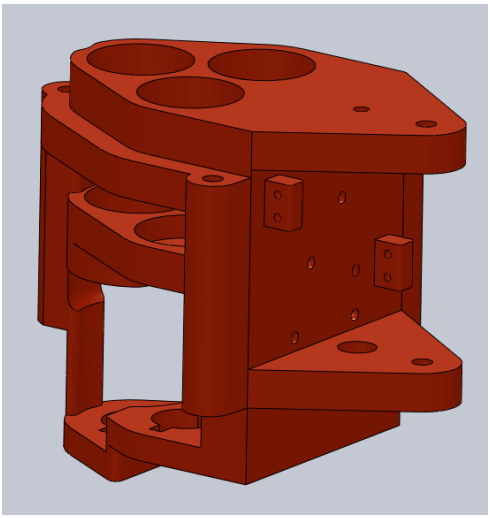


Figure 10: Final Shoulder Design

The original box design was also small and too weak. The area allotted to place three motors was about one cubic inch made from eighth inch acrylic for support. This was far too small to fit three motors in and was not strong enough to support the amount of torque that would have been placed on the structure.

As the team did more research on pulley systems, a way of rerouting the cables was discovered in a research paper. The concept is similar to that of a bicycle break line; the sheath over the line also acts as a guide. More information on the guidance channels can be found under Pulley Guidance System section.

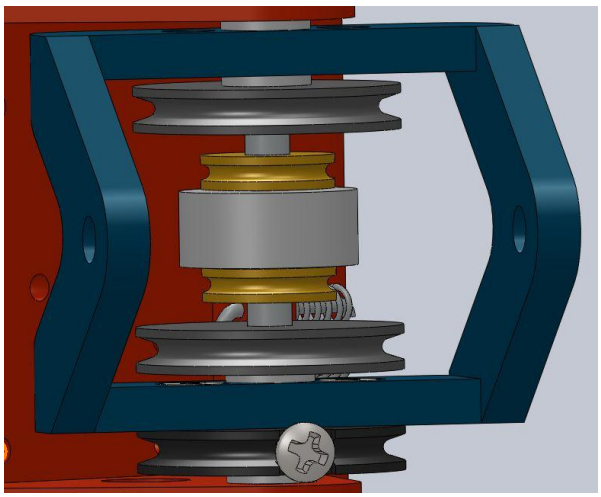


Figure 11: Shoulder Joint

The design was changed to make all of the legs joints, pulley driven with the motors set vertical. The three motors required to drive the legs make up most of the shoulder size. The motors mount onto a plate to hold them in a triangular pattern. This allows all three lines to be perpendicular to the guidance wall without placing them in a straight row, making the shoulder more compact. The rear motor in the triangle is placed lower than the other two by .25 inches to prevent the rear cable from running into the other two pulleys. The material of the shoulder quickly changed to ABS printed plastic because of the complexity of the piece.

The axle for the shoulder joint needed to be somewhat removed from guidance wall in to accommodate the rotation of the rest of the leg. To accomplish this, two holes for the shoulder axle are placed on two protrusions on the opposite side of the guidance wall from the motors. The axle holes are placed 0.75 inches away from the supporting wall to allow for 180 degrees of motion in the upper leg. On the shoulder axle, there are five pulleys that serve various purposes. Referring to Figure 11, the black pulley is used to control the motion of the shoulder joint, i.e. the forward and backward motion of the leg. The screw in the pulley is used to mount the cable to prevent the cable from slipping during operation. The grey pulleys are used as idler pulleys to redirect the cables used for upper leg motion and the gold for lower leg motion. These idler pulleys are necessary to keep the cables perpendicular to the guidance wall as well as remain in-line with the channels of the other pulleys further down the leg; see Figure 12. Without the idler pulleys the cables would begin to cut into the walls of the guidance channels and possible jump off their pulleys further down the leg. The white cylinders are spacers to keep the pulleys in line with the others.

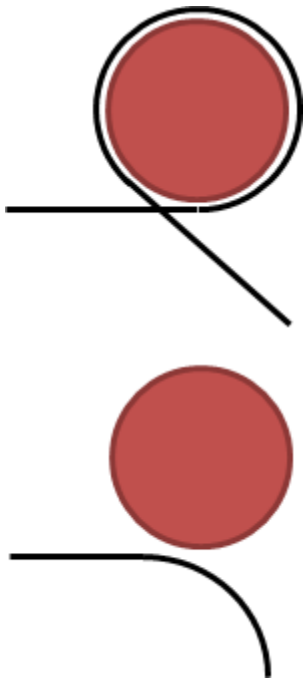


Figure 12: Idler Pulley Cable Wrapping

Below the motors, there is another surface to hold the potentiometers; which are used to measure the shaft rotation. The holes for holding the potentiometers are placed in the same orientation as the motors and slightly oversized to allow for proper alignment. The shafts of the potentiometers help to support the end of the motor shaft to prevent the shaft from being fully cantilevered. The final iteration of the shoulder can be seen in Figure 13.

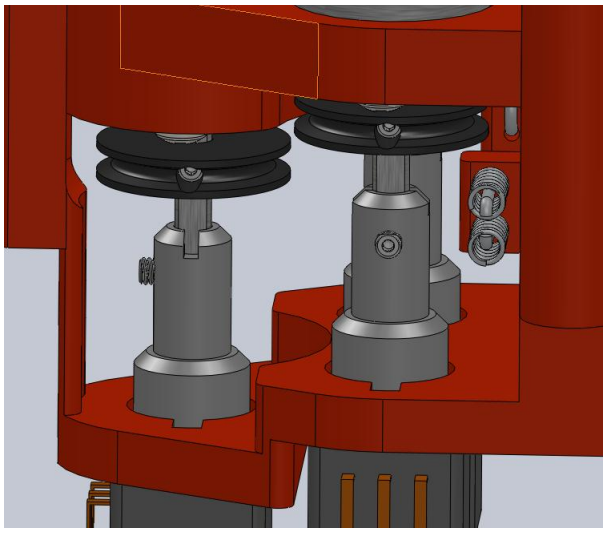


Figure 13: Motor Support

To attach the upper leg to the axle of the shoulder, a separate piece was needed; the shoulder joint, which can be seen in blue in Figure 14.

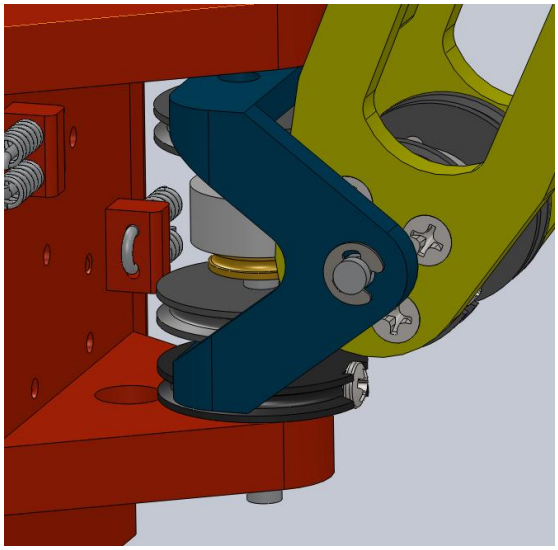


Figure 14: Shoulder Joint

This part acts as the connection point between the moving leg and the shoulder, therefore, it has to be able to withstand all forces expected to be seen by the leg. The shoulder joint also serves as a protective barrier for the rest of the robot. If the leg were to see a force much greater than was expected, the joint would give, protecting the parts in the body from the violent jerk. This is similar to the crumple zone of a car or a single use shock. The part could not interfere with the cables as the joint rotated and therefore adapted a "V-like" shape.

Upper Leg

The upper leg of the robot was loosely designed around that of the average carpenter ant. The body of the robot is much larger than that of a normal ant; therefore, the proportions of the body and leg are a little off. The length of the upper leg was more designed around necessity of function.

The original design, which can be seen in Figure 15, was made of eighth inch acrylic and glued together. Acrylic and glue were used for ease of use because the group wanted to produce an initial model quickly. The leg was built in box form which gave the structure great strength with minimal material. The initial design was drastically changed to accommodate the necessary pulleys and tensioning assemblies. As the design became more complicated, the material also changed to ABS plastic so that it could be

made in the rapid prototyper, or RP machine. Using the RP machine enable the pieces to be one solid piece. This type of construction makes the piece stronger and allows for more complicated geometry.

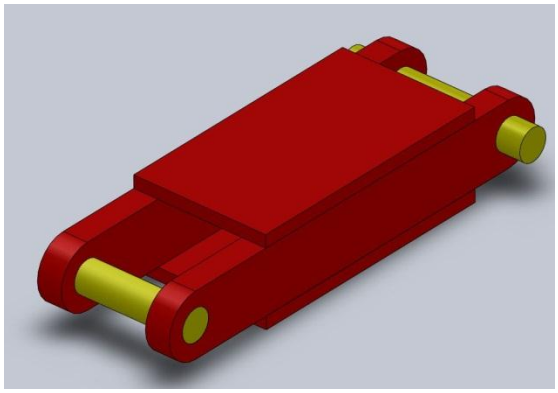


Figure 15: Initial Upper Leg

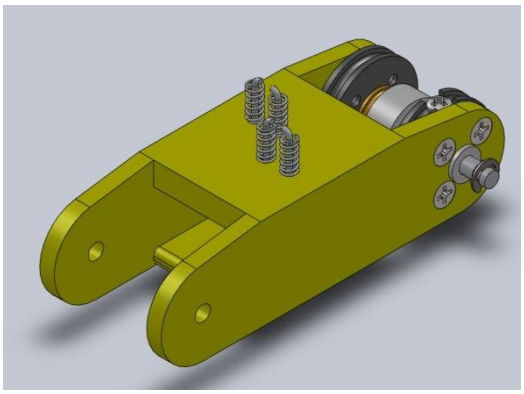


Figure 16: Latest Upper Leg

The driving factor for the size of the upper leg became cable navigation and available pulleys. The cables for the lower leg need to be kept inside the upper leg for protection, so the idler pulleys for those cables are placed in between the motion upper leg pulleys, represented in gold in Figure 17. The driving pulleys, represented in grey in Figure 17, were sized based on how they are mounted. The most effective way to mount the drive pulleys was to screw them to the side walls. Hardware includes four 3-48 screws to help distribute the load on the ABS plastic. The smallest available pulley with enough material to support the screws had a diameter of 0.75”. The width of the upper leg is dependent upon aligning the pulleys on the upper leg joint with the pulleys of the shoulder joint. The height of the upper leg is driven by the cable path from the idler pulleys on the upper leg joint to the pulley on the lower leg; the structure must not interfere with the cables.

The shaft of the upper leg joint is dead, meaning it just floats and isn't used to move any piece, to help reduce friction between pieces and to limit the use of shaft keys. Trying to use shaft keys in pieces this small would be both difficult and ineffective. Shaft keys do not do well in plastics, especially printed ABS.

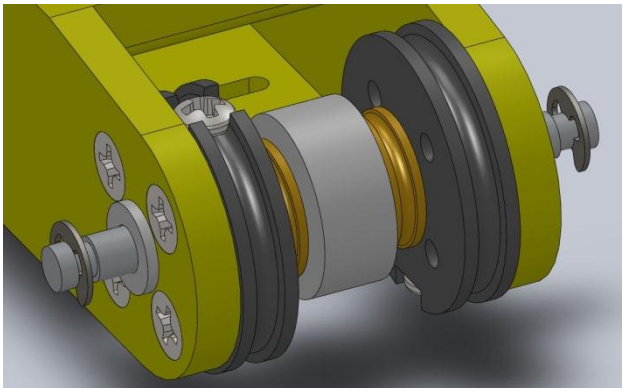


Figure 17: Upper Leg Joint

The screws in the drive pulleys are offset by 180° so that when the leg is horizontal, the screws are facing up and down. This allows for the upper leg to rotate a full 180 degrees without the cord running back over the screw. The cable that runs onto these two pulleys is done in differing directions (clockwise and counter-clockwise) allowing the leg to be pulled in either direction. The smaller 0.375", idler pulleys (seen in gold in Figure 17) serve the same purpose as the idler pulleys in the shoulder joint. The spacers help to keep the idler pulleys in alignment with the pulleys on the shoulder axle. The shaft is held in place by e-clips on either end. The shaft layout is designed to allow for some movement, but only enough so there is minimal friction between each piece.

Lower Leg

The lower leg of the ant was designed by necessity of what the robot needed to move properly. The original part, which can be seen in Figure 18, was made from acrylic. This was in part due to its ease of construction.

Figure 18: Initial Lower Leg

Figure 19: Second Generation Lower Leg

Figure 20: Latest Generation Lower Leg

As the design evolved, however, it became more complicated and the material changed to ABS plastic so that it could be printed in the rapid prototype machine. The length of the lower leg is contingent on how much clearance was needed below the robot when walking in a predetermined ideal walking gate. The walking gate was designed to put as little stress on the motors as possible. This means the legs needed to be kept as close to the body as possible to reduce the amount of torque on each joint. The initial leg was straight, but evolved into a bent leg so that the leg could reach further under the robot.

The design of the leg is meant to be one piece to make the assembly of the robot a little easier. This section does not require any external pulleys because contains a barrel that accomplishes the same task, which can be seen in green in Figure 21.

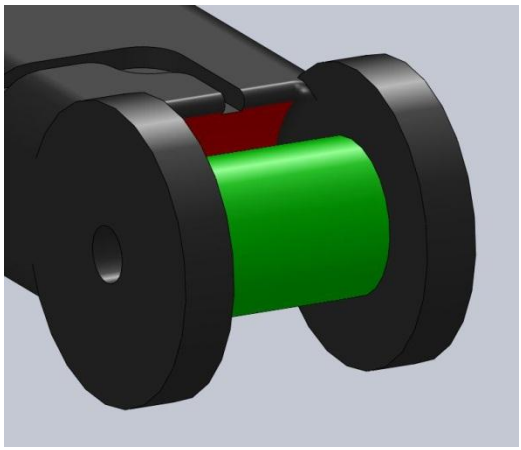


Figure 21: Lower Leg Barrel

The cable wraps around the barrel like any other pulley, but is then anchored to a bolt that goes all the way through the leg. To be able to pull the leg in both directions, the cable is wrapped around the pulley both clockwise and counterclockwise. The shaft for the lower leg joint is also passive and is an eighth inch in diameter. The only pieces on the shaft are the lower leg and two spacers used to reduce the friction between the leg sections. The spacers are there to keep the lower leg from rubbing on the upper leg and causing unnecessary friction. The shaft is held in place with two e-clips.

As the robot became weighed down with more hardware, the lower leg evolved even further. Originally the lower leg was designed to replicate the look of an ant's leg; however, the bend in the leg was not optimal for walking. A new curve was placed in the leg so that it contacts the ground at a ninety degree angle and directly below the lower leg joint. The size of the inner barrel also put much more tension on the cable than necessary. The direct length of the lower leg from the point of rotation to the tip of the

leg was 4.85”. A force tangent to the barrel of 1lbf would create a torque of 4.85in-lb on the joint. This divided by the radius of the barrel (0.1825”) creates a tension of 26.57lbf on the cable. Essentially, with the initial barrel and longer leg, the tangential force exerted on the leg is multiplied by 27.5 times in the cables. A multiplicative factor of that magnitude meant that the robot would barely be able to hold itself up if the lower legs were pointing straight out. To fix this, the diameter of the barrel was doubled to reduce the multiplicative factor to approximately 13 times. All of these calculations were done with the worst case scenario in mind with the lower legs pointing straight out. To put it into perspective, it would be similar to a human doing the iron cross. This is not a scenario the robot should encounter often, but something that needed to be considered and could be easily solved. The

Tensioning System

When using a cable driven system, slack development in the lines is a constant issue. Therefore, the team decided to implement a tensioning system for the leg cables. There were two different options the team considered: active and passive tensioners. Active tensioners continually apply tension on the line even as the line extends. This is usually accomplished by some manner of sprung system. A passive tensioning system applies a set amount of pressure once so as the line slackens, the amount of tension on the line lessens. Active tensioners are the better option because they do not require any maintenance after implementation. With the passive tensioners, someone would have to continually tighten them until the lines can no longer stretch.

The initial tensioning design implemented live tensioners so that as the robot was used and the cables wore in, the line would stay taught. The tensioners were made from compression springs and music wire. The music wire would be bent into a “U” shape and the compression springs would cause the music wire to pull taught on the cable. The cad of this can be seen in Figure 22. As the design of the rest of the robot progressed and the robot grew, the springs required to put apply enough tension to the cables were too large to fit on the upper leg. The team decided to wait to see if tensioners were needed because a passive tensioning system could be easily implemented using the mounts for the active tensioning system.

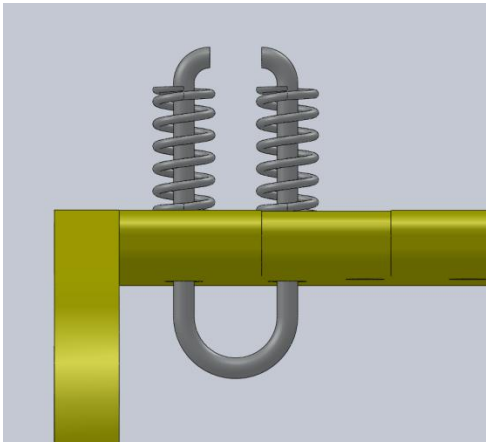


Figure 22: Active Spring Tensioner

Motor Selection

The motors selected to drive this robot were selected largely based on size and power. The ant needed motors small enough, but powerful enough to keep the robot within size and power design specifications. The motors needed to produce a maximum of approximately 2.5 in-lbs. of torque at a speed of 60 rpm. This number was calculated for the worst case scenario, i.e. running at full speed with the maximum load, both by hand and via SolidWorks. The objective was to find motors that had at least this amount of force in the smallest package possible.

Joint Torques

To calculate the torque required at each joint, the robot was broken down to simpler sub-systems. This way each joint became a simple geometry problem. With the walking gate of ants, the worst case, static, scenario will be the most force the robot will see. As the ant walks, the leg or legs that are moving do not see much force from the mass of the robot.

In Figure 21, the legs are shown perpendicular to the body, but this is not the worst case scenario. The largest load will be seen when the legs are at the beginning of the walking stroke because they are accelerating fastest.

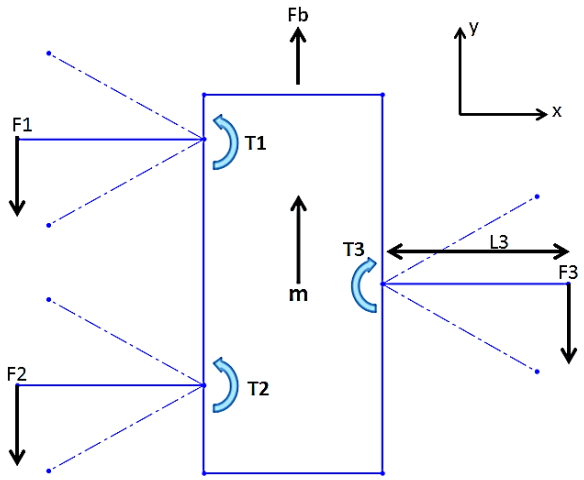


Figure 23: Shoulder Torque Free Body Diagram

$$F1 + F1 = F3$$

$$\sum Fy = Fb - F1 - F2 - F3 = 0$$

$$m = 15lb$$

$$a = 40\frac{in}{sec^2} = 3.33\frac{ft}{sec^2}$$

$$Fb = \frac{m * a}{g}$$

$$F3 = \frac{Fb}{2}$$

$$\sum M = T3 - F3(L3)$$

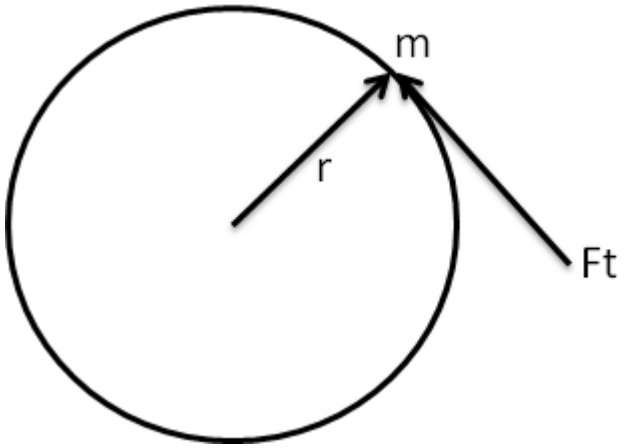
$$T3 = F3(L3)$$

T3 is the torque required by that leg to move the body forward at the desired acceleration.

Torque Required to Move Leg:

- m = 0.13lb
- center of mass is 1.72” from point of rotation
- need to move at 60 rpm to reach desired speed, $\omega = 2\pi \frac{rad}{sec}$
- reach desired speed in 0.5 seconds:

$$\alpha = \frac{2\pi \frac{rad}{sec}}{0.5sec}$$



$$T = F * L = Ft * r$$

$$T = m * \alpha * r * r = m * \alpha * r^2$$

$$Totol\ Torque\ Required = Tb + Tl$$

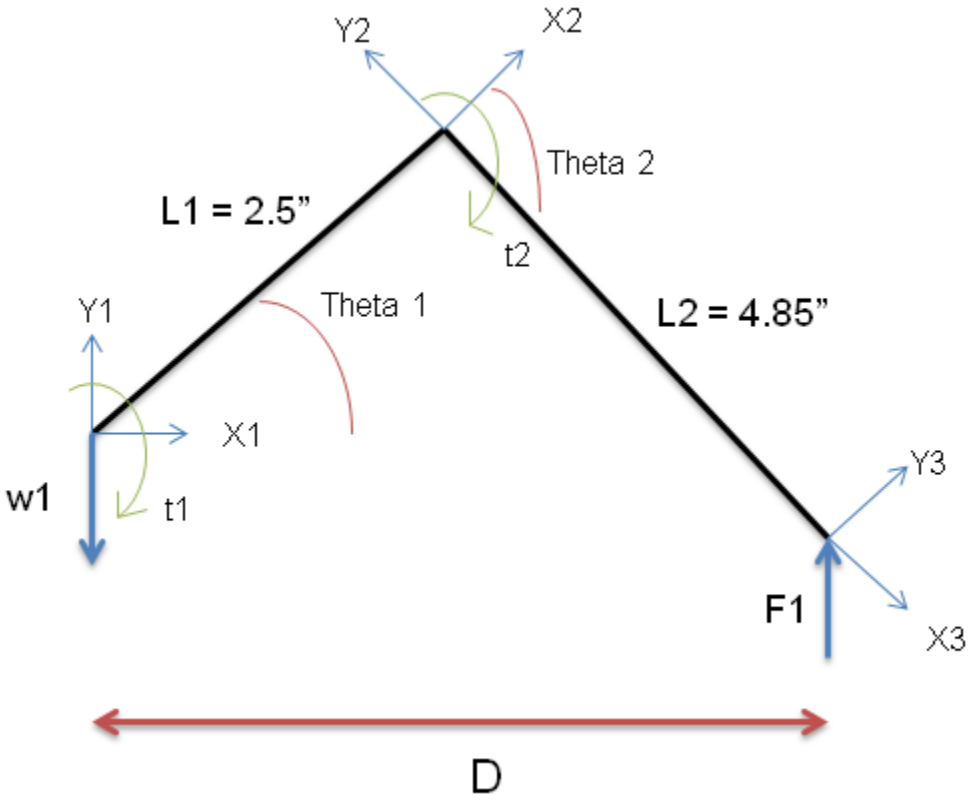
Using the variables for the worst case scenario:

- 5 pound robot
- Tri-gate walk
- Moving at 2 body lengths per second

- Accelerating to top speed in 0.5 sec
- 10 pound payload

From the equations and conditions above, the robot would require 2.337in.lb of torque from the shoulder joint motor.

Calculating the torques required by the other motors was more dependent upon position.



$$d = \cos(\theta 1) * l + \cos(\theta 1 - \theta 2) * l2$$

$$\theta 3 = \theta 2 - \theta 1$$

$$Fx3 = F1 * \cos(90 - \theta 3)$$

$$Fy3 = f1 * \sin(90 - \theta 3)$$

$$T1 = F1 * d$$

$$T2 = Fy3 * l2$$

From these equations, the robot would need over 7 in-lbs. of torque to lift itself from a full spread out position. This is far too much for a motor of small stature to accomplish. Therefore, the team implemented algorithms in the code to control how the robot lifts itself and what positions the legs go to depending on weight.

Motors

The first motor the group looked at was the Micromo 1331 with a 94:1 gear reduction. This was chosen because it produces approximately 3.5 in-lbs. of torque at 60 rpm, has an integrated encoder, and a precision gear box. This meant that no other monitoring device would need to be purchased because it already came with the motor. This is ideal because mounting the encoder or potentiometer can be a hassle and cause design problems. The motor is also a good size for this application; it is 13mm in diameter and 31mm tall. Upon further inspection of the motor, it was discovered that the motor cost around $300. Considering the robot required a minimum of 21 motors, that price would put the budget far too high.

The second motor that was considered for the robot was the BaneBot FF-050 motor with a 118:1 gear reduction. This motor does not output as much torque as the previous and is not as high quality. Despite the lesser power, the motor produces approximately 2.75 in-lbs. of torque at 60 rpm, which is just enough for this robot. This motor is slightly thicker than the first at 16mm in diameter, but is also slightly shorter. The largest draw back to this motor is that it does not come with a built in encoder, therefore, an external position monitoring device had to be placed somewhere else on the robot.

Despite the BaneBot motors lesser power, larger size, and lack of integrated sensors, the price for the motor and an external potentiometer totaled to $20. The BaneBot motor had the necessary power and fit well within the budget. For these two reasons, the BaneBot motor was chosen to be the primary drive motor for all systems.

Pulley Guidance System

As previously stated, a pulley system needed to be designed to transfer the power from the motors in the body to the joints on the legs. The most complicated design aspect was how to mount the motors such that the cables would be in line with the necessary pulleys. The team tested several configurations, but the most space efficient was having them stand vertically. This caused problems because trying to

route the cables to the necessary points required several extra idler pulleys and the shoulder mount became too large.

To get the pulley cables from the motor to the shoulder joint an alternative method needed to be found to move the six cables to their respective location in minimal space. To accomplish this, a series of .1 inch diameter tubes are incorporated into a .25 inch thick wall; which also connects the two brackets holding the motors. The tubes start tangent to the motor pulleys, travel through the wall, and end at a point tangent to pulleys on the next joint. This means the cables enter and exit the guidance wall perpendicular to the surface of the wall. The tubes move inside the wall without intersecting one another and act as small pulleys inside the wall similar to the guide sheaths on bike brake lines. Although, the reduced size comes at a cost of increased friction and wear on the part as well as the pulley cable, the alternative option was not viable. The guidance channels can be seen in Figure 24.

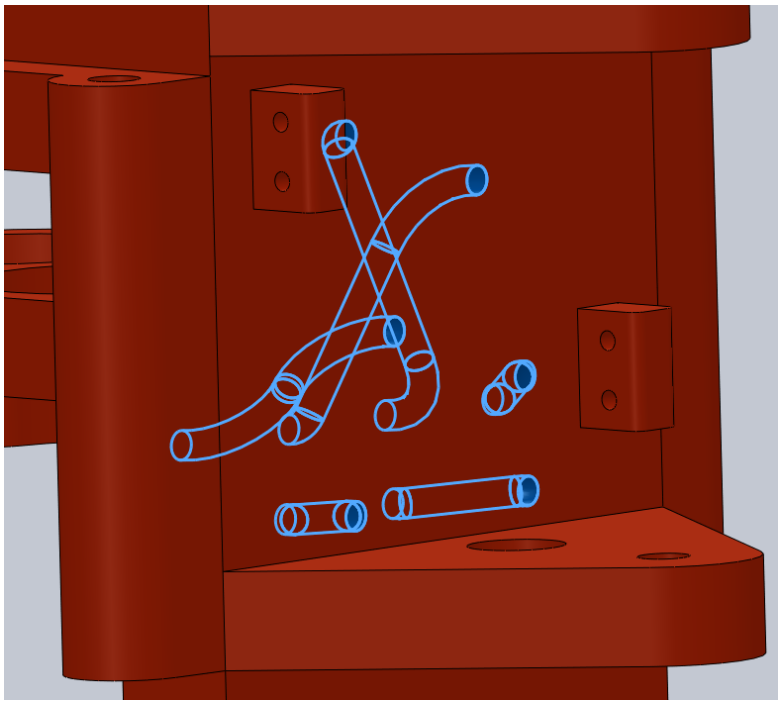


Figure 24: Guidance Channels

4.1.3.2 Tail

The tail of an ant is used for two primary functions, store pheromones and balance. For this robot, the tail is used for balance and a place to store the batteries and power management unit (PMU).

The batteries are placed in the tail for heat and weight reasons. Batteries under stress tend to put off a lot of heat and with the body being filled with motors and electronics, placing the batteries in the tail would help prevent heating issues. The batteries are potentially the densest components on the robot and therefore act as the best counterbalance for lifting. The whole point of having a counterweight is to try to keep the center of gravity (CG) as close to the center of the body as possible and therefore must be able to move.

There were two ideas in the beginning, a 2 DOF and a 1 DOF tail. The benefit of having 2 degrees of freedom is more precise control of CG manipulation. There would be vertical and horizontal motion to help the robot keep its balance. The added precision however, would make the system more complicated to produce and to control. With the 3 DOF legs and their ability to quickly react to balance the robot, the group decided to go with the 1 DOF tail.

The motion of the tail is vertical so that it can counteract a weight being picked up by the pincers. The target lifting weight of the robot is approximately two times its own weight; which is around ten pounds. Therefore, the goal is to be able to counteract the moment of a ten pound weight approximately four inches in front of the robot. Four inches came from the estimate of how long the pincer/head/neck assembly would be. There are six batteries and each only weighs 80.5 grams (0.18 pounds), which equates to only 1.08 pounds. Ideally, that means the CG of the tail would need to be ten times the distance from the center of the body as the pincers are, to combat a ten pound load. Seeing as the ant is supposed to be relatively small, having a tail that long is not plausible. Therefore, the design was concentrated largely around trying to keep the tail short while still keeping the CG of the robot within controllable ranges. If the CG were to wander outside the ant's footprint, the robot would begin to tip over and put an immense load on the front legs. Although the tail must be able to counteract a ten pound load, it must not cause the robot to unstable during operation with no load. This means the tail must have the proper shape and range of motion to keep the CG as close to the center of the body as possible when there is no load in the pincers.

The PMU is attached to the tail because it makes sense to have it close to the power supply, i.e. the batteries. There is also limited room on the body and the tail can be slightly modified to make room for the printed circuit board (PCB).

Iteration 1

The first design of the tail was an attempt to place the batteries in a compact formation with just enough room to fit the PMU. This iteration is dubbed the “Wiley Coyote” tail because it looks like a slew of rockets strapped together, as can be seen in Figure 25.

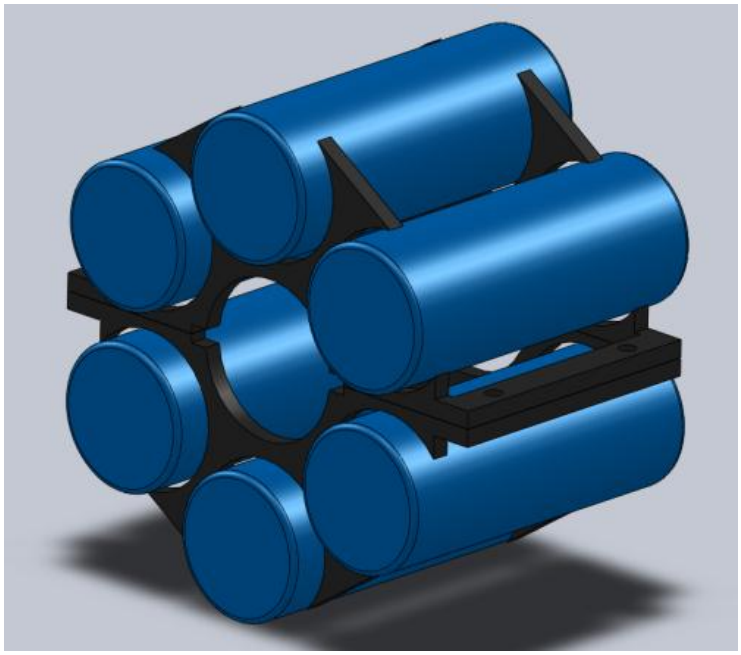


Figure 25: First Iteration of Tail

The compact layout was essentially this design’s fatal flaw; with the CG of the grouping being so close to the tail mount, it does not provide the necessary weight distribution necessary for what the robot needs.

Iteration 2

The second tail design pays more attention to the CG issue and extends the center of mass of the tail further back. The design is meant to be simple and hold the batteries with enough room for expansion; hence the vertical positioning that can be seen in Figure 26.

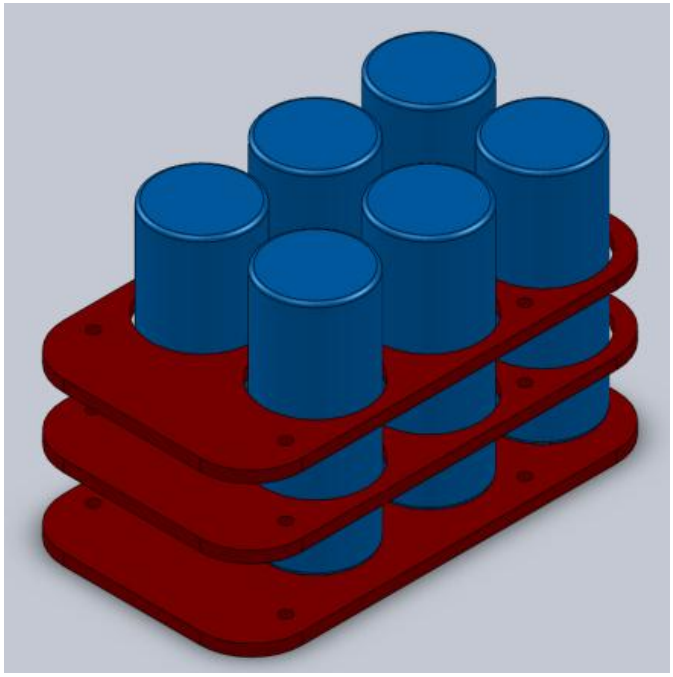


Figure 26: Second Iteration of Tail

This design, however, does not affect the CG of the robot enough to counter a ten pound load. The shape of the holding plates (seen in red) also causes problems when trying to balance the robot without load; the tail cannot get far enough over the body to discount its own weight.

Iteration 3

The third design moves the CG back even further by staggering the batteries in an attempt to keep the size of the tail down. The offset of the batteries also helps when trying to swing the tail over the top of the robot because the initial battery is set further back from the mount. The third iteration can be seen in Figure 27. Despite the attempt to move the CG further back, it was still not enough to balance the robot.

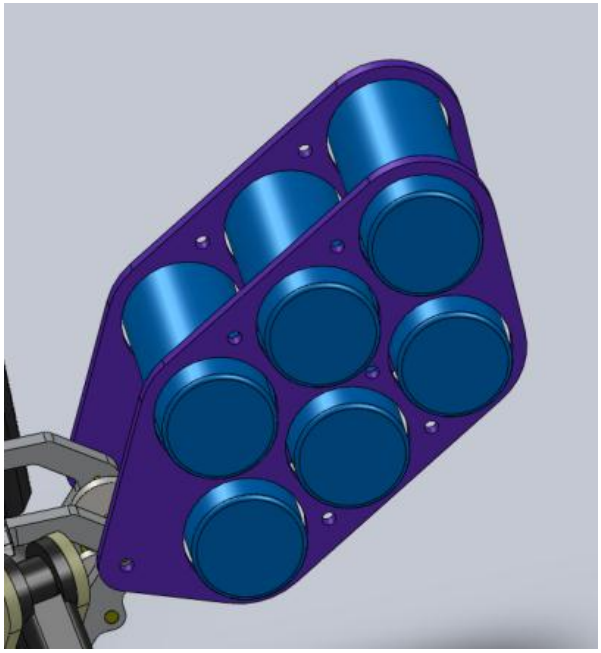


Figure 27: Third Iteration of Tail

Iteration 4

The fourth design of the tail moved the CG back further still by putting the batteries in single file; which can be seen in Figure 28. Having the batteries in a straight line moves the CG as far back as possible. As can be seen in the picture below, the last two batteries are angled 45° from the rest. This allows the CG of the tail to get far enough over the body when the tail is up to keep the overall CG of the robot close to center while still being able to counterbalance a ten pound load. The connection points of the tail are placed above the batteries for two reasons: 1) So when the tail is protruding straight back, the tail doesn't move the CG of the robot higher and 2) The tail is able to curve further over the top of the body without interfering with anything. The material of the tail would be Lexan for its shatter resistance properties.

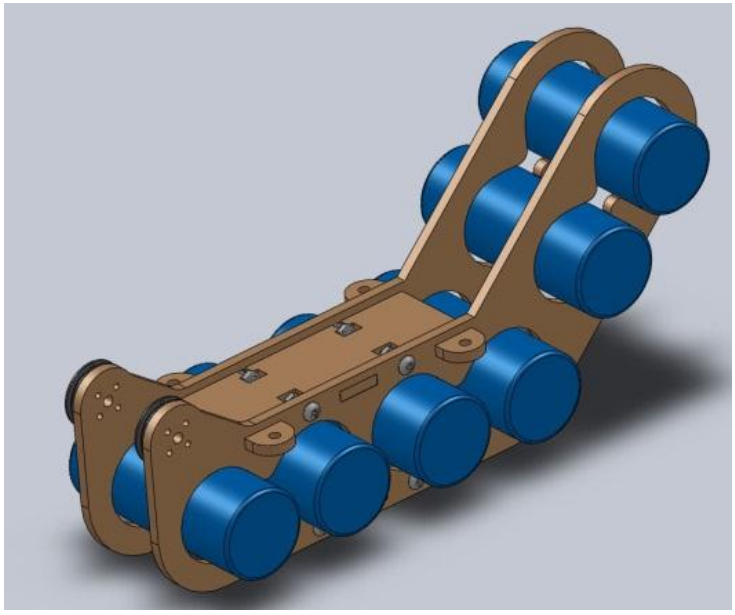


Figure 28: Final Iteration of Tail

As previously stated, each time a new tail was designed, an analysis of the CG of the robot was executed. Below are images of the three tests done for the final iteration.

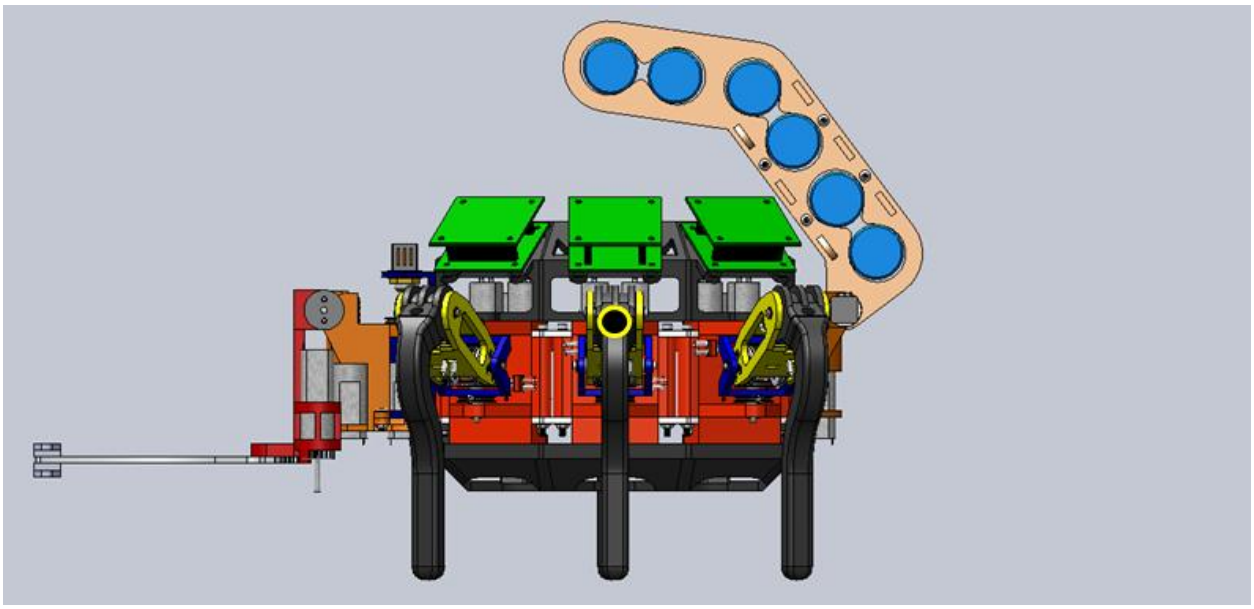


Figure 29: CG of Robot in Walking Position, No Weight

In this position, the CG of the robot is 0.04” off center.

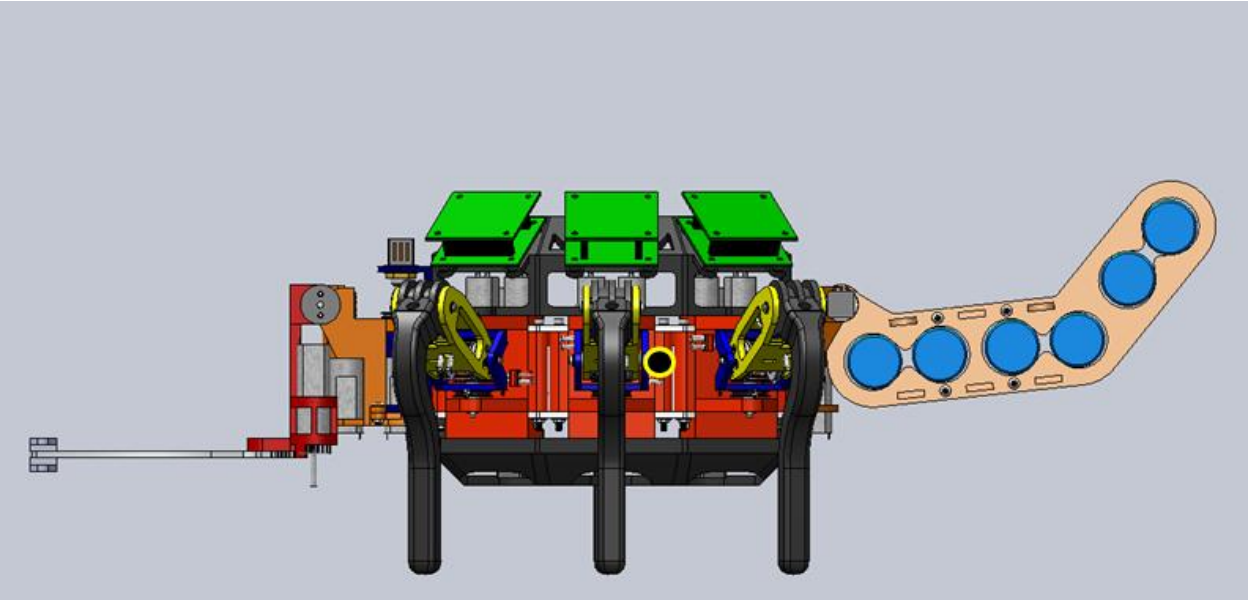


Figure 30: CG of Robot with Tail Out, No Weight

With the tail out and no weight in the pincers, the CG is 0.92” off center.

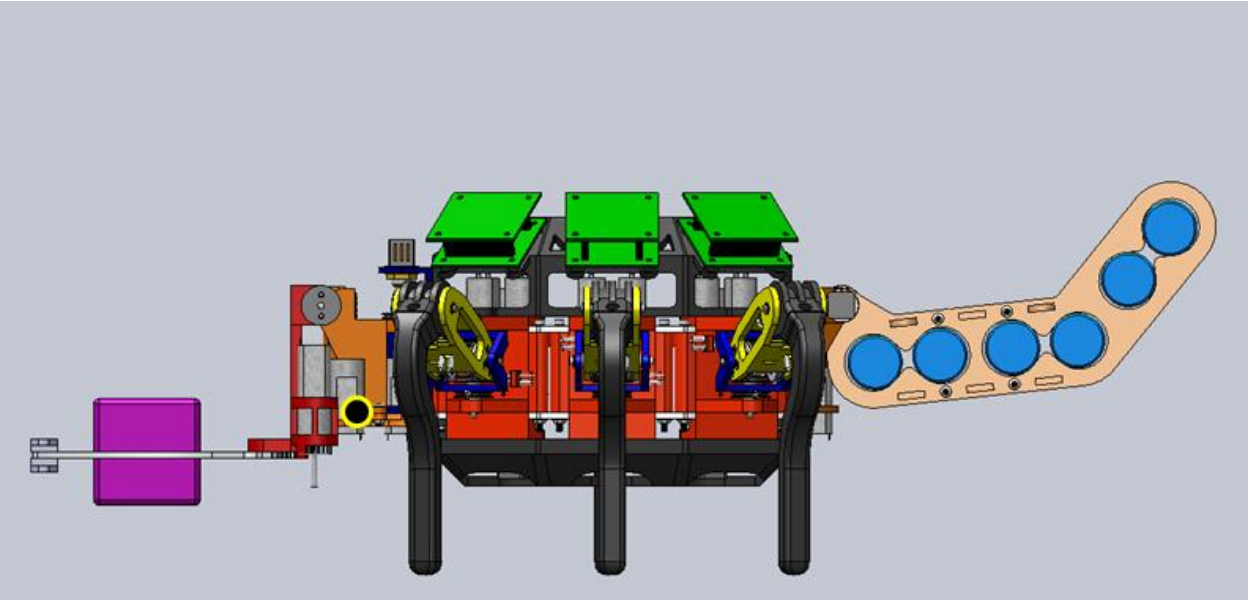


Figure 31: CG of Robot with Tail Out, 10lb in Pincers

With 10lb in the pincers and the tail out, the CG of the robot is -4.78” off center. This is very far forward, but still manageable. The CG is not yet outside the reach of the legs and therefore is still stable.

As previously stated, the ideal situation would be to have the batteries ten times further from the center of the body as the pincers are, but that's is just not feasible. Therefore, the tail was designed as best it could be with the weight and space available. There is always the possibility of adding dead weight to the tail, but that would also raise the overall weight of the robot, which is undesirable.

Tail Mount

The tail attachment needed to be sturdy enough to carry the weight of the tail as well as handle potential impacts. In the initial design, the tail mount is made from aluminum because the possible stresses that could be placed on it. The tail is driven by the motor via a worm gear to prevent back-drive. While the robot walked, the tail would only adjust for large corrections, causing the tail to remain stationary for the majority of the time. The worm drive also helps to keep the tail in position without having to continually run the motor; saving energy and preventing the motor from overheating. This iteration can be seen in Figure 32.

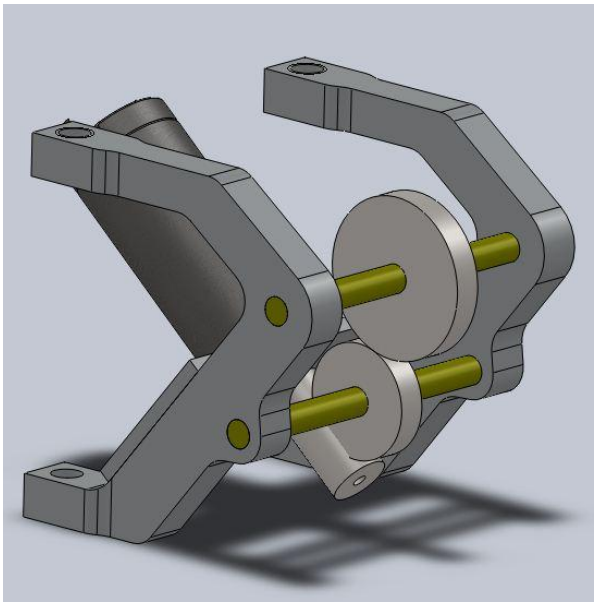


Figure 32: Initial Tail Mount

As the design of the rest of the robot progressed and the rest of the materials were being chosen, the tail mount was the only metal piece other than the mounting hardware and shafts. If the tail were to take a substantial impact, it would tear out of the body and shatter the body plate. The team decided that the mount should be redesigned so that if something were to happen to the tail, the rest of the body would not be destroyed.

Trying to make assembly more uniform, the same number of mounting points used on the shoulder was used on the tail mount. The structure is designed around motor and drive actuation. To save weight and accommodate the complicated geometry involved, it is made from ABS plastic so that it can be printed with the rapid prototype machine. The tail mount also acts as a mount for the motor that will move the tail. An image of the tail bracket can be seen in Figure 33.

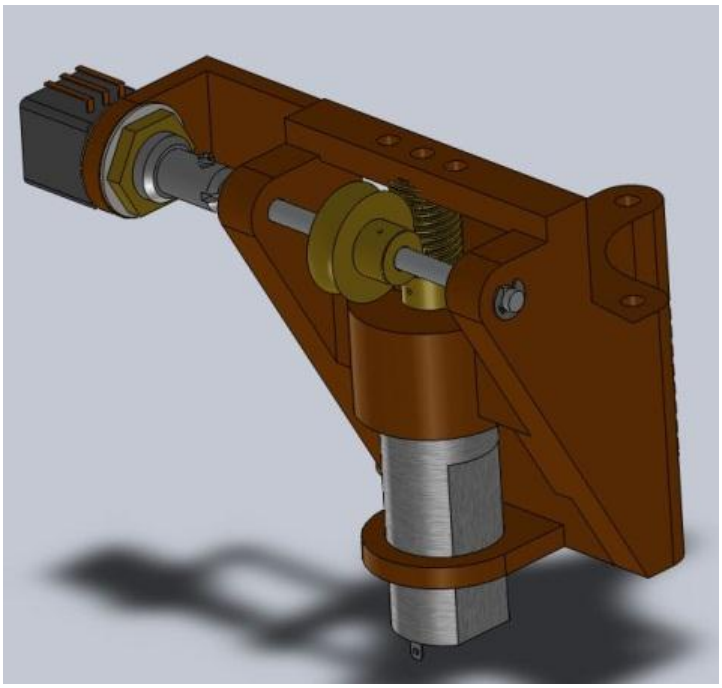


Figure 33: Final Tail Mount

4.1.3.3 Head

The head assembly was designed to lift objects and have the option to include environmental sensors. To do this a two DOF neck joint was chosen. The two motions include an up and down motion to lift objects and a side to side motion for panning the horizon. A rotational, third DOF was considered during the initial planning but was not considered necessary for the project. To accomplish the needed motion for lifting objects, the assembly was divided into three parts; the panning joint, the lifting joint, and the pincers.

The two neck joints are placed one after another so that their motion motions were compounded. This style of joint was chosen over a single joint that could provide both motions, similar to a universal joint, because it is both less complicated to design and easier to assemble. To lower the torque requirement on the lifting joint the panning joint is placed first in the assembly. This required the panning joint to

both provide the side to side motion and provide a sturdy mounting point to the body of the robot. The pincers are placed last in the series of joints so that they can move in any direction.

Panning Joint

The panning motion joint provides a side to side motion as well as a solid mounting point to the body for the head assembly. The motion is controlled by the same FF-050 motor used in the legs and tail assemblies. Unlike the tail and leg assemblies, the panning joint directly drives the next joint. By connecting the two joints together with a direct drive reduce the number of parts needed in the joint and make assembly easier. Direct drive is possible because the torque required to move the joint is low.

The motor is located in a brace where the shaft points upwards and the opposite side of the motor is flush with the lower body plate. The shaft then extends through a modified .75 inch pulley and is connected to it via a set screw. The pulley is screwed into the moving part of the joint. The moving part of the joint has outer ring that sit on an inner ring located on top of the motor bracket. This ring has thin layer of lubricant on it to prevent wear in the contact points.

On the bottom portion of the panning joint a brace is placed around the stationary part of the joint and is mounted onto the moving portion. This brace holds the moving portion of the joint along the axis of motion and also prevents the moving part of the joint from lifting off of the stationary portion. It is connected to the moving portion of the joint using two 4-40UNC-2A machine screws and nuts. This screw size was chosen because similar parts were used in other assemblies and reusing the same screw simplifies the final bill of materials. A thin layer of lubrication is also placed on the joint to prevent any wear on the contact points.

The motor is oriented vertically so a potentiometer can be placed on the shaft without sacrificing clearance on the underside of the robot. The mount for the potentiometer protrudes from the stationary portion of the assembly and extends above the end of the motor shaft. Similar to the shoulder mount, the hole for the potentiometer threads is loosely fit so the motor shaft and the potentiometer shaft can be properly aligned.

The motor brace is located outside of the central body to minimize modifications needed to the body plates. This design requires no modifications for the bottom body plate; however, to fit the bracket for the potentiometer one modification is needed in the upper body plate.

The panning joint was designed to be made out of ABS plastic, similar to the other joints. ABS plastic was used to keep weight down and reduce manufacturing labor hours. If loads exceed the strength of the joint the parts can be modified to be made out of aluminum.

Lifting Motion

The lifting joint is a single DOF joint connected to the panning joint and controls the height of pincers. To reduce cost the FF-050 motor was also used for this joint; however, the torque required to lift the pincers is greater stall torque of the motor. To compensate for the lack and torque and to lower the current usage of the motor when holding a load, a worm gear was added to control the joint. To reduce the number of different parts in the robot, the same worm and gear were used as in the tail sub-assembly. This worm and gear provide the assembly with over 2 times the needed torque to lift a payload of two times the robot's body weight.

To accommodate the motor and the worm gear, both were placed vertically, similar to the panning joint. The worm gear transfers the rotation of the shaft perpendicular to its axis allowing for the proper movement of the pincers. The gear is attached to a shaft located in two protrusions from the main part of the joint. To accommodate the potentiometer a bracket extends off of one of the two shaft protrusions. The bracket for the potentiometer is placed at a height so that it will not limit the range of motion of the panning joint.

Pincers

The pincers were designed as the robot's primary means of interacting with objects and the world. As with the rest of the head/neck assembly, the FF-050 motor was used.

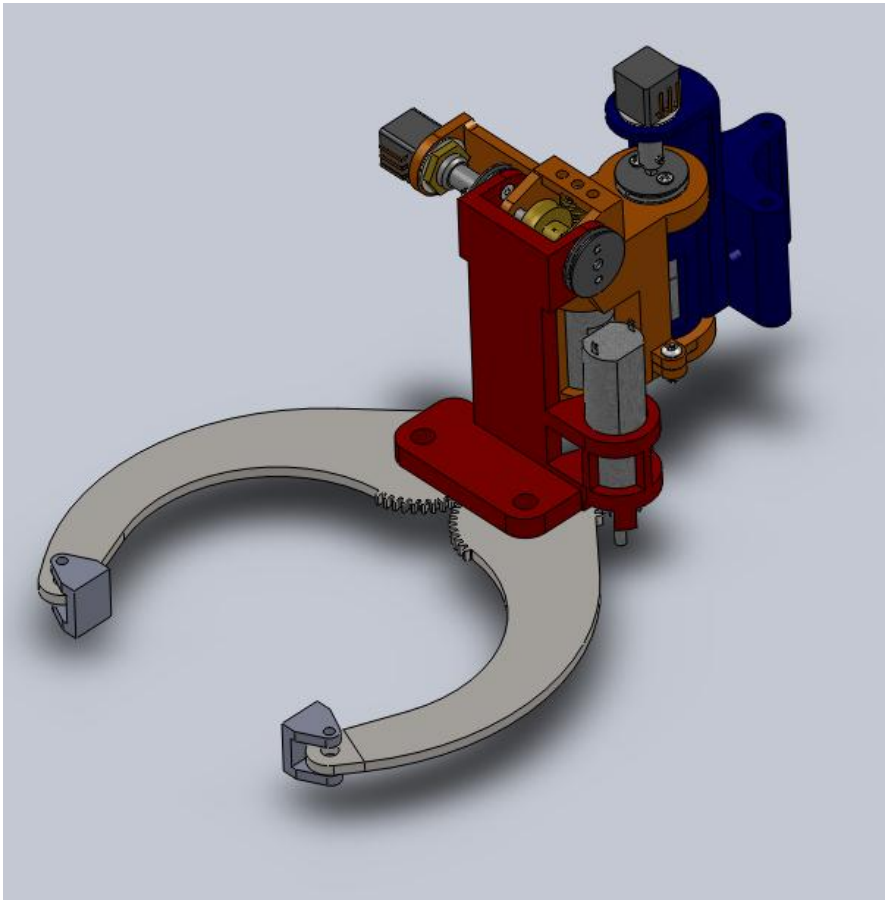


Figure 34: Pincers

In the interest of minimizing the amount of hardware required, the pincer jaws were designed using aluminum plate with involute gear teeth cut directly into them, simplifying the transmission significantly. As one jaw can drive the other at a 1:1 ratio, the drive motor only needs to drive one of the jaws directly. An 18-tooth 32 DP pinion is pressed onto the motor output shaft and drives the jaw, which has a pitch circle equivalent to that of a 48-tooth gear, giving an effective reduction of 2.667:1.

Pivoting grasping surfaces attach to the end of the jaws, providing a much larger gripping surface than the thin jaw plates themselves. In order for the pincers to be able to grasp a 10 lb. load as specified, assuming a coefficient of friction between the grasping surface and object of 0.5, a gripping force of 20 lb. is required. This equates to 82.4 in-lb. of torque at the motor, well within its specified maximum torque of 114 in-lb.

4.1.3.4 Central Body

The purpose of the central body is to connect the other sub-assemblies as well as house the electrical components. The first iteration of the body plate was minimalistic in an attempt to keep the overall size and weight of the robot down. The plates for this design are made from acrylic and are connected via metal standoffs for rigidity. Slots are cut into the front and back ends for the neck and tail mounts to slide into. The body is pocketed to reduce the weight of the assembly and can be seen in Figure 35.

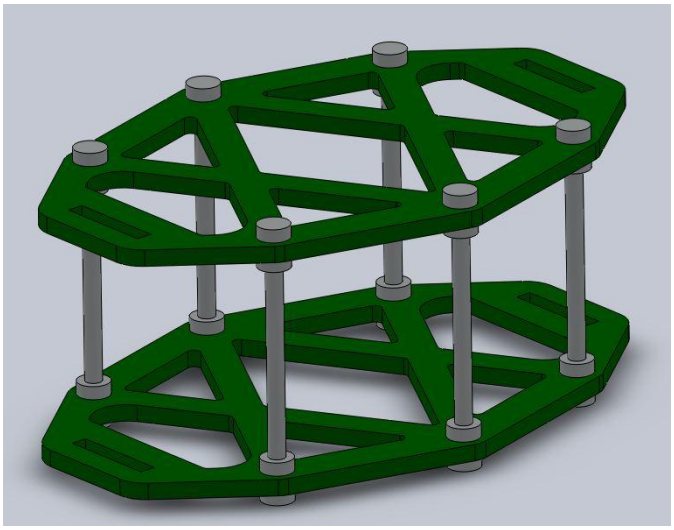


Figure 35: Initial Body Design

Once motors were chosen and the legs began taking shape, the body required major changes. The body plates grew to accommodate the larger new legs and the attachment pattern was altered. To reduce the amount of hardware necessary to attach the entire assembly, the body shares it's attachment hardware with the shoulders. The plates also have holes put in them for the motors to fit to make attachment and wiring easier. The plates are still made of acrylic which can be seen in Figure 36.

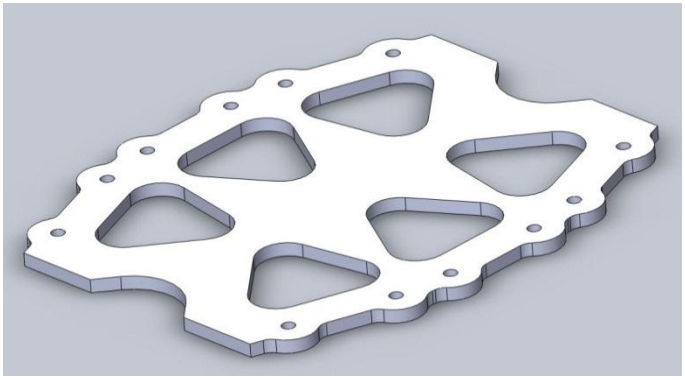


Figure 36: Second Generation Body Plate

In an attempt to make the robot lighter and more modular, the attachment locations for the legs changed drastically. The plates changed so that the legs are able to slide in and out of the body easily. The mounts for the rest of the sub-assemblies still acted as the attachments for the two body plates. The new body weighs less and is more assembler friendly than the previous version. To minimize the size of the robot the slots for the shoulder mounts are placed in a linear pattern with the forward and back shoulder mounts at a 25 degree angle, allowing the triangular shape of the shoulder mounts to fit closer to one another. The latest version can be seen in Figure 37.

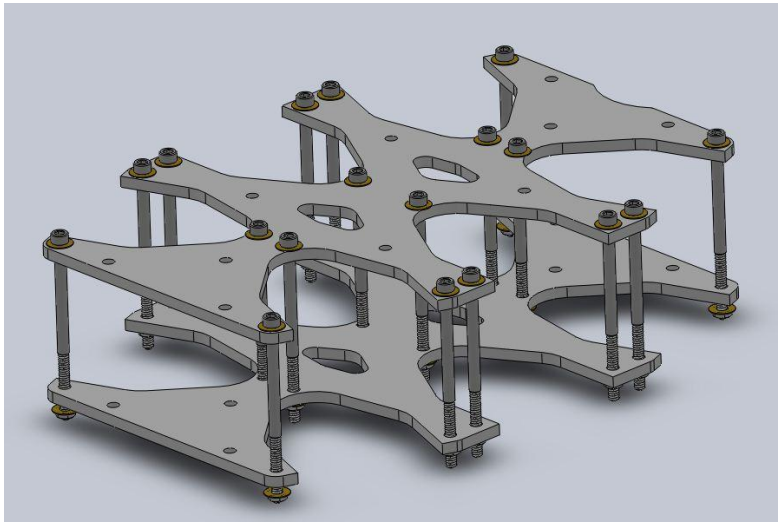


Figure 37: Final body Design

Having less material for support, different materials were tested with FEA models. Aluminum, acrylic, Lexan, and titanium were all tested. The two plastics are the lighter weight options but had potential for flexing issues. The two metals are much stiffer, but with that rigidity, comes weight. Aluminum is easier

to machine, but the plate can be made thinner if from titanium. Since the body is what gives the robot its main structure, rigidity is important. The FEA models showed more deflection in the plastic options, but that didn't take into account the support of the sub-assemblies.

There are three 4-40 machine screws that extended through the mounting area connecting the sub-assemblies to the body plates. Ridges were added to the other sub-assemblies to disperse forces through the plate as opposed to being solely transmitted screws. This allows for Acrylic plates instead of the more rigid metals.

The electronics are placed as close to the center of the body as possible so that they can be easily covered by a shell and reduce the amount of wiring. The Gumstix board is placed in the center of the body plate, in between the shoulder mounts. An acrylic plate placed on .75 inch standoffs too allow the Gumstix to be easily removed. The leg controllers are attached in a similar fashion; an acrylic plate was made to mount the leg controllers to. There are two holes on the plate that matched the two outer holes used for the shoulder mounting. This allowed the leg controllers to utilize the existing hardware and make disassembly easier.

4.1.4 Final Design

The completed design of the robot, with sub-systems connected can be seen in Figure 38.

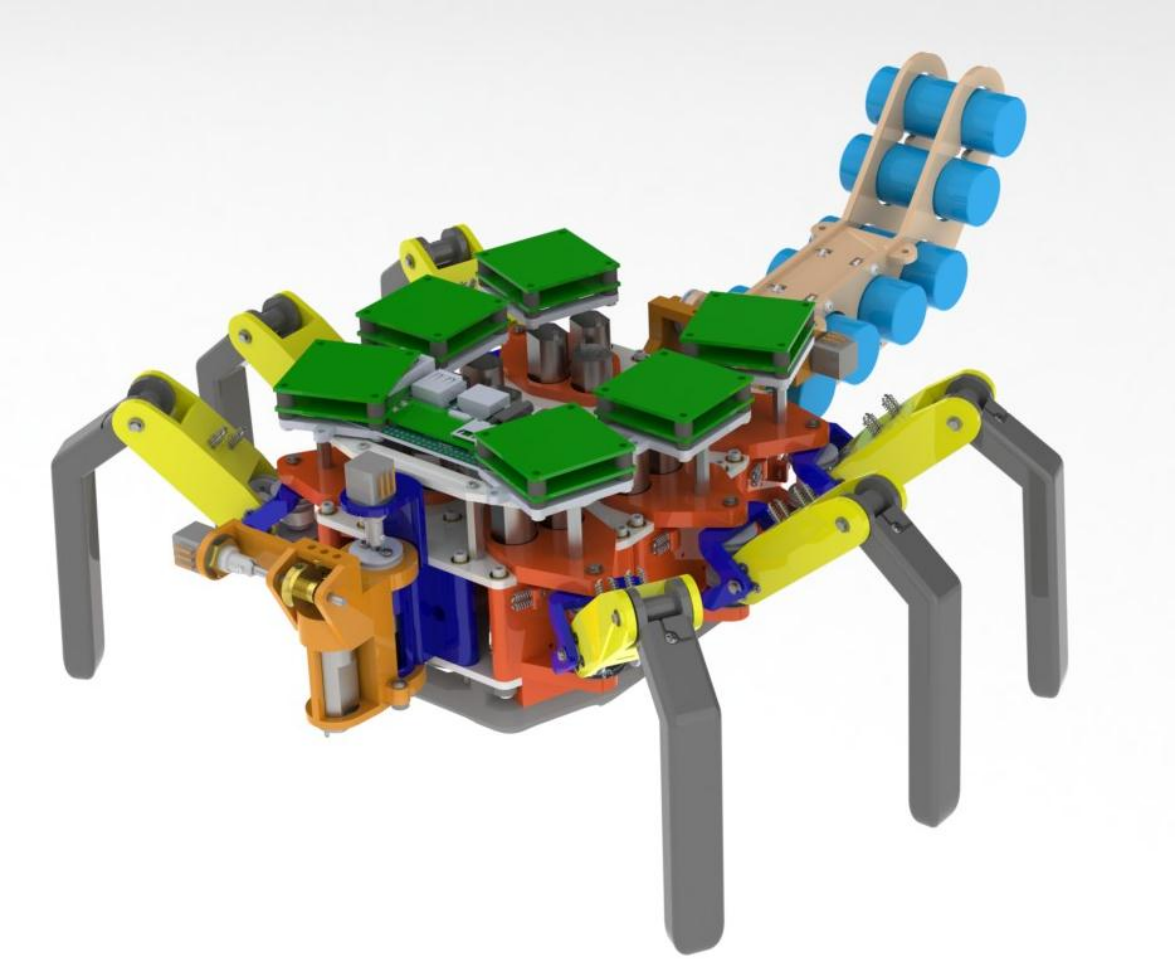


Figure 38: Full body CAD

4.2 Fabrication

4.2.1.1 Modifying and Making Pulleys

The two types of pulleys required both special considerations to be taken into account for proper manufacturing. The Delrin mounting pulleys required screw holes to be placed both on the face of the pulley and on the inset of the groove. The difficulty of placing the holes in the pulley is that the clamping force to properly secure the pulley, if held by the outer diameter, would warp and damage the pulley. A custom fixture was fabricated so that a screw could hold the center of the pulley, through the center bore. This method of mounting allowed for the pulley not be warped during manufacturing as well as

cutting multiple pulleys at the same time. Using the fixture also removed the need to “zero” the machine between every cut because the pulleys were always held in the same place. An example of this process can be seen in Figure 39.

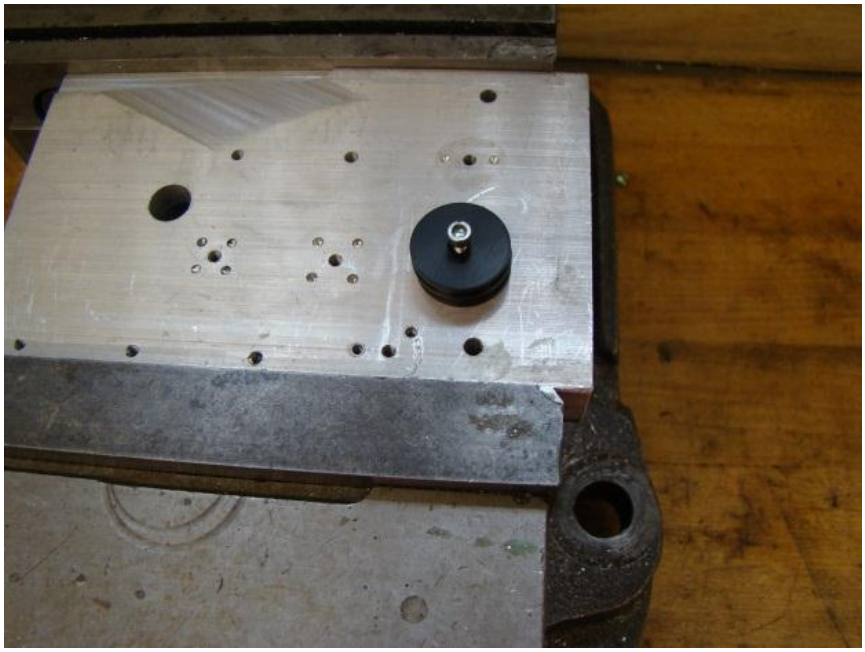


Figure 39: Example of Pulley Machining

To add the holes in the groove of the mounting pulley wheel, an additional mounting point was added to the edge of the mounting block. By placing the fixture vertically, a drill could be used to place holes to the current depth. Similar to the other use of the fixture, more than one pulley could be made at a time and the machine did not require zeroing between set ups. An image of this process can be seen in Figure 40. An image of the fixture can be seen below next to the fixture used in Figure 41. This process allowed for three pulleys to be made every ten minutes

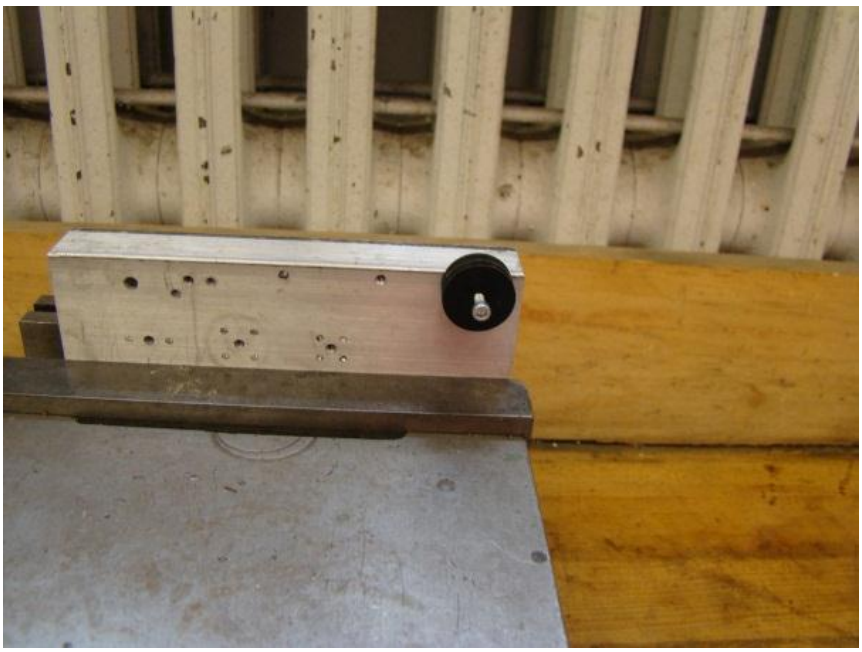


Figure 40: Drilling Set Screw Holes

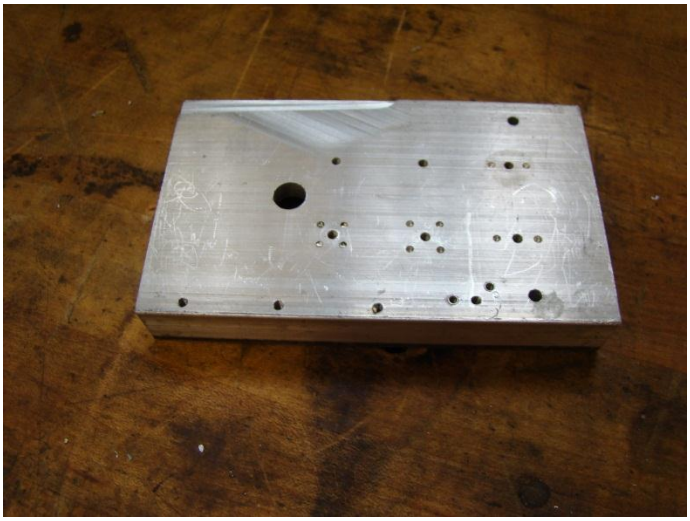


Figure 41: Fixture Plate

To make the driving pulleys, a CNC lathe was first used. A piece of stock, 5/8 of an inch diameter was placed in the chuck. Manually, the lathe was controlled to place the 3 millimeter bore in the center of the pulley. The guidance groove was then cut into the side of the stock. Finally, a cutoff tool was used to cut the pulley to the proper thickness. A mini-mill was then used to add the mounting screw point inside the guidance groove. This process limited the production of one pulley at a time. A single pulley could be made in 10 minutes of machining.

4.2.1.2 Modifying Potentiometers

To put the needed modifications in the potentiometer, a custom fixture, similar to the one used for the pulley wheels, was constructed. The fixture would clamp onto the center axle as opposed to the housing. This reduced the possibility of the sensitive electronics from becoming damaged during cutting. The potentiometers were first placed in a vertical hole. This allowed for the main mounting bore to be placed. This process can be seen in Figure 42.

Figure 42: Drilling Shaft Hole

The potentiometers were then placed in a horizontal mounting hole. This was done so the set screw hole and tap could be placed in the side of the axle. This process can be seen in Figure 43.

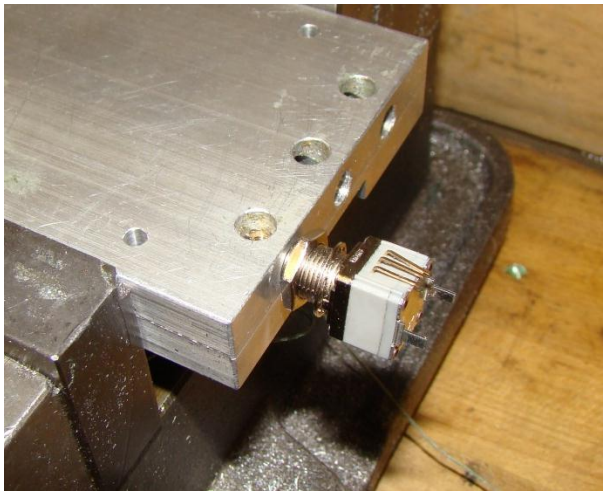


Figure 43: Drilling Set Screw Hole

If the same cut was done without using the fixture the drill would have difficult drilling into the rounded surface of the axle. The fixture allowed for the hole to be placed precisely at the outermost point of the diameter. An image of the fixture can be seen in Figure 44.

Figure 44: Potentiometer Fixture Plate

4.2.1.3 Constructing Axles

The leg axles were constructed using a manual lathe. Because of the small diameter of the stock material caused the piece to vibrate at high spindle speeds. To accommodate for this the spindle speed was reduced. This caused the vibrations to be lowered; however, if the tool was not sharp enough to cut at the low speed, the stock would break at the start of the chuck. Once the correct tool and speed was determined the axles were able to be manufactured at a rate of one axle every 5 minutes.

4.3 Controls and Electrical Design

4.3.1 Control System

When examining the control requirements of the robot, tasks can be separated into two distinct categories: low-level I/O port read/write operations, and high-level motion, navigation, and image processing. In most examples of small robots, all tasks are typically handled by one controller. However, in light of our robot's complexity, this traditional solution runs into several significant shortcomings.

With the robot having 22 controllable joints, each of which having two sources of feedback, the burden of handling the large number of controller inputs and outputs is one of the most significant factors in the control system design. For just one joint, one PWM and two digital outputs are required to communicate with the motor controller, and two analog inputs are required for position and current feedback, from a potentiometer and current sensor, respectively. This gives us a one joint total of 5 I/O ports, and when scaled to the entire robot, the control system requires 110 I/O ports!

While controlling each port individually is trivial for a controller, this would require 550 processor cycles for I/O operations alone, assuming each operation requires only one instruction cycle. When input processing and control loops for each joint are taken into consideration, the processing requirements grow to a rather daunting scale.

To solve this problem, it was decided to separate and distribute the two categories of control requirements among multiple independent microcontrollers. The highly parallel, though relatively simple I/O operations are handled by one system, while the high-level, more computationally-intensive motion, navigation, and image processing would be handled by another.

4.3.1.1 Architecture

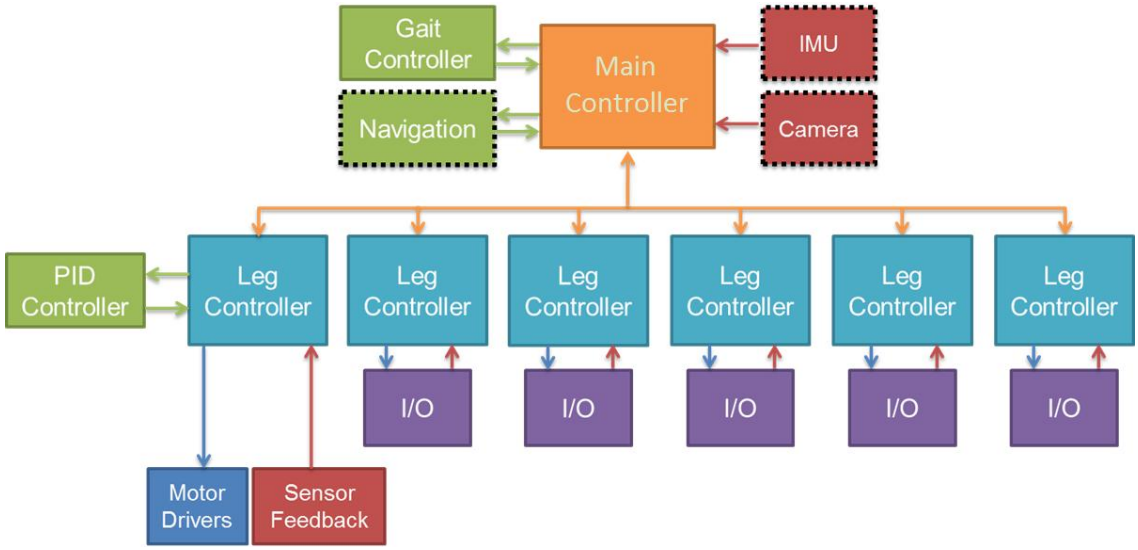


Figure 45: Robot control system diagram

The control system consists of a 32-bit ARM Cortex-A8-based main controller, combined with a group of independent, less powerful 8-bit slave controllers, with one dedicated for each leg. All are networked together over an RS-485 bus with a custom message protocol.

In the interest of modularity and compartmentalizing each leg as a complete subsystem, the low-level operations are split further into separate controllers for each leg. Each "leg controller" is responsible for joint I/O operations, as well as PID control of each joint. Aside from desired positions or velocities, each leg controller operates locally and independent of the others. The main controller, in turn, is responsible for the high-level computation, such as the gait controller, which operates on the scale of the entire robot.

Some sort of translation is then required to convert robot-scale position information to leg- and joint-scale. Once this information is broken up and formatted, it can then be passed to the leg controllers. A multipoint communication bus is used to network the individual controllers together. This enables the main controller to pass joint positions and velocities to each leg controller, as well as enable the leg controllers to transmit joint feedback information to pass back to the gait controller.

4.3.2 Main Controller

The two largest driving factors in the selection of our main controller were computational power and size. As navigation, swarm behavior and image processing are planned for the future; plenty of unused

power was desired. In addition, with timing and synchronization being critical due to the control system being inherently parallel, we wanted the controller to be capable of running an operating system at least capable of multitasking, though ideally a real-time operating system.

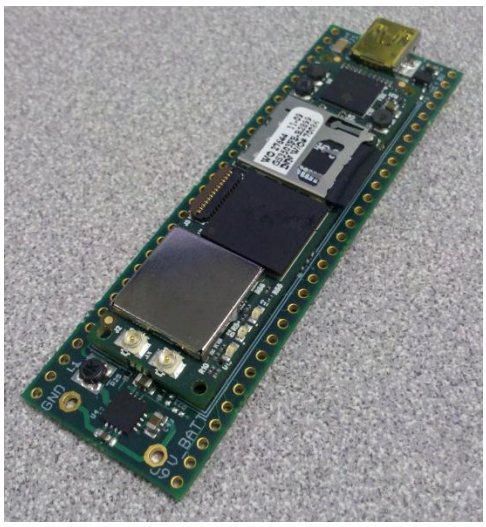


Figure 46: Gumstix Overo FE

With most I/O handled by the leg controllers, it was not a top priority for the main controller. Instead, a large number of connectivity and communications peripherals were desired to allow for interfacing with additional sensors and communicating with other robots.

Many development boards that we found initially that were intended for robotics were either too large or not powerful enough for our purposes. However, thanks to the rapid increase in popularity of smartphones, development of powerful but inexpensive microcontrollers has greatly accelerated. They tend to offer a large number of onboard peripherals, including various communications interfaces like WiFi and Bluetooth, and a large amount of RAM and storage. A few companies have started to sell development boards using these, but at this point in time, they tend to use very similar components.

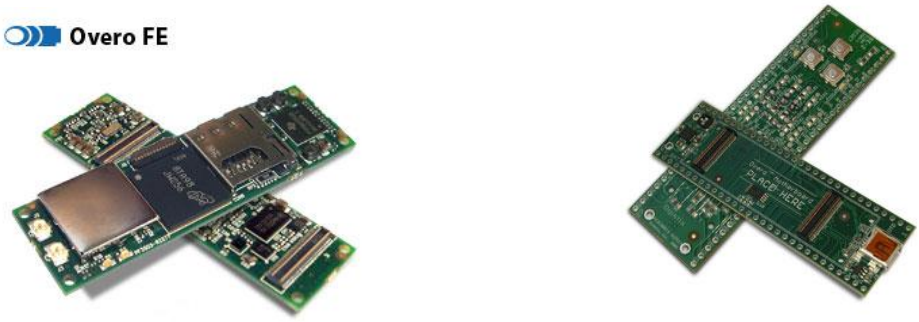


Figure 47: (a) Gumstix Overo FE COM (b) Gumstix Pinto-TH board

Eventually, we chose to use a Gumstix Overo FE computer-on-module (COM) as seen in Figure 47. The combination of high power, relatively low cost, and easy expandability have made them a popular choice for similar applications, as well as our own. A list of specifications can be found below.

Overo COMs are designed to always be used with an expansion board, and cannot function without one. Gumstix offers a number of their own design, but most are geared towards multimedia device development, featuring touchscreens and Ethernet ports. They are also much larger than the space that is available on the robot. The Pinto-TH board, however, functions essentially as a breakout board for the Overo, offering two unpopulated header rows to bring out some of the most commonly used I/O ports. In addition, it is barely larger than the Overo itself, being only 76.2mm x 23mm. One minor caveat is that the Overo COM uses 3.3V logic, whereas rest of the control system operates on 5V logic. Though the communications bus transceivers solve this problem for the communications bus, any additional I/O will need level-shifters.

4.3.3 Leg Controller

When designing the control architecture, the team decided to use individual controllers for each leg. Unlike the main controller, which runs a full operating system, the leg controller software is completely application-specific. This was done for several reasons, one of which was the number of inputs/outputs needed for the robot. Each leg requires a PWM signal for each joint which requires three ports. Each joint also has position and current feedback which take up four pins. Finally each leg needs to communicate with the main controller for gate control, which is at least four pins. To maintain all of these processes, the main controller would need at least 128 ports, not including the head or tail. A pic with that many pins would be too cumbersome to for the scope of this project. The individual leg controllers also help offload a large amount of computational weight. To have the main controller

handle gate control, 50+ sensory inputs, and PID control for all moving parts at the same time would be far too much for the processor to handle. Therefore, any task pertaining to the legs that an 8-bit microprocessor could handle was passed off to the leg controller.

4.3.3.1 PCB Design

The first thing that was done when designing the board was deciding what needed to go on the boards. To accomplish all tasks necessary for the leg controller, there needed to be an 8-bit microprocessor, motor controllers for three motors, current sensors for three motors, capacitors for signal filtering, resistors for current sensing, transceiver for RS-485, inputs for three potentiometers, input for RS-485, input for 5V input for 6V, reset button, breakout for ICSP, connectors for the three motors, and any other various components that are needed for said parts.

First the team chose the microprocessor for the controller because it was the most important component. Once that was decided upon, the rest of the parts were chosen accordingly.

Eagle CAD

Eagle CAD is a free, 2D, PCB design program that allows one to create two layer printed circuit boards. There are hundreds upon thousands of parts already in the libraries of the program which were great for industry standard parts. Eagle also had the ability to create custom parts, which was helpful for more unique components. First the parts were placed into a schematic where all of the proper pins were linked. Once all components were in the schematic, the layout of the board was laid out. First the parts were dragged and dropped into some general position where everything fit. Then the traces were drawn connecting the proper pins. During this step, all pins were connected by lines so connections were easy to keep track of. When two pins were properly connected, the line went away. The traces were drawn until all connections were made. This took a great deal of moving parts around to and redrawing traces to fit everything on the board. Once the controller was complete, a script created by SparkFun was used to produce the proper gerber files to send to a shop. The files were submitted to 4pcb.com and the boards were shipped in a week.

Space

When designing the PCB's for the controllers, one of the largest issues was space. The only space available for the controllers was a 1.625" square above the motors. The microprocessor by itself was 1.5" long and took up a great deal of the available area. The first iteration was placed entirely on one

board, but that did not nearly fit within the allotted space. Since the boards couldn't spread out any further, the team decided to go up; the leg controller changed to a tiered control board connected by pins and headers. The first layer served, more or less, as a breakout board for the microprocessor. Things like the transceiver and op-amps for the current sensor were placed on the underside of the board. The connectors for the pots, ICSP, RS-485, and 5V were also placed on this layer. The second layer contained the motor controllers, current sensors, and connectors for the 6V and motors. When choosing components, size was a large issue and therefore, most of the components were surface mounted.

Microcontroller

The selection process for the microcontroller to be used for the leg controller was driven almost entirely by I/O requirements. The two most important requirements were that there must be at least 3 independent PWM outputs and that the UART have a high enough maximum baud rate to eliminate any threat of being a bottleneck in the system. Microchip was chosen as the MCU manufacturer due to their large parts library, with numerous part configurations available and readily stocked. After dialing in on I/O requirements, as well as a few others, such as a DIP package and a free C compiler, the potential candidates were narrowed down to around one to two dozen devices from the mid- to upper-range of the PIC18F family. In the end, the PIC18FxxK22 family was selected, due to its upgraded PWM resolution and powerful UARTs. Originally the PIC18F1XK22 was chosen due to its increased memory, but during board layout, the team realized that there were not enough pins to support all components. As a replacement, the PIC18F26K22 was chosen as the cost increase was negligible, memory was similar, and offered maximum flexibility.

Motor Drivers

When selecting the motor driver, a large deciding factor was the amount of power that could be passed through it and how many motors it could drive. The chip had to have at least enough power to drive the motors maximum requirements. Dual and quad motor controllers were also more favored than single because they would require fewer chips. The first motor driver chosen for the leg controller was the Toshiba TB6612FNG dual motor driver. It is capable of supplying up to 15V and 3.2A, much more than the motors needed. This particular component was chosen for its ease of use and small size. The chip takes in a PWM signal and a direction signal. There are also break, stop, and standby modes if desired.

Since it is a dual motor driver, there only needs to be two of them and with a SSOP24-P land pattern, it does not take up much room.

There was another chip that was considered along the way because heat concerns were raised with the Toshiba. National Semiconductor's LMD18200 had enough power to drive the motors and had a large heat sync attached to it to help with heat dissipation. Although the LMD had better heat dissipation, it was only a single driver and had a much larger footprint than the Toshiba. Therefore, the team decided to stay with the Toshiba to try to keep the size of the boards as small as possible.

Current Sensor

The selection process for the current sensor was largely dependent upon signal sensitivity and noise. The current sensor needed to be able to keep a steady signal between zero and two amps, which is in the lower spectrum of most sensors. The first current sensor chosen for the board was the Maxim MAX4070 bi-directional current sensor. This particular sensor was chosen because it could work with a PWM signal, which was very important. The signal, however, did not function well at lower currents and had a tendency to walk upward. The next sensor chosen for the board was the Allegro ACS714. The sensor only had a sensitivity fluctuation of 1% over the entire range. The sensor was also a Hall Effect sensor and therefore was removed from the circuit itself and much less prone to noise. Although with a sensitivity of only 185 mV/A, an op-amp was needed to boost the signal. Since the output was only 185 mV/A and the motors, at stall, drew 2 A, the signal would never reach over 370mV. With an input voltage of 5V, the resolution could be raised significantly. To boost the signal to a more appropriate range, the op-amp would have a gain of 5, bringing the sensitivity up to 925mV/A, using 87% of the 5V input rather than 15%.

Op-amp

The op amp was needed to boost the signal of the current sensor so that the full range of the processor input could be utilized. With a stronger signal, finer monitoring and control could be achieved. The first sensor that was considered was the one used in an example diagram in the current sensor data sheet, National Semiconductor's LM321. Upon further investigation, the op-amp's output signal did not have a steady output over the full range of its use; the linearity would diminish between four and five volts. In a venture to find a more reliable op-amp, Linear Technology's LTC2054 was chosen. This chip had a much steadier output signal over its full range and minute signal degradation.

Transceiver

To convert the microcontroller's UART ports to RS-485 line levels, driver ICs needed to be used. The MAX490 driver from Maxim Semiconductor was chosen due to its full-duplex capability and ready availability.

4.3.4 Communication Bus

As the leg controllers depend on the main controller to determine necessary joint positions, a communication bus needed to be developed to facilitate transferring this information.

4.3.4.1 Physical Layer

With eighteen motors needed for driving the legs alone, electromagnetic interference (EMI) with communication signals was a significant concern. Because of this, both SPI and I2C, commonly used microcontroller busses, were quickly eliminated as potential solutions. Both were designed for communication across a circuit board, not between separate boards, and as a side effect, are quite susceptible to noise problems.

After researching several potential interface options, such as CANbus and LIN, RS-485 was selected as the physical layer of the communications bus, for several key reasons. First and foremost, RS-485 uses balanced differential-pair signaling, where two lines carry inverted and non-inverted ±5V versions of the data signal, making it highly EMI and noise-resistant. In addition, unlike many other popular busses, RS-485 only defines the electrical standard, allowing the user to use any protocol using it.

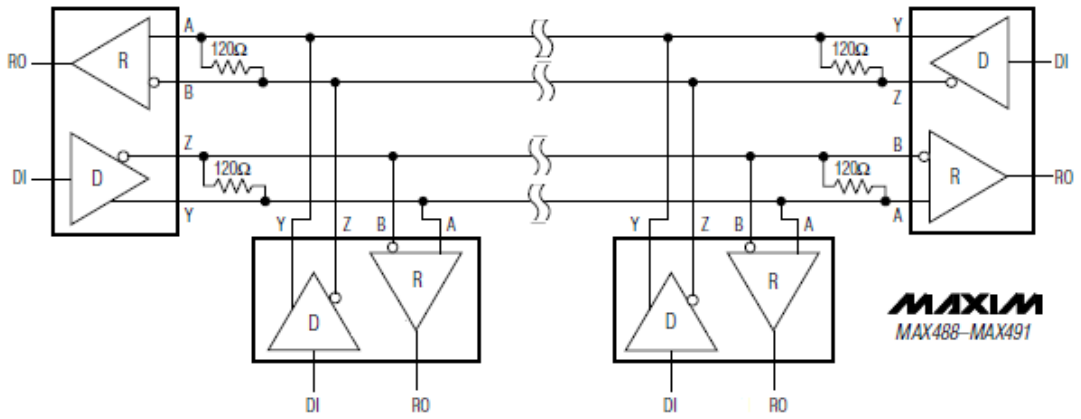


Figure 48: Typical full-duplex multipoint RS-485 system

RS-485 uses a multipoint bus topology, with one master and up to 32 slaves in a standard configuration. It is capable of both half- and full-duplex operation. In a half-duplex configuration, a transmit enable control line is required on top of the TTL RX/TX lines. For this reason, we chose to use a full-duplex network. However, in such a configuration, all slave receivers are connected to the master's transmitter, and all slave transmitters are connected to the master's receiver. As a result, slaves cannot communicate directly with one another, but thankfully this does not affect our implementation. Figure X illustrates a typical full-duplex configuration.

4.3.4.2 Protocol Overview

As RS-485 specifies no communication protocol, one needed to be developed. Thankfully, due to the popularity of RS-485 in industrial applications as result of its noise resistance, a number of protocols have been developed that are in widespread use. Among these is Modbus, one of the most popular industrial communications protocols used today. One of the reasons for this is its open and royalty-free nature, as well as its simplicity and robustness.

Seeing as Modbus was designed for interfacing industrial control systems, much of its functionality was irrelevant to the project's needs, and so the decision was made to heavily modify the message structure and tailor it to fit the project's needs, resulting in AntBus.

Byte	Field	Character	Hex	Description
1	Start Transmission	STX	0x02	Signal start of transmission
2	Leg Address	1 to 9 A	ASCII 0x41	Specific controller Broadcast to all
3	Header			See table
4	Command Type	R W D ?	0x52 0x57 0x44 0x3F	Read value Write value Returned data Ping
5	Sub-command	P V C T	0x50 0x56 0x43 0x54	Position Velocity Current Torque
6	Joint Number	0 to 9	ASCII	Joint number
7	Sign	+ or -	ASCII	Data sign
8	Data	0 to 9	ASCII	Data digit 1
9			ASCII	Data digit 2
10			ASCII	Data digit 3
11			ASCII	Data digit 4
12			ASCII	Data digit 5
13			ASCII	Data digit 6
14	End Transmission	ETX	0x03	Signal end of transmission
15	Checksum (CRC-16)	0 to F	ASCII	High byte
16		0 to F	ASCII	Low byte

Header Byte	
7	Status: ACK/NAK (1,0)
6	Command Enable (enabled when sending a command)
5	ACK/NAK Enable (enabled when responding)
4	Broadcast Enable (enabled when broadcasting to all)
3	Source Address
2	
1	
0	

Figure 49: AntBus packet structure

A complete packet is formed up of 16 bytes. For the most part, ASCII characters are used to pass data, in an effort to make packets human-readable for debugging purposes. Using a binary message format could reduce the message size to X bytes. However, the use of ASCII characters currently makes it much easier to extend the functionality of the protocol, by simply assigning different letters or other characters to new commands as they are developed.

Each leg controller is assigned a unique address, allowing filtering of messages not intended for a given leg. In addition, the master controller has the ability to send broadcast packets to be received by all leg controllers. This can be used to do things such as simultaneously stop all joints or move to a pre-programmed stance.

The header bit is used to identify what type of message the packet is. In descending order, the ACK/NAK bit is used by the leg controller to mark whether a command from the master, marked with a Command Enable bit, was successfully received or not. In the event that a message is corrupted while traveling either direction, the fall-back state of the main controller will retransmit the original command until it receives a confirmation from the leg controller or an updated command is generated by the main controller, whichever occurs first.

The Broadcast Enable bit is used if a message is intended to be received by all legs simultaneously. The lower nibble of the header byte is

At the end of the message is a provision for a 16-bit Cyclic Redundancy Check (CRC) checksum. In an effort to further protect against noise and transmission errors, leg controllers respond to the main controller, either acknowledging the message was received intact, along with any requested data, or signals that it has received a corrupt message. In such an event, the master controller can simply retransmit the message in question. Currently, the CRC-16 check is not implemented, but the necessary space has been allocated for future use.

4.4 Software Architecture

4.4.1 Main Controller

In order to accommodate future expansion of the robot's capabilities, the software should be made as modular as possible by separating the major components into separate processes. One main process can then focus on passing and routing messages between processes.

One benefit of modularizing the software is additional functionality can be easily added in the future. For example, a navigation controller and vision processing can be added and used to determine a motion profile for the robot to reach a desired position. In addition, multiple modules could be written to fill the same role, i.e. a gait simulator can be swapped for a real-time gait controller.

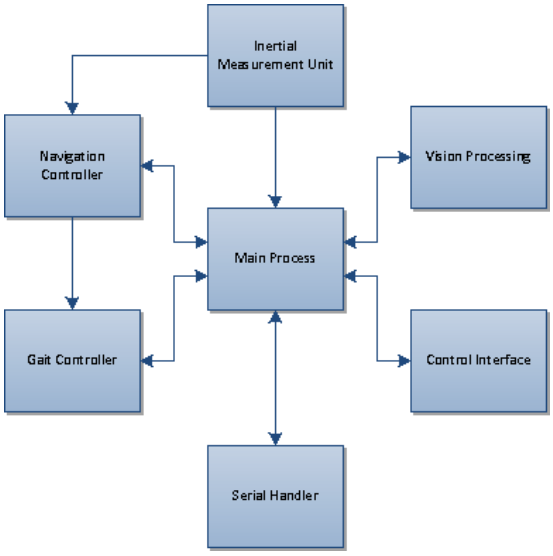


Figure 50: Main controller software architecture diagram

4.4.1.1 Main Process

The main process is responsible for coordinating and communicating between all other processes, allowing them to run independently of one another and removing the need to actively poll one another for new data.

Control parameters, such as desired speed, body orientation and gait type are taken from the control interface and passed to the gait controller, which adjusts its internal parameters accordingly. Once the gait controller has finished computing the necessary joint positions, it then passes these back to main process, which then parsed and pushed to the serial handler.

4.4.1.2 Serial Handler

The serial handler is dedicated to building, transmitting, receiving, and parsing messages to and from the leg controllers. There are actually two sub-processes running; one to handle message formatting and

translation (message handler), and another to handle interfacing with the physical serial port (transmit/receive, or TX/RX handler).

When it receives bulk joint data from the gait controller, the message handler needs to break the information up into individual messages for each joint. As these are completed, they are passed to the TX/RX handler to be sent to the leg controllers. As these messages are being transmitted, joints that have already received messages will be transmitting their responses back to the main process. The TX/RX handler then needs to pass these to the message handler for parsing. This message data is gradually saved to a buffer until all data has been received, at which point the message handler can hand it off to the main process for it to send to the gait controller.

4.4.1.3 Control Interface

As we will conceivably want to change the robot's speed and direction at some point, we need some sort of interface for the user to change these set points. When navigational functionality is added, the same interface can be used, allowing easy switching between autonomous and user-controlled movement.

4.4.2 Leg Controller

The leg controller runs in two discrete interrupt service routines. The PIC18F family provides two interrupt priorities, high and low. The high priority interrupt is able to be triggered while the low priority ISR is executing.

The high priority interrupt is triggered upon the UART receiving a byte. As the PIC UART receive buffer can only store a single byte, it is imperative that the receive register be read and cleared immediately, so that no data is lost. Originally, the ADC was updated in the high-priority ISR as well,

The main program loop runs in the low-priority ISR, which is triggered by a timer running at 4 kHz. At the beginning of each execution, the temporary UART and ADC buffers are read and copied to safe buffers. While this is being done, interrupts are temporary disabled to prevent data corruption. If a high-priority interrupt were to occur while the low-priority ISR was in the midst of reading one of the buffers, the data returned from the read would be corrupted upon exiting the high-priority ISR. Once the copy operation is completed, interrupts are re-enabled and the primary control loop begins execution. The latest ADC conversion results are directed to the PID controller feedback for the respective joints.

Once this is done, if a new command message is available, it is parsed and the requested action is performed. Any new setpoints are passed to the PID controllers for each joint, before they are updated and re-run. PWM outputs are then set accordingly. Before exiting the ISR, the timer register is reset to the correct value to ensure proper timing.

4.4.2.1 Data Management

In order to organize received commands from the main controller, as well as simplify and streamline working with the separate joints, several data structures were defined to manage this data. The most important of these are the RobotJoint, AntBusMsg, and PIDController structures.

```
typedef struct _RobotJoint{
    unsigned PWM;        //PWM channel used
    unsigned D1;         //DO port for D1 control signal
    unsigned D2;         //DO port for D2 control signal
    unsigned current;    //ADC port for current sensor
    unsigned pot;        //ADC port for potentiometer
    char direction;      //Flips direction
    int limit_low;       //Lower position limit for joint
    int limit_high;      //Upper position limit for joint
    PIDController* PID; //PID controller for the joint
} RobotJoint;
```

Figure 51: RobotJoint structure definition

The RobotJoint structure is used to store references to all I/O ports related to a given joint, soft position limits, and a pointer to the joint's PID controller instance. This makes supporting functions much easier to reuse, and allows the firmware to be easily reconfigured to use different numbers of joints (e.g. for the head and tail controllers).

```
typedef union _AntBusMsg{
    byte raw[12];
    struct {
        char start;
        byte address;
        byte header;
        char command;
        char sub_command;
        char joint;
        char sign;
        char data[4];
        char end;
        byte CRC16[2];
    };
} AntBusMsg;
```

Figure 52: AntBusMsg structure/union definition

AntBusMsg is actually a union between a structure and a byte array, done so in order to reduce the processing required to parse an incoming message. Because message packets are fixed-length, once a start byte is received, the firmware knows that the next X bytes will be part of the message. Once the end byte is confirmed, the buffer holding the received message can be dumped directly into the "raw" byte array within AntBusMsg, and the defined structure can be used to refer to the separate parts of the packet.

4.5 Gait Control

4.5.1 D-H Parameters

One of the first steps in building a model of the robot is determination of the Denavit-Hartenberg (D-H) parameters of the robot's legs. The D-H parameters are a commonly used robotics convention that represent a minimized set of terms to describe the transformation between joint reference frames.

Table 2 represents the D-H parameters for the final revision of the robot.

	d	θ	R	α
Hip	0	$90+\theta_h$	0	-90°
Shoulder	0	θ_s	0.825 in	90°
Knee	0	θ_k	2.25 in	0°
Foot	0	0°	5 in	0

Table 2: D-H parameters for current robot

These parameters can then be used to assemble transformation matrices for each joint.

4.5.2 Spatial Vectors

An alternate method of representing the parameters of the robot is through the use of 6D spatial vectors (Featherstone, 2008). Rather than using a mixture of 3x3 rotation and 4x4 transformation matrices, all parameters of a system can be represented in combined 6-dimensional vectors, containing linear and angular components. These can be used to represent motion and force.

$$\boldsymbol{v} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_x \\ v_y \\ v_z \end{bmatrix} \quad \boldsymbol{f} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ f_x \\ f_y \\ f_z \end{bmatrix}$$

Figure 53: Spatial velocity and force vectors

When using spatial vectors, mass is represented in the form of spatial inertia, which is represented by a 6x6 matrix combining the mass and rotational inertia of a body. Spatial inertia about the center of mass is defined in Figure 54. Here, $\bar{I}_{cm}$ is the 3x3 rotational inertia about the center of mass, m represents the mass of the body, and **1** is a 3x3 identity matrix.

$$\boldsymbol{I_{cm}} = \begin{bmatrix} \bar{I}_{cm} & 0 \\ 0 & m\mathbf{1} \end{bmatrix}$$

Figure 54: Spatial inertia about the center of mass

With this, we can now determine the momentum of the body, $\boldsymbol{h} = \boldsymbol{Iv}$**.** However, when defining the D-H parameters of one of the robot's legs earlier, we defined each joint axis at one end of each link, not coincident with the center of mass. As a result, the spatial inertia about this other point, O, is defined as follows:

$$\boldsymbol{I_O} = \begin{bmatrix} \bar{I}_{cm} + m\boldsymbol{cm} \times \boldsymbol{cm} \times^T & m\,\boldsymbol{cm} \times \\ m\boldsymbol{cm} \times^T & m\mathbf{1} \end{bmatrix}$$

Figure 55: Spatial inertia about an arbitrary point O

Here, $\boldsymbol{cm} \times$ and $\boldsymbol{cm} \times^T$ are the skew-symmetric matrix form of a vector representing the translation from the center of mass to point O.

Now that we know the momentum of the body in motion, we can differentiate it to find the kinetic energy, $T = \frac{1}{2}\boldsymbol{v} \cdot \boldsymbol{I}\boldsymbol{v}$.

4.5.3 Inverse Dynamics

In order to find the torques required at each joint to generate the desired motion, the inverse dynamics problem must be solved. To do so, the recursive Newton-Euler algorithm (RNEA) was used.

The RNEA solves the inverse dynamics problem by working outward down each leg, computing the velocity and acceleration of each link necessary to produce the motion desired. When this is accomplished, these accelerations are used to then compute the forces necessary, using the mass and inertial properties of each link. Finally, the algorithm works from the outside in to determine the net forces on each link back to the body.

This technique was used to approximate joint torques during early development. A model was built in MATLAB using the mass properties and D-H parameters of the robot. Approximate values from the early CAD models of the robot were used to determine the viability of the then-current design, as well as to simply give us realistic numbers to start with.

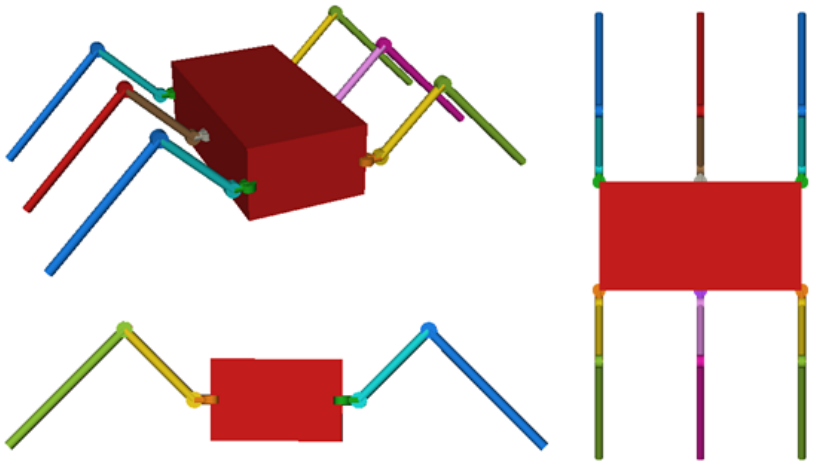


Figure 56: Model of robot standing

Initially, the robot was modeled as standing in a position approximating the joint position found in pictures of ants. A simplified 3D model of the robot was generated using the model data to verify that the joint lengths, locations and positions used as input were valid and what was expected. The result of

computing the inverse dynamics of this model showed that 120 mNm (1.06 in lb.) of torque was required at the shoulder while standing still, in line with our initial hand-calculations.

With this model complete, the next step is to calculate the torques in the time domain during a simulated gait, giving us an idea for what differences in necessary joint torques will be seen between standing still and walking.

4.5.4 Virtual Model

To solve the problem of actually determining foot placement, virtual model control, as discussed earlier, can be used to quickly build a model of the robot. Using this, we can describe and control every aspect of the robot's orientation and velocity in the world frame, and the model will determine the necessary forces or velocities at each joint.

We are able to specify the orientation of the robot's body through all six degrees of freedom, though for the sake of clarity and simplicity, this can be reduced to height, linear and angular velocity. In such a case, the virtual forces on the body would be modeled as follows:

$$
\begin{aligned}
F_x &= b_x(\dot{x}_d - \dot{x}_a) = b_x(v_d \sin\theta_d - v_a \sin\theta_a) \\
F_y &= b_y(\dot{y}_d - \dot{y}_a) = b_y(v_d \cos\theta_d - v_a \cos\theta_a) \\
F_z &= k_z(z_d - z_a) + W_{robot} + b_z(\dot{z}_d - \dot{z}_a) = k_z(z_d - z_a) - b_z\dot{h}_a + W_{robot} \\
n_x &= k_r(\theta_{rd} - \theta_{ra}) - b_r(\dot{\theta}_{rd} - \dot{\theta}_{ra}) = k_r(\theta_{rd} - \theta_{ra}) - b_r\dot{\theta}_{ra} \\
n_y &= k_p(\theta_{pd} - \theta_{pa}) - b_p(\dot{\theta}_{pd} - \dot{\theta}_{pa}) = k_p(\theta_{pd} - \theta_{pa}) - b_p\dot{\theta}_{pa} \\
n_z &= k_y(\theta_{yd} - \theta_{ya}) - b_y(\dot{\theta}_{yd} - \dot{\theta}_{ya}) = k_y(\theta_{yd} - \theta_{ya}) - b_y\dot{\theta}_{ya}
\end{aligned}
$$

The end result is a set of virtual forces on the robot's body which will act to drive the robot's body to the desired velocity and body orientation. Velocity components are included for parameters we wish to have zero velocity for the purpose of eliminating any velocity components that are introduced by instability in the system.

Before we can translate this to joint forces and positions, however, we must first determine how to distribute this load between the three stance legs currently supporting the robot. With the three parallel kinematic chains between the robot's body and the ground, a challenging is faced.

Once the force-distribution problem has been solved and forces exerted by individual legs have been determined, the individual joint torques can be computed using inverse dynamics methods described above.

Unfortunately, this system suffers from a dependence on a significant amount of reliable and accurate feedback. The robot built for this project has position feedback for each joint, and approximate force feedback via motor current sensing, but there is no body orientation information, such as would come from an inertial measurement unit, or IMU.

5 Results

The finished robot can be seen below in Figure 57.

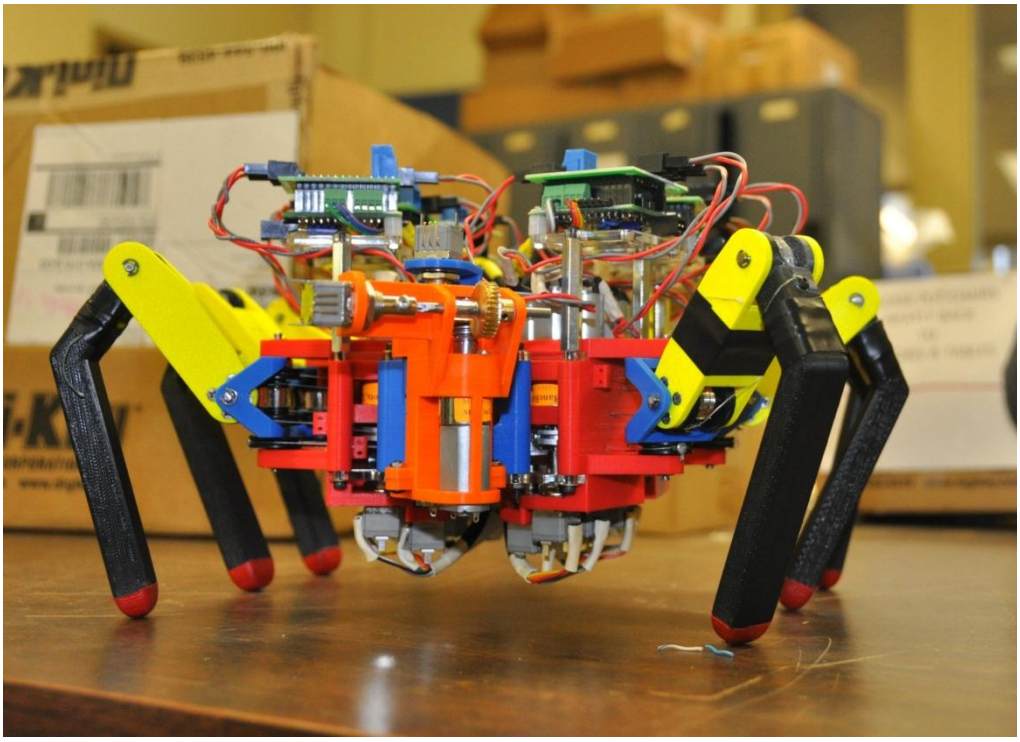


Figure 57: Completed Robot

5.1 Mechanical Results

The mechanical system was reviewed to determine the final performance specifications and decide how the current design can be further improved upon in later iterations.

5.1.1 Design

The final parts and sub-systems were compared to the initial design specifications to determine the robot's final performance specifications.

5.1.1.1 Weight

The robot weighed a total of 6.58 pounds. This is 1.58 pounds over the design specification and 0.17 pounds lighter than the predicted value generated from the computer model. The individual weight of major parts and sub-systems can be found below in Table 3: Component Weight.

Component or Sub-System	Predicted Weight (lbs.)	Actual Weight (lbs.)	Delta
Body Plates	0.08	0.08	0
BaneBots FF-050-116 Motor	0.07	0.07	0
Shoulder mount	0.13	0.105	-0.025
Shoulder joint	0.01	0.006	-0.004
Upper Leg	0.03	0.026	-0.004
Lower Leg	0.08	0.055	-0.025
Leg Sub-System	0.65	0.56	-0.09
Panning Joint	0.1	0.08	-0.02
Lifting Joint	0.05	0.039	-0.011
Head sub-system	0.51	0.429	-0.081
Battery Case	0.13	0.13	0
Battery	0.18	0.18	0
Tail Mount	0.12	0.096	-0.024
Tail Sub-System	1.48	1.346	-0.134
Motor Controller	0.1	0.07	-0.03
Complete assembly	6.75	6.58	-0.17

Table 3: Component Weight

The discrepancy between the computer prediction and the actual is due to inaccuracies in the material properties in the computer software and unknown weights of the motor controller and necessary wiring. For example, the predicted density of ABS plastic in SolidWorks is .039 lbs./in.^3 while the actual density of the printed ABS plastic is 20% less. The additional weight discrepancy can also be contributed to extra components and hardware not included in the CAD model that were added after the final design.

These figure show that the robot was overweight primarily to the weight of the motors and the addition of hardware. This additional weight will not prevent the robot from being able to walk; however, it will lower the total payload from 10 lbs. to 8.42 lbs.

5.1.1.2 Range of Motion

Due to tolerancing and additional components not in the original design, the range of motion of individual joints was reduced. The lower leg was reduced from the originally design +/- 90 degrees of motion. This was due to the limits of the pulley attached the motor not having the designed 180 degrees of motion. This situation is similar in the upper leg. The shoulder joint was limited from the originally

designed +/- 90 degrees to +/- 55 degrees. This was because of the addition of a support plate added to the shoulder mount, which will be discussed later.

With the previously mentioned changed to the range of motion of individual joints, the maximum and minimum sizes of the robot changed from the predicting size. The robot was 16 inches long with the tail up and 20 inches long with the tail down, as predicated. The maximum wingspan of the robot was 19 inches compared to the predicted 20 inches and a minimum wingspan of 9 inches. The maximum clearance under the robot was 4.5 inches and had minimum height of 7 inches with the tail down, both as predicted.

5.1.1.3 Modularity

The individual sub-systems were modular as designed. The three 4-40 bolts were able to be quickly attached and detached for sub-system repair. This modularity provided great benefit when the robot needed the pulley cables restrung on multiple occasions. The individual legs that need new cable could be detached and repaired without needing remove the other legs nor the head and tail sub-systems.

Despite the ability for the mechanical sub-systems to be modular, the electronics and wiring prevented complete modularity. The mounts for the motor controllers originally only had a single mounting point on the leg assembly. This single mounting point did not provide a secure connection to the body so a second was added off of one of the 4-40 bolts attaching the leg sub-system to the body. This prevented the leg from being detached without having to remove other components; however, this mounting method did not require the disassembly of other legs or the head and neck. The mounting point of the potentiometers in the shoulder mounts required cables to be routed through the body plates in order to be connected to the motor controllers. This prevented from directly removing the leg assembly after disconnecting the connecting bolts.

5.1.2 Strength Testing

To test the maximum load the robot could hold, individual sub-systems were tested in computer model and then verified using physical tests.

5.1.2.1 Forces in leg links

The forces in the links were as predicted. FEA models predicted that in a “locked” position the leg would have a safety factor of approximately 5 carrying a 10 lbs. load. This was confirmed when a leg was put

through a destructive test. The links of the leg were placed in a locked position, as seen in Figure 58, and force was applied the shoulder mount.

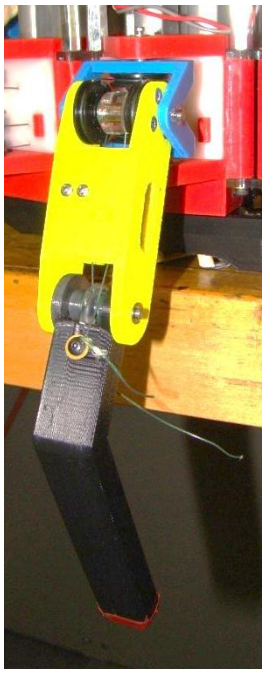


Figure 58: Leg Locked

This test was done to test a “worst case” scenario for the leg assembly reacting to shock forces. The force was increased until one of the frame components was broken. As predicted, the shoulder joint, the weakest component as shown in the FEA models, cracked first at a force of 17 pounds. Despite this crack, a force of 18 lbs. was needed to make the leg no longer usable. The break first split two layers of the printed ABS plastic around the joint connect the shoulder joint to the upper leg at the vertex of the V shaped bracket, seen in Figure 59.

Figure 59: Crack in V Bracket Around Axel Hole

The bracket then bent upward until the axle connecting the two links was able to fit through the crack. This caused the shoulder joint to break around the screw holes where a mounting pulley was attached, which can be seen in Figure 60.

Figure 60: Crack in V Bracket Around Screw Hole

This results in an approximate safety factor of 7.5 while not carrying a payload and a safety factor of 3.4 while carrying a ten pound load in a tripod gate.

5.1.2.2 Forces in Body Plate

The body plate, because of the ridges placed on the other sub-systems, was able to be made out of the inexpensive and easier to manufacture Acrylic in contrast to aluminum or Lexan. Due to the complex nature of how the sub-systems connect to the body plates, FEA models were unable to accurately depict how forces were transmitted between mounting points. To ensure that the body plates could transfer the forces from the head assembly to the legs a simulated load was applied to multiple body plates using various mounting methods. First a simulated ten pound load was applied to the head subsystem while the body plate was only connected to the leg sub-systems via mounting bolts in a wave gate, as seen in Figure 61.

Figure 61: Body Plate Test Rig

A wave gate was used to simulate the worst case scenario for walking while under load. Using a 50 lbs. force gauge, the body plate was not able to be broken. The body plate was mounted alone, without any reinforcement from the mounting ridges. This resulted the Acrylic cracking around the mounting holes at a force of 42 lbs., which can be seen in Figure 62.

Figure 62: Crack in Body Plate

5.1.3 Pulley Cable

The cable used was the spider wire 65 lbs. test coated cloth line. This line was expected to break close to its test force and degrade only minimally at the mounting points.

5.1.3.1 Physical Properties

The various fishing lines used as cable had a wide range of physical properties that altered the accuracy of leg movement, wear on structural components, and forces the legs could withstand. The three different fishing lines used were a 15 pound test plastic line, a 35 pound test cloth line and a 65 pound test coated cloth line. The plastic line stretched 12% of its initial length at the point of breaking when compared to the line under no load. By keeping the ten pound load on the line for 24 hours, the line was pre-tensioned so the total stretch was reduced to 3% stretch before breaking when compared to its no load length. Both cloth lines stretched 2% before breaking and showed no significant reduction in stretch after being pre-tensioned. Because of the small amount of strain seen in the cloth line, it was chosen as the pulley cable. Of the two cloth lines the high test was chosen so the legs could withstand high forces.

In order to connect the pulley cable to the leg joints a knot was needed around a screw. This method of connecting the cable to the leg joint drastically lowered the maximum force the line could with stand due to the stress concentrations caused by the knot. The amount the maximum force is lowered is dependent on the knot used. To test the breaking strength of the pulley cable, a force gauge was tied to one end via a noose. The knot being tested was tied to a stationary bracket that replicated the mounting point on the leg joints was paced on the other end of the cable. Force was applied using the gauge until the cable broke. The following breaking forces were seen in the 65 pounds test coated fishing line.

Knot Tested	Breaking force (lbs.)
Single Wrap	33
Double Wrap	38
Box	38
Fishing	35
Noose	42
No knot	62

Table 4: Knot Testing Results

The tests showed that noose had the least amount of stress concentrations; however, during the tests this knot is difficult to tie and hard to keep in tension. Both the box knot and double wrapping the cable showed the same breaking force and could be easily kept in tension. Double wrapping the cable around

the screw was finally chosen due to the ease of its assembly. This proved to be an adequate mounting method as walking tests continued; however, at times the 38 lbs. test was reached when applying a load. It was determined, because of this, that a similar line was need with a high maximum test force.

5.1.3.2 Pulley Wheels

The idle pulleys that were used to guide the cable through the leg joint provided a low friction surface and added no wear to the cable. Making the axle the idler pulleys were placed on dead allowed for them to spin freely when the cables were moved. The friction between the Delrin and the steel shaft was negligible to the system. Due to loose fit in the tolerances between the center bore of the pulley and the dead axle, the pulley was able to slip side to side. This slip caused slack to form in the pulley cable during assembly. Additional loose fit tolerances between the pulleys, spacers and leg links caused the pulley wheels to slide up and down. This allowed for the cable to become misaligned from the grooves and slide off the idlers, onto the axles.

The Delrin pulley wheels allowed the cable to remain in the grooves of the idler better than the aluminum pulley wheels. The Delrin would warp as the cable was screwed down. This warping would hold the cable in its desired alignment allowing for cable to remain tangent to the idlers and prevent misalignment from the pulley grooves. Despite this advantage, the Delrin pulley did not provide a ridged mounting point for the cable. The screws connecting the cable to the pulley would de-thread the mounting holes if over tightened. This would cause the cable to become un-tensioned after repeated use.

The aluminum pulleys allowed for a more rigid mounting point for the cable. The mounting screw could be tightened down without tearing the threads at the mounting point. However, the rigidness of the pulley prevented the cable from being forced into the groove. This caused the cable to become misaligned from the groove after prolonged use. Additionally, the sharp edges formed by the machining of pulley wheels promoted degradation of the cable. While no failures were seen during testing, prolonged use could lead to the cable breaking.

5.1.3.3 Guidance Channels

The guidance channels were able to redirect the cable with little friction when first under use. The friction caused by the coated fishing line added little wear to the tube over time. After prolonged use, small grooves were being cut into the plastic by the fishing line, as seen in Figure 63.

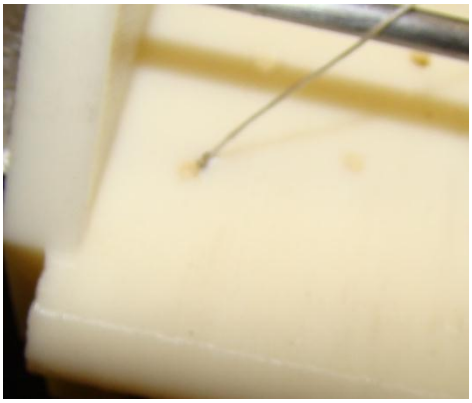


Figure 63: Wear in Guidance Channels

These grooves did not cause any noticeable changes in tensioning of the cable. After the groove grew to a certain length, failure of the tube occurred. The cable, once at a certain angle in respect to the channel, would rip through the supporting wall, which can be seen in Figure 64.

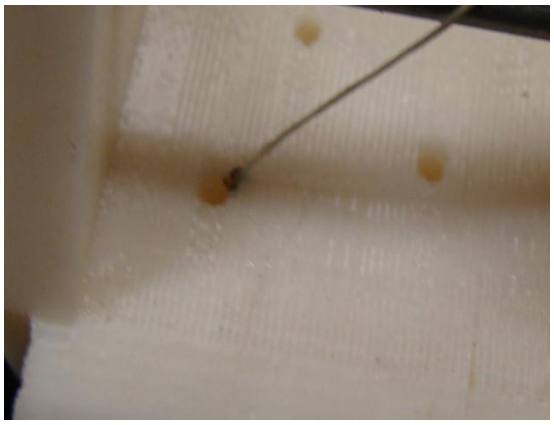


Figure 64: Failure of Guidance Channels

A Delrin plate was added to either side of the supporting wall. The Delrin plates, while increasing the difficulty of assembly and friction on the cable, removed the forces from the supporting wall thus preventing further damage to the channels. The Delrin supporting plates can be seen in Figure 65.

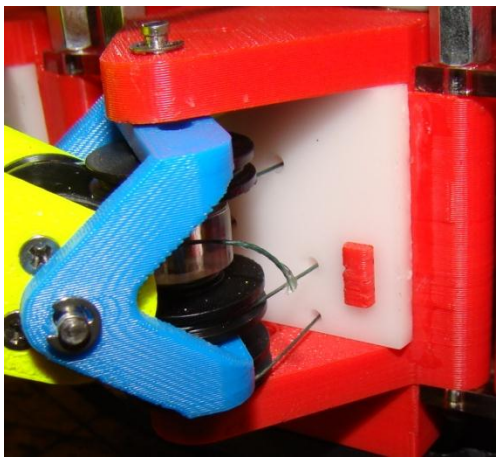


Figure 65: Delrin Support Plates

5.1.3.4 Cable Wear

As the cable was handled and underwent general use, wear began to appear. A majority of the wear appeared during handling of the cable and assembly. As cables were strung through the guidance channels the winding of the braided material began to become loose on the ends. This portion of the cable was removed before being fastened to the pulleys.

When the cable is fastened to the driving pulleys, the mounting screws degrade the outer layer of the cable. Similarly, during the testing of various knots, one end of the cable was securely attached to a stationary point while the other end was pulled with a force gauge to determine if the degradation would affect the performance of the leg joints. The cable broke at the mounting point at a similar force to the knot test. This shows the wear caused by the mounting screws does degrade the cable to the point where it breaks before the maximum force of the knot.

5.1.3.5 Tensioning

The tensioning of the cables on the shoulder joint and upper leg was determined to be unnecessary for this design after assembly was complete. The lines were able to be mounted to pulleys so that the slack remaining did not affect the accuracy of the link position. Due the difficulty associated with wiring the lower leg tensioners were still required. Unlike the initial design, passive tensioners were used. The spring force required to the cable to the lower leg actively tensioned would require a spring too large for the assembly. A passive tensioning system was put in place that used an Acrylic block with two holes for machine screws to fit through. Two machine screws were placed through upper leg assembly, using the holes for the active tensioners and then bolted to the Acrylic block. The cable is then passed through the

loop formed by the screws and the block. The block could then pull the line tighter by tightening the machine screws. This system can be seen in Figure 66. This system was able to keep the cable for the lower taunt while with little degradation to the cable.

Figure 66: Passive Tensioners

5.1.4 Motor Selection

The motors used were the BaneBots FF-050-116 DC motors. These motors had a stall torque of approximately .75 Newton meters with a gear reduction of 116:1. At maximum power output of 12 volts and 2 amps the shaft rotated at 127 rpm and provided .77 Newton meters of torque. This motor and gear box combination provided an inexpensive alternative to the Micromo 1331 Motor which had 1 Newton meter of torque but was 10 times more expensive.

5.1.4.1 Backlash in Gearbox

It was noticed that the BaneBots motor had a considerable amount of backlash because of its gear box. This means that the output shaft of the gearbox could be rotated without having to rotate the motor shaft. The backlash varied from motor to motor but averaged +/- 5 degrees of total movement. This backlash caused a decrease in the accuracy of leg placement.

This backlash had a compounding affecting in the leg sub-system similar to the effects of the controlled compounded leg motion describe earlier. Unlike the controlled compounded motion, the “slop propagation” was unpredictable. The shoulder joint could move side to side up to +/-5 degrees because of the backlash in the motor. The upper leg could rotate +/- 4 degrees up and down due to the backlash in the motor and up to +/- 9 degrees accounting for slop propagation. The lower leg could rotate up to +/- 7 degrees because of backlash in the motor and up to +/- 7degrees accounting for slop propagation.

The movement in the lower leg was mitigated by the "+/-" routing configuration described earlier; however; while there is no unpredicted rotation in the lower leg, the leg does travel. The travel was less than a total of 1/8 inch and was considered to insignificant to the system.

5.1.4.2 Heat Generated

The BaneBots motor was designed to run at a maximum of 12 volts and 2 amps. To ensure the heat generated would not damage other electronics the motor was run at the maximum power with the gear box detached. After a period of five minutes no noticeable heat generation was present. A small load was applied by pinching the output shaft. This caused a negligible amount of heat that would not affect other electronics. The power was then increased until the motor heated itself to the point of failure. The amperage was slowly increased with no load on the output shaft. At 12 volts and 4 amps the motor generated enough heat so that the shaft seized. This shows that inputting power over the factory recommended amounts will not cause immediate failure.

5.1.4.3 Motor Imperfection

The BaneBots motors commonly had flaws both out of package and after short amount of use. Out of the 25 motors purchased, one of the motors had terminal connection problems. Depending on the direction of force applied to the terminals would alter whether or not the motor received power. Additionally, after assembly three motors showed signs of seized or shattered gearboxes and were no longer usable. This was believed to be because of back-driving the motors during assembly and general handling of sub-systems.

5.2 Controls and Electrical Results

5.2.1 Leg Controller

The final printed and fabricated leg controller can be seen below in Figure 67 and Figure 68.

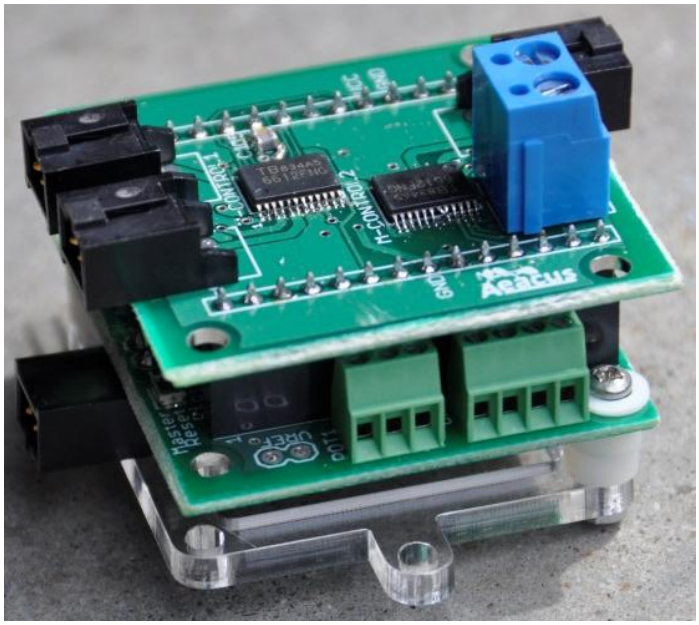


Figure 67: Leg Controller

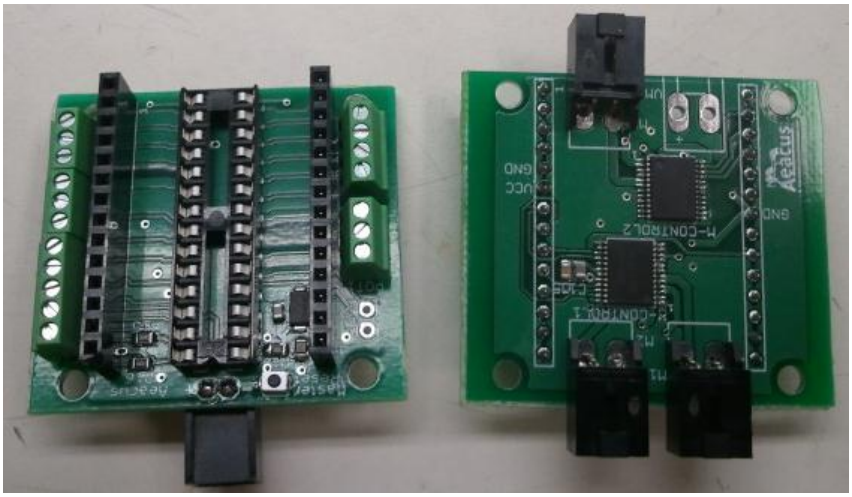


Figure 68: Leg Controller Separated

The majority of the leg controllers were functional, but there were a few flaws that needed to be fixed. There were a few simple mistakes along the lines of component selection, such as the fact that a 0805 100uF capacitor does not exist. The first major flaw was the pin allocation for PWM outputs. With this PIC, there are several channels for PWM signals but there are also several pin outs for one PWM signal. This went unnoticed during design and what was thought to be three different PWM signals turned out to be one signal on three outputs. This meant that all motors were getting the same signal. To fix this, four of the traces running from the PIC to the header were broken and wires were soldered on as

jumpers to adjust for the proper pins. Pins 23 and 24 were switched and 22 and 26 were switched. This was comparatively simple fix to a very disastrous problem.

The next issue came with the op amp circuit. The current sensor was properly reading the current running across the motors and outputting the proper voltages, but the signal coming out of the op amp was not correct. The signal entering the op amp would be around 2.53V but the signal coming out was more around 0.06V. This did not make much sense, but before the problem could be solved, it was put on the backburner so that the boards could be programmed.

The UART receiver on the PIC proved to be problematic, and we were unable to get it to function. Bytes transmitted to the PIC would fail to appear in the receive buffer, and the receive interrupt would not be triggered. The issue was reproduced on multiple PICs, and in all cases, the UART transmitter worked flawlessly at speeds of at least 1 Mbps. Further research found that a number of individuals have had major UART receiver issues with devices from the same family, often with no solution, in once case baffling a pair of product support engineers.

In the original leg controller architecture, ADC operations were performed in the high priority interrupt in the interest of not delaying the main control loop. However, the ADC proved to be triggering too quickly, and so many ADC interrupts were called in the main loop that PWM generation and timer reloading were discovered to suffer from significant interference issues. Attempts were made to slow the ADC conversion to a rate of six conversions per main loop cycle, which would be just enough to have a full set of fresh analog input data per iteration of the main loop. As a result, the ADC operations needed to be moved down to the low-priority main loop.

5.2.2 Communications Bus

Though we were unable to test the full leg controller network due to the PIC UART receiver issues, alternative test devices were able to verify that the RS-485 drivers were able to function as intended.

6 Conclusion

6.1 Mechanical Conclusions

The robot, while meeting many of the initial design specifications, could still be improved upon further. The focus of the improvements would be to make the robot lighter without degrading structural integrity, improving maximum payload and increasing power efficiency.

6.2 Controls and Electrical Conclusions

The custom leg controllers worked well except when for the serial communication. The 8-bit processors will need to be changed for the next iteration so that communication between legs can be established. The Gumstix processor can also be implemented to handle the rest of the controls.

7 Future Work

7.1.1 Robot Improvements

To reduce the overall weight of the robot, less hardware should be used. Much of the weight of the robot is built up by small pieces of hardware used unnecessarily. To remove this hardware without lowering the strength of the robot, dovetails or additional mounting ridges could be used. This includes the mounting point for the shoulder to the central body as well as the tail and neck. Furthermore, standoffs, which are the heaviest piece of hardware currently on the robot, could be replaced with plastic frame work and smaller mounting hardware. Where possible, removing unnecessary material would decrease the weight of the robot. This can be done without lowering the strength by doing FEA on individual parts and assemblies to determine what material is not needed.

An upper body shell to protect the electronics would also benefit the robot. The shell could provide a mounting point for electronics, as mentioned earlier, and protect them from damage if the robot were to topple or something to fall on it. This shell could have a fan or heat sink mounted to it for cooling of the electronics. The shell could be further developed to be enclosed, protecting against rain, dust and other debris.

7.1.2 Leg Improvements

After working with the current leg design many possible improvements were found to increase the strength of the leg frame, better the pulley system and reduce the overall weight. The most influential change in the leg design would be to use plastic molded parts rather than printed ABS plastic. This change would allow the same frame to be stronger and lighter than the current design. Failures related to the printing method, such as the breaking of the shoulder joint and poor tolerancing would be eliminated. All parts have already been designed to be made out of molded plastic with the exception of the guidance channels. The guidance channels would then be made an additional part, separate from the shoulder. This additional part would still need to be made via rapid prototyping or one time use molding. How this new part could be accommodated as well as additional changes can be found in a later section.

The next major improvement to the legs would be to include smaller potentiometers with bore mounts onto the joint links. This would give a more accurate reading of the position of each joint. Also, placing

the potentiometers on the joints rather than the motors will require a smaller lower body shell, thus increasing the clearance underneath the robot. Also, by adding limit switches to maximum ranges of motion of each joint, the potentiometers could “zeroed” automatically.

7.1.2.1 Leg Frame Redesign

To increase the efficiency of the robot while standing and walking the leg sub-system was redesigned so that the joint angles at during these times are in optimal positions. First, the shoulder joint was extended out ward so the upper leg could reach a completely vertical position. The lower leg was then redesigned so that the curve in the leg is able to wrap around the upper leg. This curve allowed the effective link to be completely vertical as well. These two changes allow the robot to be in a “resting position” where all of the robot’s weight is being transferred to the ground through frame rather than relying on the pulley system to not back drive the motors. This leg design can be seen in Figure 69.

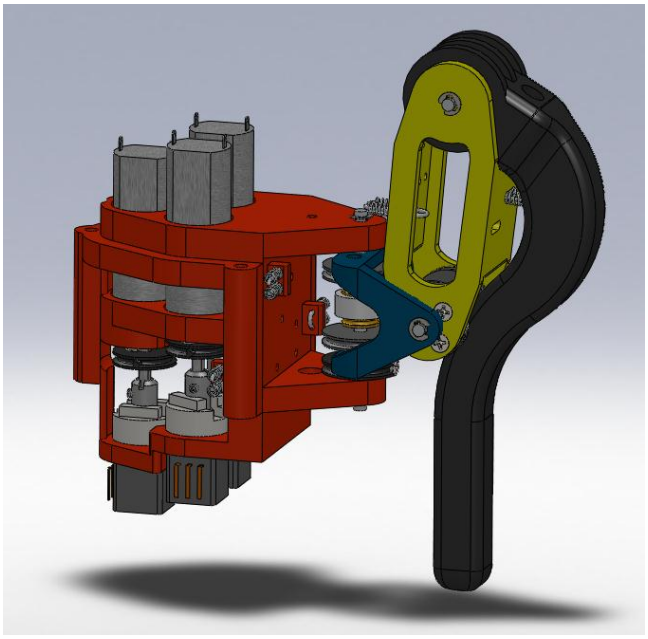


Figure 69: Leg Redesign

The change in range of motion caused by the adjustment to the leg links allow the robot to walk in a more optimal position. The required joint angle of the upper leg to give the robot walking clearance is +75 degrees, much less than + 60 degrees needed in earlier iterations. This reduces the lever arm of the leg thus requiring less torque from the motor and reducing the power requirement.

The changes in the upper and lower leg require an increase in the size of the V-bracket in the shoulder joint. The increase in length of the V bracket with also cause an increase in the amount of torque seen by the link. However, the change from printed ABS to a molded plastic would relieve the stress concentration around the axle hole. The additional space cause by the extension of the bracket was used to increase the range of motion. The redesigned shoulder design can be seen below in Figure 70.

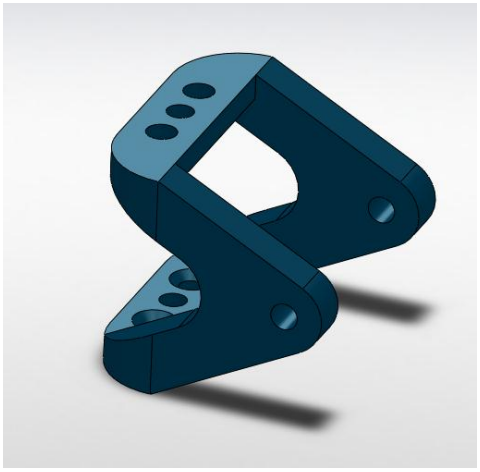


Figure 70: Shoulder Joint Redesign

The upper leg was first modified to accommodate the new passive tensioners. To do this, holes were placed in the side walls to make assembly easier. Additionally, the link wise thickened to increase the amount of tensioning possible by the new system. To accommodate the change in joint angle, notches were added to allow for a +/- 90 degree range of motion.

Improvements to the shoulder mount would include the incorporation if wiring channels and the additional guidance channel mentioned earlier. The wiring channel would allow for all wires related to the leg subsystem to be wired through the shoulder, thus removing the need to place wired through the central body. Additionally, placing the motor controllers on an upper body shell would no longer require their removal if a leg was removed. To accommodate the additional guidance tube part only slight changes would need to be made. To keep the shoulder mount manufacturable by plastic molding, a large hole would replace the current location of the tubes. An indentation around the newly made notch would be used mount the reinforcement plates. The guidance channels could then be inserted into the notch and secured in place by mounting the two reinforcement plates. This can be seen in Figure 71 and Figure 72.

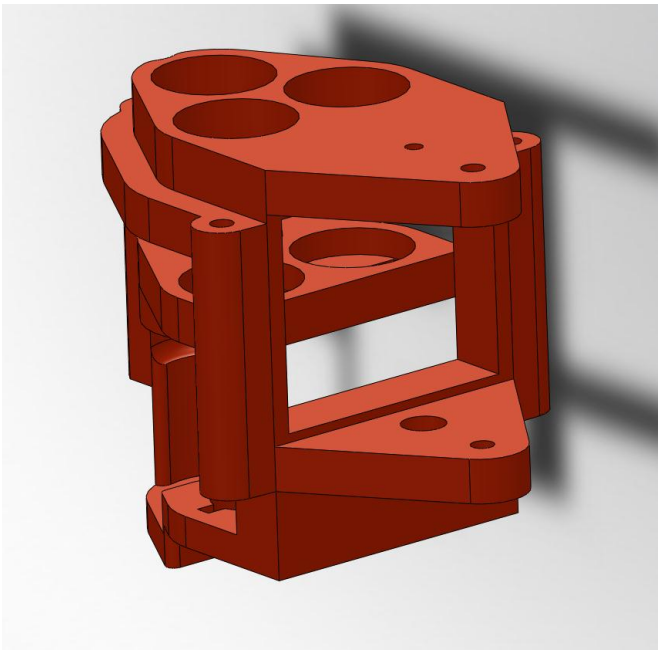


Figure 71: Shoulder Mount Redesign

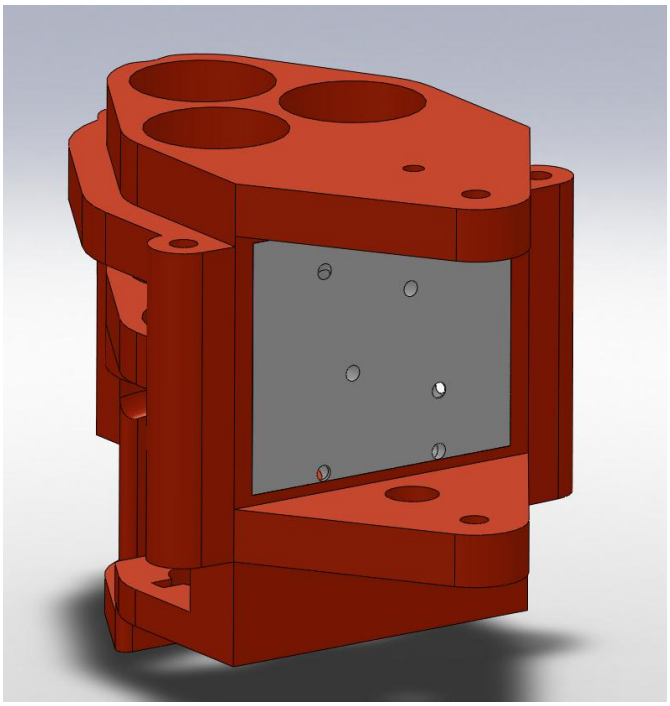


Figure 72: Shoulder Mount Assembly

7.1.2.2 Pulley System Improvements

The current pulley system would be best improved by replacing all mounting pulleys with metal pulleys with tighter tolerances than those current used. The metal pulleys, with an automated CNC process,

could be with better tolerancing and all similar. This would lower the amount of slop caused by the center bore being too large and would allow for a more secure mounting point for the cable. Tighter tolerances on the spacers between the idler pulleys will also improve tolerancing. Special order, higher test fishing line would also improve tolerancing and the maximum force the legs can withstand.

To further improve the pulley system better motors should be used. The motors had an amount of backlash that detrimental to how accurate the leg placement was. Better motors with less backlash and higher quality gears would allow the system to perform better. To further improve the system a greater gear ratio would allow the legs to lift more without lowering the speed of the robot below design specifications.

7.1.3 Future Mechanical Possibilities

With the modular and expandable design of the robot, many changes could be made in the future designs that would improve the versatility of the system. One possibility considered was the development of roles. The different roles would equate to different configurations of the platform. These roles could include “scout” robots that would have less of a payload but were quicker and more agile than the standard robot. These scouts, equipped with more sensory equipment, could look for large amounts of recyclables and have the rest of the swarm collect and sort them. Other configurations could also include “brute”. This configuration would be the opposite of a scout where it would stronger motors but was slower.

Another concept discussed was the development of central collected and recharging station, similar to an ant hill. The “Hill” would allow the robots to recharge autonomously and give them a 0,0,0 reference point for navigational purposes. The Hill would also act as a human interface center where a person could observe the state of the swarm.

7.1.4 Electrical Improvements

7.1.4.1 Main Controller

The expansion board we used with the Gumstix was unfortunately not well-suited for our application. No access was provided to the Linux debug console port, which would make in-system debugging and reconfiguration much easier.

Thankfully, the Overo COM is designed so that the end user can design their own interface board to fit their requirements. Due to time constraints, we were unable to do so. By designing a custom expansion board for the Gumstix, the limited space inside the robot could be used much more effectively by having only what is needed for the robot on the board. In addition, connectors for interfacing to leg controllers and future sensors can be included, making cable routing and wiring much simpler.

An idea discussed early on in the project was using a field-programmable gate array, or FPGA, to serve the same purpose as the network of leg controllers. The control loops for each leg and joint could be implemented in hardware, simplifying programming and increasing their performance. Image processing could also be at least partially offloaded to the FPGA, which would remove one of the largest loads from the CPU on the Gumstix.

7.1.4.2 Leg Controllers

The screw terminals used to connect sensors and communications to the board were time-consuming to use when testing required swapping connections frequently. Also, due to the stacking boards, the screw terminals were inaccessible when the top board was installed, requiring disassembly to change the connections. Replacing these with edge-mount connectors would make this process much faster.

References

Acosta-Avalos, D., Wajnberg, E., Oliveira, P., Leal, I., Farina, M., & Esquivel, D. (1999). Isolation of magnetic nanoparticles from pachycindyla marginata ants. *Journal of Experimental Biology*(202), 2687-2692.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms.* Springer.

Gordon, D. M. (1997). Networking ant: how do ants find the cake crumb you dropped. *Natural History Magazine*.

Hu, J., Pratt, J., Chew, C.-M., Herr, H., & Pratt, G. (1998). Adaptive Virtual Model Control of a Bipedal Walking Robot. *IEEE International Joint Symposia on Intelligence and Systems.* Rockville, MD.

Pratt, J. E. (1995). *Virtual Model Control of a Biped Walking Robot.* Masters Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge.

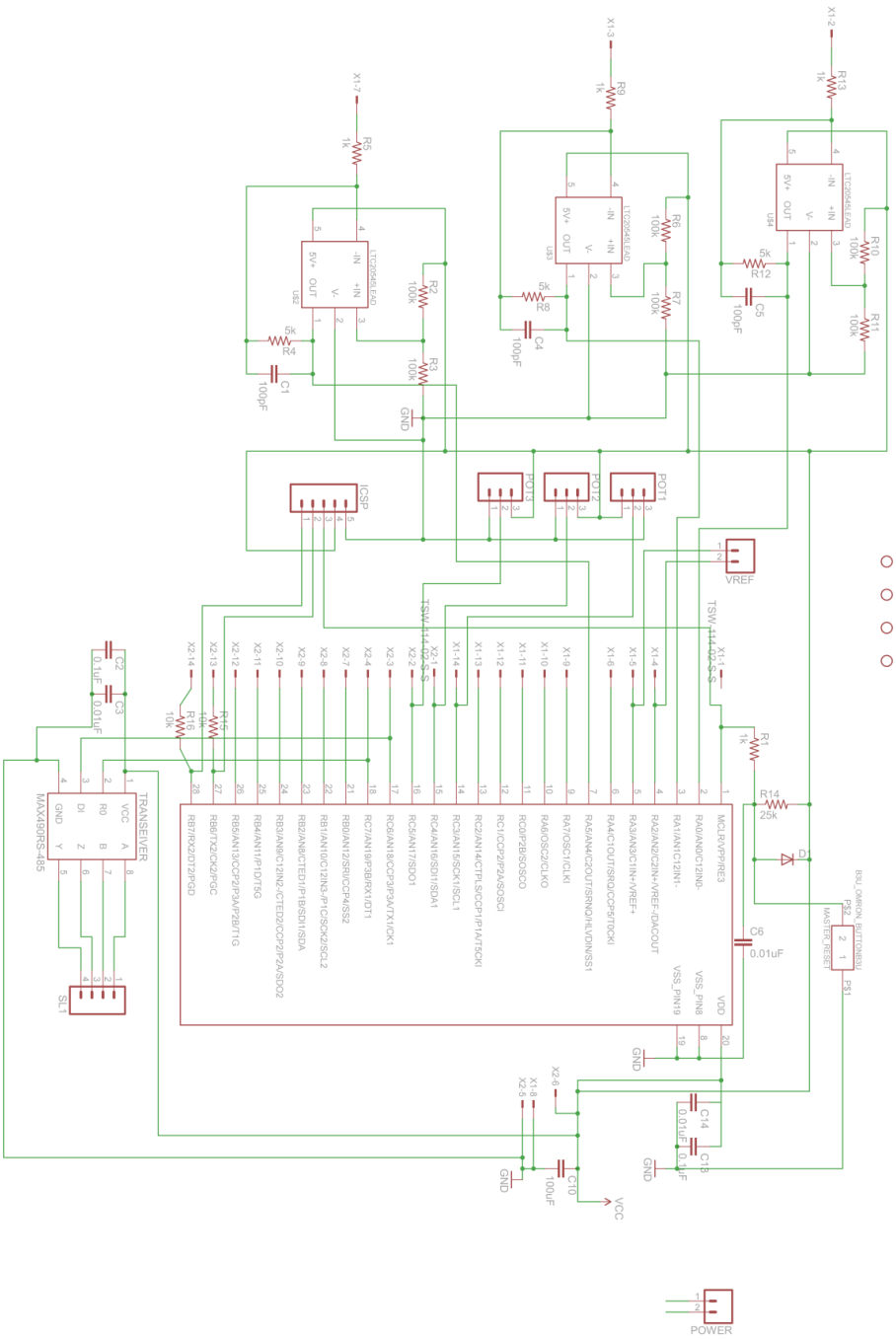
Ruiz, M. (2006, June 16). *File:Scheme ant worker anatomy-en.svg.* Retrieved from Wikipedia: http://en.wikipedia.org/wiki/File:Scheme_ant_worker_anatomy-en.svg

Wehner, R. (2003). Desert ant navigation: how miniature brains solve comlex tasks. *J Comp Physiol A*(189), 579-588.

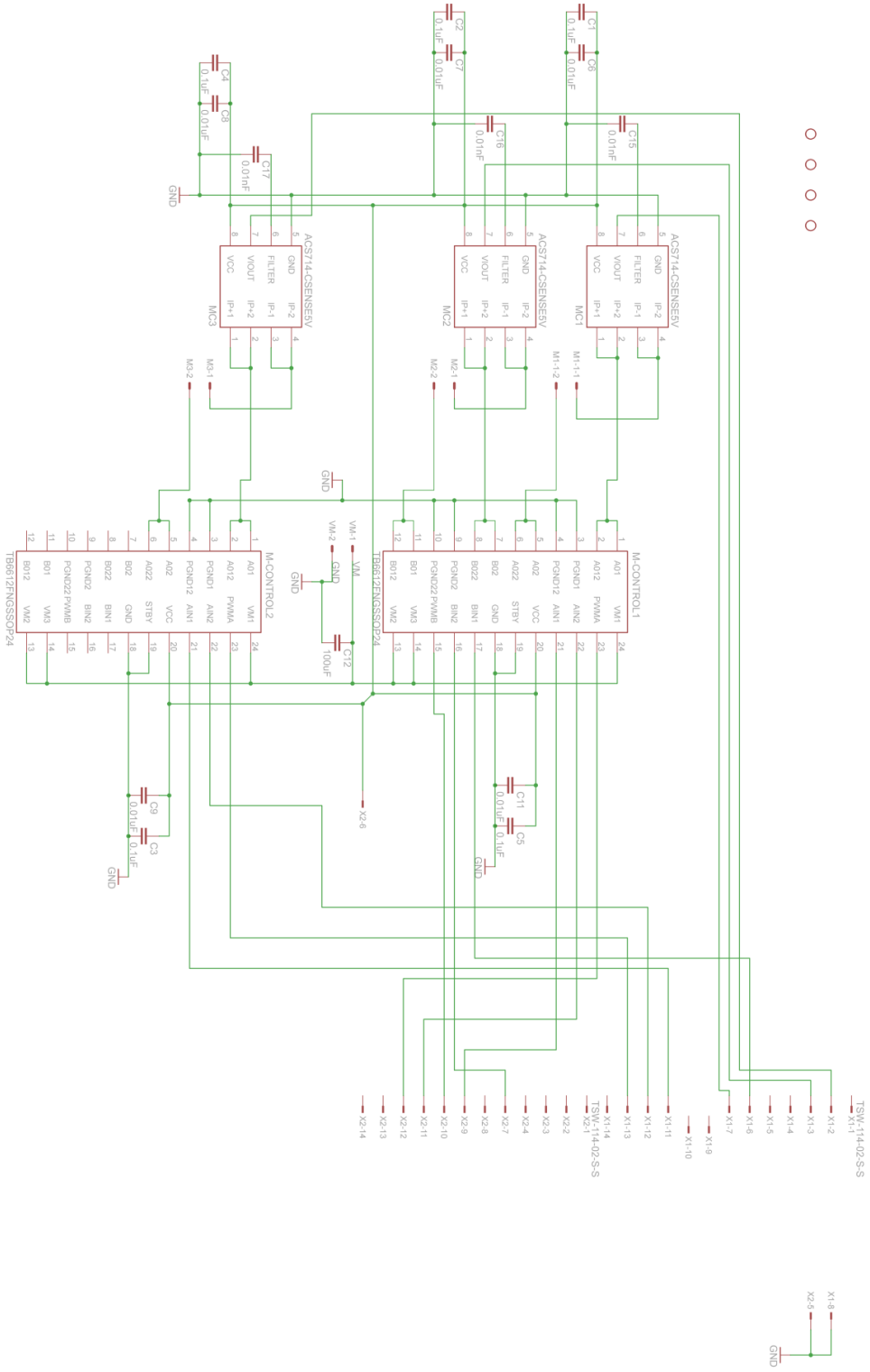
Wehner, R., Barbara, M., & Antonsen, P. (1996). Visual Navigation in Insects: Coupling of egocentric and geocentric information. *Journal of Experimental Biology*(199), 129-140.

8 Appendix A: Leg Controllers

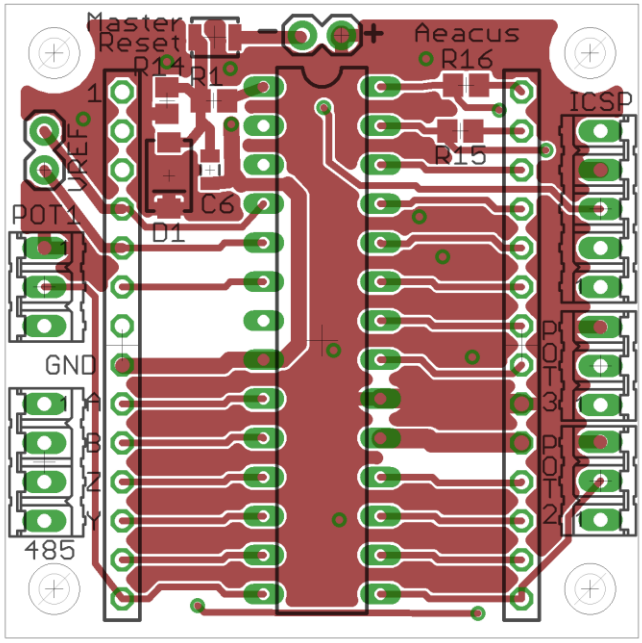
8.1 Board 1 Schematic



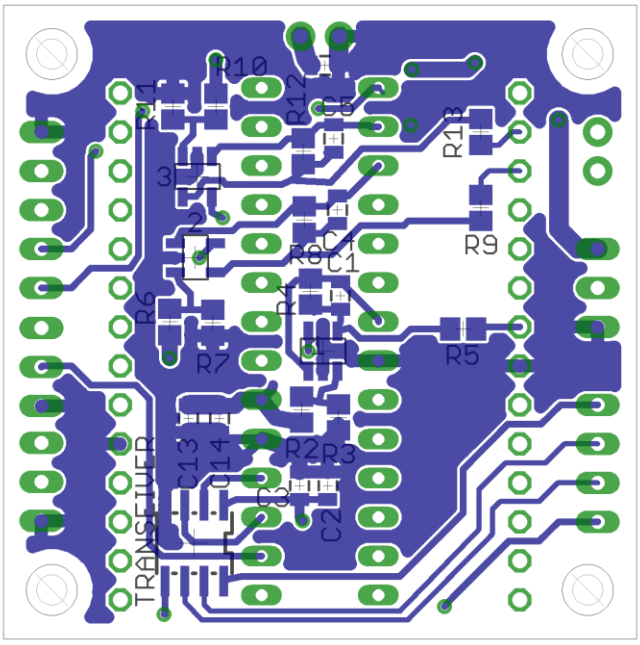
8.2 Board 2 Schematic



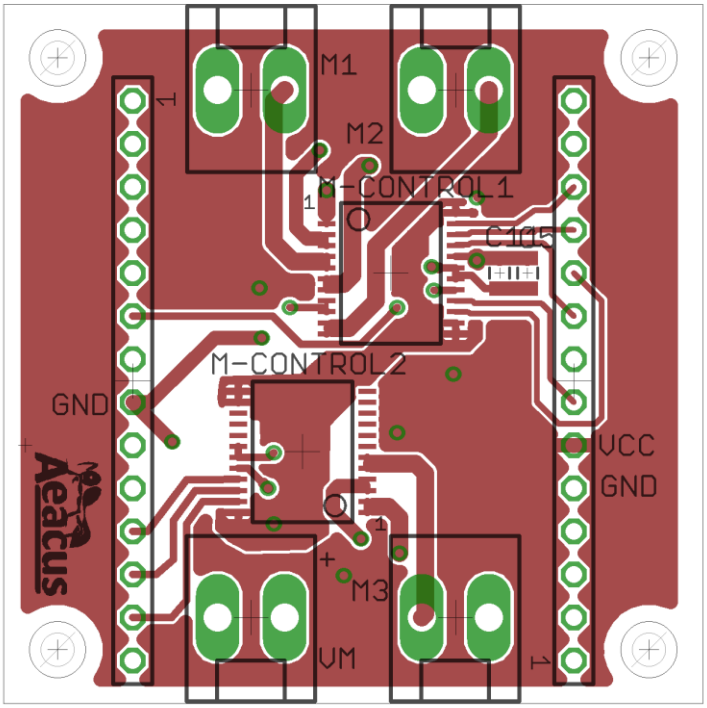
8.3 Board 1 Top Side



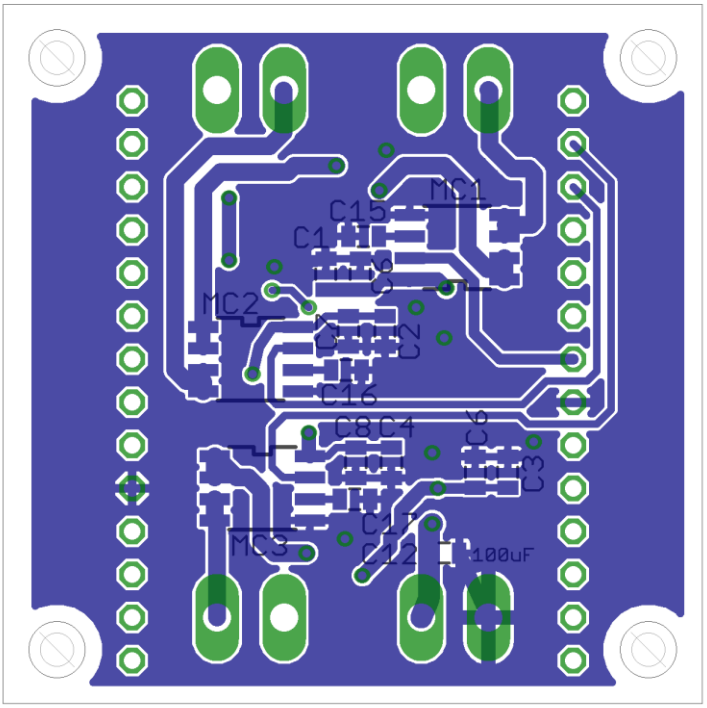
8.4 Board 1 Bottom Side



8.5 Board 2 Top Side



8.6 Board 2 Bottom Side



9 Appendix B: Bill of Materials

9.1 Robot

ITEM NO.	PART NUMBER	SW-Title(Title)	QTY.
1	Core Body Assembly		1
	BP-X1-CL		1
	BP-X2-CL		1
	BS-X1-CP		1
	Shoulder with motors		6
	LB-X1-CP	Shoulder Bracket	1
	XX-M1-PB-MS-16118-050	.75 Nm Motor	3
	9434K12	Shoulder Tensioning Spring	8
	LB-C1-CF	Shoulder Mount Tensioning Clip	4
	LB-A1-CF	Shoulder Axel	1
	XX-E1-PM-98408A116	1/8" E-Clip	2
	91375A074	2-56 Motor Set Screw	6
	XX-N1-CF	Encoder Support Nut	3
	XX-W1-PM-95229A480	Encoder Washer	3
	XX-V2-MD-P260T-D1BS3CB100K	Potentiometer	3
	90272A076	2-54 Pulley Screw	9
	LB-P1-MM-3434T32	5/8" Diamter Motor Pulley	3
	Bolt and nut assembly		20
	91251A162		1
	98032A421		2
	90730A005		1
	Bolt and nut assembly - long		4
	91251A162		1
	98032A421		2
	90730A005		1
	Left Right Neck		1
	HN-X1-CP		1
	XX-M1-PB-MS-16118-050	.75 Nm Motor	1
	90272A076	2-54 Pulley Screw	2

	XX-V2-MD-P260T-D1BS3CB100K	Potentiometer	2
	XX-N2-PG-1JLR9		2
	XX-W1-PM-95229A480	Encoder Washer	4
	91375A074	2-56 Motor Set Screw	2
	Tail Bracket		1
	TM-X1-CP		1
	XX-M1-PB-MS-16118-050	.75 Nm Motor	1
	XX-G2-PS-S1D93Z-P048SS		1
	XX-G1-PS-S1C83Z-P048B030S		1
	TM-A1-CF		1
	XX-E1-PM-98408A116	1/8" E-Clip	2
	XX-V2-MD-P260T-D1BS3CB100K	Potentiometer	1
	XX-W1-PM-95229A480	Encoder Washer	2
	XX-N2-PG-1JLR9		1
	91375A074	2-56 Motor Set Screw	1
	top body shell		1
	BS-X2-CP		1
	motor controller arrangement 2		8
	Motor Controller Board		2
	XX-W2-PM-94639A199		4
2	Shoulder		6
	LS-X1-CP	Shoulder Movement Bracket	1
	LX-P1-PM-3434T31	.375" Diameter Pulley	2
	LS-P1-PM-3434T33	.75" Diamter Pulley	2
	LS-P2-MM-3434T33	.75" Diamter Shoulder Movement Pulley	1
	LX-W1-PM-95630A435	.0325" Spacer	4
	LX-W2-CF	.21" Custom Spacer	1
	90272A076	2-54 Pulley Screw	1
	LS-S1-PM-91253A092	3-54 Hex Mounting Screw	2
	LS-X2-CP	Shoulder Movement Bracket	1
3	Upper leg		6
	LU-X1-CP-2	Upper Leg Frame	1

	LU-P2-MM-3434T33	.75" Diameter Upper Leg Movement Pulley	2
	LX-W1-PM-95630A435	.0325" Spacer	2
	LX-P1-PM-3434T31	.375" Diameter Pulley	2
	LU-A1-CF	Upper Leg Axel	1
	XX-E1-PM-98408A116	1/8" E-Clip	2
	LX-W2-CF	.21" Custom Spacer	1
	91771A092	3-54 Mounting Screw	8
	9434K12	Shoulder Tensioning Spring	4
	90272A076	2-54 Pulley Screw	2
	LU-C1-CF	Upper Leg Tensioning Clip	2
4	Lower leg		6
	LL-A1-CF	Lower Leg Axel	1
	XX-E1-PM-98408A116	1/8" E-Clip	2
	LL-X1-CP-4	Lower Leg	1
5	Battery Case		1
	TP-X1-CL		2
	TP-X2-CL		1
	Battery		6
	TP-P1-MM-3434T33	.75" Diamter Pulley	2
	91375A074	2-56 Motor Set Screw	4
	90272A076	2-54 Pulley Screw	8
	TP-X3-CL		1
	90730A005		8
	91255A108		8
6	Up Down Neck		1
	HN-X2-CP		1
	HN-X3-CP		1
	92949A108		2
	90730A005		2
	XX-M1-PB-MS-16118-050	.75 Nm Motor	1
	XX-G2-PS-S1D93Z-P048SS		1
	XX-G1-PS-S1C83Z-P048B030S		1
	HN-A1-CF		1
	90272A076	2-54 Pulley Screw	4
	XX-V2-MD-P260T-D1BS3CB100K	Potentiometer	1

	XX-W1-PM-95229A480	Encoder Washer	2
	XX-N2-PG-1JLR9		1
	91375A074	2-56 Motor Set Screw	3
	HN-P1-MM-3434T33	.75" Diamter Pulley	1
7	Pincers		1
	HN-X4-CP		1
	HN-P1-MM-3434T33	.75" Diamter Pulley	2
	LS-S1-PM-91253A092	3-54 Hex Mounting Screw	4
	XX-M1-PB-MS-16118-050	.75 Nm Motor	1
	LX-W1-PM-95630A435	.0325" Spacer	2
	HN-X5-CP		1
	HN-X6-CP		1
	HN-X7-CP		1
	HN-X9-CP		2

9.2 Leg Controllers

9.2.1 Board 1

Part	Value	Package	Library	Position (inch)	Orientation
Cap	100pF	0805	SparkFun	(0.76 0.88)	MR270
Cap	0.1uF	0805	SparkFun	(0.7925 0.3925)	MR90
Cap	0.01uF	0805	SparkFun	(0.865 0.3925)	MR90
Cap	100pF	0805	SparkFun	(0.7675 1.1)	MR270
Cap	100pF	0805	SparkFun	(0.7775 1.2825)	MR270
Cap	0.01uF	0805	SparkFun	(0.525 1.2)	R270
Cap	100uF	0805	SparkFun	(0.8 1.47)	MR0
Cap	0.1uF	0805	SparkFun	(1.15 0.565)	MR270
Cap	0.01uF	0805	SparkFun	(1.07 0.565)	MR270
Diode		SMA-DIODE	SparkFun	(0.42 1.185)	R270
ICSP		05P	con-amp-quick	(1.5275 1.1)	R90
MASTER_RESET	B3U_OMRON_BUTTONB3U	BUTTON	Aeacus2	(0.5375 1.54)	R180

POT1 03P con-amp-quick (0.1 0.9) R270

POT2 03P con-amp-quick (1.5275 0.4025) R90

POT3 03P con-amp-quick (1.5275 0.6975) R90

POWER-Con 1X02 SparkFun (0.8625 1.545) R180

Res 1k R0805 resistor (0.535 1.3775) R180

Res 100k R0805 resistor (0.86 0.5875) MR90

Res 100k R0805 resistor (0.765 0.5675) MR270

Res 5k R0805 resistor (0.8375 0.89) MR270

Res 1k R0805 resistor (0.445 0.795) MR180

Res 100k R0805 resistor (1.1975 0.8125) MR90

Res 100k R0805 resistor (1.0875 0.81) MR270

Res 5k R0805 resistor (0.8525 1.075) MR270

Res 1k R0805 resistor (0.4 1.105) MR270

Res 100k R0805 resistor (1.08 1.365) MR270

Res 100k R0805 resistor (1.19 1.365) MR90

Res 5k R0805 resistor (0.855 1.25) MR270

Res 1k R0805 resistor (0.4 1.3) MR90

Res 25k R0805 resistor (0.415 1.3775) R270

Res 10k R0805 resistor (1.1675 1.3) R180

Res 10k R0805 resistor (1.1825 1.4175) R180

Con 04P con-amp-quick (0.1 0.45) R270

TRANSEIVER MAX490RS-485 MAX490 Aeacus2 (1.135 0.245) MR270

PIC PIC18F26K22_SMALLSPDIP 28LEAD_SPDIP Aeacus2 (0.8125 0.7625) R270

Op-amp LTC20545LEAD 5-LEAD_TSOT-23 Aeacus2 (0.805 0.7375) MR0

Op-amp LTC20545LEAD 5-LEAD_TSOT-23 Aeacus2 (1.1325 0.9775) MR270

Op-amp LTC20545LEAD 5-LEAD_TSOT-23 Aeacus2 (1.1275 1.185) MR180

Con 1X02 SparkFun (0.1 1.2) R90

Header TSW-114-02-S-S TSW-114-02-S-S con-samtec (0.3 0.75) R90

Header TSW-114-02-S-S TSW-114-02-S-S con-samtec (1.325 0.75) R270

9.2.2 Board 2

Part Value Package Library Position (inch) Orientation

Cap 0.1uF 0805 SparkFun (0.875 0.9975) MR90

Cap 0.1uF 0805 SparkFun (0.7375 0.865) MR270

Cap 0.1uF 0805 SparkFun (0.4525 0.5375) MR90

Cap 0.1uF 0805 SparkFun (0.7225 0.56) MR270

Cap 0.1uF 0805 SparkFun (1.2175 1) R270

Cap 0.01uF 0805 SparkFun (0.795 0.9975) MR90

Cap 0.01uF 0805 SparkFun (0.8225 0.865) MR270

Cap 0.01uF 0805 SparkFun (0.805 0.56) MR270

Cap 0.01uF 0805 SparkFun (0.5275 0.5375) MR90

Cap 0.01uF 0805 SparkFun (1.1525 1) R270

Cap 100uF 0805 SparkFun (0.6 0.35) MR0

Cap 0.01nF 0805 SparkFun (0.7875 1.085) MR180

Cap 0.01nF 0805 SparkFun (0.8275 0.775) MR0

Cap 0.01nF 0805 SparkFun (0.8075 0.475) MR0

M-CONTROL1 TB6612FNGSSOP24 SSOP24 Aeacus (0.9 1) R270

M-CONTROL2 TB6612FNGSSOP24 SSOP24 Aeacus (0.695 0.585) R90

con KK-156-2 con-molex (0.575 1.425) R0

Con KK-156-2 con-molex (1.05 1.425) R0

Con	KK-156-2	con-molex	(1.05 0.2)	R180	
Cur.Sense	ACS714-CSENSE5V	8-PIN_SOIC	Aeacus2	(0.57 1.06)	MR90
Cur.Sense	ACS714-CSENSE5V	8-PIN_SOIC	Aeacus2	(1.05 0.8)	MR270
Cur.Sense	ACS714-CSENSE5V	8-PIN_SOIC	Aeacus2	(1.0225 0.5)	MR270
con	KK-156-2	con-molex	(0.575 0.2)	R180	
Con	TSW-114-02-S-S	TSW-114-02-S-S	con-samtec	(0.3 0.75)	R90
Con	TSW-114-02-S-S	TSW-114-02-S-S	con-samtec	(1.325 0.75)	R270