eclipse technologies

System Performance Specifications
Project Lycanthrope

REVOLUTIONIZE EMPOWER DESIGN

TABLE OF CONTENTS

COMPANY IDENTITY

VISION STATEMENT

Eclipse Technologies strives to drive change and surpass limitations. The world will not progress into a better place if changes aren't made. We aim to do the impossible in order to motivate others.

MISSION STATEMENT

To design and create innovative technology deemed impossible that is beneficial to all; To gain purposeful experiences as high school students; To exhibit professionalism, intelligence and realism.

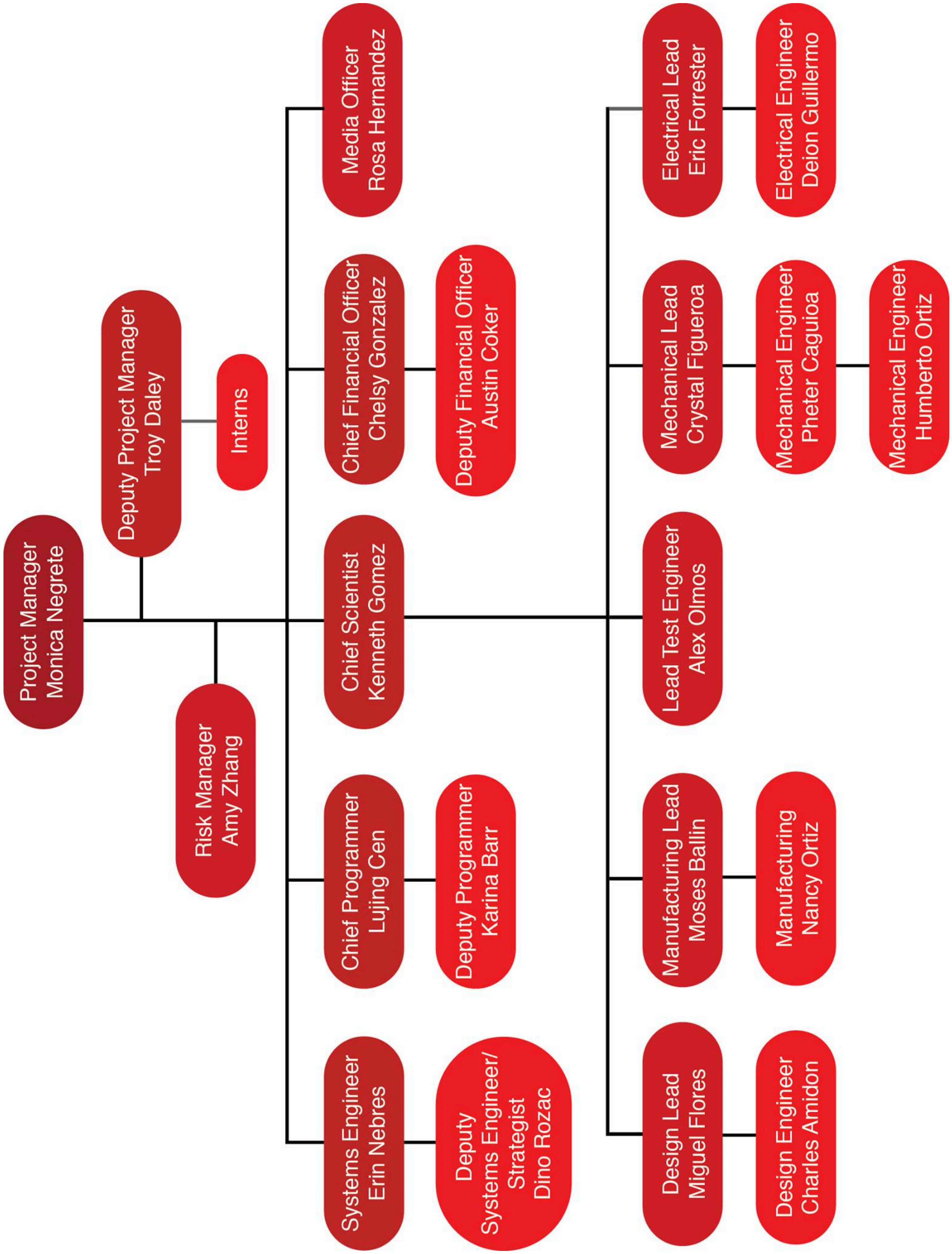
ABOUT US

Eclipse Technologies was founded by 22 high school students from the California Academy of Mathematics and Science (CAMS). Before tackling the Lycanthrope Project, Eclipse Technologies came up with 5 core values: embrace the adventure, simplify, ship, champion the mission, and inspire all.

Staying true to their core value of inspiring all, Eclipse Technologies plans to implement an outreach program to teach middle-school students about the STEM field. Eclipse Technologies believes it is necessary to inspire younger generations to consider the STEM field as a pathway because they are the future of the world. Our vision is to surpass the limitations of current technology and younger minds will be the ones creating that new technology. Everyone at Eclipse Technologies has been inspired by someone in STEM at some point in their lives and we intend to be that inspiration for the next generation.

Eclipse Technologies will partner with different middle schools to provide an engaging and eye-opening experience to their students. We plan to teach them the basics of engineering and give them a hands-on experience applying these principles.

TEAM ORGANIZATION CHART



JOB DESCRIPTIONS

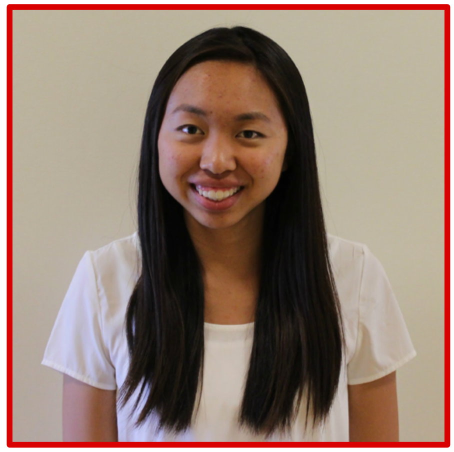
Project Manager (PM): Monica Negrete

The PM is the person in charge of leading the company to success. The PM is responsible for the planning and completion of Lycanthrope Project. She is the backbone of any project and is responsible for the team. She governs the direction of the team and the project and is responsible for making sure the requirements are met. As PM, she must manage and approve schedules, assign tasks to the different systems, and maintain contact with the customer. The PM must be able to speak and understand the personalities of every member of the company. She has the authority to switch team positions, if needed. The system's lead reports any progress or complications to the project manager. If any problems arise between team members, the PM is in charge of finding a solution. The customer will only contact her if there are any changes to the project. In order to take this role, she must be ready to exhibit leadership, take responsibility for downfalls, and be the support system that the team can rely on.

Deputy Project Manager (DPM): Troy Daley

While working closely with the PM, the DPM helps oversee different aspects of the project. The DPM is also in charge of the company's internship and outreach programs. The internship program gives younger high school students the opportunity to work with an engineering company to help develop their skills while working on their own independent project. They also are given the chance to work with members of the company. Through the outreach program, Eclipse teaches fundamental engineering concepts to a younger audience in the hopes of promoting and inspiring the next generation to join STEM fields.

Risk Management Officer: Amy Zhang



The Risk Management Officer identifies and analyzes all risks in order to minimize their effects on the company's success. She ensures that the uncertainty involved with the project will not hinder its execution. This is achieved using Six Sigma and Lean practices, which are systematic methods of managing variability, upholding quality, and reducing waste in the manufacturing process. She is also responsible for implementing safety protocol and documenting workplace accidents.

Chief Scientist: Kenneth Gomez

The Chief Scientist ensures that the final product design has the capability to efficiently complete the mission. The Chief Scientist leads the company through the design process and oversees the engineering portion of the company. It is the Chief Scientist's responsibility to understand every technical aspect of the product in order to effectively lead the engineering teams throughout the project. The Chief Scientist works with the Systems Engineer to successfully integrate every component into the final design, while working with the PM and DPM to maintain order and peace within the company.

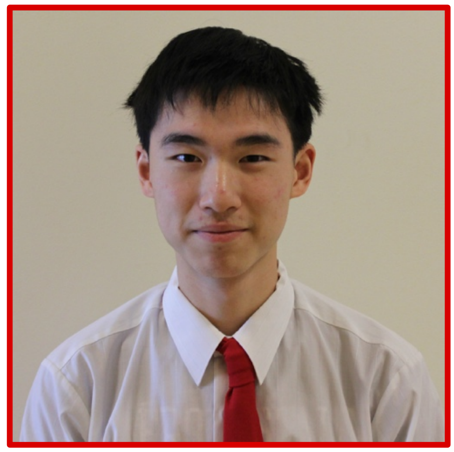
Systems Engineer: Erin Nebres

The Systems Engineer is responsible for taking the team's designs and ideas for the product and organizing them into an efficient and realistic plan of action. This entails working closely with the Chief Scientist in order to make the best design possible within the given constraints and communicating with all sub-teams. The Systems Engineer must utilize various tools to coordinate sub-teams, such as Gantt charts, work breakdown schedules, and other visual representations of the plan for the project.

Deputy Systems Engineer/ Strategist: Dino Rozac

The Co-Systems Engineer/Strategist is responsible for assisting the Systems Engineer with synthesizing the team's ideas, creating the most efficient plan of action, working with the Chief Scientist to design the most effective solution, and communicating with all sub-teams. The Co-Systems Engineer will also utilize the various tools used by the Systems Engineer, for example, Gantt charts and work breakdown schedules. In addition, the Co-Systems Engineer will anticipate strategies employed by competition and plan corresponding countermeasures.

Chief Programmer: Lujing Cen



The Chief Programmer is responsible for overseeing the programming team. The Programming Team ensures the functionality of high-level software interfacing with sensors and servos which allow the robots to autonomously perform desired actions. The team will also utilize languages such as Python, C++, and JavaScript to develop efficient monitoring, processing, and movement modules. The Chief Programmer works closely with the Electrical and Mechanical teams to design algorithms for controlling quadrupedal and bipedal locomotion. In addition, the Programming Team will report to the Lead Test Engineer frequently for code verification and hardware-software interface testing.

Deputy Programmer: Karina Barr

The Deputy Programmer is responsible for assisting the Programming Team. The Programming Team ensures the functionality of high-level software interfacing with sensors and servos which allow the robots to autonomously perform actions. The team will also use languages such as Python, C++, and JavaScript to develop efficient monitoring, processing, and movement modules. The Deputy Programmer works closely with the Electrical and Mechanical teams to design algorithms for controlling quadrupedal and bipedal locomotion. In addition, the Programming Team will report to the Lead Test Engineer frequently for code verification and hardware-software interface testing.

Chief Financial Officer: Chelsy Gonzalez

The Chief Financial Officer is responsible for the management of the company's funds. He is expected to record purchases made by the team, ensure team members are reimbursed for their purchases, and research opportunities for maximum return on investment. The Chief Financial Officer is responsible for ensuring the company has sufficient funds to operate. With the assistance of the Deputy Financial Officer, he is expected to help compile the company's bill of materials, manage all fundraisers held by the team, and manage the solicitation of funds from other parties. In addition, the Chief Financial Officer will help spearhead Eclipse Technologies' media campaign. He will coordinate all press releases.

Deputy Financial Officer: Austin Coker

The Deputy Financial Officer is responsible for assisting the financial team. The financial team is responsible for the management of the company's funds. The financial team is also expected to record purchases made by the team, ensure team members are reimbursed for their purchases, and research opportunities for maximum return on investment. The team is responsible for ensuring that the company has sufficient funds to operate. The financial team is also expected to help compile the company's bill of materials, manage all fundraisers held by the team, and manage the solicitation of funds from other parties. In addition, the Deputy Financial Officer will work with the Manufacturing Lead to create the bill of materials, and find the least expensive materials. The Deputy Financial Officer will coordinate all items bought by Eclipse Technologies.

Media Officer: Rosa Hernandez

The Media Officer's main task is to develop the team's corporate identity. The media officer does this by designing all company logos, badges, banners, and business cards. Other responsibilities of the media officer include supervising presentations of all documents, creating the company website, organizing all formal presentations, designing the Trade Expo display, and producing the company's documentary. The Media Officer also documents the team's progress by way of pictures and videos for the documentary, website, presentations, etc.

Lead Design Engineer: Miguel Flores

The Lead Design Engineer is in charge of managing the design team. The role of the Design Team is to collect the ideas brought up by the team. The goal for the Design Engineer is to use ideas to make a product that is capable of solving the problem. The ideas collected have to be used to make all of the components of the product to work. The goal is to design a product that will work and follow the constraints.

Design Engineer: Charles Amidon

The Design Engineer works closely with the Lead Design Engineer. Design team is responsible for collecting the ideas of the team and making the product successful. They are responsible for making the design work, ensuring all systems work together, and meeting constraints.

Lead Manufacturing Engineer: Moses Ballin

The Manufacturing Team manages the materials that will be used to make the robot. To do this, the Lead manufacturing engineer will communicate directly with the company's Chief Financial Officer to discuss whether parts and materials can be bought. It is the Manufacturing Team's goal to make the project. The Manufacturing Team is also responsible for managing equipment and teaching others how to use the equipment.

Manufacturing Engineer: Nancy Ortiz

The Manufacturing Engineer is responsible for assisting the manufacturing team with the fabrication and assembly of the robotic mechanism. The team also determines the most cost-effective means of fabrication and assembly. The team coordinates machine time in order to make the parts.

Lead Test Engineer: Alex Olmos

The Test Engineer closely follows the creation and assembly of the mechanisms to ensure the safety of the robot and its purposeful functions. She is in charge of overseeing the work of all the other engineers and contributing to the making of the robot. Her main role is to work with the Systems Engineer and Chief Programmer to conduct adequate testing procedures.

Lead Mechanical Engineer: Crystal Figueroa

The Lead Mechanical Engineer is in charge of the mechanical process that goes into the making of Wolfgang and the PackBOTs. She keeps an engineering notebook up to date, guides the Mechanical Team, and collaborates with the other sub-teams for the development of the robots. The Mechanical Team leads the construction of the robot.

Mechanical Engineers: Pheter Caguioa and Humberto Ortiz

The Mechanical Team focuses on the overall dynamics of Wolfgang and the PackBOTs. The function of the team is to ensure that the robots are well-built and well maintained. It is also the job of the Mechanical Team to help with the construction of the robot's drive and articulation.

Lead Electrical Engineer: Eric Forrester

The Lead Electrical Engineer is in charge of managing the Electrical Team and designing all electrical components of the robot. He acquires parts, designs schematics, and chooses the best methods and equipment necessary to fix problems. He is in charge of powering the robot as well as powering the systems incorporated with the robot, such as the motor and sensor controls. The Electrical Engineer is also in charge of robot communication through wired and wireless controllers. The Electrical Engineer will collaborate with the Chief Programmer and Systems Engineer to ensure cooperation between the different systems.

Electrical Engineer: Deion Guillermo

The Electrical Team is responsible for assisting the Lead Electrical Engineer with the systems within the robot. This includes tasks such as wiring the motors and sensors, maintaining the functionality of the electrical systems, and ensuring efficient power distribution.

Interns: Represented by Shayna Weldon

The lead intern's responsibilities include overseeing the intern team, delegating responsibilities, scheduling the project of the PreyBOT from its beginning to its completion, forming teams, monitoring the schedule, leading focus and brainstorming meetings, monitoring the progress of the team, and making adjustments when necessary. The lead intern coordinates and reports to the DPM.

Team Dynamic Assessment													
Name	Systems	CAD	Pro-gramming	Manu-facturing	Sen-sory	Arti-culation	Ele-ctrical	Strate-gist	Media	Commu-nication	Out-reach	Fi-nance	Total
Monica Negrete							1	1	1	1	1	1	6
Troy Daley	1	1			1	1	1	1	1	1	1		9
Amy Zhang						1		1	1	1			4
Erin Nebres	1	1		1						1	1		5
Dino Rozac	1		1	1			1	1		1			6
Lujing Cen	1	1	1		1	1	1					1	7
Karina Barr			1				1		1		1		4
Kenneth Gomez	1	1			1	1		1	1				6
Chelsy Gonzalez		1		1						1	1	1	5
Austin Coker										1	1	1	3
Rosa Hernandez				1					1		1	1	4
Miguel Flores		1							1		1		3
Charles Amidon	1	1		1		1							4
Moses Ballin				1		1	1						3
Nancy Ortiz		1						1	1		1		4
Alex Olmos				1					1	1	1		4
Crystal Figueroa	1	1	1	1		1			1	1	1		8
Pheter Caguioa	1					1		1					3
Humberto Ortiz		1							1	1	1		4
Eric Forrester	1	1					1	1					4
Deion Guillermo		1					1		1	1			4

MISSION OVERVIEW

PROBLEM STATEMENT

The primary focus of Eclipse Technologies is to design and produce three robotic devices to be used in military S&C missions. Currently, the military uses soldiers to complete S&C missions that have a low chance of a successful extraction. The lives of thousands of soldiers are being risked because of these missions. Eclipse Technologies hopes to decrease the number of lives lost during S&C missions by designing devices that will work together to identify the enemy and successfully transport the targeted enemy to a designated holding cell.

DESIGN BRIEF

The CAMS Advanced Research Projects Agency (CARPA) Initiative has challenged Eclipse Technology to design a Distributed Autonomous Robotic System (DARS) to complete a search and capture (S&C) mission. One of the team members, codenamed Wolfgang, must be a dual-mode, polymorphic robot that manipulates a steel ball in order to open the field and allow the other two members of the team, the PackBOTs, to search and either return or destroy the prey. The purpose of the challenge is to design, build, and test an autonomous, dual-mode robotic system, capable of performing the autonomous S&C mission, culminating in the return of a specified "enemy" target.

The mission will begin with Wolfgang in a four-legged state. It must traverse the field and find its way into the morphing chamber, where it will be exposed to ultraviolet (UV) light and a specific sound frequency. These will trigger Wolfgang's transformation into a two-legged walker. After three minutes, Wolfgang will revert back into a four-legged state. While in its two-legged state, Wolfgang will be tall enough to activate the switch that gives the PackBOTs access to the rest of the field. After the barrier is lowered, the PackBOTs will be allowed to enter the field and must scan each PreyBOT's Radio Frequency Identification (RFID) tag in order to return the correct enemy target. The PreyBOTs will be constructed by an anonymous entity and the design will be revealed on the date of the mission. If the PackBOTs are unable to capture and return the prey, the PreyBOTs can hit a kill switch to destroy the prey. CARPA would prefer if the PackBOTs returned the prey alive to a holding cell, but if the PackBOTs choose to destroy the prey, the mission will still be considered a success.

REQUIREMENTS FOR THE LYCANTHROPE PROJECT:

WOLFGANG:

- Must not exceed a height of 20 centimeters (cm) while in a four legged state.
- Must not exceed a height of 30 cm in a two legged state.
- Must not exceed a total weight of 8 kilograms (kg).
- Must automatically transform back onto four legs 3 minutes after transformation.
- Must be able to detect color, UV light, and sound frequencies.

PACKBOTS:

- Must not exceed a height of 20 cm.
- Must not exceed a weight of 4.5kg.
- Must be able to wirelessly communicate with each other, Wolfgang, and the mission control center.
- Must be able to read a passive RFID chip.
- Must capture and return proper PreyBOT to a designated area.

EXECUTIVE SUMMARY

Eclipse Technologies is currently developing technology that would save the lives of thousands of American soldiers who carry out highly dangerous search and destroy missions. We have designed a Distributed Autonomous Robotic System (DARS) capable of completing a similar search and capture (S&C) mission. Three walking robots -- codenamed Alpha, DOG-1E5, and DOG-4S1 -- must work together to capture two specific “PreyBOTs” from a group of four.

Alpha, initially on four legs, must morph into a two legged robot and place a steel ball in a cup. This triggers a mechanism that opens the rest of the field, allowing the DOGs to capture a PreyBOT. The Telekinesis Module allows Alpha to manipulate the steel ball, while the Lykos Module controls transformation. DOG-1E5 and DOG-4S1 are four legged walkers with maximized hunting capabilities through their Theia and Ares Modules, which are responsible for vision processing and PreyBOT capture, respectively. They will use a spring trap and a trailer for PreyBOT capture and transport.

PreyBOTs will be completely designed and manufactured by a team of interns hired by Eclipse Technologies. These interns will work at their own discretion, while assisting the main team if necessary.

Eclipse Technologies is responsible for raising and managing all financial assets throughout the course of the entire project, and will approach third party companies for sponsorships. The financial team will maintain a budget and ensure that responsible fiscal procedures are properly authorized, appropriated, executed and recorded.

Our company will also reach out to prospective engineering students in middle schools to educate and inform them about possibilities in STEM fields. We believe in creating a lasting impression through these young, driven students.

At Eclipse Technologies, we strive to drive change and surpass limitations through research, teamwork, and imagination. By taking on this project, we aim to do what seems impossible, and motivate others to contribute to worldwide technological innovation.

RISKS AND ASSUMPTIONS

LOCOMOTION

Eclipse Technologies assumes that the distance *Alpha* must walk in the field will not be problematic, which permits that speed need not be considered. Furthermore, Eclipse also assumes that the field does not introduce any sources of harm to the robot, such as excessive heat or rough terrain.

Every robot is required to walk on four legs, so the team must assume that the robot will be capable of balancing. Each robot's system of locomotion assumes that all four legs and each IMU sensor will always be fully functioning.

ARTICULATION

In order to activate the switch that releases the PreyBOTs, *Alpha* will use electromagnets to acquire the steel ball. The use of magnets introduces the possibility of the magnetic field affecting the chipsets. To lessen this risk, the Design Team will place all processors a sufficient distance away from the electromagnets in the arm.

A major assumption is that the team can successfully acquire the steel ball. Since the ball is composed of stainless steel, it may not contain enough magnetic material for the electromagnet to pick it up. To compensate for this risk, the team has backup methods that do not rely on magnetism. Successful steel ball manipulation is imperative as the team cannot complete the mission without the release of the PackBOTs.

The Pangolin design of *Alpha* assumes that the robot will be able to balance on two legs, with weight evenly distributed between the body and the tail. The tail, which consists of a system of pulleys, is essential to the design. The use of pulleys as the means of articulation introduces the risk of the string breaking or tangling. Therefore, the material that Eclipse uses for the string must be able to withstand the force of the tail, and the pulley system within the tail must be easily accessible to allow for necessary maintenance.

ROBOT COMMUNICATION

As all three robots are autonomous, communication from each robot to the control center and communication between the three robots are crucial to the successful execution of the mission. Eclipse Technologies assumes that all three robots will be able to communicate effectively with the command center, as the actions of the robots rely on human input from the command center. Without effective communication, the productivity and usability of the robots would be greatly impaired.

Furthermore, the team assumes that all sensors and cameras will be fully functioning throughout the duration of the mission, which includes the successful transfer of data from each device back to mission control. This assumption is imperative as failures in sensory input would dramatically increase the level of difficulty in completing the mission. Without knowledge of the specific location or position of the robots, it would be extremely difficult for human control to accurately guide the robots.

CAPTURE STRATEGY

Eclipse Technologies' primary capture method relies on the assumption that the PreyBOTs will walk on the nets that the PackBOTs place onto the field. If the PreyBOTs fail to walk on the traps, the team will not be able to continue with the mission. To minimize the risk of this assumption, the PackBOTs will aid in herding the PreyBOTs into the nets after deploying them. Additionally, the low nature of the nets ensures that the PreyBOTs' sensors will not be able to detect the nets as objects, further minimizing the probability of this risk.

Furthermore, Eclipse assumes that the PackBOTs will be able to deploy the nets without detonating the mechanism that releases the trap. This assumption is essential to the capture method's success, and the design of the PackBOTs will account for this assumption. Using a net to capture the PreyBOTs also assumes that the net will be strong enough to withstand the movement and force of the PreyBOTs. When designing the nets, the team assumes the PreyBOTs will be maximum weight capacity to account for this risk.

In regards to the trailer method, the same assumption is applicable in that the PreyBOTs must walk into the trailer to ensure a successful capture. If they don't, the team cannot continue the mission. However, as with the net capture method, the low design of the trailer will lessen the possibility of this risk occurring, and the PackBOTs will also aid in herding the PreyBOTs into the trailer. Additionally, the same assumptions regarding the vital nature of communication are applicable as the trailer must also communicate with mission control to alert the PackBOTs that the trailer is ready for transportation of the captured PreyBOTs. To eliminate the risk that the PackBOTs cannot recognize and reunite with the trailer after initial deployment, the trailer will be brightly colored and easily identifiable using vision tracking.

MANUFACTURING

Due to the small size of Eclipse Technologies' intended products, the company will manufacture a portion of the robots using three-dimensional printing. In order for Eclipse to meet manufacturing deadlines, the company assumes reliable access to at least two functioning 3-D printers. Both printers are privately owned; one by a member of the team and the other by a personal contact.

Since Eclipse is limited in the amount of stock material available, the team will need to order stock material from outside sources. This introduces several risks involved in the shipment process, including orders being lost, or particularly slow lead times if the order is being shipped from a far location. In order to compensate for these risks, Eclipse will place all orders well before deadlines to compensate for shipping incidents or slow shipping times.

To manufacture parts that Eclipse Technologies cannot 3-D print, the team assumes regular access to the California Academy of Mathematics and Science Machine Shop and Computer Lab, and that both facilities are fully functional during the manufacturing period. As with shipping times, the team plans to anticipate this risk by scheduling manufacturing times ahead of schedule. This will also make up for oversight in the manufacturing process and allow time for correcting errors.

Risk Acceptability Table				
	Severity			
Likelihood	**Trivial**	**Minimal**	**Significant**	**Critical**
Frequent	Moderate	Uncertain	Intolerable	Intolerable
Possible	Acceptable	Moderate	Uncertain	Intolerable
Occasional	Acceptable	Acceptable	Moderate	Uncertain
Improbable	Acceptable	Acceptable	Moderate	Uncertain

Composite Risk Management Worksheet					
Risk	*Consequence*	*Likelihood*	*Severity*	*Acceptability*	*Action*
Robots cannot remain on four legs	Robots may lose ability to move	Improbable	Critical	Uncertain	Minimize probability of robot losing balance, i.e. use of IMU sensor
Use of electromagn ets	Magnets interfere with chipsets	Improbable	Minimal	Acceptable	Ensure that chipsets are a sufficient distance from all chipsets
Insufficient magnetic content in steel ball	Cannot acquire ball and release PackBOTs	Possible	Significant	Uncertain	Invent backup methods that do not rely on magnetism
Nets are not be successfully deployed	Unable to proceed in capture method	Occasional	Significant	Moderate	Improve design of net deployment to eliminate possibility of failed deployment
PreyBOTs fail to walk into nets/trailer	Unable to capture PreyBOTs	Occasional	Significant	Moderate	Use PackBOTs to help herd PreyBOTs into nets/trailer, increase number of nets
String failure in Wolfgang's pulley system	Tail unable to raise, issues in balance	Occasional	Significant	Moderate	Ensure that string can withstand the weight of tail, use adequate material for string
Ordering parts from outside sources	Shipping delays, defective product	Occasional	Minimal	Acceptable	Reserve excess time to anticipate delays or other complications
Access to 3D printers and Machine Shop	Inability to machine and build parts	Possible	Trivial	Acceptable	Reserve excess time to compensate for limited access to printers or machines

SCOPE OF WORK

There will be five Critical Design Review (CDR) periods, where the teams will present their major deliverables. At each CDR, the major deliverables will be assessed and a winning team will be designated. The team that amasses the most points during the CDRS will be awarded the contract and determined to be the overall winner for the 2016 CARPA Challenge. Following are the deliverables and the deadlines for each CDR:

SOFT DELIVERABLES

CDR 1: System Performance Specifications (10/20/2015)

- Proposal Document: Preliminary Design
- Proposal Presentation
- Yearlong Schedule
- Work Breakdown Structure
- Professional Input/Survey
- Bill of Materials/Budget
- Test Schedule and Procedures

CDR 2: Proof of Concept (POC) Presentation and Demonstration (01/19/2016)

- POC Document: Revised Design
- Revised Design Presentation
- Prototypes
- Analysis of Test Results

CDR 3: First Article and Technical Data Package (TDP) (04/25/2016)

- TDP Document

- Final Design and Bill of Materials
- Test Requirements

CDR 4: IDP Trade Expo and Presentation (05/27/2016)

- Functional Prototype
- Engineering Notebook
- Team Portfolio
- Computer Program
- Documentary
- User Manual

CDR 5: Mission Performance (06/07/16)

- Completion of given scenario with fully functioning prototype

HARDWARE DELIVERABLES

CDR 4: IDP Trade Expo and Presentation

- The Lycanthrope Pack: Wolfgang and 2 PackBOTs
- Packaging
- Accessory Components

ENGAGEMENT MANAGEMENT

Throughout the course of Project Lycanthrope, Eclipse Technologies will go through various performance reviews from the CARPA Initiative.

IN PROCESS REVIEWS (IPR)

Informal weekly meetings highlighting each team's progress throughout the week. IPR's plan out the following week's progress and ensure the company keeps on schedule.

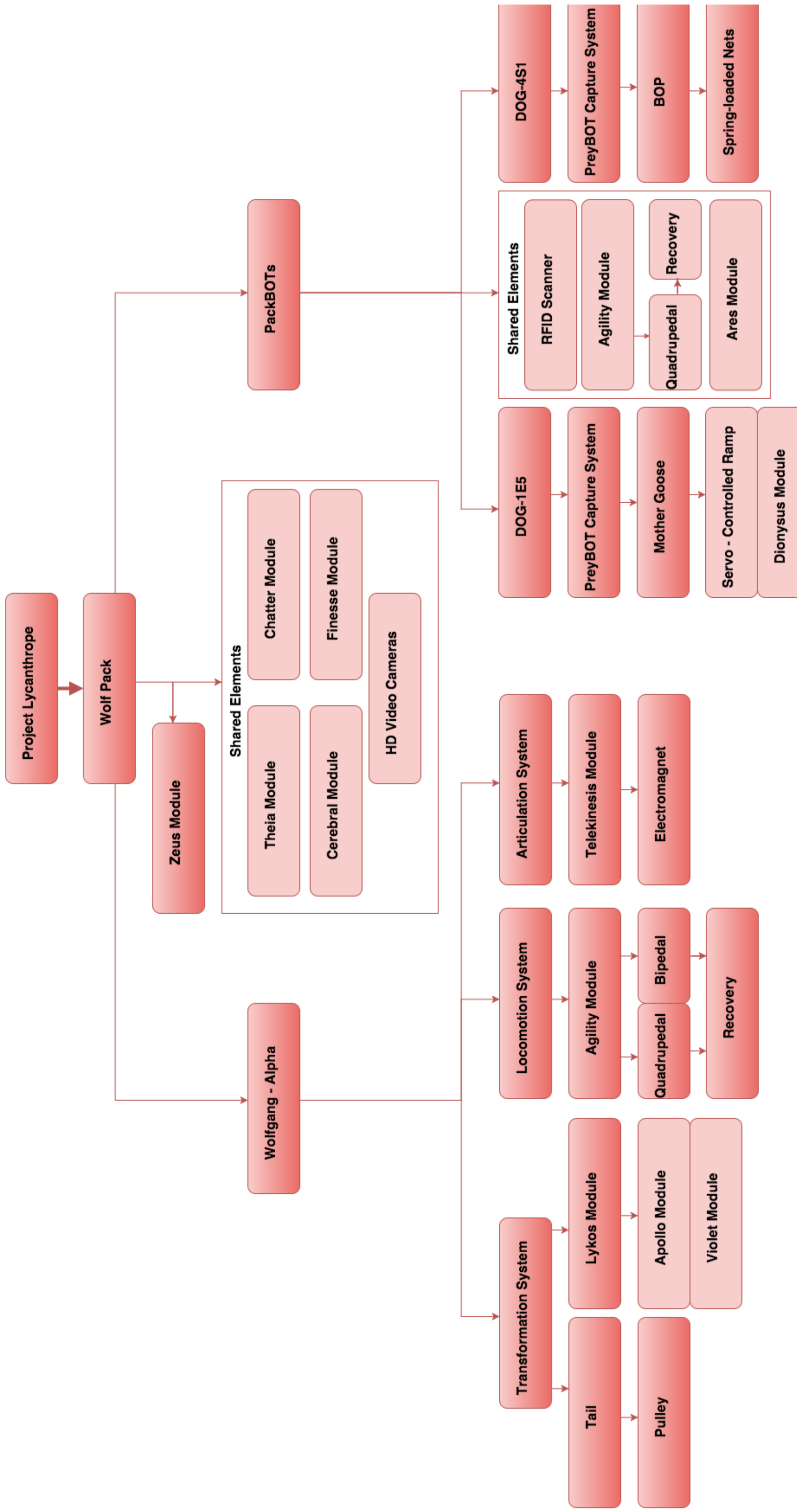
PERIODIC DESIGN REVIEWS (PDR)

Formal monthly meetings that goes in depth with each sub-team. Presents a clearer view of how the company's progress is coming along. The management team will closely review all the work done by the sub-team to make sure it is up to the company's standards.

CRITICAL DESIGN REVIEW (CDR)

The most important of the reviews, CDRs will occur five times over the course of Project Lycanthrope. A panel of auditors appointed by the CARPA Initiative will review the work done by Eclipse Technologies. The five CDRs that will take place are the System Performance Specification (SPS) Proposal Document, Proof of Concept (POC) Document and Presentation, Technical Data Package (TDP) Document, Trade Show Consumer Marketing, and Project Lycanthrope Performance.

WORK BREAKDOWN SCHEDULE (WBS)



YEAR LONG SCHEDULE

Eclipse Technologies' Schedule

Item	Task	% Complete	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June
1	**CDR 1: Systems Performance Specifications**	100%										
	Company Profile											
	Problem Statement											
	Research											
	Brainstorming - Wolfgang											
	Brainstorming - PackBOTs											
	Scope											
	Work Breakdown											
	Schedule											
	Design Matrices											
	Proposed Solution/ Design											
	Risks and Assumptions											
	Bill of Materials											
	Proposed Budget											
	Professional Survey											
	SPS Rough Draft											
	SPS Presentation											
2	**CDR 2: Proof of Concept**	10%										
	Final Computer Aided Designs											
	Prototype- Wolfgang											
	Prototype- PackBOTs											

Item	Task	% Complete	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June
	Prototype- Net											
	Prototype- Ramp											
2	**CDR 2: Proof of Concept**	10%										
	Sub-systems Testing Procedures											
	Test Designs											
	Analyze Results											
	Revisions to Designs (1)											
	Test Revised Robots											
	Analyze and Summarize Results											
	Final Revision											
	POC Rough Draft											
	POC Rough Draft Presentation											
	POC Preparation											
3	**CDR 3: Technical Data Package**	0%										
	Finalized Wolfgang Design											
	Finalized PackBOTs Designs											
	Robot Assembly											
	Finalize Part Drawings											
	Finalize Bill of Materials											
	Test Requirements											
	Rough Draft of Data Package											

	Finalized Technical Data Package											

Item	Task	% Complete	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June
4	**CDR 4: Trade Expo**	0%										
	Wolfgang PackBOTs											
	Engineering Notebooks											
	Performance Scenario/ Strategy											
	Packaging											
	User Manual											
	Software Programs											
	Documentary											
	Team Portfolio											
5	**CDR 5: Mission Performance**	**0%**										
	Scenario Completion											
	Sensor Expo											

PROPOSED DESIGN SOLUTION

DESIGN OVERVIEW

The first of the three robots, Wolfgang, must be a dual-mode, polymorphic robot. It will manipulate a steel ball in order to open the field and allow the other two members of the team, the PackBOTs, to search and either return or destroy the prey.

To complete the mission, Eclipse Technologies will develop one Wolfgang, codenamed Alpha, and two PackBOTs, code-named DOGs when referred to together, and DOG-1E5 and DOG-4S1 respectively. Each robot has a specific role in ensuring the success of the mission. Alpha must be able to walk on both four and two legs. It will utilize a locomotion system where each limb will have three degrees of freedom, allowing Alpha to maneuver the field efficiently. To aid the transformation process, Alpha will have a heavy, segmented tail, which can curl up and down to shift the balance of the robot. Alpha will be equipped with electromagnets in its forelimbs to manipulate the steel ball.

The DOGs must be able to identify and capture the prey. They will utilize a similar locomotion system as Alpha, with three degrees of freedom (DOF) per limb. They will be equipped with RFID scanners to identify prey, along with two cameras for vision processing and an IMU to maintain balance. The DOGs will use three tools to capture the PreyBOTs: a trailer, spring loaded nets, and jaws. The trailer is a mobile trap which will sit until a prey wanders inside. After a PreyBOT is inside, the trailer will close, allowing a DOG to scan the RFID tag. They will utilize a publish and subscribe system between each robot and mission control to receive tasks and deliver status reports. Mission Control will direct the robot to perform specified tasks aided by vision processing. If it's the correct prey, then the DOG will drag the trailer back to the holding cell. It the wrong prey then it will be released. The spring-loaded nets are trapping mechanisms that sit on the field until triggered. When triggered, the net will then immobilize the prey, allowing it to be scanned and returned if necessary. The jaws on the DOGs are methods of transporting immobilized PreyBOTs.

SHARED ASCPECTS AND DESIGN PRINCIPLES

ASYNCHRONOUS DESIGN

Eclipse Technologies takes an innovative approach to solve the problem of robotic multitasking. The nature of the challenge requires the robots to execute tasks quickly and, in some cases, simultaneously. Most robotic frameworks operate synchronously; each individual operation is completed before the next operation begins. The diagram below models one such synchronous operation.

A = Capture Video
B = Capture Audio
C = Process Video
D = Process Audio

```
|-------A-------|-----B-----|--------------C--------------|-----D-----|
```

A problem is quickly evident from the synchronous approach: if one operation takes too long, it blocks the execution of subsequent operations. Eclipse Technologies exploits Python 3.4's new *asyncio* module, which allows functions to operate asynchronously while maintaining the same execution time for each individual operation.

A = Capture Video
B = Capture Audio
C = Process Video
D = Process Audio

```
|-------A-------|
                |--------------C--------------|
|-----B-----|
            |-----D-----|
```

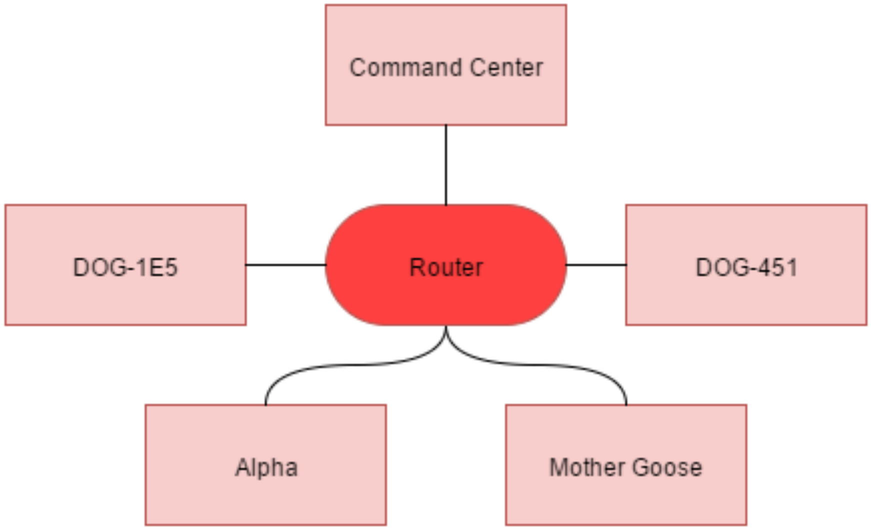
It is evident that asynchronous execution of operations increases the efficiency of the processor and reduces the total execution time of all tasks. In addition, the delay of a non-dependent operation will not block the execution of any other operations. All of the communication and processing modules obey this design principle. Asynchronous design allows the robots to perform complex, time-consuming operations, such as vision processing, without degrading other crucial operations, such as balancing and reaping sensor data.

COMMUNICATION

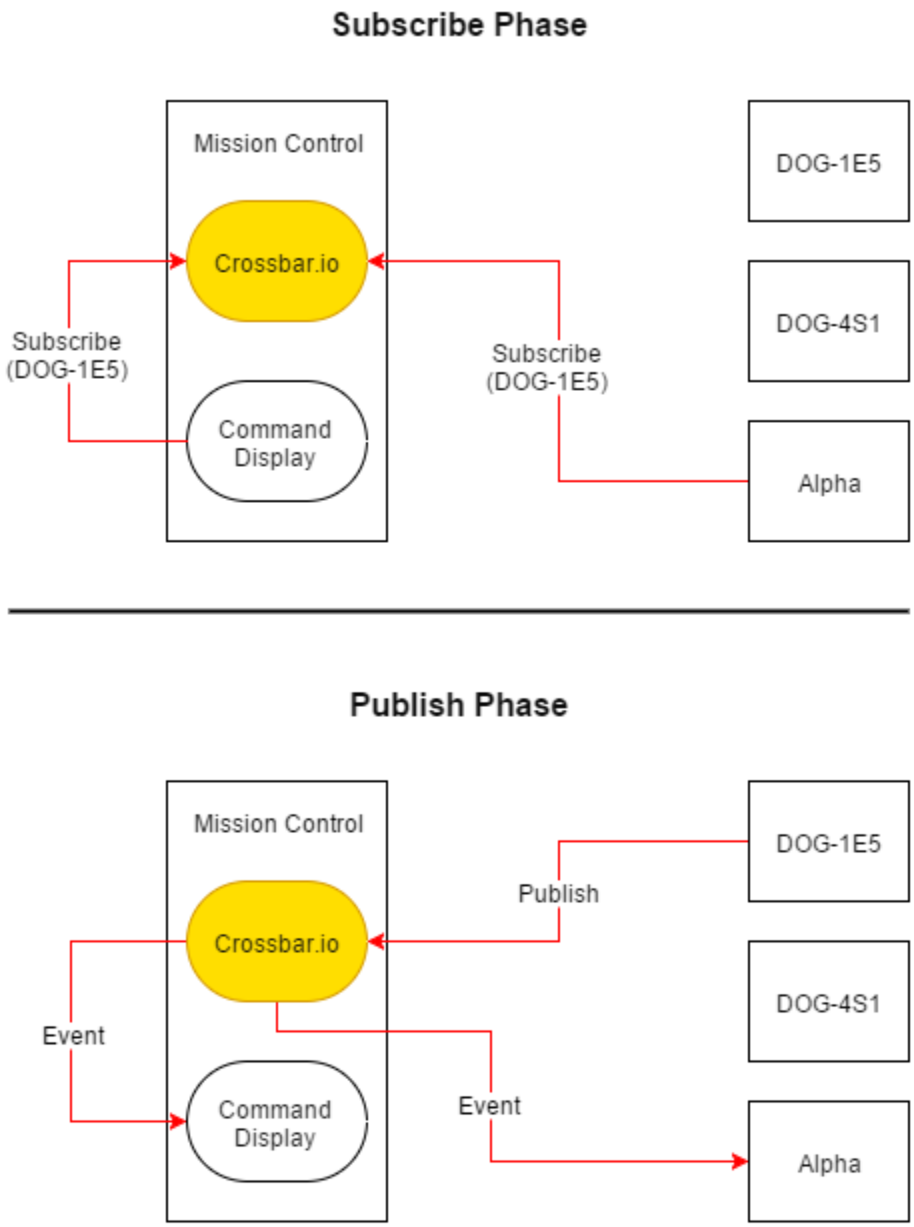
PROGRAMMING - CHATTER MODULE

All robots and the command center communicate using Wireless Local Area Network (Wi-Fi). Wi-Fi has several advantages over specialized wireless communication chips like XBee. Data transfer rate is crucial in a computation and data intensive mission such as the Lycanthrope Project. Standard Xbee ZigBee modules have a max data rate of 250 Kbps, which is insufficient to stream video, share computation tasks, or communicate large amounts of data. Wi-Fi is a well-tested and mature solution for wireless communication and interfacing. With a data rate of around 10-20 Mbps, a Wi-Fi based solution greatly improves communication flexibility and allows more interaction between devices.

Eclipse Technologies takes an innovative approach to solve the complexity of robot-to-robot and robot-to-server communications. All devices will interact primarily using Web Application Messaging Protocol (WAMP). WAMP is a real-time communication protocol built upon WebSocket, a set of specifications for versatile communication over an established Transmission Control Protocol (TCP) connection. The gaming community already uses WebSockets heavily to handle real-time multiplayer games. Eclipse Technologies will tailor this technology for use in robotics. At the center of all communication is a Wi-Fi router with support for 802.11n (the wireless network protocol with the best range). All devices, including the command center, will connect to the router to form a simple but well-protected intranet.



The command center acts as a server that processes, displays, and relays information. A WAMP server powered by Crossbar.io running on Python will stand by to accept connections as soon as the mission starts. All robots and controlled devices will asynchronously connect to the server and identify themselves. After the identification phase, each device will enter a subscription phase where it subscribes to information that it wants to receive. At any point in the mission, a device can publish data and the data subscriber(s) will receive the pertinent data. This flexible Publish & Subscribe pattern improves communication between robots.



A second option that is sometimes used is the Remote Procedure Call pattern. For example, a robot may need to invoke the movement commands of another robot. At the beginning of the mission, each robot registers available functions for other robots to call. At any time during the mission, robots are allowed to invoke the shared and registered functions remotely and asynchronously.

The WAMP protocol and Crossbar.io supports the sending and receiving of binary data. This allows robots with video or audio feeds to send data to the command center. Due to the diverse community and abundant plugins, WAMP allows for polyglot applications. For example, the mission control display powered by HTML5 and JavaScript can communicate with robots powered by Python and C++.

MICROCONTROLLERS

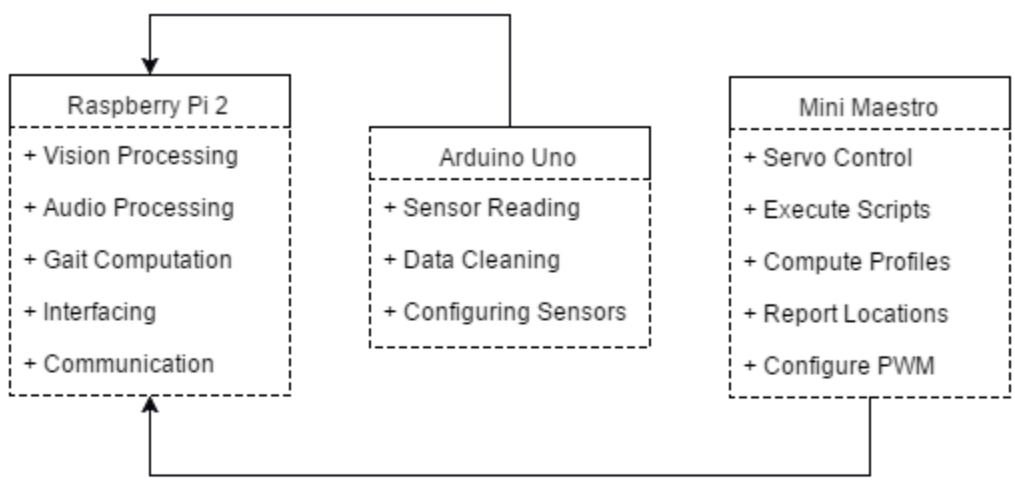
PROGRAMMING - CEREBRAL MODULE

The Lycanthrope Project requires high-speed processing and accurate hardware interfacing. Thus, all Eclipse Technologies robots will be equipped with a Raspberry Pi 2 Model B, an Arduino Uno, and a Pololu Mini Maestro 18-Channel Servo Controller. Each device plays a unique role in ensuring that the robot can perform desired tasks with accuracy and dynamically respond to changes in the mission field.

The Raspberry Pi 2 is equipped with a 900MHz quad core ARM-A7 processor. This processor architecture allows a variety of libraries and software, such as OpenCV and Python 3, to be used. The Pi serves as the communication center for the robot. It links up additional controllers and interfaces with the command center. A majority of CPU-intensive processing is done on the Raspberry Pi as the Arduino and Mini Maestro lack a powerful processor. Vision, audio, gait, communication, and interfacing modules are all available on the Pi.

While the Raspberry Pi has 40 GPIO pins, most of them are not hardware-based, so they are prone to inaccuracies and fidgeting, but even a 500 microsecond difference in PWM is detrimental to servo performance. Thus, all robots are also equipped with the Mini Maestro Servo Controller. The controller has a dedicated processor capable of setting velocity and acceleration for each servo. In addition, the Servo Controller has the capacity to execute premade scripts with extreme precision. Communication with the Raspberry Pi is made through USB. The Raspberry Pi sends servo positions, velocities, and acceleration, and the Servo Controller reports back its status.

The Arduino Uno is used mostly for reaping sensor data. Many sensors have pre-made libraries for Arduino, and there are some designed specifically for Arduino. The 16MHz ATmega328P processor is incapable of performing complex, iterative computations. Instead, it will assist with cleaning and preprocessing sensor data before sending it to the Raspberry Pi via USB. The Arduino will also send configuration data to the sensor, if necessary, before or during the mission to compensate for unexpected changes.



The Eclipse Technologies infrastructure will be programmed mostly using Python 3. The complexity and magnitude requires a programming language that minimizes development time and maximizes robot performance. Python 3 was chosen due to its ability to interface with C/C++ as well as its rapid development time. Most of the processing done on the Raspberry Pi is written in Python. However, high intensity tasks such as vision processing and inverse kinematics rely on C++ functions called from Python.

In addition to the main language, C, C++, JavaScript, HTML5, CSS, and the Maestro Scripting Language will be used to enhance the functionality of the robots and the command center. Arduino Uno will run on C. The command center will use a combination of JavaScript, HTML5, and CSS for its front-end processing and Python and C++ for its back-end processing. Finally, the Servo Controller will use the Maestro Scripting Language to execute precompiled routines for repetitive and predictable actions.

INVERSE KINEMATICS

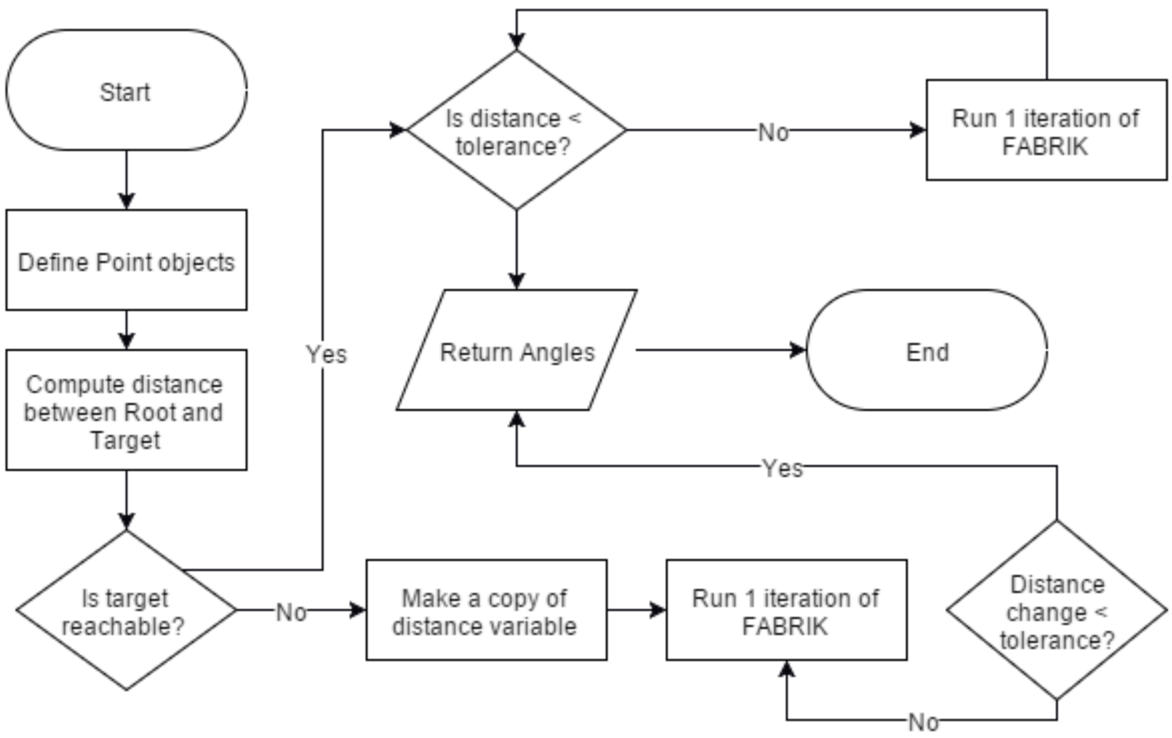
PROGRAMMING - FINESSE MODULE

Legged animals have multiple joints per leg that can bend and rotate. Eclipse has modeled the robots after animals, with one servo per joint to create a simplified 1 DOF joint mechanism. Servos are controlled through relative angles where each servo's position and angle depends on the angle and position of the previous servo(s) (excluding the first servo). Given the angles for each servo, it is possible to compute the location of the end effector.

However, the movement of a robot requires the exact opposite. Given a point in space, a configuration must find the correct angles for each joint so that the end effector reaches the target. This is not an easy task. Conventional methods such as the Jacobian Transpose, Jacobian Pseudo Inverse, Cyclic Coordinate Descent, and Triangulation are either too slow or produce impossible angle configurations that would not benefit movement.

FORWARD AND BACKWARD REACHING INVERSE KINEMATICS (FABRIK)

Eclipse solves the difficult task of real-time Inverse Kinematics (IK) by implementing a modified version of the FABRIK algorithm by Andreas Aristidou. In testing environments, FABRIK finds a solution over 1,000 faster than the classical Jacobian Transpose even under constrained conditions. Eclipse has created an improved version of FABRIK for real-time movement and advanced constraining of 1 DOF chained joints with a single end effector.

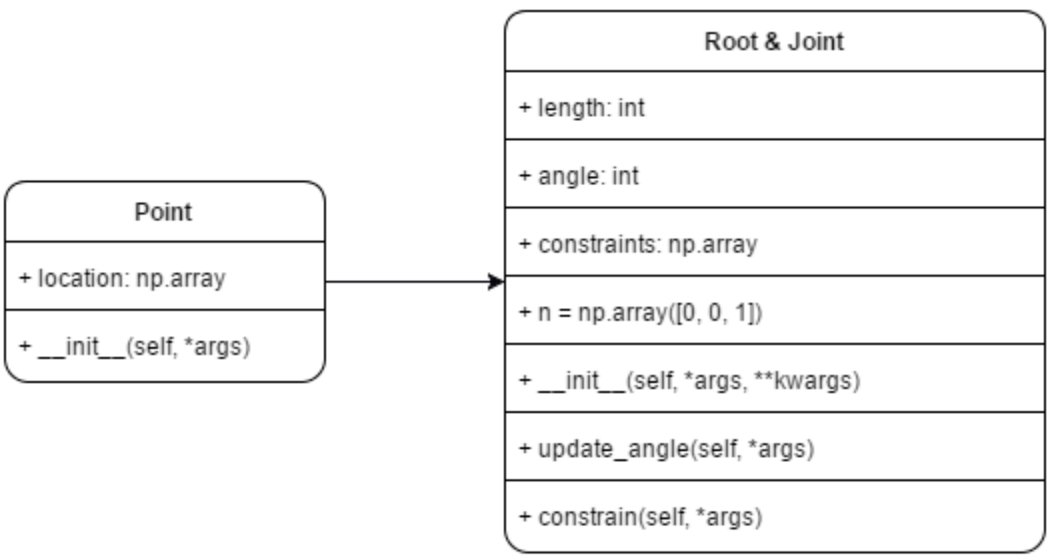


A single iteration of FABRIK is simple and effective. It can be broken down into thirteen main steps.

1. Give joint locations as points (x, y, z) in 3D space. The first point is p_0 and the last point is p_{n-1} where n is the number of joints.
2. Let t be a point (x, y, z) such that t is the target for the end effector.
3. Let p_n be the end effector.
4. Let $d_i = |p_{i+1} - p_i|$ for $i = 0, 1, \ldots, n - 1$.
5. Move p_n to t. p_n is now p'_n.
6. Construct a line l between p'_n and p_{n-1}.
7. Move p_{n-1} to p'_{n-1} such that p'_{n-1} is d_{n-1} from p'_n and lies on line l.
8. Repeat until p_0 is move to p'_0.

9. Move p'_0 to p_0. Now p'_0 is p''_0.
10. Construct a line l between p''_0 and p'_1.
11. Move p'_1 to p''_1 such that p''_1 is d_0 from p''_0 and lies on line l.
12. Repeat until p'_n is p''_n.
13. Check if $|p''_n - t| < tolerance$. If so, return all points. Else, repeat $5 - 12$ until such condition is satisfied.

Understanding that each joint is unique, Eclipse has developed an efficient constraining algorithm that works for any theoretical 1 DOF per joint configurations. First, all points (root, joints, and end effectors) are objects (which extend the *Point* class) with a *constrain* function that accepts the previous and next points as parameters.



Each iteration of FABRIK does not set an object's point unless that object is the end effector or root. Otherwise, it will call *update angle* which will then internally call *constrain* to ensure that the newly placed joint position does not violate an angle constraint. However, the vanilla version of FABRIK does not address this problem in an efficient manner. Often times, the original constraining algorithm will fail to converge even though a solution exists. Eclipse solves this problem through the use of analytical and numerical constraining which shows little damage to the performance.

In 3D space, it is possible to compute the angle formed by three points using vectors and the dot product. The constraint process is effective but non-trivial. A joint's *constrain* function accepts p_{i-1} and p_{i+1} as parameters. The basic procedure is as follows:

1. Define vectors $\vec{a}$ and $\vec{b}$ such that:

$$\vec{a} = p_i - p_{i-1}$$

$$\vec{b} = p_i - p_{i+1}$$

2. Compute Θ.

$$\Theta = \arccos\left(\frac{\vec{a} \cdot \vec{b}}{|a||b|}\right)$$

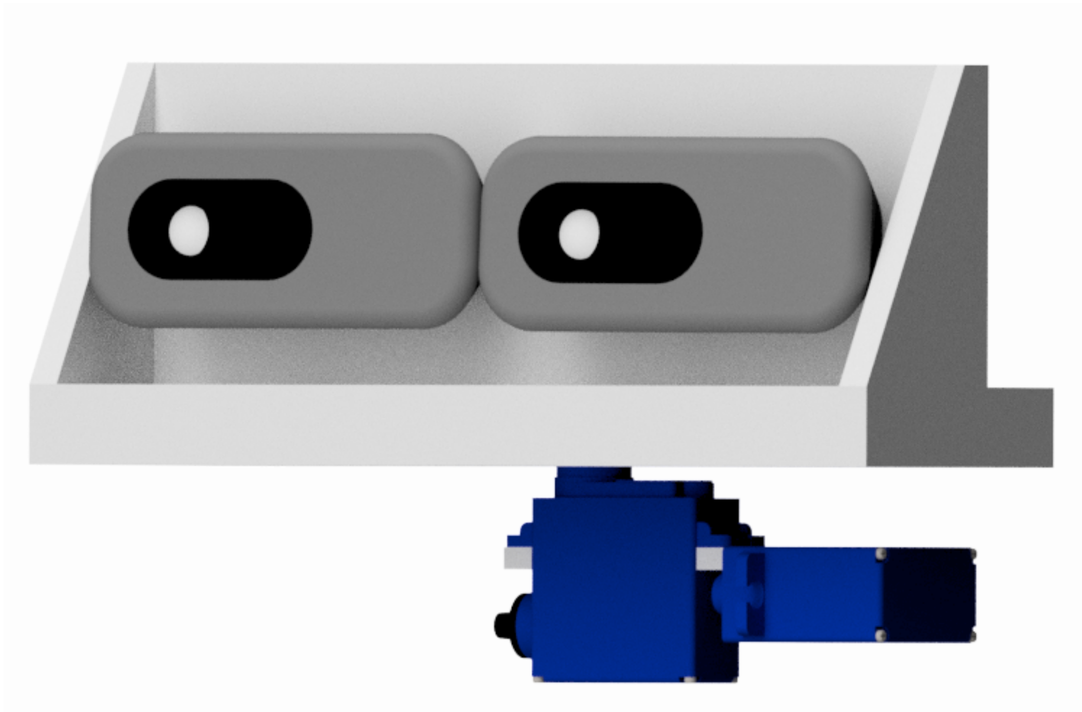
3. Determine the sign of the angle by combining a dot product and a cross product. Note that $\vec{n}$ is the plane normal to $\vec{a}$ and $\vec{b}$. If below is true, Θ is negative.

$$\vec{n} \cdot (\vec{a} \times \vec{b}) > 0$$

4. If Θ is within the bounds defined by the joint, then do nothing and exit.
5. If Θ is not within the bounds defined by the joint, move p_i such that Θ would be within bounds. This is done by computing a rotational matrix and rotating p_i about the appropriate point.

VISION

MECHANICAL



Alpha's vision will be through two HD webcams. Two webcams instead of one gives the ability of depth perception, a skill that will become useful for tasks such as picking up the steel ball. Alpha's vision will have two degrees of freedom, utilizing two micro servos. The purpose of one of the micro servos is so the webcams can rotate on the z-axis, expanding its viewing range. The purpose of the second micro servo is so the webcams can rotate on the x-axis. It allows the cameras rotate so when Alpha is walking on two legs it can see straight ahead instead of up. The estimated maximum torque needed from these servos is shown below:

Servo for Z axis rotation

=0.076454m * (1.36kg*9.8m/sec2) *cos(0) = 1.02N*m

Servo for X axis rotation

= 0.1118362m * (1.38kg * 9.8m/sec2) * cos(90) =1.51N*m

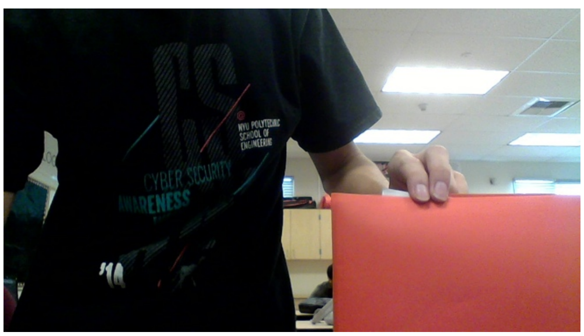
PROGRAMMING - THEIA MODULE

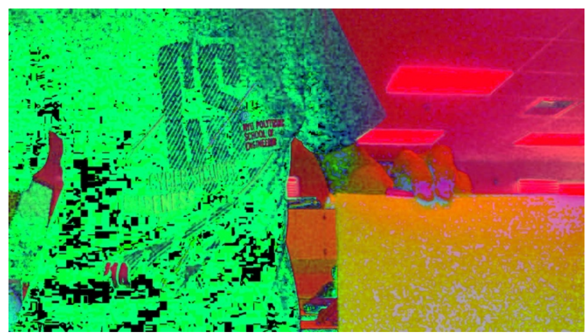
Eclipse takes the versatility of sight and uses it to its full capacity. All robots are equipped with two HD cameras to assist in locomotion, target tracking, and field reconnaissance. Many of the robots' autonomous features rely heavily upon the Theia module. Like other modules, Theia is asynchronous. However, because vision processing is a highly CPU consuming task, Theia is given its own thread for execution and freedom to utilize other processors in order to complete processing.

The primary purpose of vision is to track distinctly colored objects in the mission area. To accomplish this, Theia takes advantage of *OpenCV*, a mature and featureful computer vision library with Python bindings. *OpenCV* contains a variety of highly optimized image processing functions which allow the PackBOTs to easily track color. Performance wise, *OpenCV* is extremely fast. On a Raspberry Pi 2, it is capable of processing 720p HD video at 30 FPS. However, vision processing does not need to be at such a high frame rate. 10 FPS is more than sufficient to track moving targets effectively while conserving computation power.

Theia does not rely on the standard RGB color space. RGB is too sensitive to lighting changes - any minute differences on the mission field will immediately cause detection to fail. Instead, Theia utilizes the HSV color space. HSV separates the luma (lighting/intensity) from the chroma (color). A given chroma in HSV can still be detected at different lumas with high accuracy.

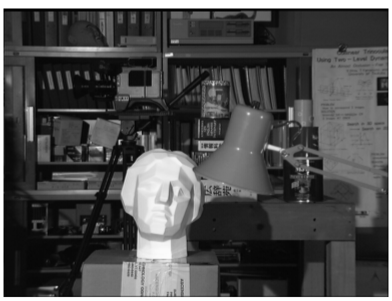
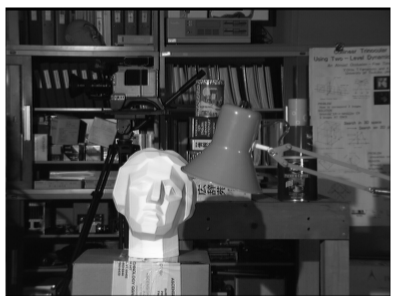
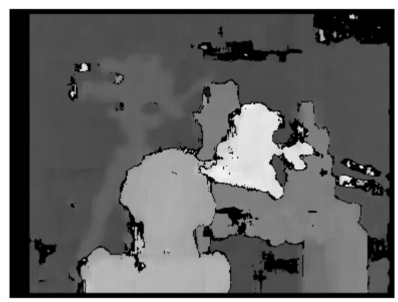
The vision module goes through a phase-by-phase detection, analysis, and reporting cycle. It directly interacts with the movement module to provide the robot with fluid motion. An iteration of HSV is fairly straightforward. First, an image is captured and converted from RGB to HSV. A gaussian blur is applied to reduce noise. Then, an HSV filter is applied to extract the relevant entity(ies) in the image. Finally, the image is eroded and dilated to obtain an accurate mask for the object. Theia will then compute the area of the white areas, ensuring that they are large enough to be the actual robot. It relays the information to other modules for decision making.





In most cases, the vision module will take the final masked image, locate its center of mass using *OpenCV*, and return the coordinates for further processing. Theia also satisfies the video feed requirements. Upon capturing an image, the raw binary data for the image will be encoded in Base64 and published to the receiver (the command center). The asynchronous nature of the communication system as well as the vision module allows smooth video feeds of one or more robots.

However, detecting the x and y coordinates of an object is not very helpful as the robots must also have a way to measure the z coordinates of objects. To do this, Eclipse Technologies applies the same principle of human and animal vision: disparity mapping. The two cameras on the robot are mounted a certain distance from each other and calibrated before the mission. Theia takes one image from each and computes a depth map. The left image shows the capture from the left eye and the right image shows the capture from a right eye (simulated). The third image is the disparity where lighter colors indicate which object is closer.

The support for 3D vision makes Theia a crucial component of the robots as it allows for fully autonomous navigation and manipulation of arbitrary objects on the mission field. More specific uses for the Theia Module are discussed in later sections.

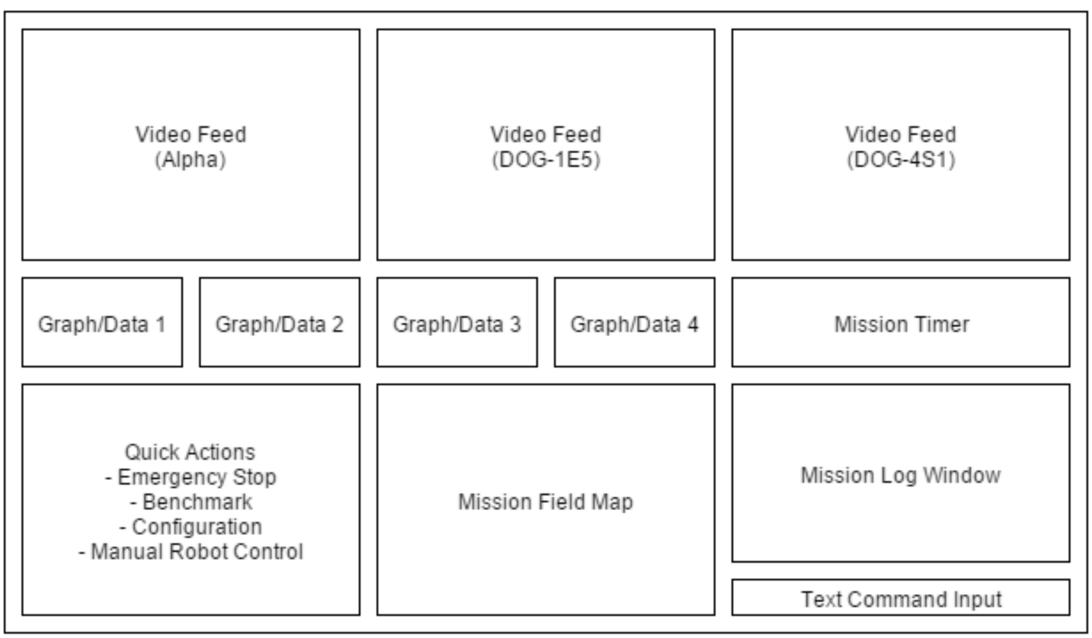
MISSION CONTROL

COMMAND CENTER & UI

PROGRAMMING - ZEUS MODULE

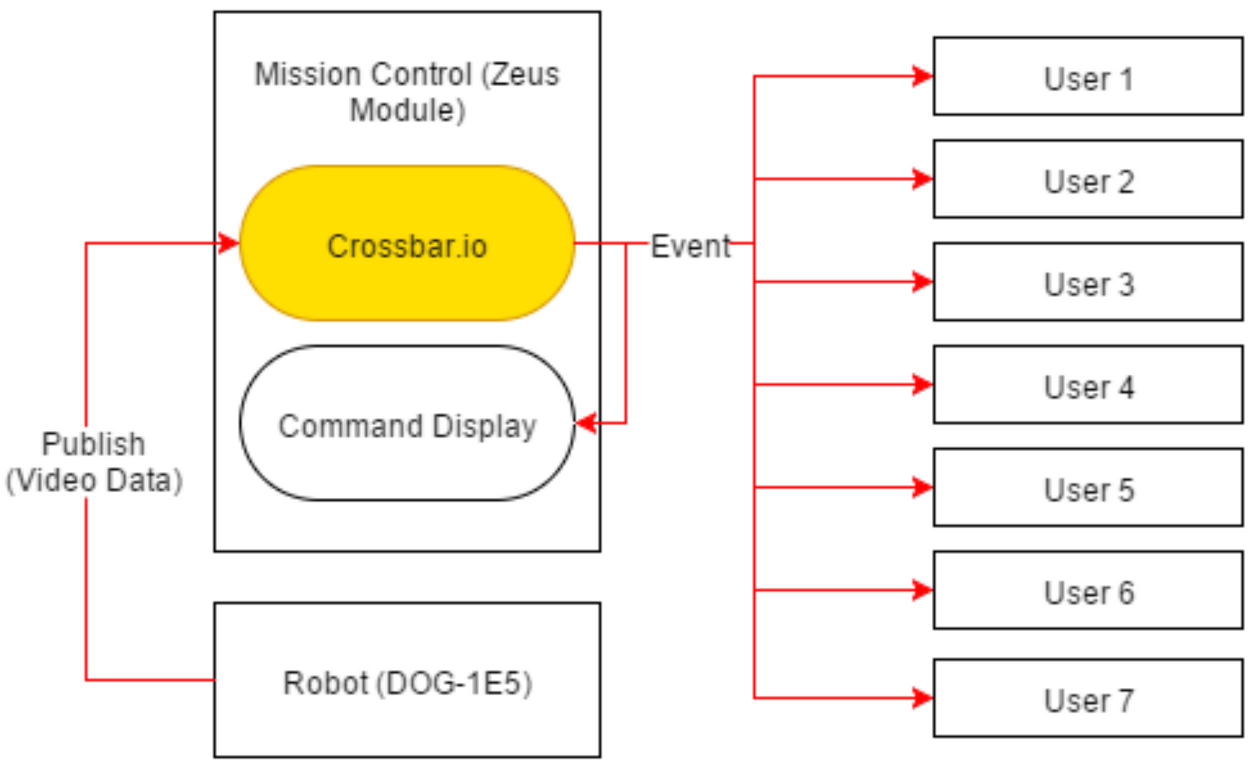
The command center, also known as the Zeus Module, is the center of all communication and data. All robots connect to the command center and all information passes through the command center before it is sent to its recipients. Zeus runs on a laptop computer connected to the main router. On the computer, two main applications exist: the WAMP Server and Command Display.

The Command Display is a Graphical User Interface (GUI) application designed to display information sent from the robots and provide non-autonomous commands to the robots. It will be designed using HTML5, CSS, and JavaScript. The main advantage of designing the Command Display in web-based languages is that it reduces development and testing time and improves aesthetics. In addition, WAMP already has a library made for JavaScript that can communicate with Python-powered robots. A proposed interface layout is shown below.

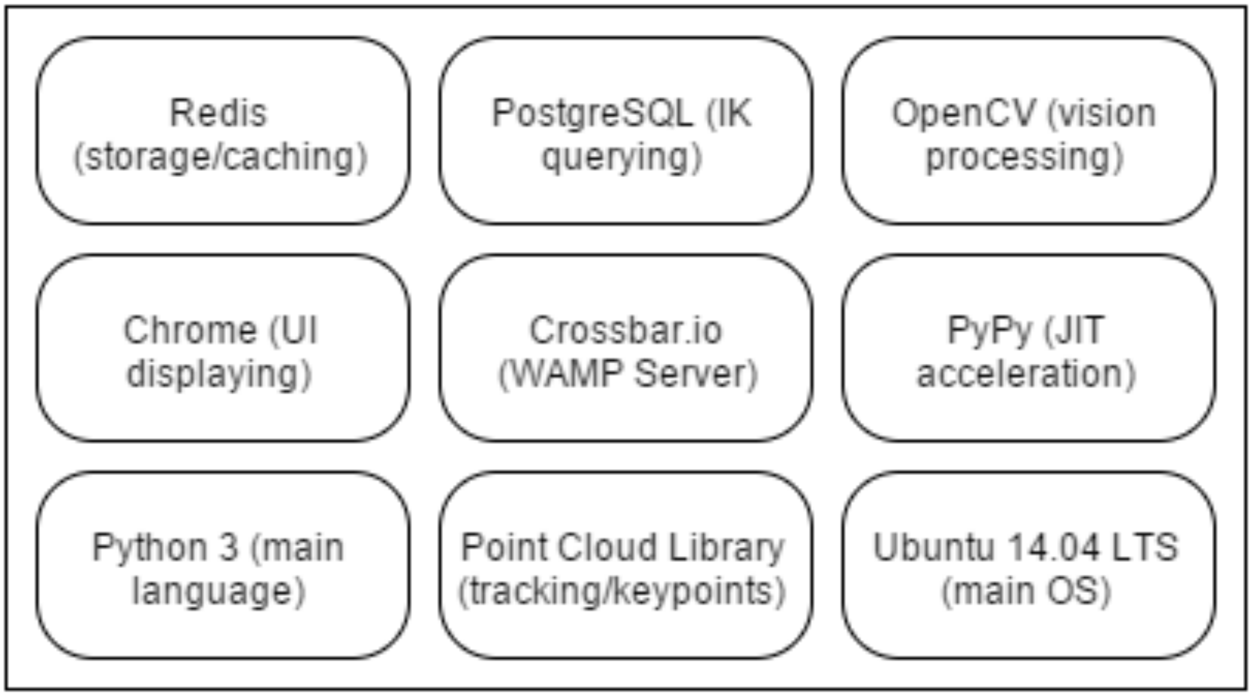


To comply with mission specifications, the Command Display allows for users to confirm or overwrite certain actions. A location for text command input will allow the controller to send specific commands to any robot or to configure the interface. A manual robot control area will serve as a backup system in case autonomous systems fail.

Eclipse Technologies has also developed a method for guest viewers to access mission data from anywhere in the world. The WAMP Server on the command center also serves as a relay station for additional users connected to the server. This allows any other participants with access to the internet to connect to the command center and view exposed data. Due to the rapid send/receive speed of Crossbar.io, Zeus can send mission data nearly instantaneously to hundreds of additional users.



In addition, Zeus is equipped with additional programs to assist in processing and storage (described later). While important, the robots can complete the mission without using all of these high level features. However, some are crucial to mission functionality and others improve mission speed and server performance.

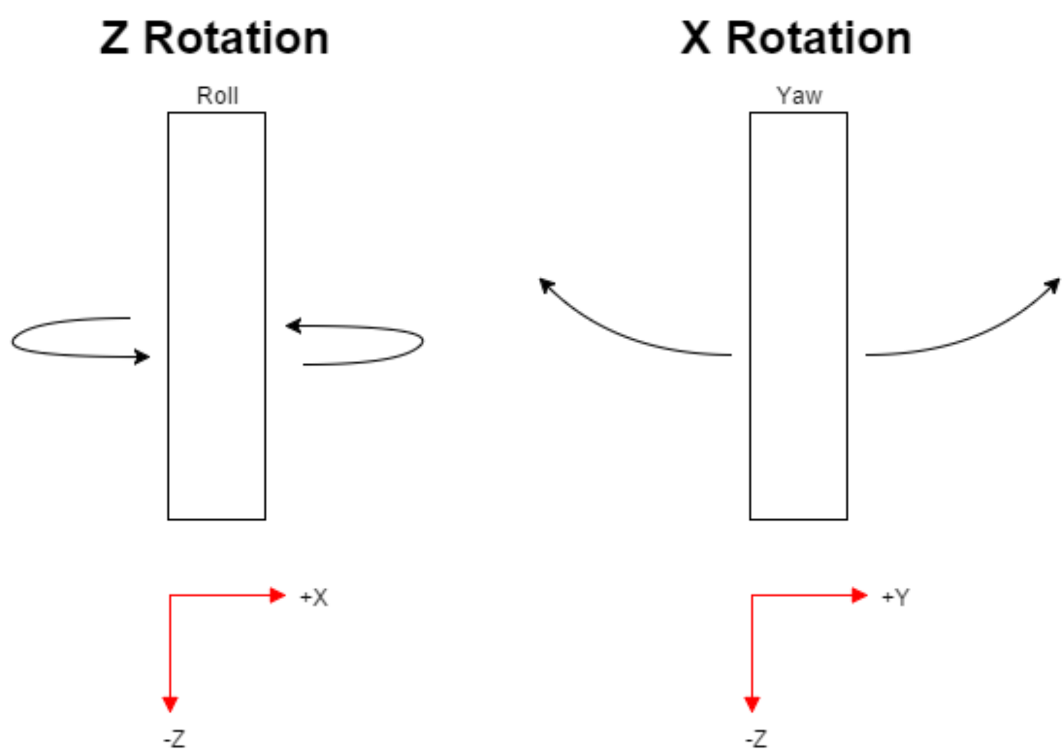


WOLFGANG

LOCOMOTION

MECHANICAL

Alpha will be walking quadrupedally (on four legs), then transform to walk bipedally (on two legs). The hind legs will be heavier than the forelegs to maintain balance when Alpha is in bipedal mode. Each leg will have a total of three servos in order to create three degrees of freedom in each leg. The hind legs will have degrees of freedom on both the x and y axis as well as on the z rotational axis. This means these legs will be able to rotate on the x axis which will become useful for turning when Alpha is in bipedal mode. The forelegs will also have three degrees of freedom, however, instead of the z rotational axis, it will be on the x rotational axis. This means that these legs will be able to move sideways, which will make turning less difficult when Alpha is in quadrupedal mode.

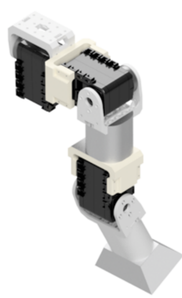


The servos on the shoulder joint would need at most a torque of 2.45 N*m in order to pick up the entire arm. The servos on the hip joint would need a torque of at most 7.84 N*m in order to pick up the entire leg. Torque calculations are performed using the following equation:

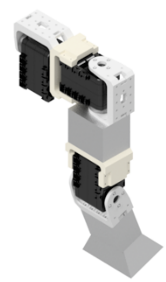
$$\text{Torque} = \text{radius} * \text{Force} * \cos(\theta)$$

where θ is the angle between the lever and the point of rotation

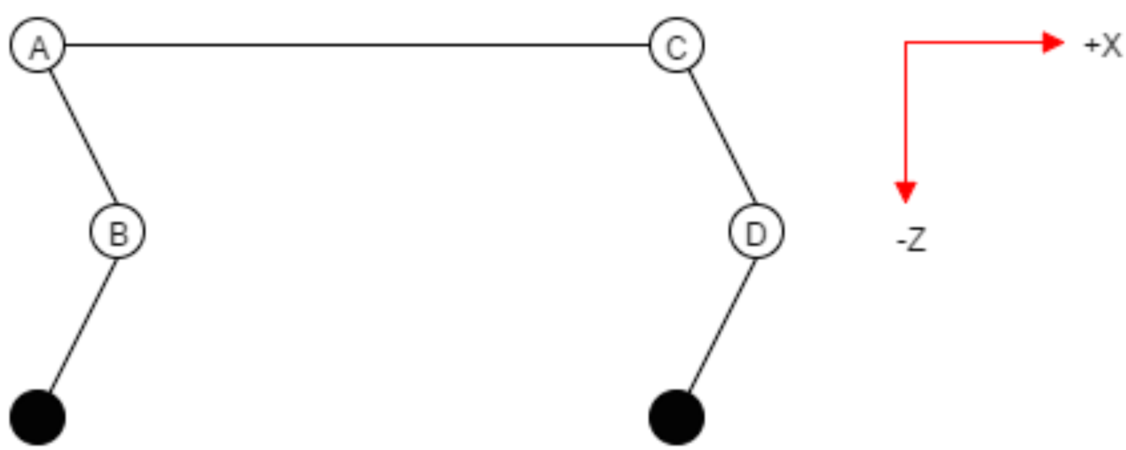
Not only will the hind legs be thicker than the forelegs, but the back feet will also be bigger than the front feet. This means that the surface area of the back feet will be greater than that of the front feet. When Alpha is in bipedal mode, the large surface area of the hind feet will help keep Alpha balanced, as well as increase traction between the robot and the ground.

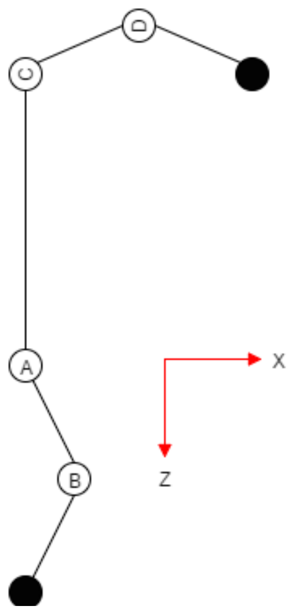


Front Leg

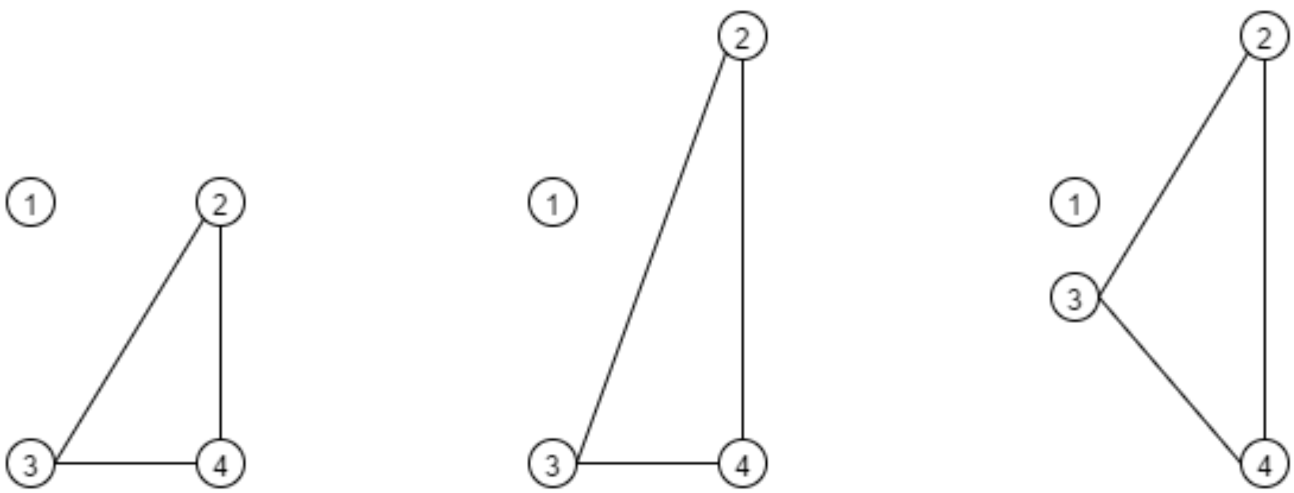


Back Leg





The legs will be in a neutral layout with their knees facing forward on the positive x-axis. Smaller front legs allow for an ease of mobility in the front legs, making manipulation of the steel ball easier. Joint C would need to rotate from -90 to 90 degrees relative to the z-axis while joint A would need to rotate from -135 to 135 degrees relative to the z-axis with the added degrees being needed for the bipedal transformation. B and D will rotate at -170 to 10 degrees and act as knees while the robot walks.



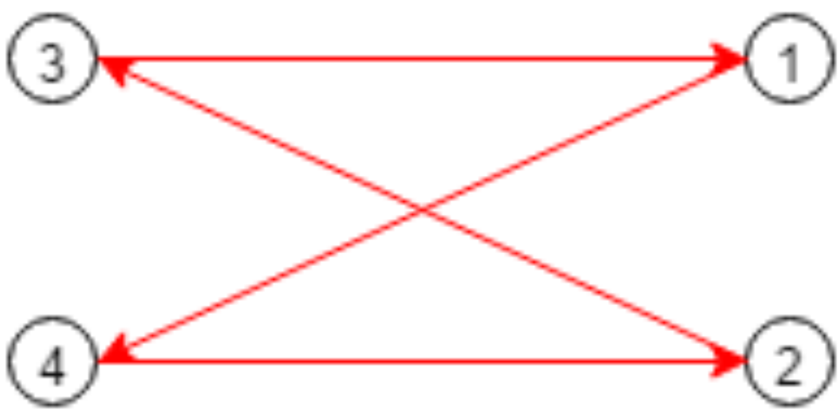
In order to avoid falling while in quadrupedal mode, the center of mass must remain in the triangle depicted in the figure above. The three vertices of the triangle are the legs on the ground as the robot is walking. The far left triangle represents the robot standing upright and it has the smallest area. The center of mass will be calculated to be on the hypotenuse of the left triangle. This would allow the center of mass to always be in the triangles of any leg configuration.

PROGRAMMING - AGILITY MODULE

Movement is perhaps the most programmatically complex component of Alpha. However, since Alpha will most likely not encounter unpredictable events, most of its movements will come from pre-generated step and gait sequences. The robot will still call upon sensor data for balancing and will be able to readjust its own gait/step to accommodate changes in its environment. It simply does not rely heavily upon real time inverse kinematics as it is not necessary.

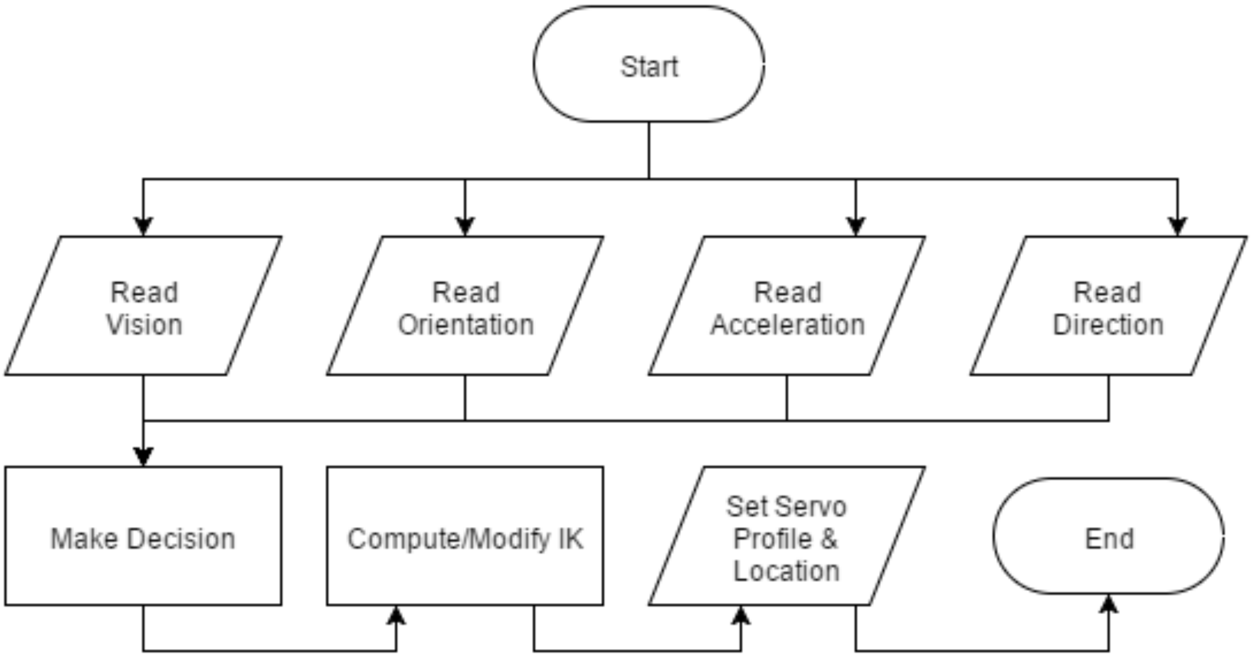
FOUR LEGGED GAIT

While on four legs, Alpha will rely on the crawl gait, which is the most stable gait for moving in the forward direction. The gait itself moves opposite legs and alternates between front and back to maximize stability. For example, the front right leg will move forward, followed by the left back leg, the left front leg, and finally the right back leg.



Inverse kinematics solutions are pre-made for the robot. Most of the time, it simply needs to execute a sequence of commands stored in the database on Cerebral. During pauses or disturbances, Agility will attempt to find the closest point in the command sequence and set the servos to that sequence before proceeding.

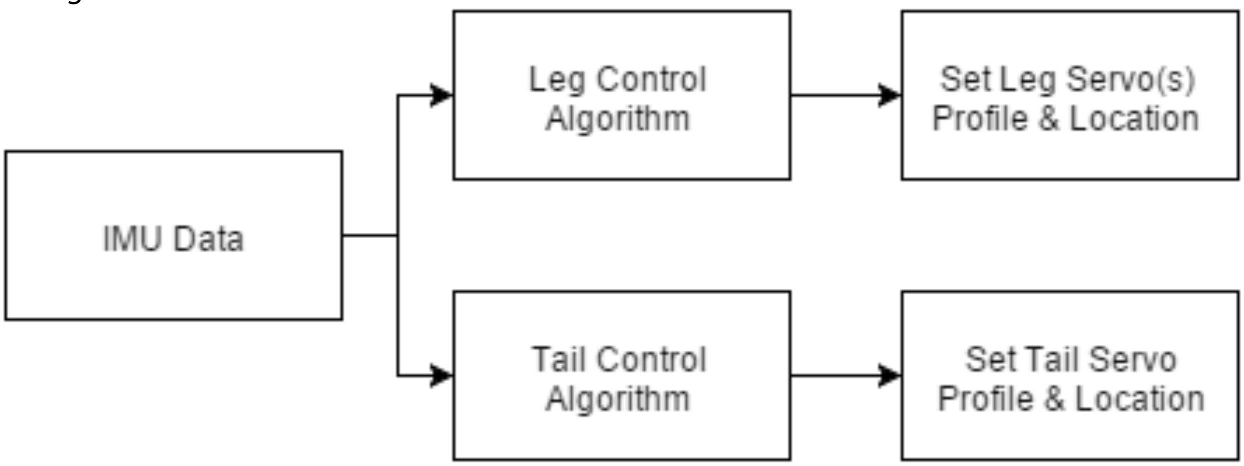
However, Eclipse Technologies understands that unexpected circumstances can occur. Thus, the Agility module is also given data from a 9-axis IMU (MPU-9150) capable of measuring direction, orientation, and acceleration. IMU data is combined with vision data for the Theia module to make decisions regarding movement.



It is also important that Wolfgang has the ability to turn on four legs. Wolfgang will turn through the execution of a predefined lifting and placing combination of legs 1 and 2 on the y axis. This ability to turn in place gives Wolfgang a significant advantage as it can now easily maneuver to the morphing chamber or even assist in the mission field as a backup.

TWO LEGGED GAIT

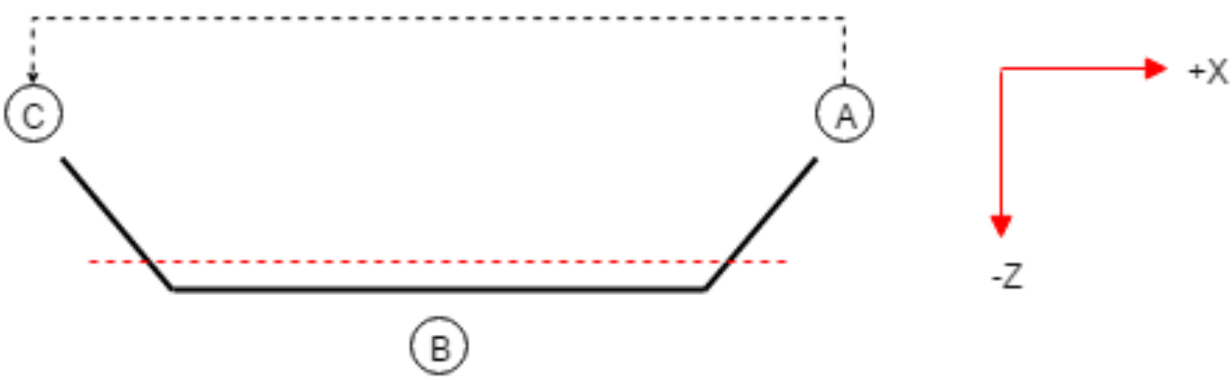
After transformation, Alpha will walk on its two hind legs. Balancing on two legs is significantly more difficult than balancing on four legs, so the Agility module will also control Alpha's tail when it is in a two-legged state to improve stability and balance. Tail control is an asynchronous task. When walking, Wolfgang must be able to ensure that it does not fall forwards, backwards, or to the side. IMU data is shared between two separate tasks to position the leg and tail.



To turn on two legs, Wolfgang uses a predefined set of servo profiles and locations. However, it will return the joints to a root position before turning because rotation of a back leg requires extreme precision. Throughout the turning process, the Agility module will prepare to set down the other leg if it detects changes in the IMU that indicate a possible fall.

FORWARD PROPULSION

Alpha relies on a simple linear method of forward propulsion. For simplicity, all movement paths are viewed from the reference frame of the robot and not the outside. Thus, the velocity of the robot does not affect the joint path of the robot.



The diagram shows one sharp power stroke of one leg. The effector of the leg will move from point A to point C. Upon reaching point C, the end effector will be moved to A again for another power stroke. The dotted red line symbolizes the ground plane formed by the three legs that are not in the air. A general power stroke can be described in 3 steps.

1. Descent (A) - Before contacting the ground, the end effector will be in the air and approach at a steady angle. The angle will be determined by Agility depending on the downward force that the joint must apply in order to lift the robot to a desired height.
2. Drag (B) - After contacting the ground, the end effector will attempt to move below the ground plane (from the robot's perspective). This additional downward distance translates to the robot moving upwards since the ground plane does not move. The upward motion of the robot combined with the rotational motion of the servos will create the desired forward movement. This occurs over a long period of time usually with three legs.
3. Return (C) - When the main stroke is completed, the end effector will lift up. The angle of lift off is not crucial in this case. Agility will then set the new target position back to point A in order to prepare for another power stroke.

Upon generation of a path through linear functions, Agility will call the Finesse Module to generate an inverse kinematics solution for execution. Around 30 different joint positions will be created along with their respective velocity and acceleration profiles. The data is compiled into assembly script for parsing by the servo controller.

RECOVERY

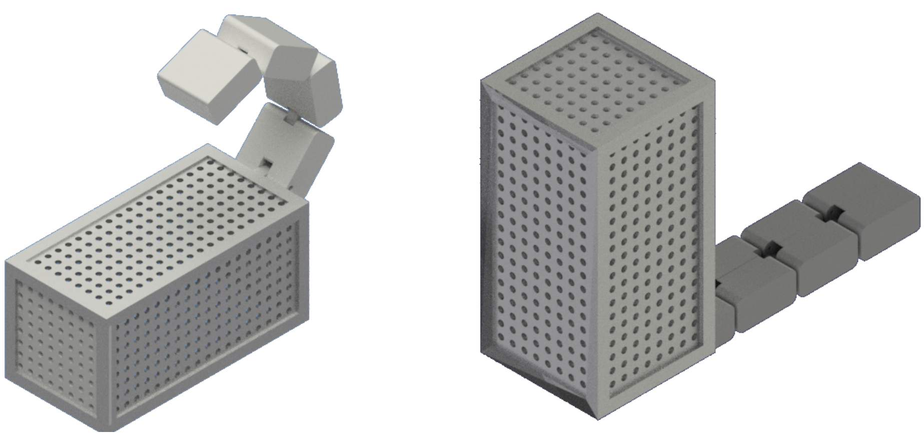
Alpha has the ability to recover from a fall. The front joints have significant freedom to rotate. When the robot falls and the IMU detects the fall, Agility proceeds to compose a recovery sequence by moving joints and the tail such that the robot can correct itself (to two-legged or four-legged state). The recovery sequence involves rolling the front joints 180 degrees rapidly and at different times so that the robot is able to get back into its original position.

TRANSFORMATION

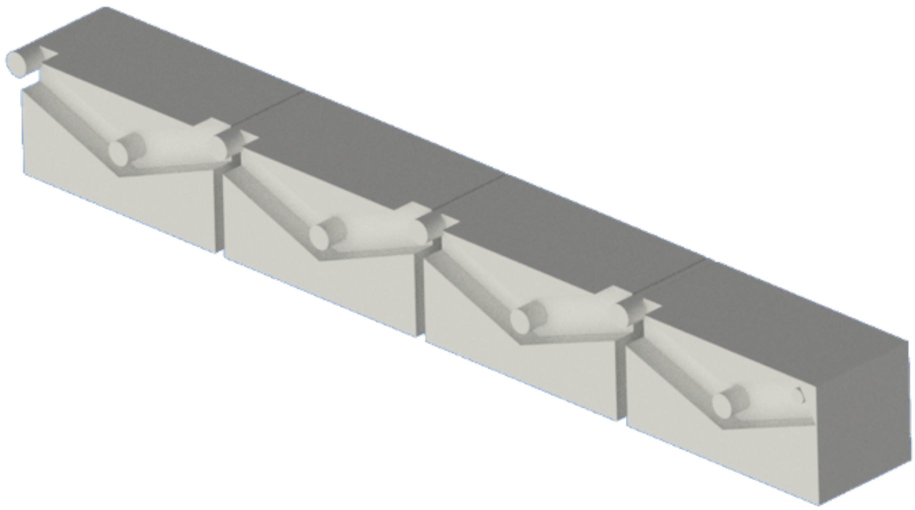
MECHANICAL

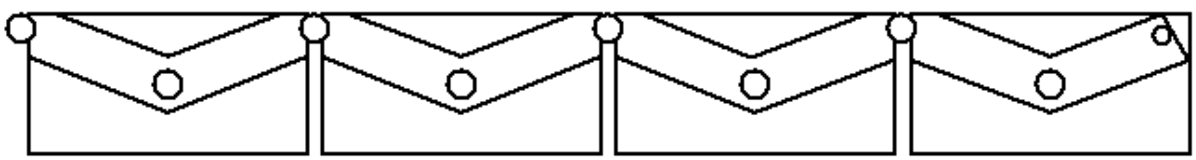
Alpha, an original four-leg walker, with a tail on the top of its body, will enter the chamber room. When it detects sufficient UV light and the G note in the song "Werewolves of London", a servo in Alpha's tail will quickly rotate outwards in order to help Alpha balance on two legs. After the tail rotates, Alpha pushes up off its front legs. During this motion, the servos in hips of the hind legs will rotate around the y-axis. The combination of both forces will make Alpha stand on two legs, bringing the front legs up, and the tail down.

TAIL



The tail is the main component in Alpha's transformation from quadrupedal to bipedal movement. The tail is responsible for the change in the distribution of Alphas total weight. The tail will only be able to move vertically into a curled position similar to the movement of a snake. The tail works by using a system of pulleys, string will be running through the four section of the tail. One end of the string will be attached to the tip of the tail and the other end will be wound around an axle that is attached to a servo motor on the back end of Alpha. while Alpha is in quadrupedal movement the tail will be in the curled position, meaning that the tail will be hovering over Alpha's back the weight of the tail will help ensure that Alpha stays on 4 legs and will help with quadrupedal movement giving Alpha more stability

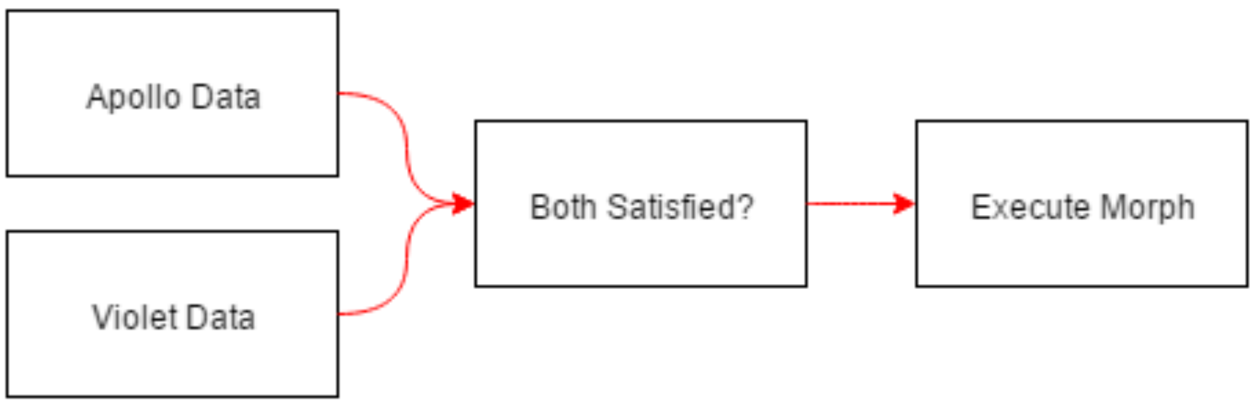




Once Alpha enters the transformation chamber and is allowed to transform into a bipedal walker, the front legs will spring upward and cause the tail that is resting on Alphas back to fall backward and hit a stopper that is connected to Alpha's body. This will make the front half of Alpha's body to become erect because Alpha's lighter front half. After Alphas has transformed into bipedal form the tail will be dragging on the floor behind Alpha as he walks, this will stabilize Alphas body as he walks in his bipedal form. The tail will be approximately 20 cm long by 10 cm wide by 10 cm tall.

PROGRAMMING - LYKOS MODULE

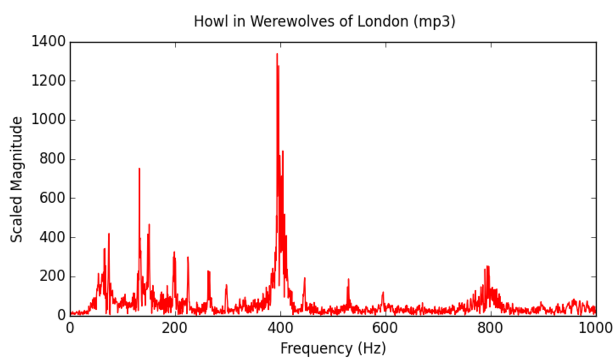
Wolfgang will only be allowed to transform once it reaches the morphing chamber and receives two stimuli simultaneously: a howl from the "Werewolves of London" and sufficient UV light. To accomplish this, Eclipse Technologies has equipped Alpha with the Apollo and Violet submodules (detailed later). When both modules report that the conditions for transformation are satisfied, a transform function is called which will initiate the morphing from a four-legged to a two-legged state.

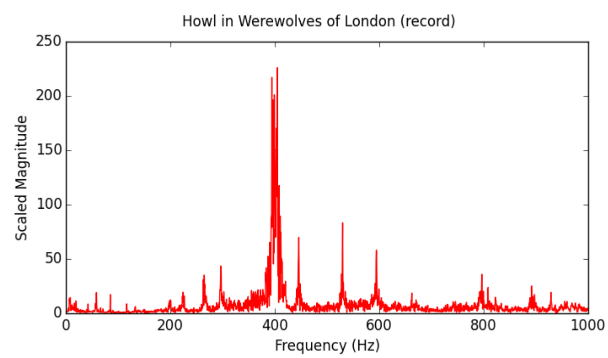


The transform function is a prepare-and-execute procedure. The robot pauses all of its movement activities and reads sensor data. It will then utilize the sensor data to create a motion path for all of the servos. Every step of the movement for every servo is stored into the memory and executed from the beginning after the entire preparation is complete. This will ensure that the robot has a smooth and controlled transformation, as attempting to compensate while standing is tasking on the processor and servos. The transformation when the sequence ends and the IMU reports that Alpha is now standing.

AUDIO - APOLLO SUBMODULE

In order to detect specific audio frequencies, Alpha is equipped with an integrated microphone on a camera and the Apollo submodule. Apollo is an asynchronous frequency analyzer. It utilizes PyAudio to perform non-blocking audio collections from the microphone and processes audio using a fast one-dimensional Fast Fourier Transform (FFT). In simple terms, FFT finds the magnitude of each frequency range in a given audio segment. The mission specifies that the robot must be able distinguish the howls in the song "Werewolves of London." Analysis into the howl shows that it is a relatively consistent G note (392 Hz).





The image on the left shows the howl in "Werewolves of London" from an mp3 file while the image on the right shows the same howl, but recorded using a low quality microphone at a distance of 1 foot from the speakers. In both graphs, the predominant signal is in the 390 - 405 Hz range. This indicates that it is possible to utilize frequency analysis to detect howls.

Detection must be fast, as the shorter howls last slightly less than two seconds. Apollo relies on *numpy*, a highly optimized C computational library to assist with analysis. The algorithm behind Apollo can be broken down into the following steps:

1. Obtain audio data (in mono) every 0.2 - 0.5 seconds in the form of an 1D array of signed 16-bit integers. The resolution of the output is given by the equation:

   resolution (Hz) =sample rate (Hz) / N

   Audio should be captured at a sample rate of at most 44,100 Hz with a FFT length (N) of 8192 to maintain a frequency detection resolution of around 5 Hz.

2. Perform a quick analysis on the audio to make sure that the volume of the detected sound is above a certain threshold. This eliminates false positives from white noise.
3. Take the FFT using *numpy* to get magnitudes for each frequency bin.
4. Obtain a sequence of linearly spaced frequency coefficients for the FFT data.
5. Find the maximum magnitude of the absolute value of the FFT (this will be the frequency that has the greatest magnitude, and the absolute value is necessary because FFT is taken in the complex plane).
6. Multiply the respective coefficient in step 4 for the maximum found in step 5 by the sample rate to obtain the frequency.
7. Compute the number of steps the note is away from A (440 Hz) using the equation:

   n=loga (f440 Hz) where f is the frequency and a is the 12th root of 2

8. Return the data and look for locations where n is the correct number of steps from the 440 Hz A, which in this case would be -2 (for the note G, which is 2 half-steps below A).

To increase detection accuracy, Apollo takes multiple samples and checks if the note G appears more than two times in a period of five data reaps. This ensures that no false positives occur as the program can occasionally err from white noise or background instruments. The data is returned to the Cerebral module for additional processing.

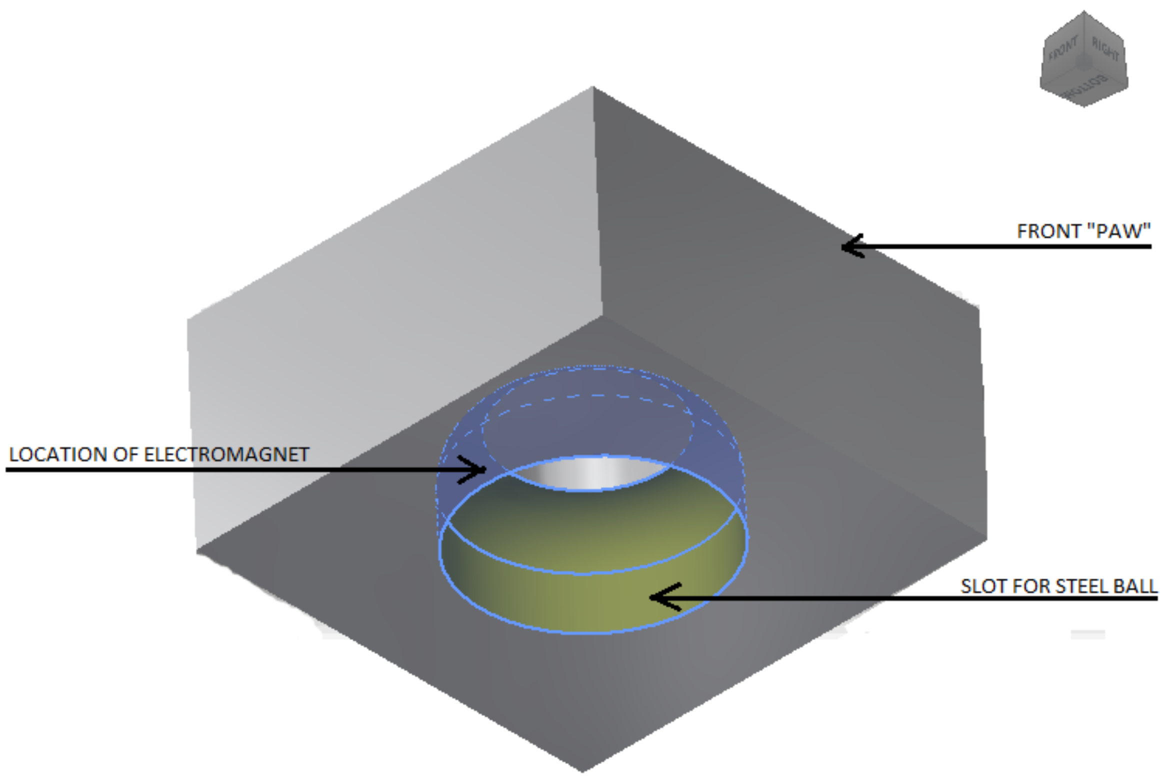
UV - VIOLET SUBMODULE

Alpha must be able to transform upon detecting ultraviolet light in addition to a G note in "Werewolves of London." For this tasks, Eclipse Technologies will use a simple UV Sensor Breakout attached to the back of the robot. The sensor reports analog data which can be easily interpreted by the Arduino Uno.

ARTICULATION

MECHANICAL

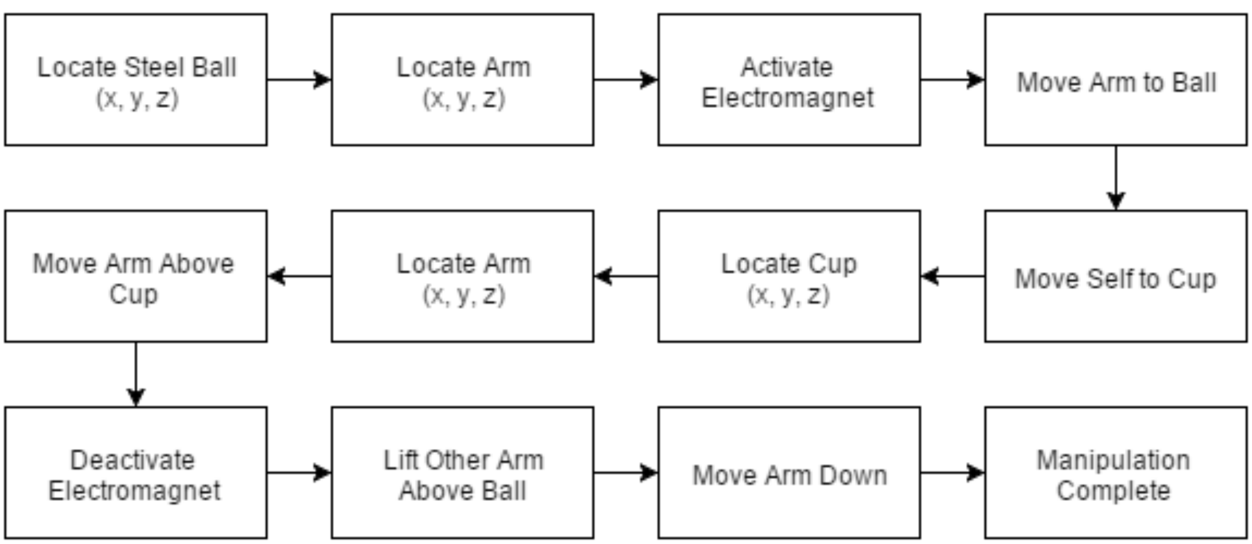
Alpha needs to transport a 1 in steel ball from a specified location (which we assume is on the floor) to a cup elevated 18 cm off the ground. Alpha will be in bipedal mode during this whole action. In order to complete the mission, the "palms" of both forearms will have electromagnets capable of lifting the 1 in diameter steel ball. In other words, as seen in the picture below, an electromagnet will be embedded inside each front foot. A hole inside the foot will be made in order for the foot to maintain a flat surface area while it is walking.



Both or one of the front legs will be able to transport the ball into the cup. Alpha would bend down its joints so that the front legs can reach the steel ball. Once it has captured the ball, it would return to its normal position which is enough to reach the cup. Alpha while in bipedal mode has enough reach to drop the ball into the cup. The moment the ball is dropped, Alpha's goal is complete and the rest of the mission can progress.

PROGRAMMING - TELEKINESIS MODULE

Alpha must have the ability to manipulate a steel ball and place it into a cup. The Telekinesis module is designed for this purpose. It utilizes vision data and servo locations to determine how to move an object to its destination. The front legs are embedded with electromagnets which will be activated programmatically when Telekinesis detects that it is close enough to the steel ball. After the ball is successfully obtained, the robot will walk over to the cup while holding the ball and calibrate its position and arm to drop the ball into the cup.



Telekinesis is not directly responsible for balance and movement. It simply relays instructions to the Agility module for processing. The extra weight will not present a problem to Wolfgang as Agility has the ability to use IMU data to readjust the robot's center of mass while walking.

VISION

PROGRAMMING - THEIA MODULE

Alpha's Theia module extends the shared Theia module. It adds additional features which are necessary for navigation and steel-ball manipulation. No details are given regarding the appearance of the transformation chamber or the cup. While color is a good indicator, Theia will have difficulty distinguishing similarly colored objects without extensive testing and configuration. The steel ball will be problematic as its luster and reflection can be distracting and cause color detection to fail.

To supplement detection accuracy, Alpha's Theia module is also equipped with a shape detection feature. Shape detection is done through a polygon approximation function in *OpenCV*. To Wolfgang, the steel ball will appear to be a circle. Circles, or any simple polygon of n-sides can be detected by first finding the contours in the image and then approximating a polygon that fits the contour. The pseudocode below shows this operation.

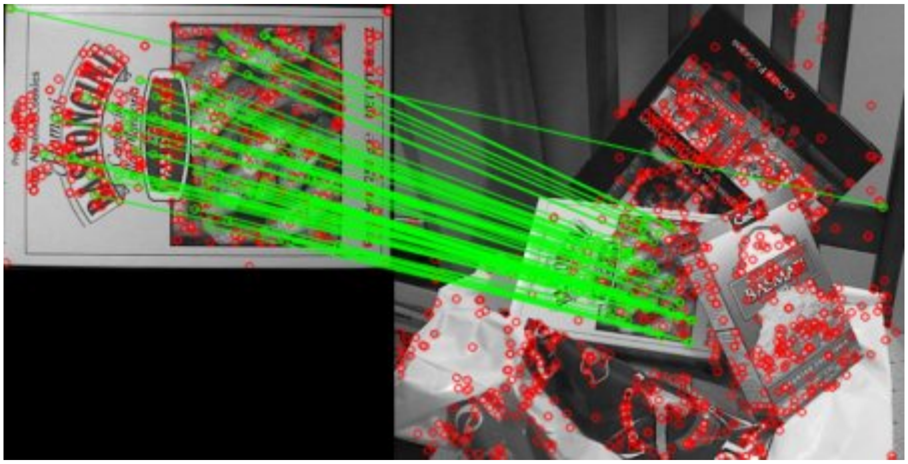
```
% Capture image from OpenCV
% Find all contours in the image
for contour in contours:
    % Approximate a polygon for the contour
    % Count the number of distinct edges
    if len(edges) > 15:
        # Circle has been detected
    elif len(edges) == 4:
        # Triangle has been detected
```

```
elif len(edges) == 4:
    # Rectangle has been detected
elif len(edges) == 5:
    # Pentagon has been detected
```

(image from OpenCV project)

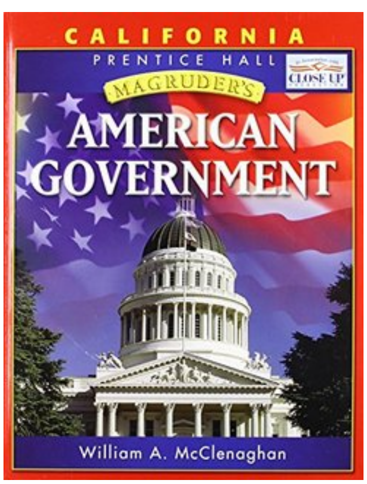
Data from the shape detection will be combined with data from the color detection to ensure that the robot is in fact targeting the steel ball. Before any of this can be done however, Wolfgang must be able to reach the morphing chamber. The features of the chamber are not yet known. However, it can be assumed that in addition to being a regular, detectable shape, the morphing chamber will also have some distinct and distinguishing features in terms of texture.

Wolfgang's Theia is also equipped with a ORB detection algorithm. ORB is an *OpenCV* feature extractor. From an object or picture, it can extract relevant information on edges, contours, and details in the object/picture that make it unique. The data can then be stored and used to detect that same object/picture again, regardless of orientation or rotation. An example of the detection powers of ORB-based detection is shown below.



(image from OpenCV project)

ORB can take an arbitrary picture of the object (taken before the mission) and use it to detect the same object in the real world. This highly accurate and optimized algorithm allows Wolfgang to locate and move to desired targets without worrying about disruptions of shape and color. The image below shows an example of a single image of a book being detected in a cluttered scene. Both images were processed in grayscale, so color was not considered during feature detection. Feature matching was done using a Bruteforce Hamming Matcher paired with RANSAC homography followed by a simple perspective transform.

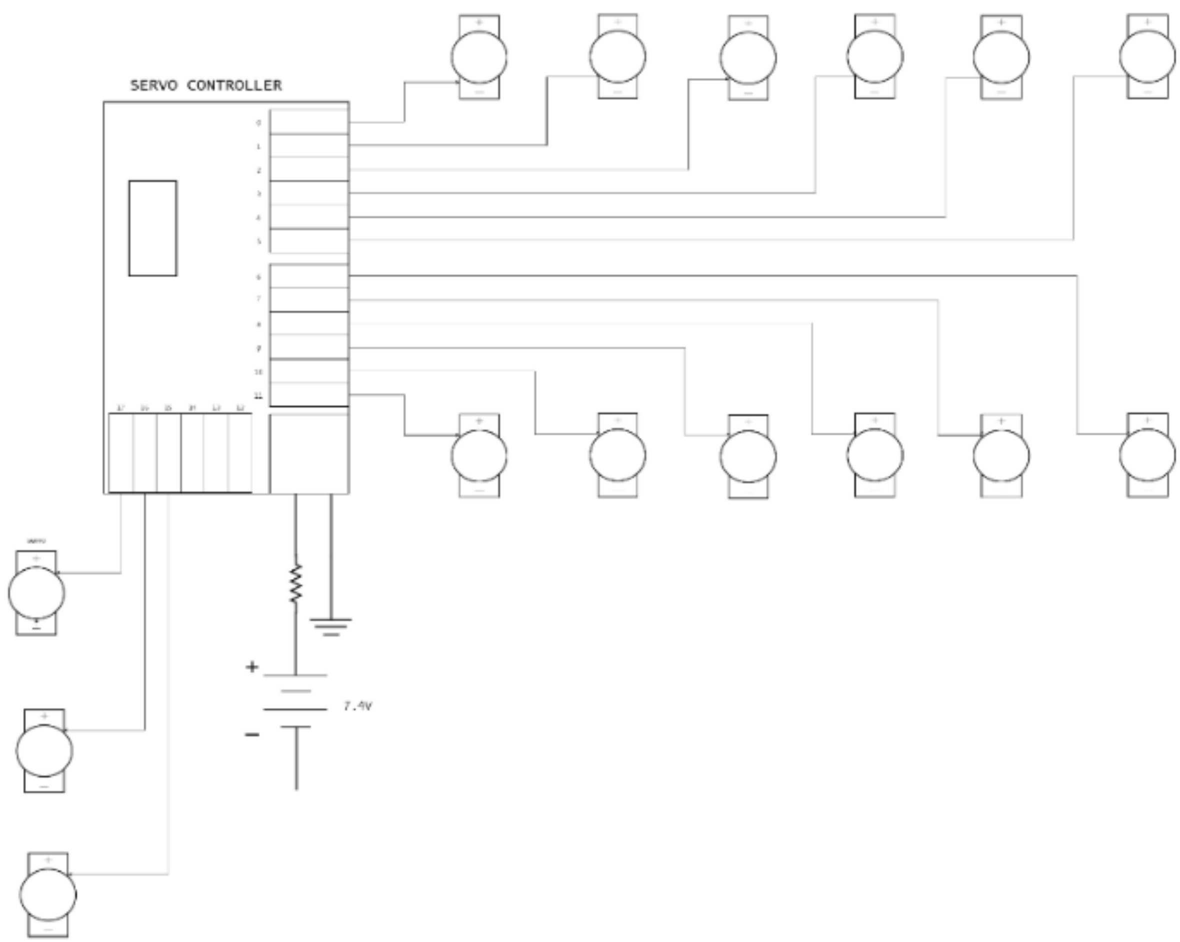


Using a combination of color detection, shape detection, and feature detection, Wolfgang's Theia module will allow for accurate and quick movements to a desired target location. Even if part of the object is obstructed or exposed to bad lighting, Wolfgang will still be able to distinguish it from its surroundings. Theia will also allow for smooth manipulation of the steel ball with high repeatability.

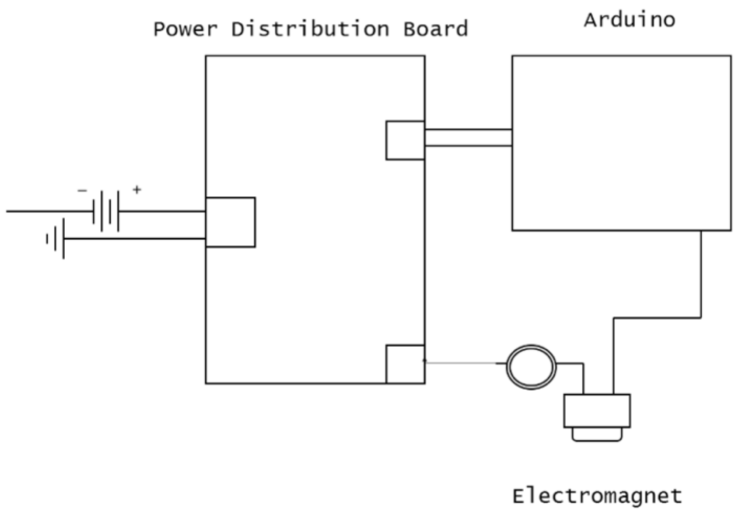
POWER AND WIRING

ELECTRICAL

Wolfgang will be equipped with 13 6-volt (v) servos to achieve locomotion, with each leg containing three servos. All servos will be wired directly from a 7.4-volt battery to a servo controller, a complex distribution board, and then stepped down to 6v using a voltage stepper. Communication will be worked through an 18-port servo controller, which will communicate with the IMU sensor and vision system to plan and walk the most efficient route.



Wolfgang will be equipped with an electromagnet that will have a pull force of 50 pounds at its max voltage- 12v DC. It will be powered using 7.4v AC which will then be converted to 12v AC using a stepper. It will then be changed to DC using an AC to DC converter. The magnet should therefore be fed with 12v DC.



TOTAL DRAW FROM THE ROBOT

Each robot will be equipped with similar items shown below in the chart. These are the major if not all the electrical components of the robot.

Item	Voltage	Amperage	# Per robot	Total Amperage
Raspberry Pi 2	5	1.8A	1	1.8A
DFRobot 270 Degree Servo	6	100mA	13	1.3A
IMU Breakout	3	40mA	1	.04A
HD Webcam	5	500mA	2	1A
Mini Maestro Servo Controller	5	4mA	1	.04A
SparkFun UV Sensor	3.3	.0003A	1	.0003A
Micro Servo	5	50mA	2	.1A
Arduino Uno	5	40mA	1	.04A
Electromagnet	7.4v	.5A	1	.5A

The total amperage for Wolfgang is 4.82A. Components with an unknown amperage nor the minor components not listed cannot be assumed negligible. To accommodate this, another 2 amps will be added to the total amperage; it will be a safe bet to assume that the max amperage is 6.82A. However, this may be off as all components will not work continuously. For example, the servo will not all work at the same time.

THE BATTERY

As said above, the assumed maximum amperage of the robots will be 6.82A. Thus, the batteries' minimum discharge rate needs to be 6.82A so the robot doesn't burn out if the robot happens to pull that many amperes. While none of the items require 7.4V, the battery will be 7.4 volts as it is more versatile and more accessible than lower voltage batteries, which tend to be Lead Acid batteries anyways. Similarly, a battery rated at 6.82AH should be able to run a device running 6.82A for an hour. The ideal battery seems to be a 7.4V Lithium Polymer battery with 6600mAH of run time by Tenergy. The battery provides the robot with a little under an hour of run time (58 Minutes) and has a sufficient discharge rate to fit our needs. The voltage may seem as if it will be too high for most components and to low for the electromagnet. But the 7.4V battery is the most efficient weight-wise and the voltage can easily be brought down to the proper voltage needs of the robot, and still provides the electromagnet with sufficient power to do its job.

DISTRIBUTION

A distribution board will be wired directly to the battery and will work similarly to a panel. The main feed will be soldered to a common bussing that will be shared with all electrical components. The voltage will be 7.4V all the way through the wiring and distribution board and then will be resisted down to its proper voltage right before the electrical component. This will be the case with all electrical components except for the servos, which will share a separate distribution board altogether. This distribution board for the servos will also be considered the servo controller as it can turn the servos on and off as necessary. Before the distribution board, the voltage will be lowered to the common 6V that all servos share. This will be done for efficiency for both price and manufacturing time.

CABLING

A protective sheath will be put over all wires to make sure no abrasive parts inside the robot cut open the cables and cause a short circuit. Components inside the body of the robot will be an easy point A to point B, but the difficulties arise when one must wire through the moving parts such as the legs and the head. The protective sheath will then work to relieve tension among the wires. Slack in the wire will also be required so that the wire does not prevent full movement of the robot and its servos. Simply put, extra wire will not be secured to the robot so that it can move without fear of cables being torn.

FABRICATION AND MATERIALS

MANUFACTURING

The overall frame for Alpha will be made from .39 in thick Aluminum. This thickness will be used in order to keep the frame lightweight but still provide rigidity. The frame will have .098 in Aluminum sheets with holes in order to mount the different components. The tail will be made from ABS plastic in order to minimize the weight and optimize accuracy when manufacturing. The tail will be used to help support Alpha during bipedal locomotion so if needed extra weight will be added inside the tail in order to help the robot morph from 4 legs to 2 legs. The back legs will be made from .16in Aluminum. This thickness will be used because the back legs will need to be strong enough and have a large enough surface area to hold up Alpha when walking on 2 legs. The front legs will be made out of .10in Aluminum. This thickness will be used to keep the front legs as lightweight as possible in order to help in the morphing process. The front feet will be 3-D printed using ABS plastic because it is the best and fastest method that insures less inaccuracies. It will also help keep Alpha lightweight and ease the morphing process. An electromagnet capable of pulling 10lbs will be embedded inside the front feet.

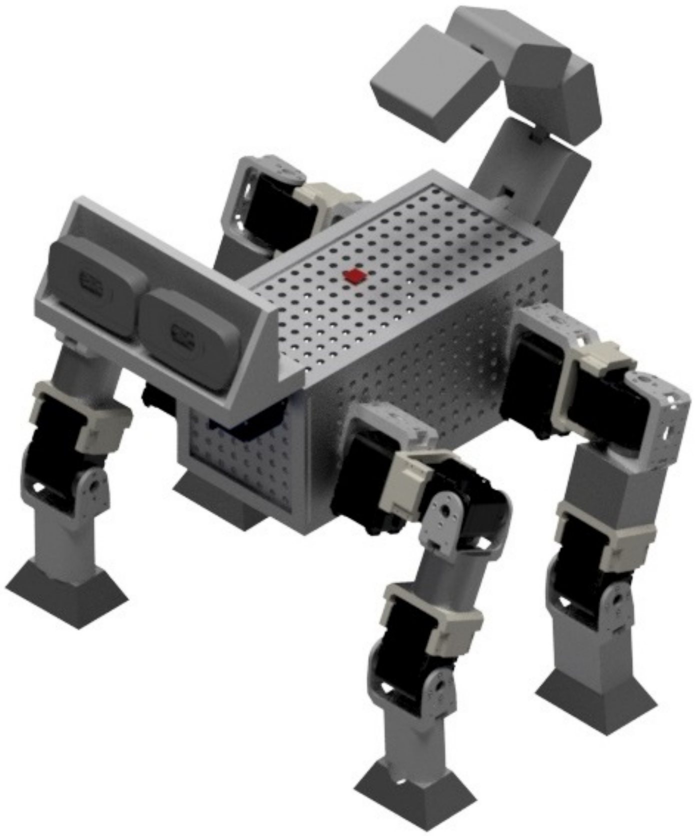
Round Electromagnet	Diameter (inches)	Height (inches)	Watts	Pull (lbs)	Weight
Item#70A0003	0.75	1.25	1.5	10	0.0482 kg
Item#70A0008	1	0.718	1	10	0.0539 kg

The table below demonstrates a comparison of pros(+)and cons(-) regarding 3 types of metals: Aluminum, Titanium and Steel. Through the decision matrix, Aluminum proved to be the number one choice due to its factors of being a strong, lightweight, easily machinable and cheap. In second place, we have Titanium, which is very strong and lightweight, however it's price exceeds our budget. Lastly, we have steel which is strong like the other two metals. It's cheap like Aluminum, but it is heavy giving a disadvantage since we have weight restrictions regarding our robots.

Metal	Strength	Lightweight	Machinability	Price	Total
Aluminum	+	+	+	+	4
Titanium	+	+	-	-	2
Steel	+	-	-	+	2

Below are listed all of the materials that will be used in Alpha with their calculated weights.

Item	Weight
Frame	0.318 kg
Small Sheet (x2)	0.0344 kg
Long Sheet (x4)	0.36468 kg
Tail Section (x6)	0.17142 kg
Servo (x13)	0.767 kg
Micro Servo (x2)	0.009 kg
Servo mount (x14)	0.5712 kg
Foot (x4)	0.339 kg
Foot connectors (x4)	0.03628 kg
Head	0.189 kg
Camera (x2)	0.116kg
Head servo (x2)	0.0308 kg
Thigh pieces Front (x2)	0.0248 kg
Servo rotating piece	0.043 kg
Raspberry Pi 2	0.045 kg
IMU Breakout	0.0028 kg
Arduino Uno	0.025 kg
Mini Maestro	0.003 kg
Battery	0.311 kg
Wiring	0.03554 kg
Electromagnet	0.0482 kg
Back thigh pieces (x2)	1.174 kg
Total	4.686 kg



Wolfgang

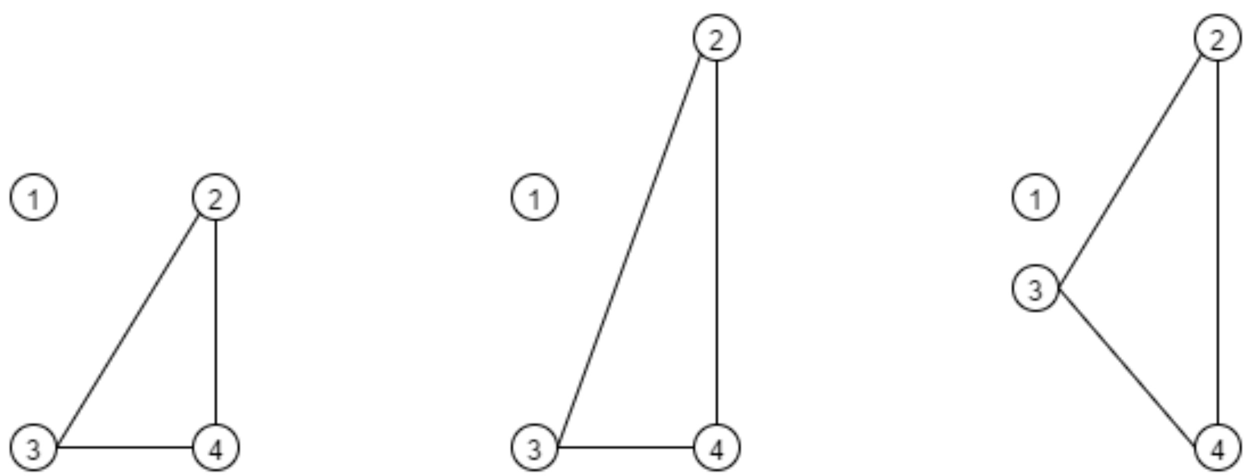
PACKBOTS

LOCOMOTION

MECHANICAL

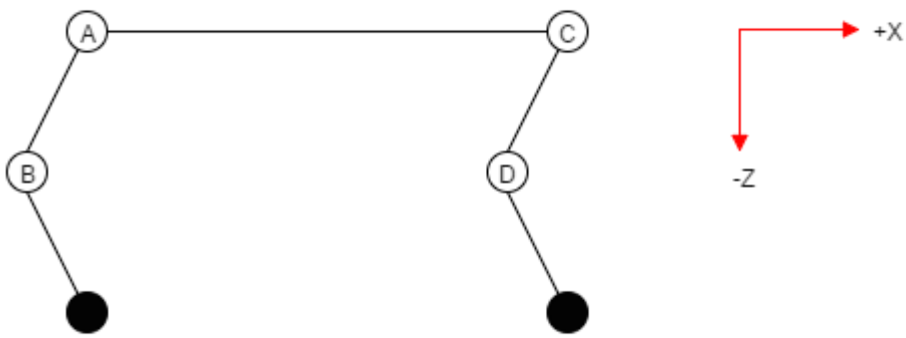
The PackBOTs, also known as DOG-1E5 and DOG-4S1, are robots designed to identify, capture, and retrieve the correct prey. Their legs are designed to allow smooth and efficient movement. There are four legs on each robot, each with three degrees of freedom. In order to obtain three degrees of freedom, a servo will be mounted on each joint. Both the front legs and back legs move in the x and z planes. The front legs can perform yaw rotation, rotation about the z axis, and the back legs can perform both yaw rotation and roll, rotation about the x axis. These degrees of freedom allow the robot to move with agility and versatility.

BALANCE



In order to avoid falling, the center of mass must remain in the triangle, depicted in the figure above. The three vertices of the triangle represent points where the robot's legs touch the ground, while the remaining point represents a leg that is lifted up. The center of mass is usually located at the center of the hypotenuse of the triangle. This allows the center of mass to always be in triangles during any configuration.

In order for the PackBOTs to be more agile, the knees will face the back of the robot. The hip joints (A and C) will rotate from -170 to 100 degrees relative to the z-axis. The front knee joints (B and D) will rotate from -10 to 170 degrees relative to the lines formed by the joints A & B and C & D respectively. This large range of motion allows for efficiency in our recovery method. In the image below, positive x is the forward direction.



TORQUE

As the PackBOTs walk, they will use servos to move their joints. Eclipse Technologies has decided to use 12 high powered servos (3 in each leg) to maintain consistency throughout the robot. In the table below are calculations of the average torque the robot would need as it walks.

PROGRAMMING - AGILITY MODULE

The movement of PackBOTs is crucial in capturing moving PreyBOTs on the field. PreyBOTs have an advantage as they are allowed to move on wheels while the PackBOTs must walk. Thus, Eclipse has designed a powerful PackBOT movement module to ensure that DOG-

1E5 and DOG-4S1 can outmaneuver, outsmart, and outrun the PreyBOTs. The PackBOTs' Agility module is designed exclusively for four-legged motion. However, it is optimized for speed, real-time computation, and movement while turning.

FOUR LEGGED GAIT

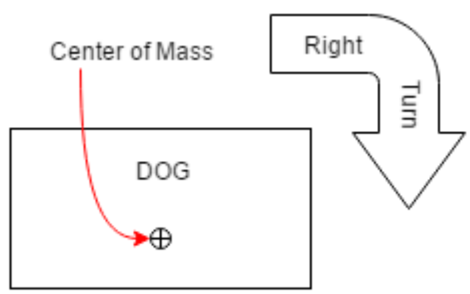
PackBOTs must be highly versatile when it comes to movement. Certain conditions call for different types of gaits. Thus, PackBOTs' Agility module is equipped with three different gait patterns: trot, crawl, and 1243/1342 rotary gallop. Agility can switch between these three gaits at any time during the mission to best accommodate the situation.

PackBOTs must be versatile, fast, and maneuverable. So PackBOTs will use the trot gait for most of the mission. The trot gait involves standing on two opposite legs and moving two opposite legs. This gait makes PackBOTs fast but also maneuverable, as they can turn while walking using the trot gait by shifting their center of mass. The pair of legs are lifted and placed down for the same amount of time.

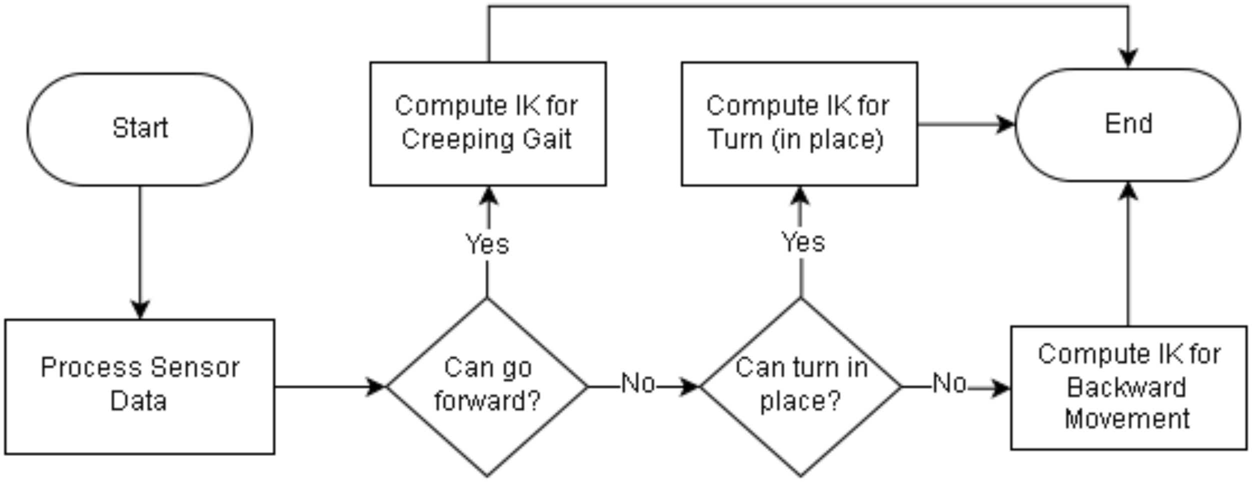
2 & 3 Lifted   1 & 4 Lifted

Leg 1	Down	Down	Up	Up
Leg 2	Up	Up	Down	Down
Leg 3	Up	Up	Down	Down
Leg 4	Down	Down	Up	Up
Time	0 - 1/4	1/4 - 2/4	2/4 - 3/4	3/4 - 1

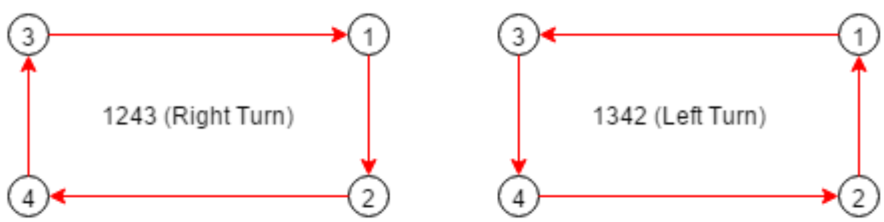
In order to execute turns while moving, Agility will adjust the center of mass of the robot to match the direction of the turn. This method of turning is modeled after quadrupeds, which is currently the most efficient and effective method. Each of the four joints in the legs will rotate so that the robot tilts and the center of mass shifts.



Data from the IMU will be collected to determine the center of mass and momentum. Agility will generate the necessary gait patterns to reach the target location. When the DOG cannot move forward and turn at the same time (when it is near a wall or a target), Agility tilts each of the four legs individually so that the robot can turn in place. In extremely rare conditions, DOGs' Agility module can even generate a backward movement sequence for the robot to walk backwards.



Sometimes, even the trot gait is not enough. For example, if the target is at a distance and attempting to run away, Agility will shift its gait to the 1243 and 1342 rotary gallop. Depending on the target's initial position (left or right of robot), Agility will choose one of the gaits more suited for that direction of turning. However, this does not mean that the robot will be unable to turn in the other direction using the less preferred gait. The timing diagram shows the 1243 rotary gallop.



Leg 1	Down	Down	Up	Up	Up	Up
Leg 2	Up	Down	Down	Up	Up	Up
Leg 3	Up	Up	Up	Up	Down	Down
Leg 4	Up	Up	Up	Down	Down	Up
Time	0 - 1/6	1/6 - 2/6	2/6 - 3/6	3/6 - 4/6	4/6 - 5/6	5/6 - 1

Eclipse Technologies also ensures that the DOGs will have maximum longitudinal stability when dragging prey. To do this, Agility switches to the 1423 crawl gait. This gait maximizes forward moving stability and ensures that the DOG carrying the PreyBot will not fall and lose its prey.

Leg 1	Down	Down	Down	Up
Leg 4	Up	Down	Down	Down
Leg 2	Down	Up	Down	Down
Leg 3	Down	Down	Up	Down
Time	0 - 1/4	1/4 - 2/4	2/4 - 3/4	3/4 - 1

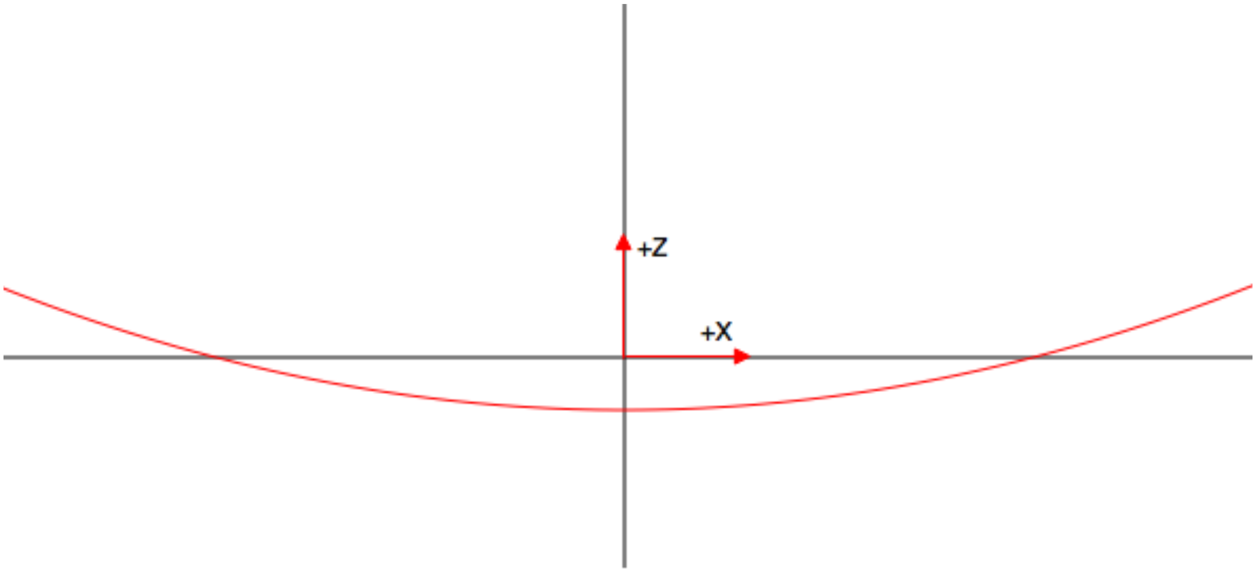
The timing in all gait patterns is relative. A total time for one cycle can be defined and Agility will compute individual leg down and leg up times.

FORWARD, BACKWARD, AND SIDEWAY PROPULSION

PackBOTs rely on the sharp power stroke for their crawl gait (refer to Alpha's Agility module). However, the PackBOTs rely on a smooth power stroke for the trot and rotary gallops. A smooth power stroke is generated from a quadratic function and allows for a bouncing movement to maximize speed and servo efficiency. Comparatively, the smooth power stroke works best for lighter robots like the DOGs.

The general form of the quadratic curve is defined in the x-z plane as:

$$z = ax^2 - c$$



The x-axis is the ground plane relative to the robot before the leg enters the power stroke. The root of the leg is located somewhere along the positive z axis. The portion of the curve on or below the x axis represents the time period in which the leg is in its “down” state. The c in the equation is simply the height by which the leg should lift the robot. Thus, a must be determined given the amount of time or in this case the arc length and a desired velocity.

Due to the lack of a b, the curve will always be an even function. Thus it is possible to quickly compute the roots of the function. Only the positive root is necessary for arc length computations.

$$x = \frac{-b \pm \sqrt{b^2 - 4a(-c)}}{2a} = \frac{\sqrt{4ac}}{2a} = \frac{\sqrt{ac}}{a}$$

$$r = \frac{\sqrt{ac}}{a}$$

It is now possible to use the arc length formula to find the length of the parabolic arc from $(-r, 0)$ to $(r, 0)$ by taking twice the arc length from $(0, -c)$ to $(r, 0)$.

$$L(a,c) = 2\times \int_0^r \sqrt{1+[z'(x)]^2}\ dx$$

$$L(a,c) = 2\times \int_0^r \sqrt{1+4a^2x^2}\ dx$$

$$L = 2\left[\frac{2ar\sqrt{4a^2r^2+1}+\sinh^{-1}(2ar)}{4a}\right]$$

$$L = \frac{2\sqrt{ac}\sqrt{4ac+1}+\sinh^{-1}\left(2\sqrt{ac}\right)}{2a}$$

However, it is necessary to compute a given L and c in order to generate the quadratic curve. A problem arises in that this equation contains a transcendental function and analytical methods of solving for a are nontrivial. Obviously, there is at most one solution but that solution cannot be determined analytically. Due to computational capacities, it is feasible and highly efficient to use Newton's method to estimate a solution to the transcendental function given an initial guess.

$$0 = \frac{2\sqrt{ac}\sqrt{4ac+1}+\sinh^{-1}\left(2\sqrt{ac}\right)}{2a} - L$$

The function behaves erratically when $a = 0$ because the denominator causes the equations to become undefined. Obviously, a can never be zero because the quadratic equation would become a line. To assist with convergence, it is better to remove the denominator (identified from testing). The derivative is simplified by assuming that all variables are positive real numbers.

$$f(a) = 2\sqrt{ac}\sqrt{4ac+1}+\sinh^{-1}\left(2\sqrt{ac}\right) - 2aL$$

$$f'(a) = 2\left(\sqrt{c\left(\frac{1}{a}+4c\right)} - L\right)\ for\ a, c, L > 0$$

$$a_1 = a_0 - \frac{f(a_0)}{f'(a_0)}$$

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)}$$

An error function can be defined by taking the difference between the a_n and a_{n+1}. This is extremely useful for speed as the program can exit once the error goes beneath a certain threshold.

$$\delta = |a_{n+1} - a_n|$$

Applying around 30 iterations with $a_0 = 1$ can provide an extremely accurate value for a on the order tens of microseconds. Speed if further accelerated by using a JIT compiler to translate the entire function into optimized assembly before execution.

```
@jit
def f1(a, c, l):
    '''Returns f(a).'''
    return (2 * (a*c)**(1/2) * (4*a*c+1)**(1/2) + math.asinh(2 * (a*c)**(1/2))) -
2*a*l
@jit
def f2(a, c, l):
    '''Returns f'(a).'''
    return 2 * ((c * (1/a+4*c))**(1/2) - l)
```

```
@jit
def find_a(c, l):
    '''Approximates a using Newton's method.'''

    # The initial guess for a.
    a = 1.0

    # The error term.
    error = 0

    # Run 30 iterations of Newton's method
    for i in range(30):
        a1 = a - f1(a, c, l) / f2(a, c, l)
        error = abs(a1 - a)
        a = a1

        # Convergence completed early.
        if error < 3e-10:
            return a

    return a
```

Benchmarks indicate that the entire process of determining a can be completed in under 3 microseconds for nearly all valid c and L values. A lambda function generator can be made so that the parabola can be passed to the Finesse module for parsing.

```python
def generate(c, l):
    '''Returns a lambda function for quick evaluation of z(x).'''

    a = find_a(c, l)
    return lambda x: a * x**2 - c
```

In the PackBOTs, a desired velocity v, time on the ground t, and height c are passed to the *generate* function in the following manner so that the resulting parabola is a curve where the time the leg is on the ground can be controlled:

$$L = vt$$

$$generate(c, L)$$

To perform sideway propulsion, the same parabolic curve is used. However, the curve is translated along the y-axis. This is very simple transformation.

$$[x, 0, z(x)] \rightarrow [x, y, z(x)]$$

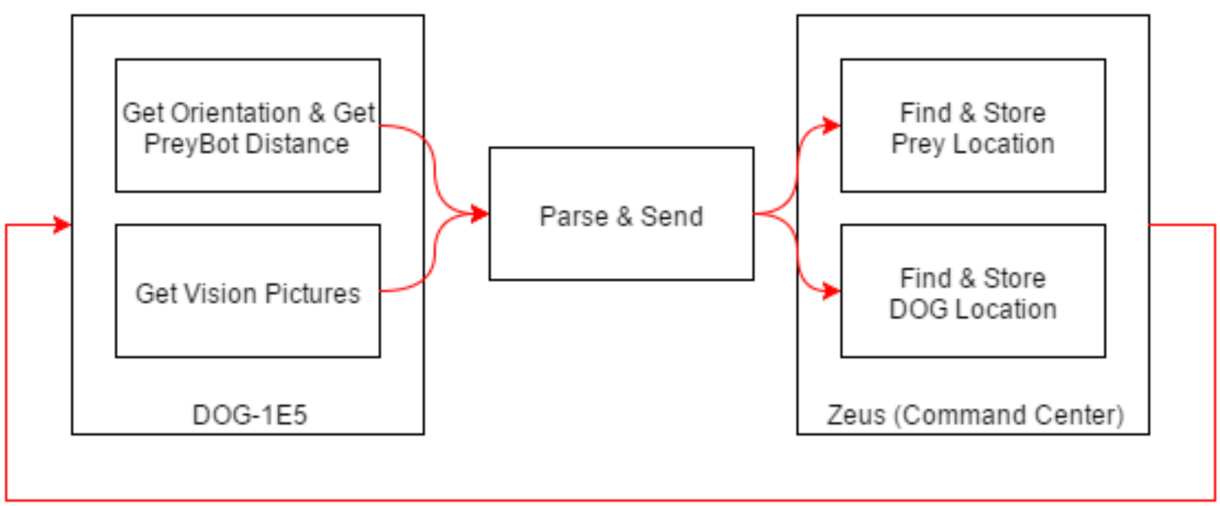
RECOVERY

Both DOGs are equipped with eight wide angle servos. If the IMU detects that the robot has fallen, Agility will generate a simple recovery sequence by flattening all joints to be parallel to the body and then rotating two selected joints quickly by 180 degrees to flip the robot back to it's original positions. The joints can then reorient themselves to their pre-fall state.

VISION

PROGRAMMING - THEIA MODULE

In order for PackBOTs to capture their targets, they must have the ability to react quickly to changes in the motion of PreyBOTs. For this reason, the DOGs will only use color and stereoscopic vision to determine the location and distance of their targets. However, in order for the robots to collaborate, it is necessary that they understand each other's positions and the positions of the PreyBOTs. Unlike Alpha's Theia module, the DOGs' Theia module is equipped with quick field mapping and position system. Along with the position data of the PreyBOTs, each DOG will also send the pertinent image capture to Zeus asynchronously for processing. Zeus will parse the image and combine it with the robot's orientation data to determine the position of the PreyBOT(s) in the robot's field of view and the position of the robot.



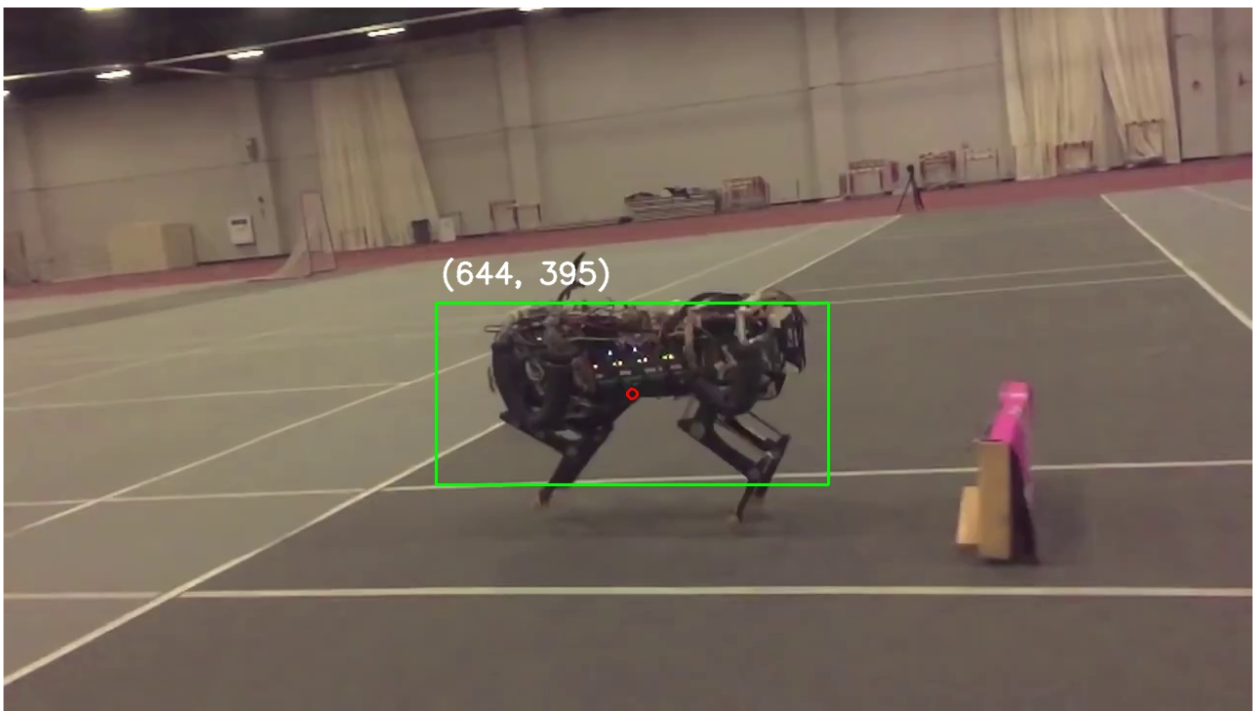
To perform a location computation, the command center will process the orientation to determine which wall or walls the robot is facing. Then, it will determine the distance to the wall(s) by performing stereoscopic depth map analysis. Combining the data with the orientation, it will be possible to determine the absolute position of the DOG on the field. The robot will also send the PreyBOT's distance. From the DOG's position, the server can then compute the PreyBOT's location as well. This is not a crucial operation and is done when there are resources to spare. This operation does not block any other operations and will not occur if robot or server load is high.

AUTONOMOUS & ADAPTIVE OBJECT TRACKING

No features or color descriptions of the desired targets are given prior to mission day. This makes it relatively difficult to configure HSV and vision manually. To combat this problem, Eclipse Technologies has developed an adaptive and fully autonomous object tracking system that can locate and follow a desired region of interest. This system relies on a simple prerequisite - that the PreyBOTs will move on their own without external stimuli.

At the beginning of the mission, the PackBOTs will remain motionless for around two seconds to observe the moving PreyBOTs. Once a PackBOT detects movement, it will create a mask by comparing two different frames and automatically select the ROI, which should be a PreyBOT. Once Theia successfully stores this ROI, it can begin applying the CAMshift algorithm to track the object. As a backup, it is possible to manually input an ROI image for use with the algorithm.

One major advantage of this object tracking system is that it does not require specific descriptions of the desired moving entity. CAMshift relies on a pixel distribution based on Hue (and possibly Saturation). Upon locating the ROI, it finds the location in the next frame where the most concentrated presence of pixels in the ROI are located. It can then compute a new ROI if necessary for the next frame. This method is orientation-independent. Whereas features and color could degrade from different angles, CAMshift's reliance upon the previous frame's ROI ensures accurate detection as long as the tracked entity does not move out of the FOV and completely alters its orientation/appearance before the PackBOT can regain the target.



In addition, the vision framework uses the Kalman filter to predict the location of the target even it it becomes obscured for a certain amount of time. This makes the PackBOTs capable of locating and tracking arbitrary prey or targets.

HUNTING

MECHANICAL

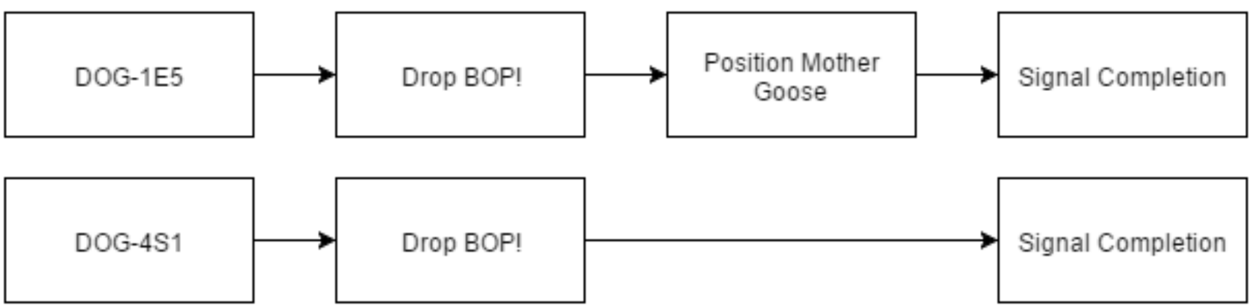
In addition to the two tools, BOP! and Mother Goose, Eclipse Technologies decided upon using a jaw mechanism to immobilize and transport the PreyBOTs. The jaw would be located on the front half of the robot in the field of view of the camera. This allows the robot to locate its targets and determine where to position the jaw. It would consist of two metal claws fixed on a servo that moves the jaws in a snapping motion. Each metal claw has three teeth that each have a length of 1 cm. The tips of the teeth will be rounded.

The jaw will be able to open up at least 8 cm wide so that the PreyBOTs can fit inside. It will also be able to "bite" into the target 4 cm deep. A servo will be attached to the rounded rectangular end to power the clamping of the jaw. In between the two plates, there will be a scanner prepared to read the RFID tag embedded in the PreyBOTS to ensure that the right target is captured. If the wrong target is captured it will be released by opening the jaws. The whole jaw would be attached to a servo-powered joint that will be able to move 180 degrees about the y-axis. This will enable the robot to carry the robot from the ground to a higher position if necessary. Eclipse Technologies has decided that the jaw mechanism is a secondary option for capturing and transporting the PreyBOTs.

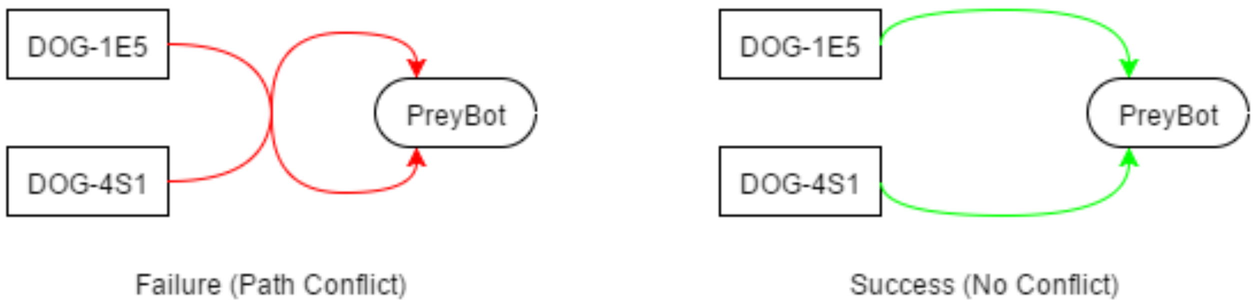
PROGRAMMING - ARES MODULE

One DOG may not be able to attain the required target. Thus, Eclipse Technologies has equipped both DOGs with Ares - a strategy, capturing, and collaboration module. Ares depends heavily on the Theia module. Vision data is sent to the command center for asynchronous processing and data can be retrieved from the server at any time. The main purpose of Ares is to have the DOGs provide the location of PreyBOTs on the field, ensure that both DOGs can collaboratively seek out the correct target, and bring the desired target back.

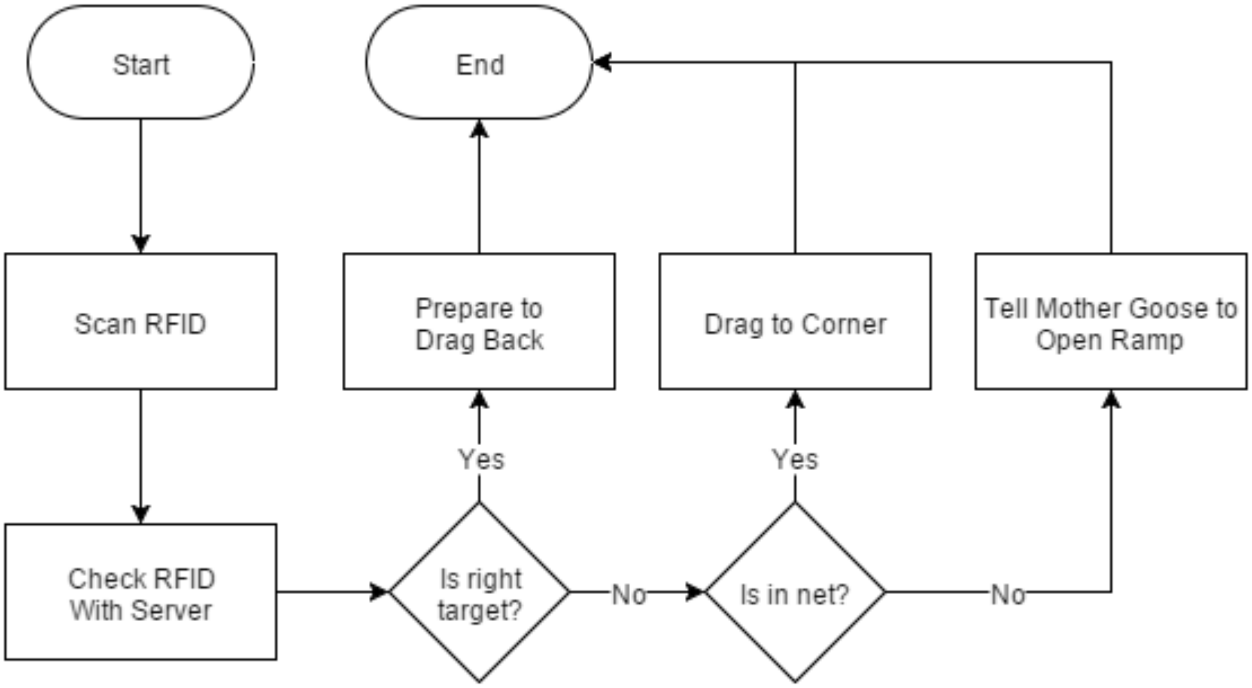
At the beginning of the mission, Ares ensures the deployment of BOP! and Mother Goose. DOG-1E5 will carry one BOP! and Mother Goose while DOG-4S1 will carry two BOP!s. Upon entering the mission field, DOG-1E5 and DOG-4S1 will proceed to deploy BOP! at three preset locations equally spaced on the field. This separation ensures the highest probability of capturing the targets. Ares will then direct DOG-1E5 to line up the trailer with a corner of the field. Finally, Ares signals for Mother Goose to open and prepare for capture.



The robots are programmed to avoid nets using vision data. Ares directs the robots to enter a watch mode and wait for the PreyBOTs to run into the nets. If a net is set off, Ares detects the change in the color of the net and directs both robots to scan and capture. If nothing occurs after a preset time (30 seconds), Ares will direct the DOGs to seek out a target and attempt to corner it at the location of the trailer or force it to run into a BOP!. During certain operations, such as target seeking, Ares will pull position data from Theia (explained in Theia Module). Each robot will then generate a path in attempt to corner the robots. Both DOGs will exchange path data to ensure that they are not colliding or ending at the same location. If such is the case, the robots will negotiate a functional cornering plan and execute the plan.



Once a PreyBOT is trapped in BOP! or in Mother Goose, one of the DOGs will move to locate and scan the RFID tag on the PreyBOT. It then checks with Zeus to see if the RFID belongs to the correct target. If it is the wrong target, and the target is in the trailer, the DOG will signal a release call to free the PreyBOT from Mother Goose. If the incorrect trapped prey is in BOP!, it will be dragged to a corner of the field to not distract vision processing. However, if the target is the correct one, both DOGs will drop their current tasks and work on getting the PreyBOT to its holding cell.



Ares also has the ability to control and manipulate tools. In the case of Mother Goose, Ares can signal the back ramp to open or close. In addition, Ares also helps the robot align with the trailer to bring it back to the holding area. For BOP!, Ares helps the robots avoid stepping into the traps, collect the traps when it detects that it is closed, and drag the trap back to the holding area.

In most cases, a capture by BOP! or by Mother Goose will be sufficient, and the planned method of either throwing the net into the holding cell or pushing the robot from Mother Goose will work. However, in worst case scenarios, when the DOGs cannot move the net or cannot lift the net into the holding cell, Ares will take control of Alpha. It will direct Alpha to morph and come on to the mission field to assist with lifting the PreyBOTs since Alpha has a height advantage.

POWER AND WIRING

TOTAL DRAW FROM THE ROBOT

Each robot will be equipped with similar items shown below in the chart. These are the major electrical components of the robot.

Item	Voltage	Amperage	#Per robot	Total Amperage
Raspberry Pi 2	5	1.8A	1	1.8A
DFRobot 270 Degree Servo	6	100mA	13	1.3A
IMU Breakout	3	.40mA	1	.04A
HD Webcam	5	500mA	2	1A
Mini Maestro Servo Controller	5	4mA	1	.04A
Micro Servo	5	50mA	3	.15A
USB RFID Reader	5	30mA	1	.03A
Arduino uno	5	40mA	1	.04

The total amperage per PackBOT is 4.40A. Components with an unknown amperage nor the minor components not listed cannot be assumed negligible. To accommodate this, another 2 amps will be added to the total amperage; it will be a safe bet to assume that the max amperage is 6.40A. However, this may be off as all components will not work continuously. For example, the servos will not all work at the same time.

THE BATTERY

As said above, the assumed maximum amperage of the robots will be 6.40A. Thus, the batteries' minimum discharge rate needs to be 6.40A so the battery doesn't burn out if the robot happens to pull that many amperes. While none of the items require 7.4 volts, the battery will be 7.4 volts as it is more versatile and more accessible than lower voltage batteries, which tend to be Lead Acid batteries anyways. The ideal battery seems to be a 7.4V Lithium Polymer battery with 6600mAH of run time by energy. The battery provides the robot with 62 minutes of run time and has a sufficient discharge rate to fit our needs. Although the voltage is higher than what is required, it can be brought down using resistors and voltage regulators.

DISTRIBUTION

A distribution board will be wired directly to the battery and will work similarly to a panel. The main feed will be soldered to a common bussing that will be shared with all electrical components. The voltage will be 7.4V all the way through the wiring and distribution board and then will be stepped down to its proper voltage right before the electrical component. This will be the case with all electrical components except for the servos, which will share a separate distribution board altogether. The distribution board for the servos can also be considered the servo controller as it can turn the servos on and off as necessary. Before this distribution board, the voltage will be lowered to the common 6V, using a 6V voltage stepper that all servos share. This will be done for efficiency of both price and manufacturing time

CABLING

A protective sheath will be put over all wires to make sure no abrasive parts inside the robot cut open cables and cause a short circuit. Connections inside the body of the robot will be an easy point A to point B, but the difficulties arise when one must wire through the moving parts such as the legs and the head. The protective sheath will then work to relieve tension among the wires. Slack in the wire will also be required so that the wire does not prevent full movement of the robot and its servos. Simply put, extra wire will not be secured to the robot so that it can move without fear of cables being torn.

FABRICATION AND MATERIALS

MANUFACTURING

The overall frame of these robots will be made out of Aluminum with sheets of ABS plastic placed on the sides to allow them to move quicker. Both of these materials are good for the final product because they are lightweight and sturdy. The frame will be made out of Aluminum in order to provide more stability to the PackBOTs. ABS plastic will be used to keep the weight as low as possible. The weight needs to be low in order for the PackBOTs to carry the capture tools.

Material	Strength	Lightweight	Machinability	Price	Total
Aluminum	+	+	+	+	4
ABS Plastic	-	++	+	+	4

Total weight of PackBOTs without capture methods

Item	Weight
Raspberry Pi 2	0.045 kg
DFRobot 270 Servos (x13)	0.767 kg
Micro Servo (2x)	0.018 kg
IMU Breakout	0.0028 kg
HD Webcam (x2)	0.204 kg
Mini Maestro	0.003 kg
Arduino Uno	0.025 kg
SparkFun RFID USB Reader	0.00567 kg
Battery	0.311 kg
Wiring	0.03554 kg
Frame	0.4427 kg
Hook	0.00315 kg
Servo Mounts (x15)	0.612 kg
Jaw Parts	0 .2087 kg
Small Sheet (x2)	0.0344 kg
Long Sheet (x4)	0.1139 kg
Total	2.847 kg

PackBOT	Total Weight
DOG-1E5	4.446 kg
DOG-4S1	3.847 kg

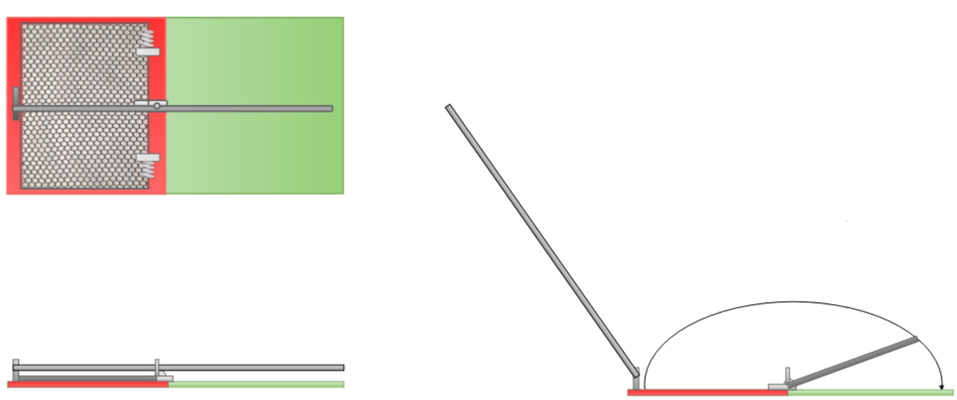
CAPTURE TOOLS

BOP!

MECHANICAL

The main method for immobilization of the PreyBOTs is to utilize thin, spring-activated nets, known as BOP!, to capture the PreyBOTs from below, completely surrounding them. Capturing the PreyBOTs this way will prevent the PreyBOTs from shaking off the net as well as prevent the net from falling off. For deployment, BOP! will detach from the PackBOTs rear end and lay waiting to capture PreyBOTs.

The net is essentially an enlarged mousetrap. The fly lever will have a mesh net attached to it so that instead of just flipping and missing the PreyBOTs, it will capture them within the net. It will utilize a similar trip mechanism to a mousetrap, however, the rod will be extended to increase the chances of a PreyBOT tripping it. Knocking the trip lever off of its resting place it will cause the trap to go off. The net on the fly lever will be neon green on the untripped side and red on the tripped side. This will allow PackBOTs to notice that the trap has gone off so they can go check the trap. From there, the PackBOTs will scan the RFID, and will either drag the PreyBOT away or leave it there depending on the results of the scan.



MANUFACTURING

The net will be made of nylon due to its strength and durability. The net will have a mesh size of 0.75 inches and mesh thickness of 1.32mm. The net will withstand 125 pounds of force before breaking. The wood structure will be made of 1/16 inch thick plywood because it offers resistance to cracks, breaks, and twisting. Plywood also resists high impacts, which is essential in BOP!, as the levers will be slamming into the wood at high speeds. Both the nylon net and plywood base are cost effective in proportion to their high strength.

Types of Net	Lightweight	Strength	Price	Total
Nylon	+	+	+	3
Wool	-	+	+	2
Polyester	+	-	+	2

Types of Wood	Lightweight	Strength	Price	Total
Plywood	+	+	+	3
Balsa Wood	+	-	+	2
Pine Wood	+	+	+	3

Item	Weight
Board	0.535kg
Steel Rod	0.151kg
Nylon Netting	0.0454kg
Springs	0.0227 kg
Miscellaneous	0.0227kg
Hinges	0.0454
Total	0.822kg

MOTHER GOOSE

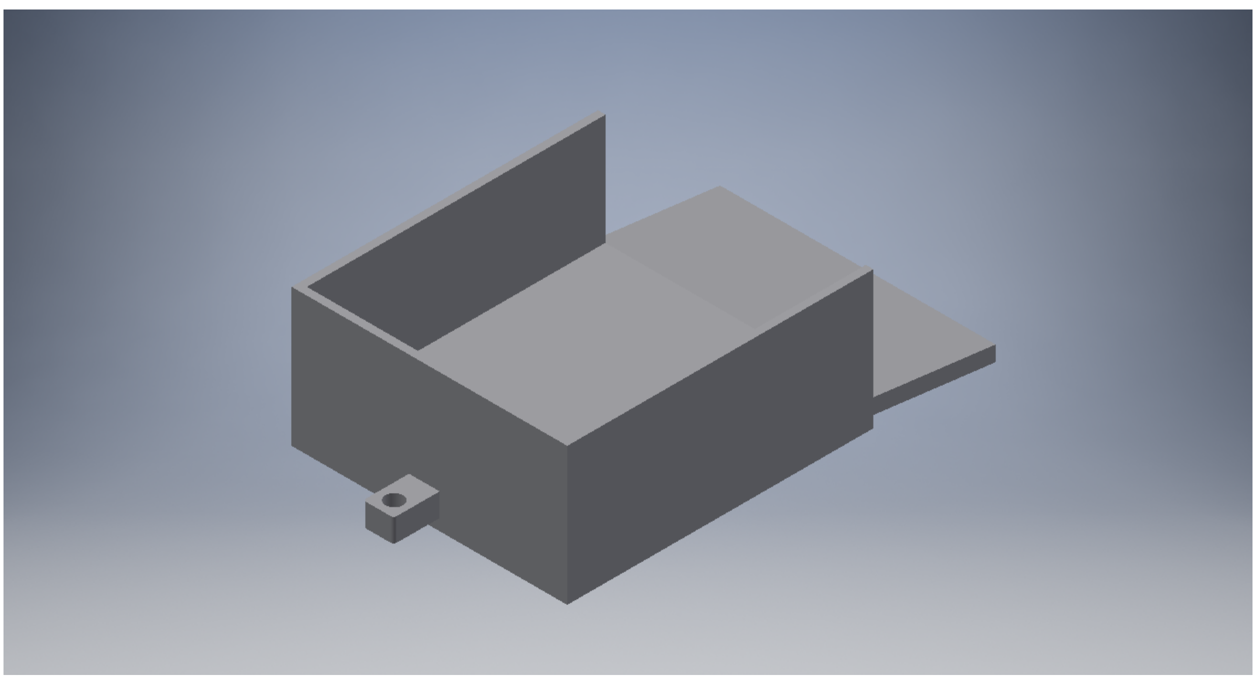
MECHANICAL

The PackBOTs, code-named *DOG-1E5* and *DOG-4S1,* will use a trailer system named Mother Goose to trap and transport PreyBOTs into the “Holding Cell”. Mother Goose will consist of 4 wheels, a ramp and a box made out of lightweight material. In the beginning of the mission it will be deployed with DOG-1E5. During the mission, the trailer can detach and reattach itself from the DOGs through a simple hook mechanism.

The walls of the container will raise to a sufficient height such that a PreyBOT will not be able to escape once it is contained by the trailer. At the back of the trailer, a servo-controlled ramp creates a smooth incline for the PreyBOTs to enter, either spontaneously or through the use of force. DOG-1E5 will place Mother Goose at one corner of the field to minimize alignment computations and the possibility of a misalignment due to unpredictable trailer motion during capture. Both DOGs will attempt to corral a selected PreyBOT into Mother Goose. Due to the low-sitting nature of the corral, the PreyBOT will not recognize it as an obstacle using conventional ultrasonic sensors. Consequently, it will move into the trailer.

As soon as Mother Goose, using an ultrasonic range finder, detects that something has entered, the trailer will spin the servo so that the ramp now acts as a barricade to prevent the PreyBOT from escaping. Upon closing, Mother Goose broadcasts a capture signal to command, which relays the signal to the DOGs. DOG-4S1 (or DOG-1E5 if the former is not available), will approach the trailer and scan the imprisoned PreyBOT. If it detects that the captured entity is not the desired one, the DOG broadcasts a signal to open the trailer and extracts the PreyBOT. This process repeats until the correct PreyBOT is captured.

Upon capturing the correct PreyBOT, DOG-1E5 will reattach to Mother Goose and position the trailer a specified distance from the designated drop-off box. A signal will then activate the ramp which falls down on the side of the drop-off box, acting as a slide for the PreyBOT. If the prey is alive, both DOGs will attempt to coerce the PreyBOT into moving up the ramp and into the capture box by itself. However, if that fails, one of the DOGs will use a predesigned push mechanism in the box to try and push the prey into the capture box. Finally, if both methods fail, Wolfgang_Alpha will be summoned to lift the box or the PreyBOT so that the PreyBOT ends in the capture box.



ELECTRICAL

Item	Voltage	Amperage	# of item	Total Amperage
DFRobot 270 Degree Servo	6	100mA	1	.1A
Raspberry Pi 2	5	1.8A	1	1.8A
Ultrasonic range finder	5	2mA	1	.02A

POWER

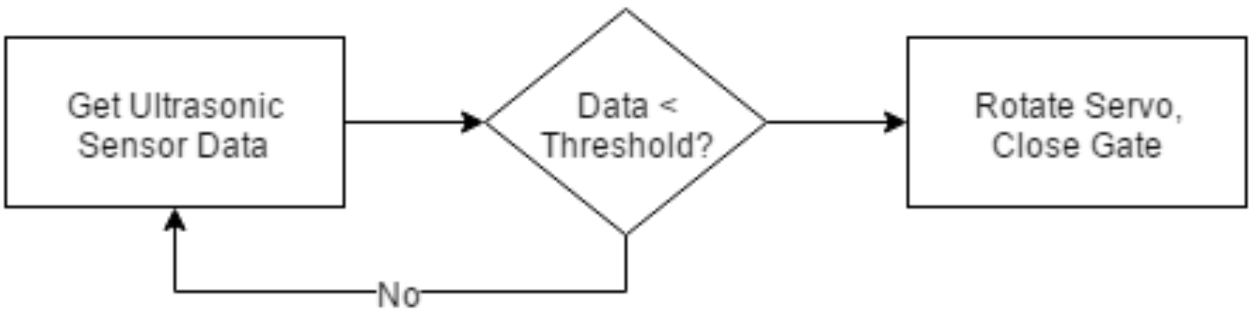
Mother Goose is independent from the PackBOTs and needs its own power supply. If everything ran continuously there would be a draw of 1.8A. Seemingly the most effective battery would be a 5V battery running with 2000mAH. The weight is so little it is practically insignificant and will allow just over an hour of runtime. The actual runtime should be more as the servo will not continuously run.

WIRING

A 3-port distribution board will be used to feed all components, and a 6V voltage stepper will be used to bring the power up to 6V for the servo. Communication wiring will go from the servo to the Raspberry Pi to the ultrasonic range finder and all will work in conjunction to close the gate when there is a PreyBOT inside Mother Goose.

PROGRAMMING - DIONYSUS MODULE

Mother Goose will be powered by a Raspberry Pi 2. The processor will constantly reap data from the ultrasonic range finder. If the distance detected by the ultrasonic range finder decreases at any point, the Raspberry Pi will signal for the servo to close the back. A signal will be broadcasted to all robots that a target has been captured. Mother Goose also obeys instructions from the PackBOTs. When a PackBOT signals for it to open the ramp, Mother Goose will execute the action.



MANUFACTURING

The container will be made out of ABS or High-Density Polyethylene (HDPE) plastic to keep the trailer lightweight. The alternative would be to use aluminum because of its strength but this will be used as a last resort since it will be heavier than plastic. The estimated weights for each material is stated below.

Item	Total Weight (kg)
Solid Aluminum	3.8
Shelled 0.5cm Aluminum	2.8
Shelled 0.5cm ABS	0.8

Plastic	Impact Strength	Lightweight	Machinability	Rigidity	Price	Total
ABS	+/-	+	+	++	+	5.5
HDPE	+/-	+	+	++	+	5.5
Polycarbonate	+	+	+	+	+	5

Item	Weight
Servo	0.045 kg
Wheel (x4)	0.028 kg
Ultrasonic Range Finder	0.005 kg
Raspberry pi	0.045 kg
Frame	0.976 kg
Eye Hook	0.0003402 kg
Total	1.099 kg

TESTING

SENSORS/ TRANSFORMATION

Alpha has a host of sensors to detect the proper time for transformation. The sensors do this by detecting ultraviolet light and specific sound frequencies.

The purpose of the ultraviolet (UV) light sensor is to inform the robot if it is inside the morphing chamber, allowing Alpha to morph into a 2-legged walker. The UV light sensor will be activated by the UV light inside the chamber. The Arduino Uno interprets the sensor's analog data, so that the robot can report to the command center. To determine whether or not the UV sensors are fully functional, tests will begin by placing a piece of metal in front of the sensor. Its reading should be 0 or under a given threshold. The UV light should be tested at different distances to see at what distances the sensor can detect UV light. Since the Sun emits UV light, testing outside can lead to inaccurate data. Therefore, we must test the UV light sensor outside as well, to make sure that the UV light from the sunlight and regular lights does not trigger Alpha to morph.

The purpose of the sound sensors is to register the sound of the G note in the song "Werewolves of London" to trigger Alpha to stand along with the UV Light sensor recognition. Sound sensors will be tested by playing notes of different frequencies at different distances to see if the sensor's detection is correct. Also, we will be testing the sound sensors under noisy conditions, such as a room full of people talking.

ASYNCHRONOUS DESIGN

The testing for asynchronous processing will consist of measuring how long it takes for asynchronous mode to process a given number of tasks and comparing it to the time taken for synchronous mode to process the same tasks. We will also see how efficient each mode is by loading them with varying amounts of tasks and seeing how much each mode slows down.

LOCOMOTION

Each leg will have 3 motors: 2 in the “hip” and 1 in the “knee.” Alpha will need to walk on 4 legs and on 2 legs, but the PackBOTs only walk on 4 legs. All robots will be tested to see how successful they are at mimicking the motion of a human or animal. The test will consist of measuring how strong the robots’ servos are by using a spring scale on each leg, and hooking the scale to the legs and then pulling the spring downwards. Once the motors begin to turn, it means that the robot's servos have reached their maximum strength. The second factor that must be tested is the strength of the legs. We will know the strength of the legs by placing different amounts of weight on top of the robot. Once the robot begins to slow down, it means it has reached its highest strength capacity.

Another test that must be done is for the movement path of the legs. To do this, a line will be drawn on a sheet of paper, and the leg will be configured programmatically to trace the line, and we will see how closely the robot can follow the line.

To test the timing sequence of the gaits, we need to measure the time it takes for the legs to descend, stroke, and return. The times measured should match with the theoretical times given in the gait diagrams.

2-legs

Testing the two-leg walker will involve placing the robot on different courses, such as flat surfaces, inclines, and steps, to calculate the efficiency of its walking and determine the stability of the walking mechanism. We will have the robot walk a certain distance to see if the tail and legs need to be adjusted to improve balance. The servo on the tail is a key factor of the robot's ability to stand and walk on two legs. The servo will be used to lift and release the robot's tail, making it possible for the robot to transform from a four-leg to a two-leg walker. This is essential to the two-leg walker because without balance, walking is impossible. Furthermore, we will test the robot's ability to recover from falls by actually making the robot fall with a hand push and then recording the robot's reaction. Since the mission is restricted to a 3-minute time period, testing the walking speed is necessary. The test will consist of having the 2-leg walker walk 10 feet for several trials and recording the time. If the robot walks too slowly, then we must adjust the rotations per minute in the servos.

VISION (WEBCAMS)

The vision test will involve determining the robot's ability to determine an object's distance from the robot and identifying distinguishable colors. For the test, we will present the robot with colors, preferably neon, with a series of different backgrounds to determine if the robot is able to identify the colors by publishing the image to the Zeus Model. We will also present the robot with similar colors, such as red and orange, to determine how accurate and precise the robot is at identifying and tracking the moving colors. Furthermore, the distance it takes the robot to identify the colors will be tested by presenting the colors at certain distances and recording the distances at which the robot was and wasn't able to identify the colors. To make the robot more accurate and precise, we will be testing the stereoscopic by analyzing how accurate the range finding software's data is compared to known distances

ELECTROMAGNET/STEEL BALL MANIPULATION:

The electromagnets will be tested by placing a steel ball at different distances to determine the range of the electromagnet. We will also be testing the holding strength of the magnets by placing non-metal material in between the one-inch steel ball and the magnet. Another test would be placing steel material with different masses and analyzing whether or not the electromagnet is able to carry it.

RFID SCANNER

Testing the RFID scanner will consist of presenting the sensor with the RFID tags at different distances to determine the range of the RFID scanner. Also, since we don't know what material the PreyBOTs will be made of, it is essential to test the RFID tags in different environments with different materials obstructing the sensor, such as steel, aluminum, nylon, and plastic. The materials will also be tested at different ranges and with different thicknesses.

CAPTURE METHODS: (NET, JAW, TRAILER)

Net

We will be testing the net trap by having a toy car pass through the trap and activate the trigger and analyzing the reaction time and force of the net closing. This will determine if the springs will have to be tightened or loosened to accommodate the PreyBOTs. This is important because it must close quickly and powerfully enough to capture a PreyBOT of any size or speed within the constraints without damaging the electronics.

Jaw

The purpose of the jaws is to secure the robots. The strength of the jaws must be tested to determine if the robot be able to complete its duty. We will place different sized weights on the floor. Then we will measure the time it takes for the robot to pick up the weight and how far it can drag the weight. Furthermore, we will be testing how far the jaw opens, both in degrees and centimeters, to determine the size range of the objects that the jaw can successfully grab.

Trailer

The trailer, also known as Mother Goose, will be tested by placing different materials inside the trailer to determine the sensitivity of the ultrasonic sensors. Also, the ability of objects to enter the trailer smoothly must be tested. This test will consist of having the robots carry a prototype PreyBOT and placing it inside the trailer. Once placed inside the trailer, we will need to observe the time it takes for the ramp to close and whether or not the ramp hits or impairs the PackBOTs.

The strength of the ramp when it is used as a wall blocking the PreyBOTs' escape must be tested. We will test the force of the ramp by hooking the end of a spring scale to the ramp and pulling on the other end. Once the servos begin to turn, it means that the ramp has reached its maximum capacity. Furthermore, the overall structure of the trailer, such as the location of the wheels, the dimensions, and the materials, should be tested to see if it is capable of containing and transporting PreyBOTs. We will be testing all of this by using prototypes to give us an idea of what will happen when the PreyBOTs are loaded into the trailer.

COMMAND CENTER-ZEUS MODEL/COMMUNICATION

The PackBOTs and the trailer must be able to communicate with each other during the mission course through a system called the Zeus module. To test this, Zeus, PackBOTs, and Mother Goose will all ping each other with different information types such as strings and binary data. We will also see if the ping delay increases if Zeus or any other robots ping each other at the same time and if any ping packets are lost when this happens.

MICROCONTROLLERS - CEREBRAL MODULE

Microcontrollers are essential and therefore each microcontroller, such as the Raspberry Pi 2 Model B, Arduino Uno, and Pololu Mini Maestro 18-Channel Servo Controller, will be tested.

The Raspberry Pi 2 Model B will be tested to see how much CPU it has remaining under heavy load. Interfacing between Raspberry Pi 2 and the Arduino Uno and Pololu Mini Maestro is tested by sending ping packets to both devices from the Raspberry Pi and obtaining responses to ensure that USB communication is functional. Communication between the Raspberry Pi 2 and the command center is tested by benchmarking transmission rates for video frames and sensor data.

The robots will contain many sensors in order to accomplish their tasks, and the Arduino Uno will be recording the data from the sensors. Thus, the Arduino Uno will be tested to see how well it can interface with sensors. The Arduino should report accurate data from both the IMU and the UV Sensor.

The Mini Maestros will control the servos, execute scripts, compute profiles, report location, and configure PWM. Controlling the servos consist of a Raspberry Pi 2 command that will be tested. For the Mini Maestro, communication is key to its function. Testing the communication between the microcontrollers and determining whether or not the Maestro is doing its job is important.

INVERSE KINEMATICS - FINESSE MODULE

Inverse kinematics can be tested through simulation and physical tests. Simulations involve placing a joint (with constraints) in 3D in space and directing the module to find a solution. The module will be benchmarked to see if it can find a solution for the joint in a given time (under 200 microseconds). Once a physical model is available, the test involves directing the leg of the robot to move to certain locations under constraints and ensuring that the module is error free when it is called by other modules.

RESTRICTIONS

- Weight constraints: Wolfgang (8 kg), PackBOTs (4.5 kg)
- Height constraints: Wolfgang (30 cm), PackBOTs (20 cm)
- All height and weight constraints must be met.

ASSEMBLY

All of the robots must be assembled according to specifications and all parts must be in their appropriate places. Tolerances will also be measured, and parts will be deemed defective or not by seeing if their dimensions fall within constraints. There should be enough space to add all features into the robots. In addition, the servos must be placed in the correct orientation and their positions must be set by programming before assembly to ensure that the correct range of freedom can be achieved. Testing the servos involves moving a servo to its two extremes and ensuring that it can reach the desired angles.

RESEARCH

MARKET SURVEY

In order to get a broader sense of thought on designs, Eclipse Technologies surveyed 100 professional engineers and college engineering students. Their responses and comments were taken into careful consideration throughout the team's design process. Their input has helped shaped ideas that are incorporated onto all three robots in the Lycanthrope pack to better the product for potential consumers.

1. One objective of Wolfgang is to be able to intake a one-inch steel ball, lift it, and deposit it within a container. Which of the following solutions do you believe is best suited to perform this task?

Option	Votes (out of 100)
Electromagnet	63
Claw	21
Front arm extension	18
Conveyor Belt from foot	4
Conveyor belt from tail	1

2. One key consideration in the construction of Wolfgang is the body design. Which of the following solutions do you believe is best suited to perform this task?

Option	Votes (out of 100)
Pengolin	89
Human	11

3. Wolfgang has to have accurate and precise movements. Which of the following solutions do you believe is best suited to perform this task?

Option	Votes (out of 100)
2 DOF	3
3 DOF (rotation)	37
3 DOF (translation)	19
4 DOF	24
6 DOF	1
7 DOF	11

4. After Wolfgang has deposited a steel ball into a specific container, the gates protecting the target “PreyBOT” will open. Once this is complete, two “PackBOTs” will attempt to capture the target PreyBOT. Which of the following solutions do you believe is best suited to perform this task?

Option	Votes (out of 100)
Box	16
Harpoon	10
Net	40
Jaws	35
Forklift	0
Catapult	0

5. Once the PreyBOT has been successfully captured, it will need to be retrieved and placed in a holding area. Which of the following solutions do you believe is best suited to perform this task?

Option	Votes (out of 100)
Jaw	45
Transferrable cage	20
Nets	30

6. A fundamental idea in the design of Wolfgang is balance. To accomplish this, the use of a tail is recommended. Which of the following tail designs is best suited to perform this task?

Option	Votes (out of 100)
Flexible	30
Segmented	50
One-Joint	15

PATENT SEARCHES

ROBOTIC WALKING LEGS

Leg structure for walking robot

US 5484031 A

A leg structure for a walking robot that contains multiple points of input and outputs. This allows for a larger range of motion. This leaves a 180° boundary between the leg and arm operating states and the driving mode. It changes from low speed and high torque to high speed and low torque.

ELECTROMAGNET

Combined permanent magnet and electromagnet

US 3089064 A

Combined permanent magnet and electromagnet is the arrangement of a pair of magnets that are placed in a pattern, having one pair of magnets as a closed magnetic circuit. An electromagnet will be embedded in Wolfgang's foot to pick up and transfer the metal ball into the designated container.

AUTONOMOUS WALKING ROBOT

Method for planning path for autonomous walking humanoid robot

US 9069354 B2

A method for planning the path of an autonomous, walking, humanoid robot. The robot takes steps while being aware of the environment around it, initializing path information.

Wolfgang must morph from a four legged walking robot to a two legged walking robot. Wolfgang may be modeled after a bipedal robot that walks like a human.

AUTONOMOUS HUNTING CAPABILITIES

Autonomous coverage robot

US 8972061 B2

Autonomous coverage robot is a robot that contains a drive system designed to maneuver floor surfaces. This robot contains a cleaning system with a sensor. Wolfgang and PackBOTs must be able to autonomously maneuver and communicate with each other to perform required tasks just as the autonomous coverage robot must autonomously maneuver to complete its task.

VISION PROCESSING (CAMERA)

Video camera

US 8358357 B2

A video camera that is designed to capture, compress, and store data at a rate of 23 frames per second. Wolfgang will be outfitted with a camera to provide a live video feed of the mission.

MOUNT FOR CAMERA

Mount system for attaching camera to a sport board

US 8014656 B2

The attachment of a camera to a sports board via a mount. The camera is attached to the mount and the mount can be attached to an object via an attachment device like a suction plug. Camera positioning can be located at front or back of board with camera facing either direction. Camera mount angle can be changed to a variety of angles. A camera must be mounted to Wolfgang to provide a live video feed of mission.

TRANSFORMING ROBOTS

Rescue robot for polymorphic real-time information transmission

CN 203046783 U

A robot containing a wheel-foot conversion mechanism, a flying device, a real-time image collection and transmission device, and a control system. This robot will be used in various situations, such as earthquakes, coal mines, terrorist bombing. It is built for agility, has significant mobility, and is convenient and practical.

SENSORS

UV sensor

UV detection devices and methods

US 8044363 B2

A portable UV detection system. Able to detect UV light and will be integrated into Wolfgang's design.

IMU (ACCELEROMETER, GYROSCOPE)

Integrated GPS/IMU method and microsystem thereof

US 6480152 B2

An integrated Global Positioning System (GPS)/Inertial Measurement Unit (IMU) method and MicroSystem. Measures and reports a craft's velocity, orientation, and gravitational forces, using a combination of accelerometers and gyroscopes. An IMU will be integrated into the robots so that they are able to walk without falling over.

AUDIO FREQUENCY SENSOR

Electronic sensing and measuring apparatus for signals in audio frequency range

US 4144490 A

An electronic sensing and measuring system designed for measuring the pitch of various sound frequencies. A sound sensor will be integrated into Wolfgang so that the robot can distinguish the howl in Warren Zevon's song, "Werewolves of London," signaling Wolfgang to transform from a four-legged walker into a two-legged walker.

SEGMENTED TAIL SYSTEM

Amusement device

US 6672934 B2

A toy with an animal shaped body and remote-controlled parts. The toy, shaped like a lizard, has a segmented tail that can curl up and flatten out. This same segmented tail system will be implemented in Wolfgang to act as a counterbalance when the robot morphs from a four-legged walking robot to a two-legged walking robot.

HUNTING NET TRAP SYSTEM

Safety disposable mouse trap

US 20060064922 A1

An old-fashioned spring-loaded mousetrap. Normally equipped for exterminating unwanted rodents, the mechanism by which the mouse trap captures rodents will be modeled into the PackBOTs' arsenal to capture the PreyBOTs.

Autonomous navigation system for a mobile robot or manipulator

US 5758298 A

A navigation system for a mobile robot or manipulator that is supposed to guide the robot through a designated area without fully identifying obstacles or boundaries. This autonomous navigation capability will need to be integrated into both Wolfgang and the PackBOTs to navigate the mission field

FINANCIAL REPORT

BUSINESS CASE

Eclipse Technologies is a defense contractor. Entering the competitive private defense technology sector, its main source of revenue will rely heavily on government contracts awarded to corporations that successfully carry out government-proposed challenges. To be a competitive player in this industry, Eclipse will position itself as a deliverer of advanced, autonomous, robotic technologies. It is essential that Eclipse consistently delivers cutting-edge autonomous technologies to keep pace with the high demand of the military.

To bring advanced technologies to the armed forces, Eclipse will execute the following:

1. IDENTIFY MILITARY NEEDS

Eclipse will use the challenges and requests provided by the federal government as guidelines to deliver cutting-edge technology. The technology that Eclipse delivers will be made to the specifications of the government.

2. INNOVATE AND OVER-DELIVER

At Eclipse, engineers and designers are challenged to create technology that has never existed before. The Eclipse team will be in constant communication with government agencies and authorities to improve upon the specifications already given by the government. To be successful, it is important that the products that Eclipse produces not only meet, but exceed the expectations of its clientele.

3. FUNDRAISE

As a start-up company, it is essential that Eclipse obtain sufficient funding required to bring a high-quality product to its clients. Eclipse has two major components of its fundraising plan. First, Eclipse will partner with other companies and solicit funding from these companies. Also, Eclipse will be in close contact with companies established in the defense technology and engineering sectors to gain insight into the market. Second, Eclipse will approach investors to fund the company. Investors will only legitimize the work being done at Eclipse and attract more attention to the company. Eclipse hopes to become a nationally recognized brand of advanced technologies.

4. MANUFACTURE AND DISTRIBUTE

At its core, it is necessary for Eclipse to ship its technologies. It will find way to quickly manufacture products that do not compromise the quality of its products. Since a very large percentage of the technology created at Eclipse will be for defense agencies, Eclipse has a two-part approach to manufacturing and distribution. Part one entails Eclipse developing viable technologies that seamlessly fit the requests of defense agencies. During the process, Eclipse will work to streamline manufacturing processes. Once the technology satisfies the desires of the clients, Eclipse will take purchase orders for the exact number of products wanted by these agencies. This way, there is no excess inventory and funds are not carelessly spent.

5. REVENUE

An extremely large percentage of Eclipse's revenue will come from contracts awarded by government defense agencies. However, Eclipse will explore additional application of its technologies to generate multiple streams of revenue.

6. EXPLORE COMMERCIAL APPLICATION

Commercial application may be a very profitable source of revenue for Eclipse. After Eclipse technology is created for authorities, all applications of the technology will be considered. Given the current mission to create a polymorphic search and destroy/rescue robot, there are a few obvious applications the technology could be used for. Law enforcement officers can use the Lycanthrope Pack robots to conduct property searches for suspected criminals to minimize the amount of risk to law enforcement officers. Additionally, these robots can be sent out during natural disasters to aid in the relief campaigns that follow these drastic events.

7. MARKETING CAMPAIGN

Branding and marketing are very important to the success of Eclipse Technologies. It is imperative that Eclipse brands itself as a leader in autonomous robot innovation. Eclipse will brand itself starting with the "why" Eclipse exists. Eclipse believes that people who are passionate and work together can drive change and surpass limitations. Following this fundamental belief, Eclipse will market its technologies to the public to gain notoriety as one of the most revolutionary robotics companies in the world. To drive this marketing campaign, Eclipse will employ a combination of traditional and guerilla marketing tactics, such as television ads and social media campaigns, to keep fans of Eclipse technology engaged.

8. PR CAMPAIGN

Eclipse will look to maintain its pure, untainted image by keeping good relations with the public. Eclipse will conduct its business and work with the utmost integrity and uphold its ethics at all costs. If an issue arises concerning these matters, Eclipse will work cooperatively with outside parties to resolve these issues. Eclipse technologies is planning an aggressive press campaign in the next several months, targeting local media outlets like the Daily Breeze and Long Beach Press Telegram. As the project continues, Eclipse will be actively seeking out more nationally recognized media outlets like TechCrunch, Forbes, Bloomberg, and the Wall Street Journal. The more press Eclipse obtains, the more legitimate the company appears to the public, which garners respect from others in the defense industry and from government agencies looking for companies to award their contracts to.

BUDGET

MISCELLANEOUS

Vendor	Item Number	Description	QTY	Unit	Unit Price	Total Price
TheBookFactory	1	Engineering Notebooks Red Cover with Imprinting	5	1	$23.00	$115.00
Staples	2	Custom Banner 6'x2.5'+ Ourdoor Vinyl+ Gromnets	1	1	$80.00	$80.00
Fedex	3	Printing Services Book Manuels	2	1	$80.00	$160.00
Amazon	4	Acrylic Red Plexiglass Plastic Sheets 1/8" 12x12	1	1	$10.00	$10.00
OfficeDepot	5	Avery Pin Style White Name Tags with Flexible holder	2	48	$5.99	$12.00
Staples	6	Printing Services Brochers	50	1	$0.15	$7.50
Food4Less	7	Food and Drinks	100	1	$1.00	$100
CustomInk	8	Paraphenalia	23	1	$550.00	$1,650.00
VistaPrint	9	Business Cards	200	1	$2.00	$400.00
						$2,534.50

TOTAL SPENDING

Miscellaneous	**$2,534.50**
Lycanthrope	$3,517.45
Shipping Cost	$550.00
Total	**$6,051.95**

BILL OF MATERIALS

ELECTRICAL

Vendor	Item Number	Description	Part Number	QTY	Unit	Unit Price	Total Price
Battery Space	1	Battery 5~10ah, 7.4 V	18650	3	1	$64.95	$120
Amazon	2	Distribution Board	QAV250	3	1	$20	$60
Amazon	3	Electromagnet	B000701BVK	1	1	$15	$15
Amazon	4	Wire Coils - 16 Gauge	B008XKU0VS	2	1	$6	$12
Amazon	5	Wire Coils - 18 Gauge	B00XKHS3QS	1	1	$6	$6
Amazon	6	16 Gauge Fuse Holder	0400ATCFH16-5	3	1	$7	$21
Amazon	7	Rat Trap	M201	25	1	$3	$75
Amazon	8	Nylon netting	KNN154	5	1	$10.00	$50
Amazon	9	Neodymium Magnets	M1414CYLSmCo	10	1	$11.38	$114
Home Depot	10	1/16 inch plywood	161640	10	1	$3.00	$30
Amazon	11	Steel Ball Bearings	B004PX9KO0	40	1	$0.00	$0.00
BatteryJunction	12	Tenergy 18650 6600mAh 7.4VIon Battery	TENERGY-74-6600-PCB	1	1	$59.80	$59.80
Home Depot	13	5ft Fiberglass rod Rod	203153095	6	1	$4	$24
Lowes	14	Brass Hinges	4222351	8	1	$2	$16
SparkFun	15	1/4 Watt Resistors 20 pack	10969	2	1	$9	$18
SparkFun	16	5v Li PO battery 2000mAH	8483	1	1	$10	$10

Vendor	Item Number	Description	Part Number	QTY	Unit	Unit Price	Total Price
Amazon	17	5V Voltage regulator	s7v7f5	10	1	$5	$50
Amazon	18	6V voltage regulator	s18v20f6	3	1	$5	$15
Amazon	19	Ac/Dc converter	Adams rite 4603	3	1	$11	$33
Amazon	20	Stepup voltage regulator 12v	u3v12f12	3	1	$4	$12
Amazon	21	3.3V voltage regulator	s7v8f3	5	1	$6	$30
							$770.60

MANUFACTURING

Vendor	Item Number	Description	Part Number	QTY	Unit	Unit Price	Total Price
McMaster-Carr	1	Aluminum Shim Stock or Sheets .005 .007 .008	89015K173	1	1	$25	$25
APW Company	2	Round electromagnet	EM075-6-122	2	1	$30	$60
McMaster-Carr	3	Steel Plates	6544K22	1	1	$20	$20
McMaster-Carr	4	Titanium Sheets or Shim	9502K51	1	1	$100	$100
McMaster-Carr	5	ABS plastic SHEETS	8712K204	1	1	$15	$15
McMaster-Carr	6	HDPE SHEETS PLASTIC	8671K39	1	1	$10	$10
McMaster-Carr	7	Polycarbonate sheets	8574K31	1	1	$30	$30
Gourock	8	Nylon Netting	KNN154	1	1	$15	$15
							$275.00

PROGRAMMING

Vendor	Item Number	Description	Part Number	QTY	Unit	Unit Price	Total Price
element14	1	Raspberry Pi 2, Model B, 1 GB + MicroSD	RASPBERRYPI-2-MODB-1GB	4	1	$40	$160
Arduino Store	2	Arduino Uno, Rev 3	a000066	4	1	$24.95	$99.80
Amazon	4	Logitech HD Webcam C270 (w/ microphone)	960-000694	6	1	$21.99	$131.94
Amazon	5	Wifi Dongle Adapter	KT-RPWF	4	1	$7.99	$31.96
Best Buy	7	GoPro - HERO HD Waterproof Action Camera	N/A	1	1	$129.99	$129.99
SparkFun	9	SparkFun RFID USB Reader	9963	4	1	$24.95	$99.80
SparkFun	10	SparkFun UV Sensor Breakout - ML8511	ML8511	2	1	$12.95	$25.90
Adafruit	11	Adafruit 10-DOF IMU Breakout	1604	4	3	$29.95	$119.80
Amazon	13	SainSmart HC-SR04	B004U8TOE6	1	1	$9.98	$9.98
SparkFun	14	SparkFun RFID USB Reader	9963	3	1	$24.95	$74.85
Amazon	15	AmazonBasics 4-Port USB 2.0 Ultra-Mini Hub	HU2K44N3	3	1	$6.99	$20.97
							$904.99

MECHANICAL

Vendor	Item Number	Description	Part Number	QTY	Unit	Unit Price	Total Price
SparkFun	8	SparkFun 9 Degrees of Freedom Breakout - MPU-9150	11486	4	1	$34.95	$139.80
Pololu	6	Mini Maestro 18-Channel USB Servo Controller	1354	4	1	$39.95	$159.80
DFRobot	3	DFRobot 270 Degree 15KG Metal Geared Standard Servo	Standard servos	42	1	$20.48	$860.16
AdaFruit	12	Sub-micro Servo - SG51R	2201	18	1	$5.95	$107.10
							$1,266.86

LYCANTHROPE

Electrical	**$770.60**
Manufacturing	$275.00
Programming	$904.99
Mechanical	$1,266.86
Shipping	$300.00
Total	**$3,517.45**

INDEX

ACRONYMS

CAMS – California Academy of Mathematics and Science

STEM – Science Technology Engineering and Mathematics

PM – Project Manager

DPM – Deputy Project Manager

CARPA – CAMS Advanced Research Projects Agency

S&C – Search and Capture

DARS – Distributed Autonomous Robotic System

UV – Ultraviolet

RFID – Radio Frequency Identification

CDR – Critical Design Review

SPS – Systems Performance Specifications

POC – Proof of Concept

TDP – Technical Data Package

IPR – In Process Reviews

PDR – Periodic Design Reviews

WBS – Work Breakdown Schedule

DOF – Degrees of Freedom

Wi-Fi - Wireless Local Area Network

WAMP - Web Application Messaging Protocol

TCP - Transmission Control Protocol

DOF - Degrees of Freedom

Root - The first entity in a series of entities (in Kinematics)

End Effector - The final or resultant entity in a series of entities (in Kinematics)

IK - Inverse Kinematics

FABRIK - Forward and Backward Reaching Inverse Kinematics

FPS - Frames per second

JIT - Just in Time

WORKS CITED

"17.1. Threading — Thread-based Parallelism¶." *17.1. Threading*. Python Software Foundation, 31 Aug. 2015. Web. 20 Oct. 2015.

"7.4v Lithium Ion Battery Packs." *7.4v Lithium Ion Battery Packs*. Mega Batteries, n.d. Web. 20 Oct. 2015.

Ackerman, Evan. "Spot Is Boston Dynamics' Nimble New Quadruped Robot." *IEEE Spectrum*. IEEE Spectrum, 10 Feb. 2013. Web. 22 Oct. 2015.

Ackerman, Evan. "Video Friday: Quadcopter With Quadruped, Magnetic Microbots, and Baxter Flips Pancakes." *IEEE Spectrum*. IEEE Spectrum, 18 Sept. 2015. Web. 24 Sept. 2015.

"Adafruit 10-DOF IMU Breakout - L3GD20H + LSM303 + BMP180." *Adafruit*. Adafruit, n.d. Web. 22 Sept. 2015.

"Adafruit's Raspberry Pi Lesson 8. Using a Servo Motor." *Adafruit*. N.p., 09 Jan. 2009. Web. 22 Sept. 2015.

"All-Battery.com: AT: Tenergy Li-Ion 18650 7.4V 6600mAh Rechargeable Battery Module W/ PCB Protection and Bare Leads." *All-Battery.com*. N.p., n.d. Web. 15 Oct. 2015.

"All-Battery.com: AT:Tenergy Li-Ion 18650 7.4V 8800mAh PCB Protected Rechargeable Battery Module W/ Bare Leads." *All-Battery.com*. N.p., n.d. Web. 15 Oct. 2015.

"ArduinoBoardUno." *ArduinoBoardUno*. Arduino, 2015. Web. 20 Oct. 2015.

Armstrong, Shain. "Improve Your RFID Read Range with These Six Factors." *RFID Insider*. AtlasRFIDstore, 01 July 2013. Web. 29 Sept. 2015.

"Audio." *Python*. N.p., n.d. Web. 22 Sept. 2015.

Bodwell, Donald J. "Team Norms." *Team Norms*. Donald J. Bodwell, 9 May 2014. Web. 29 Sept. 2015.

"Bolas - Forgotten Hunting Tool." *YouTube*. YouTube, n.d. Web. 01 Oct. 2015.

Bradley, Arthur. "Build Your Own Faraday Cage. Here's How." *TheSurvivalistBlognet*. M.D. Creekmore and TheSurvivalistBlog.net, 07 June 2014. Web. 29 Sept. 2015.

Calin, Dragos. "Guide To Choose Suitable Battery To Build A Robot | Into Robotics." *Into Robotics*. Into Robotics, 20 June 2013. Web. 29 Oct. 2015.

Callin, Dragons. "How to Detect and Track Object With OpenCV | Into Robotics." *Into Robotics*. N.p., 01 Apr. 2013. Web. 20 Oct. 2015.

"Can I Use the GPIO for Pulse Width Modulation (PWM)?" *Raspberry Pi Stack Exchange*. N.p., n.d. Web. 22 Sept. 2015.

"Cast Net Materials." *Brunsonnet: , Netting Supplies*. N.p., n.d. Web. 01 Oct. 2015.

"Chrome Steel Balls (Ball Bearings)." *SgtKnots.com*. N.p., n.d. Web. 06 Oct. 2015.

"Color Detection & Object Tracking." *OpenCV Tutorial C++*. N.p., n.d. Web. 22 Sept. 2015.

"Controlling Servo Motors." *O'Reilly*. N.p., n.d. Web. 22 Sept. 2015.

"Corrective and Disciplinary Action." *Washington State University*. N.p., n.d. Web. 01 Oct. 2015.

"Detect & Record Audio in Python." *Stack Overflow*. N.p., n.d. Web. 22 Sept. 2015.

Falconer, Jason. "Designing a More Human-Like Lower Leg for Biped Robots." *IEEE Spectrum*. N.p., 30 May 2013. Web. 22 Sept. 2015.

"Feature Detection and Description¶." *OpenCV*. N.p., n.d. Web. 22 Sept. 2015.

"Formula for Frequency Table." *Physics of Music*. N.p., n.d. Web. 22 Sept. 2015.

"Frequency Detection Using the FFT (aka Pitch Tracking) With Source Code." *BJORG*. N.p., 22 July 2012. Web. 22 Sept. 2015.

Fur, Umber. "Arduino-Controlled Wolf Tail." *Instructables.com*. Instructables, n.d. Web. 20 Oct. 2015.

"Gait." *Wikipedia*. Wikimedia Foundation, n.d. Web. 22 Sept. 2015.

"GE: Our Company: Leadership, History, Culture, Advertising." *General Electric : Our Company : What Is Six Sigma?* General Electric Company, n.d. Web. 22 Oct. 2015.

Graves, Jada A. "The Best Team-Building Exercises." *U.S News*. N.p., 24 Sept. 2014. Web. 08 Oct. 2015.

"Groups v Teams." *Groups v Teams*. Leadership-development-coaching, n.d. Web. 29 Sept. 2015.

Guizzo, Erico, and Evan Ackerman. "How South Korea's DRC-HUBO Robot Won the DARPA Robotics Challenge." *IEEE Spectrum*. N.p., 09 June 2015. Web. 22 Sept. 2015.

Hackerspace Adelaide. "Dynamixel XL-320 Servos." *YouTube*. YouTube, 4 Dec. 2012. Web. 20 Oct. 2015.

Haughey, Duncan. "Planning a Project Using a Work Breakdown Structure and Logic Network." *Project Smart*. N.p., n.d. Web. 22 Sept. 2015.

HerkuleX DRS-0101/ DRS-0201 User Manual. 20 Aug. 2011. Manual. 3166 N Lincoln Ave #218, Chicago.

"HERO." *GoPro*. GoPro Inc., n.d. Web. 24 Sept. 2015.

Hil. "A Tentacle Mechanism." *Spirits Dancing*. Wordpress, 21 Feb. 2005. Web. 22 Oct. 2015.

Hooper, Rich. "Robot Inverse Kinematics." *Learn about*. Learn About Robots, 18 July 2014. Web. 22 Sept. 2015.

"How Do Magnets Work?" *Cool Magnet Man*. N.p., n.d. Web. 22 Sept. 2015.

"How Subscriptions Work." Crossbar.io, 2 Oct. 2015. Web. 22 Sept. 2015.

"How to Build Positive Relationships Among Your Team Members." *Small Business*. N.p., n.d. Web. 08 Oct. 2015.

"How to Improve Your Relationships." *WikiHow*. N.p., n.d. Web. 08 Oct. 2015.

"ISensor MEMS Inertial Measurement Units." *Analog Devices*. Analog Devices, Inc., n.d. Web. 24 Sept. 2015.

Jacowski, Tony. "Creating a Work Breakdown Structure (WBS)." *Project Smart*. N.p., n.d. Web. 22 Sept. 2015.

Jagatia. "Maximum Number of Servos?" *Maximum Number of Servos?* Arduino, 5 Feb. 2012. Web. 22 Sept. 2015.

Jdogrich1. "Quadruped Robot Walking." *YouTube*. YouTube, 20 May 2010. Web. 22 Sept. 2015.

JimRD. "Gyroscopic Precession Robot (Version 2)." *Instructables.com*. Autodesk, Inc., n.d. Web. 24 Sept. 2015.

Johnson, Chad. "Send and Receive Binary Data over Web Sockets in Javascript?" - *Stack Overflow*. Stack Exchange Inc, 23 Apr. 2011. Web. 20 Oct. 2015.

Johnson, Sophie. "Punishment In Organizations." *Business & Entrepreneurship*. N.p., n.d. Web. 01 Oct. 2015.

Kaifi, Ahmad. "Multiple Object Detection with Color Using OpenCV." *YouTube*. N.p., 01 Dec. 2014. Web. 22 Sept. 2015.

Katie. "Plastics – a Big List of the Different Types of Plastic." *A Guide to Living Plasticless*. N.p., 19 June 2012. Web. 29 Sept. 2015.

Katzenbach, Jon R., and Douglas K. Smith. "The Discipline of Teams." *Harvard Business Review*. N.p., 01 July 2005. Web. 20 Oct. 2015.

Kim, Sangbae. "MechE Connects: Talking Shop with Assistant Professor Sangbae Kim." *MechE Connects Talking Shop with Assistant Professor Sangbae Kim Comments*. Massachusetts Institute of Technology, n.d. Web. 22 Sept. 2015.

Lassiter, David. "The Importance of Emotionally Intelligent Teams." *Executive Coaching*. David Lassiter, n.d. Web. 24 Sept. 2015.

"A Library for Audio Labelling." *Aubio*. N.p., n.d. Web. 22 Sept. 2015.

"Light Alloys and Metals Information." *IHS Engineering 360*. IHS, n.d. Web. 29 Sept. 2015.

"Locomotion Skills for Simulated Quadrupeds." *YouTube*. Mvandepanne, 09 May 2011. Web. 20 Oct. 2015.

Logitech. "Robot Check." *Robot Check*. Amazon, n.d. Web. 22 Oct. 2015.

"Long-Tailed Pangolin: The Animal Files." *Long-Tailed Pangolin: The Animal Files*. Theanimalfiles, 1 Aug. 2014. Web. 22 Sept. 2015.

Lundh, Fredrik. "Thread Synchronization Mechanisms in Python." *Effbot*. N.p., July 2007. Web. 22 Sept. 2015.

MacCarthy, Eugene M. "9.3: Evolution of Pangolins and Armadillos." *Macroevolution.net*. N.p., n.d. Web. 22 Sept. 2015.

"Magnets." *K&J Magnetics Products*. N.p., n.d. Web. 06 Oct. 2015.

"Master of Business Administration - Resource Articles." *Our Lady of the Lake University*. N.p., n.d. Web. 08 Oct. 2015.

"Materials Chart." *Spiratex*. The Spiratex Company, n.d. Web. 29 Sept. 2015.

"Materials for Robot Building: An Introduction." *Robotics Universe*. N.p., n.d. Web. 29 Sept. 2015.

Matthijs. "Reducing Voltage with Resistors." - *Electrical Engineering Stack Exchange*. Stack Exchange Inc, n.d. Web. 22 Sept. 2015.

McCarthy. "9.3: Evolution of Pangolins and Armadillos." *9.3: Evolution of Pangolins and Armadillos*. N.p., n.d. Web. 22 Oct. 2015.

Miller, Ben. "How to Build a Laser Tripwire With Arduino - Tuts+ Computer Skills Tutorial." *Computer Skills Tuts+*. Envato Pty Ltd., 27 June 2014. Web. 22 Oct. 2015.

Mortensen, Peter, and Skyler. "Why Does Water Damage Low-voltage Microelectronics so Easily?" - *Electrical Engineering Stack Exchange*. Stack Exchange Inc, n.d. Web. 22 Oct. 2015.

"Music21: A Toolkit for Computer-Aided Musicology." *Music21*. M.I.T, n.d. Web. 22 Sept. 2015.

NatGeoWild. "World's Weirdest - Pangolin." *YouTube*. YouTube, 6 Dec. 2012. Web. 22 Sept. 2015.

Naveenagrawal, and Swagatam. "Types of Kinematic Joints in the Design of Machines." *Brighthub Engineering*. Bright Hub Inc., 28 July 2009. Web. 22 Oct. 2015.

"Next-generation Application Messaging." Crossbar.io, 2 Oct. 2015. Web. 22 Sept. 2015.

NowWatchThisVideo's Channel. "Quadruped Robot, Arduino Walker (controlled with Wii Nunchuk) 2." Youtube, 22 Apr. 2011. Web. 22 Sept. 2015.

"Numba¶." *Numba — Numba*. Continuum Analytics, 14 Sept. 2015. Web. 20 Oct. 2015.

"OpenRAVE | Ikfast Module | OpenRAVE Documentation." *OpenRAVE*. OpenRAVE Documentation, n.d. Web. 22 Sept. 2015.

"ORB (Oriented FAST and Rotated BRIEF)¶." *OpenCV*. N.p., n.d. Web. 22 Sept. 2015.

"Pangolin Facts." *Pangolin Facts*. Softschools, n.d. Web. 22 Sept. 2015.

Pchan. "Raspberry Pi 2 Model B GPIO 40 Pin Block Pinout." *Raspberry Pi 2 Model B GPIO 40 Pin Block Pinout Version History*. Element14, 28 Jan. 2015. Web. 22 Sept. 2015.

Pitan, Dustin. "How to Strengthen a Permanent Magnet." *EHow*. Demand Media, n.d. Web. 22 Sept. 2015.

"Pitch Detection Algorithm." *Wikipedia*. N.p., n.d. Web. 22 Sept. 2015.

"Plastics as Strong as Steel." *Materials Today*. Elsevier Ltd., 11 June 2012. Web. 29 Sept. 2015.

"PortAudio V19 Python Bindings." *PyAudio*. N.p., n.d. Web. 22 Sept. 2015.

"Pressure Sensors." *Mouser Electronics*. N.p., n.d. Web. 06 Oct. 2015.

Prideout. "Does AutobahnJS Support Binary Messages?" *Autobahn*. 2015 Stack Exchange Inc, 14 Sept. 2013. Web. 20 Oct. 2015.

"Pros and Cons of Aluminum." *ElementAluminum*. N.p., n.d. Web. 06 Oct. 2015.

"Pulleys." *Science Information*. N.p., n.d. Web. 06 Oct. 2015.

"PWM Outputs." *Raspberry Pi • View Topic - PWM Outputs*. N.p., n.d. Web. 22 Sept. 2015.

"Quadruped Robot (Stag Mk2) - Autonomous Navigation and Trotting." *YouTube*. StagRobotics, 13 Apr. 2013. Web. 22 Sept. 2015.

"Reducing Voltage with Resistors." - *Electrical Engineering Stack Exchange*. Electrical Engineering, n.d. Web. 20 Oct. 2015.

"RFID Reader ID-20LA (125 KHz)." *SparkFun Electronics*. SparkFun Electronics, 2015. Web. 22 Sept. 2015.

"RFID Tag Maximum Read Distance." *SkyRFID*. SkyRFID, n.d. Web. 22 Sept. 2015.

Richards, David J. "CHAPTER 10 TEAM STANDARDS OF EXCELLENCE." *Team Standards of Excellence*. David J Richards, 25 Jan. 2014. Web. 29 Sept. 2015.

Robertson, David. "Pangolin Walking." *YouTube*. You, 05 July 2014. Web. 01 Oct. 2015.

Robot Shop. "HerkuleX DRS-0101 Smart Robot Servo." *Robot Shop*. Robot Shop, n.d. Web. 26 Sept. 2015.

"ROBOTIS E-Manual V1.25.00." *Overview of Communication*. ROBOTIS, 2010. Web. 22 Sept. 2015.

Rosebrock, Adrian. "Accessing the Raspberry Pi Camera with OpenCV and Python - PyImageSearch." *Py Image Search*. N.p., 30 Mar. 2015. Web. 22 Sept. 2015.

Rosebrock, Adrian. "Install OpenCV and Python on Your Raspberry Pi 2 and B+." *PY Image Search*. N.p., 23 Feb. 2015. Web. 22 Sept. 2015.

Rosebrock, Adrian. "OpenCV and Python Color Detection - PyImageSearch." *PyImageSearch*. N.p., 04 Aug. 2014. Web. 22 Sept. 2015.

Russell, Linda. "Gantt Charts, PERT Charts: What Use Are They?" *Project Smart*. N.p., n.d. Web. 22 Sept. 2015.

S0uraV_DAS. "Handheld EMP Device." *Instructables.com*. Autodesk, Inc., 21 Oct. 2015. Web. 22 Oct. 2015.

"Science News." *ScienceDaily*. N.p., 29 Jan. 2009. Web. 22 Sept. 2015.

Sheffler, Tom. "Realtime Streaming Using Autobahn Websockets." *Realtime Streaming Using Autobahn Websockets*. N.p., 2 Dec. 2013. Web. 20 Oct. 2015.

"Sign up for Our Monthly Newsletter!" *K&J Magnetics Blog*. N.p., n.d. Web. 01 Oct. 2015.

"Simple IR Distance Sensor Tutorial." *Arduino Playground*. Arduino, n.d. Web. 29 Sept. 2015.

Skinner, J. D., and Cristian T. Chimimba. "The Mammals of the Southern African Sub-region." *Google Books*. Cambridge University Press, n.d. Web. 22 Sept. 2015.

Skyler, and Peter Mortensen. "Why Does Water Damage Low-voltage Microelectronics so Easily?" - *Electrical Engineering Stack Exchange*. Stack Exchange Inc, 5 Sept. 2013. Web. 22 Oct. 2015.

Smith, Julius. "Audio Applications of the Short-Time Fourier Transform (STFT)." *CCRMA*. N.p., n.d. Web. 22 Sept. 2015.

"SparkFun UV Sensor Breakout - ML8511." *SparkFun*. N.p., n.d. Web. 08 Oct. 2015.

Spröwitz, A., M. Ajallooeian, A. Tuleu, A. Ijspeert, M. Vespignani, E. Badri, and F. L. Moro. "Accessibility Links." *Cheetah-Cub*. EPFL, 27 Mar. 2014. Web. 22 Sept. 2015.

StagRobotics. "Quadruped Robot (Stag Mk1) - First Steps." *Youtube*. N.p., 12 Mar. 2013. Web. 22 Sept. 2015.

"Stainless Steel Metal Densities and Weight Conversion (Based on 300 Series Stainless)." *The Falcon Metals Group*. Falcon Metals Group, n.d. Web. 29 Sept. 2015.

Suksant. "Fast Colour Based Object Tracking with OpenCV | Suksant Sae Lor (Hui) – 罗光辉." *Suksant Sae Lor*. N.p., 26 Apr. 2013. Web. 20 Oct. 2015.

"Surface_matching. Surface Matching¶." *OpenCV*. N.p., n.d. Web. 22 Sept. 2015.

"TEAM LEADERSHIP AND SELF-MANAGED TEAMS." *TEAM LEADERSHIP AND SELF-MANAGED TEAMS*. N.p., 21 Sept. 2009. Web. 29 Sept. 2015.

"Tenergy 18650 6600mAh 7.4V Protected Lithium Ion (Li-ion) Bare Leads Battery - Bulk (31008)." *BatteryJunction.con*. N.p., n.d. Web. 15 Oct. 2015.

"Titanium & Aluminum." *TiLite*. TiLite, n.d. Web. 29 Sept. 2015.

Toner, Matt. "Showcase of Projects and Ideas." *Stag Robotics*. N.p., 2013. Web. 22 Sept. 2015.

Travisdewolf. "Inverse Kinematics of 3-link Arm with Constrained Minimization in Python." *Studywolf*. Wordpress, 11 Apr. 2013. Web. 22 Oct. 2015.

Umberfur. "Arduino-Controlled Wolf Tail." *Instructables.com*. Autodesk, Inc., n.d. Web. 22 Oct. 2015.

Voute, Christopher. "Making Electromagnetic Weapons: EMP Generator, Part One." *Fear Of Lightning RSS*. WonderHowTo, n.d. Web. 24 Sept. 2015.

"WAMP Programming." *Autobahn|Python*. Autobahn|Python, 6 Sept. 2015. Web. 22 Sept. 2015.

Wantanabe, Laurie. "Aluminum Vs. Titanium." *Mobility Management*. 1105 Media, Inc., 1 Mar. 2010. Web. 29 Sept. 2015.

"Welcome To Exploring Electrical Engineering." *Welcome To Exploring Electrical Engineering*. N.p., 30 Jan. 2005. Web. 24 Sept. 2015.

"Welcome to PyPy." *RSS*. Pypy, 4 Sept. 2015. Web. 22 Sept. 2015.

"What Is Lean?" *Lean Practices*. Lean Practices, LLC, 23 May 2015. Web. 22 Sept. 2015.

"What Is Six Sigma?" *ISixSigma*. ISixSigma, n.d. Web. 22 Sept. 2015.

"Wiring Diagram." - *How to Make and Use S*. Smart Servo, n.d. Web. 20 Oct. 2015.

Wolf Tronix. "Covering the World of Personal Robotics." *Life Size Animatronic Wolf Tail*. ServoMagazine, 9 Sept. 2015. Web. 22 Sept. 2015.

WoodFord, Chris. "Resistors." *How Do Resistors Work? What's inside a Resistor?* Explain That Stuff, 6 Aug. 2015. Web. 20 Oct. 2015.

"World's Weirdest - Pangolin." *YouTube*. NatGeoWild, 06 Dec. 2012. Web. 01 Oct. 2015.

Xu, Wenfu, Bin Liang, Cheng Li, and Yangsheng Xu. "Autonomous Rendezvous and Robotic Capturing of Non-cooperative Target in Space." *Robotica*. N.p., n.d. Web. 24 Sept. 2015.

Yao, Ming. "Mathematics for Inverse Kinematics." *Mathematics for Inverse Kinematics* (2013): n. pag. 10 Oct. 2013. Web. 22 Sept. 2015.

Yegulalp, Serdar. "5 Projects That Push Python Performance." *InfoWorld*. Infoworld, 9 Feb. 2015. Web. 22 Sept. 2015.

CARPA APPROVAL

Rules, conditions, and requirements are subject to change by the CARPA authority. All changes will be discussed in class, and be presented to the company teams in writing. Changes are negotiable and deadline extensions must be agreed upon by both teams.

Project Manager	Date
Deputy Project Manager	Date
Chief Scientist	Date
Systems Engineer	Date
Chief Programmer	Date
Chief Financial Officer	Date
CARPA Authority	Date