Senior Design Project

Arboreal Ascender

21 September 2023

“I pledge my honor that I have abided by the Stevens Honor System.”

The report has been prepared by:

1. Thomas Roff [T.R.]

2. Marc Finelli [M.F]

3. Jason Pagan [J.P]

4. Daniel Goldberg [D.G]

5. Paul Gomez [P.G]

Mission Statement (IDE401)

Background (IDE401)

Analysis of Stakeholders Needs (IDE401)

Introduction

Background

In 2023 alone, a staggering 43,899 wildfires ravaged a total of 2.33 million acres (Center for Disaster Philanthropy, 2023). This project hopes to prevent this tragedy by engineering a fully operational tree-climbing robot that can aid in forest fire suppression efforts. Given the urgent and hazardous nature of forest fires, finding a means to minimize harm to both wildlife and first responders is imperative.

On the 19th of September, four brave British-Columbian wildfire fighters were combating the Canadian wildfires. Due to treacherous road conditions caused by the disaster, they lost their lives on their way home. This case is just one of many where courageous first responders have lost their lives in the line of duty. There should be more safety measures put in place to protect first responders by using robots to conduct their most dangerous tasks.

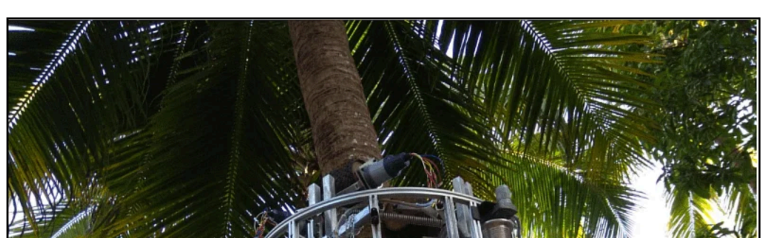
A compelling solution to alleviate these statistics emerges from robots that possess a unique talent, the ability to climb trees. Although there are numerous existing tree-climbing robots, each comes with its own set of trade-offs, including but not limited to efficiency, cost, or ethical considerations. Furthermore, none of these robots are designed to provide substantial assistance or fully replace the roles of these wildfire fighters, thus removing them from harm's way. Attaching a mechanism intended to combat flames from the treetops could rid the need for direct human interaction with the disaster site. The creation of a robot capable of addressing the pressing challenges posed by wildfires, while simultaneously mitigating the mentioned trade-offs, is something that the group seeks to make feasible.

State of the Art Review

As previously stated, there are existing robots capable of tree climbing. During the concept selection process, it's crucial to analyze and draw insights from existing successful designs, aiming to enhance and build upon them. To do so, a State of Art Review was conducted on various designs that bring about interesting concepts to think about when considering the design of the Arboreal Ascender.

Figure 1: Inchworm Bot, Ref 5

The first robot observed (as shown in Figure 1) utilizes talon-like limbs to adhere itself to the tree, and a body that compresses and expands to slowly inch itself upward, similar to that of an inchworm. The creator of this robot used a lot of workarounds for the price, such as repurposing parts from another tech, but a loose price estimate would range from \$300-\$500. There are a few significant drawbacks to this design, notably its apparent difficulty in supporting its weight, let alone the added burden of a payload, as well as the excessively slow ascension speed and inability to travel along the horizontal axis. Nonetheless, this robot provides valuable information to the group regarding the feasibility of the form of locomotion used along with general ways to keep the cost of the robot in check.



The second robot observed (shown in Figure 2) is designed to replace the treacherous job of coconut tree harvesters. It utilizes a total of 8 mecanum wheels, 4 on the top and 4 on the bottom which are attached to a body that fully wraps around the tree. While both the price and feasibility that go into making a robot like this are favorable, this design suffers from significant limitations when it comes to its applicability to various tree types. It works perfectly for palm trees which have no branches, but that is not the case for most other trees. One could argue that the built-in saw included in this design used to cut the leaves could be repurposed to cut the branches in its path. However, this proposition raises ethical concerns as it would inflict irreparable harm to the tree. Despite the aforementioned negatives included with this design, the use of mecanum wheels provides an interesting view into different ways to allow the robot to achieve full 360-degree mobility around the tree's base.

Figure 2: Coconut Harvesting Bot, Ref 6

The third design observed (pictured in Figure 3) utilizes a similar form of locomotion to the one used in the inchworm bot (Figure 1). This robot first attaches its lower half to the tree, extends its spine, secures its upper half to the tree, and subsequently releases its lower half while decompressing its spine to slowly make its way up the tree. The stark difference between the two designs is that rather than utilizing sets of individual talons as its form of adhesion, it uses a set of claws that can be fixed to the tree independently. This modification may raise the cost of manufacturing but also increases its overall maneuverability due to the fact it doesn't rely on a system of talons as seen in Figure 1 or a set of wheels as seen in Figure 2 to achieve static stability. These claws enable the robot to securely grip a broader range of tree sizes, accommodating slender and wider trunk diameters. This specific example also utilizes a more flexible spine, thus allowing

Figure 3: Inchworm Bot w/ Claws, Ref 7

the robot to turn at slight angles. The demo of this design shows that it can carry a 1.75kg payload, which may seem very light but is a step further in the right direction compared to the inchworm bot. This doesn't necessarily mean that claws allow for a higher payload than talons, rather it gives insight into how different combinations of adhesion and locomotion may provide notable differences in weight capacity and maneuverability, while not straying extensively far in price.

Project Scope and Resources

The overall scope of the project is to design and build a swarm of robots that can surround and contain a forest fire. This will be done by having each drone climb its tree around the border of the fire and dispense a fire suppressant downwards to stop the spread of fire. This is unfeasible based on the allotted budget and time frame, but to work towards that endpoint, the initial plan is to make a robot that can climb a tree autonomously. There are two potential designs for our project, one assuming the default budget of $700 and the other assuming significant additional funds. With both, the goal is to have a robot that can ascend a tree dynamically.

For each degree of freedom, accounting for the actuator and controller a module would most likely be around $75 at minimum. This means that with a base budget, a maximum of around 10 degrees of freedom would be allowed. With an increase in budget, more degrees of freedom would be available which allows for an increase in potential mobility. The available resources will primarily change the amount of dynamic movement and overall strength which will drastically change the speed and efficiency of the robot.

Project Plan (IDE401)

Requirements

Product Perspective and Intent

To relieve the risk accrued by forest firefighters as well as the expanse of destruction caused by the spread of damage produced by forest fires, this robot will be able to ascend a tree of sufficient height to then discharge fire suppressant over a significant area of land threatened by the growth of the fire. In this function, the robot will reach its target height and affix itself to the tree to then release the firefighting agent it has carried up to this point. The interaction that occurs between the tree and the robot will be designed in a way to not damage or kill the tree since halting destruction is its primary intent. Through this act, this robot provides a humanitarian option for wildfire suppression because humans would no longer have to risk their lives in the line of forest fire fighting.

Primary Functions

This robot must be capable of dynamically ascending its target tree, which can be broken down into three key aspects. These aspects are the motion of the robot on the tree, the adhesion of the robot to the tree, and the sensing ability of the robot to detect any irregularities in its path of ascension. The robot must maintain a complex function consisting of these three components to accomplish its end goal of forest fire suppression.

The first of the key aspects of our robot design is the way the robot moves and ascends once attached to the tree. To accomplish this goal effectively, the robot should have motion capable of at least three axes of motion. For example, the robot should be able to move left and right, up or down, and forward and backward from its point of reference. Otherwise, the robot should have the ability to rotate around the tree depending on the design of the end effector on the robot arm.

The second essential aspect of the intended function of this robot is its method of adhesion to the tree it is attempting to climb. To accomplish this goal, it has been determined that the robot will have sufficient adhesion to remain entirely stable during its gate of motion as it moves upward along the tree. Additionally, its method of adhesion during its motion will not be inherently destructive to resist damaging or even killing the tree while it is climbing to a sufficient height.

The third and final key element to the ideal function of our robot design will be the sensing and pathfinding that the robot performs while it moves up the tree to its target destination. In its function, the robot must be able to detect any irregularities or changes in the direction of the tree. To accomplish that, the robot must be able to sense its grip location, strength, and stability. As well, it should be able to detect the surface topology of the tree that it is climbing to inform its path.

Operating and User Environments

During this robot's intended end goal function, the tree climbing robot is expected to perform preventative measures to help direct and or control a forest fire's spread to aid in the reduction of damage and risk to human life. Therefore the robot should be able to resist some degree of high temperature. However, it is not intended to be exposed to direct open flame temperatures. As such, the components of this robot should be composed of materials that will withstand a reasonable amount of heat and the wiring insulated in a way to resist damage at these same temperatures.

The ultimate end goal of this robot design is to be created as a drone swarm, which would require a master controller to direct the primary path of each swarm group to the necessary location to effectively control or subdue the fire. Therefore, there would be minimal human-to-robot interaction and little to no risk to the user. By setting the necessary areas of importance in the path of the fire, the master control platform or brain of the robot should successfully perform the necessary functions. Ideally, this robot concept can be improved to be fully autonomous in its functions with no need for human involvement in its mission.

System Requirements

The system requirements for the robot are defined to ensure the robot can successfully climb a tree without damaging it. The robot will need to understand the tree's shape and structure to properly plan a path to climb it, and it needs to not damage the tree while climbing it. These requirements can be seen with appropriate weighted importance, desired values, and determination of functional and nonfunctional components in the following Table 1.

Table 1: System Requirements Table

	System Requirements Table		
#	Requirement	Importance	Desired Value
1	Ability to climb the most common types of trees in the tri-state area	1	Y
2	Capable of staying at the top of the tree for a period of time	2	1hr
3	Payload Capacity greater than goal	2	2x
4	Capable of sensing and understanding tree topology	2	Y

NF1	Needs to fit within an overall space of 1 m^3	2	Y
NF2	The system must be safe around human operators	3	Y
NF3	Must be non-destructive to the tree's surface	1	Y
NF4	Must not leave any permanent effect on tree's lifespans	1	Y

Functional Requirements Descriptions:

1. To create a successful prototype, the robot will be designed around climbing the most common tree type in the tri-state area. Later on, the design can be modified to climb other tree types.
2. To conduct firefighting operations, the robot needs to remain stable at the top of the tree for a length of time.
3. The robot needs to lift twice its body weight to allow for transportation of the firefighting system.
4. The robot needs to be able to sense the tree's shape and texture to create a strategy to climb it.

Non-Functional Requirements Descriptions:

1. The robot will be used in a swarm, so the smaller it is the easier it should be to transport. Therefore, it should be able to fit in at least a 1 m^3 space.
2. Although it is designed for remote use, it should still be safe around humans in case it is used close to residential communities.
3. The robot will be designed to save trees, so being as non-destructive as possible is essential to ensure the trees are not damaged.

4. The robot should not harm or damage the tree in such a way as to reduce the tree's lifespan.

Hardware Requirements

The hardware requirements for this robot center around achieving a stable gait capable of climbing the tree. The hardware needs to be able to achieve a static gait climbing up the tree while using as few degrees of freedom as possible to reduce the system's cost. Since the tree surface is unpredictable, the robot should be able to adjust its gait in such a way as to reach a variety of grip positions. In a situation where one grip location is impossible to use due to the tree shape, the robot should be able to extend or shorten its standard gait to use an alternative grip position to climb the tree.

Table 2: Hardware Requirements Table

	Hardware Requirements		
#	Requirement	Importance	Desired Value
1	Can accurately determine grip strength	1	Y
2	Can determine branch layout	2	Y
3	Can achieve a stable open loop gait to climb	1	Y
4	Can achieve its gait through a variety of grip positions	2	Y

5	Can rotate end effector to ensure proper contact with tree	1	Y
6	Can reach static stability at least once per gait cycle.	2	Y
NF1	Cost within budget	1	700
NF2	Top vertical speed is higher than goal.	2	0.1in/sec
NF3	Battery longevity is higher than goal	3	10 hrs.

Functional Requirement Descriptions:

1. Determining grip strength is essential for the robot to determine its stability. If it can not achieve high grip strength, it should move its gripper until it can achieve high grip strength.
2. Determining the branch layout through lidar or camera sensors will allow the robot to plan its path up the tree.
3. Although closed-loop gaits can be more efficient, having a stable open-loop gait is important to improve reliability.
4. By increasing the maneuverability of the robot, it can climb the tree using various grip positions. This is important if the robot encounters a position where achieving a strong grip is impossible, it can navigate around that grip position to still ascend the tree.

5. Most end effector designs only work properly if the end effector is perpendicular to the tree surface. Since the surface is curved, being able to adjust for that change is essential to ensure proper contact.
6. Achieving static stability at least once per gait cycle will allow the robot to stop and reassess its path up the tree if it encounters a grip position that is not as effective as predicted.

Non-Functional Requirement Descriptions:

1. Keeping the cost of the hardware within the budget is essential, as that is the main expense for this project.
2. The faster the robot is, the more time is left for firefighting operations after its assent.
3. Having a long battery life is important if the robot must remain at the top of the tree for an extended period of time.

Software Requirements

The robot's software needs to be capable of moving the robot between various points using a variety of algorithms. It needs to predict future grip positions, generate a path between these positions, and then use various feedback mechanisms to maneuver the robot between the points in space. The functional and nonfunctional software requirements that lead to this behavior are shown below.

Table 3: Software Requirements Table

Software Requirements

#	Requirement	Importance	Desired Value
1	Capable of determining future grip positions	1	Y
2	Can evaluate the feasibility of future grip positions and rank them	2	Y
3	Capable of generating a path of grip positions up a tree	1	Y
4	Capable of navigating around obstacles using alternative grip positions	2	Y
5	Can actuate joints to achieve the generated robot motion with acceptable deviation	2	+/- 1 inch
6	Actuate joints to the desired setpoint within accepted error margin using closed-loop feedback	1	+/- 5%
NF1	Can operate autonomously without user intervention	1	Y
NF2	Can run completely on onboard hardware	2	Y
NF3	Relies only upon data collected during the assent.	3	Y

Functional Requirement Descriptions:

1. To climb the tree, the robot must be able to identify potential grip positions higher on the tree so that it can use them as anchors to ascend.

2. By locating potential grip positions, it can then determine their possible strength so the robot can create a path with the most stable grip positions up the tree.
3. Once the robot has evaluated good grip positions, it can then weave them together into a continuous path up the tree according to its gait cycle.
4. With its list of grip positions, the robot should be capable of generating detours around branches by relying on backup grip positions.
5. Once the robot has determined its ideal grip positions, it needs to actuate its joints properly to achieve this path.
6. To actuate the joints properly, the robot needs control systems capable of reaching low steady-state error to ensure the joint variables are correct to reduce error in the workspace.

Non-Functional Requirement Descriptions:

1. The robot should operate autonomously, without user intervention.
2. The robot will be used in remote environments, so the computational power needed to calculate its path and actions needs to be housed onboard the robot itself.
3. Since the robot ascends trees in the wilderness, it is unlikely to have any pre-generated data before starting its ascent. It will have to generate its map of the tree as it ascends the tree.

Constraints and Assumptions

The primary constraints of this project are cost, feasibility, and weight capacity. An ideal tree-climbing robot would be fairly cheap to create so that it can be produced in numbers, very feasible in design to prevent complications not only in the construction process but also in the field, and lastly be capable of carrying not only its weight but additionally the weight of its payload. As the majority of on-the-market designs, as seen in the state-of-the-art review, are both slow and cannot support a significant payload, the primary goal of the project is to support a large payload and or go up the tree fast. Additional constraints for the robot would include limited internet and signal in the woods, and limited access to power aside from potential solar cells or generators. In addition, as it is in remote parts of the woods, there is a low likelihood of repair if anything breaks down, either needs to be extremely cheap or extremely consistent.

Some assumptions that are to be made are that we will be able to filter through most climbable trees. The goal is not to climb every single tree, but to climb the majority of the trees, and in most cases, it would be enough to encapsulate the fire properly. The robot will be aiming to climb a tree that has not yet been set ablaze as the goal is surrounding the fire and stopping the spread, not putting out the fire, thus we can assume that the tree itself is not on fire. Finally, there is an assumption that there is access to a base station which would allow for computation to be offset.

Applicable Codes and Standards

In order to maintain the safety of users and manufacturers alike, there are certain standards and regulations at play in the design and development of robots or robotic parts. As well, there are safety standards and codes at play in the use of certain chemicals. More specifically, there are particular standards when it comes to the use of chemical fire suppressants in the environment. In the development and design of the robot, the American National Standard Institute's standard R15 (ANSI R15) will be followed to maintain safety for us as designers and for the future potential users of the robot. For the selection of chemicals to use in the end function of fire suppression, the standard set in place is the National Fire Protection Association's standard 17 (NFPA 17), which will determine that the chemical used will function accordingly.

Concept Development and Selection

The group began concept development by studying nature and emulating climbing techniques available in nature. The group identified a bunch of mechanisms for adhering to the tree surface, for locomotion up the tree, and for generating the feedback necessary for closed-loop control. Once these lists of different ideas were generated, the group combined these individual “ingredients” into recipes for various different robot designs. Based on this thought process, ideas of the movement from the robot that were considered were articulated manipulator, rectilinear, friction wheel, winch, Stewart table, linkages, SCARRA, sails, and jump. Adhesion revolves around how the robot will remain intact to the tree as it climbs it, in the table the ideas included are claws, talons, drills, micro-spline, squeeze, and bolt/hook. The last column entailed the path planning and the open/closed system. Certain robots could achieve a closed-loop gait, while others were locked into an open-loop gait due to restrictions in their movement. The group then took these various “ingredients” and mixed them to create various different “recipes” for possible robot design. Table 4 shown below shows the table with all of the ideas generated.

Table 4: Concept Generation Table

Abbreviation	Description	Movement	Adhesion	Control
AAA	Monkey	Articulated Manipulator	Claws	Open
AAS	Scorpion	Articulated Manipulator	Claws	Open
			Squeeze	
¢	Centipede	Articulated Manipulator	Talons	Open

$¢	Rotating Centipede	Friction Wheel	Squeeze	Open
		Rectilinear		
RBR	Rubber Band Bracing	Friction Wheel	Squeeze	Open
			Bolt/Hold	
BAB	Belt Arm Bot	Linkages	Belts	Open
Air$	Air Snake	Articulated Manipulator	Air Bladder	Open
Win	Winch Jawn	Winch	Bolt/Hold	Open
Dril	Drill Rack And Pinion	Rectilinear	Drills	Open

The group then took all of the above-generated ideas and used a simple Pugh matrix to filter the ideas down into three ideas that the group believes are well suited to the task. This matrix uses affordability, feasibility to prototype in the project timeline, payload capacity, and flexibility in its movement to evaluate the promise of each design.

Table 5: Plus or Minus

Abrev	Affordability	Feasibility	Capacity	Flexibility	Branch Angle	Total
AAA	-1	0	0	1	1	1
AAS	-1	-1	0	0	0	-2

¢	1	1	1	-1	-1	1
$¢	0	0	1	0	0	1
RBR	1	0	1	-1	-1	0
LAB	1	1	1	-1	-1	1
Air$	1	-1	0	0	0	0
Win	1	1	1	1	1	5
Dril	0	1	1	0	1	3

Of these designs, a few are impractical. The drill-based design does not work as it would be destructive to the tree, and the Air Snake would be too complex to be viable considering the difficulties inherent with air bladders. The Winch idea is just generally outside of the spirit of the project and is potentially dangerous for the design fair, as it involves shooting a bolt into the top of the tree and rappelling up to the top, sidestepping the interesting question of locomotion up a vertical surface. This narrows down the list to the following: monkey, centipede, and the linkage arm bot. The monkey idea (concept #1) is shown in Figure 4 below.

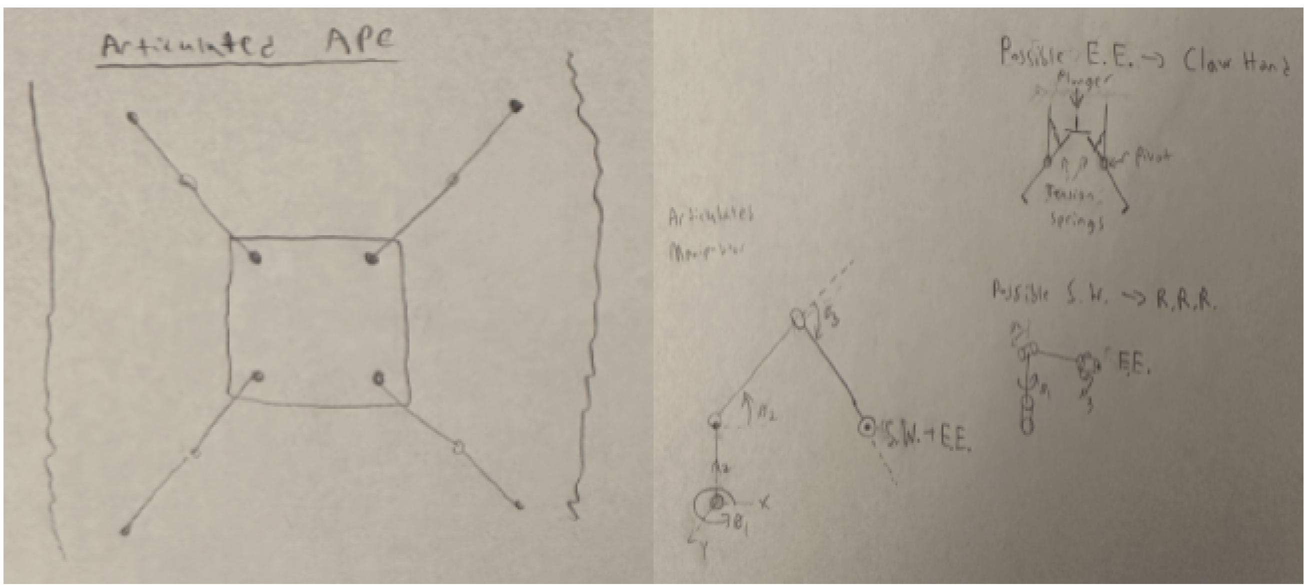


Figure 4: *Monkey (Concept #1)*

The monkey concept is the most capable design that the group has considered. This is due to the degrees of freedom and ability to be able to get up the tree regardless of what obstacles (branches) it might come across. The monkey is made up of four articulated manipulators, shown in Figure 4. At the end of each manipulator is a claw which acts as a hand to grasp the tree surface. Each manipulator is 6 DOF. This results in a total of 24 degrees of freedom in this design, which translates to a much more costly design. The rectangular base contains all the computational power needed to drive the robot. The flexibility provided by the multiple manipulators will allow the robotic monkey to use its arm to lift itself up and over the branch similar to a real-life monkey. While it could climb as well as a monkey would, the issues that arise from having a high DOF system with a constrained budget make the group hesitant to choose this design.

The robotic centipede design (concept #2) involves six limbs that will compress into the tree. For the centipede to move up vertically, it must maintain four points of contact on the tree while the other two limbs extend. These limbs act in sets of two, where one on each side of the tree squeezes the tree trunk to stay fixed to the tree. The number of DOFs is much lower compared to the monkey design, which allows for more feasibility to create this design within the budget. Comparatively, the centipede bot will only have two articulated manipulators. This leads to the system having 11 DOFs in total. Lowering the DOFs will make it lack the flexibility to climb a tree, as it will only be able to go up vertically and will need an open path of 180 degrees. Branches and other obstructions being within a 180-degree distance will be detrimental to its ability to climb the tree, as it must span a 180-degree distance along the tree surface to stay fixed to the tree. The limbs would behave similarly to a rack and pinion mechanism to adjust themselves as they climb. Two hands or legs will be held in place while the rest of the limbs stretch and clamp to get to their new position. This design needs an ideal condition of not having anything in its pathway, restraining its overall capability. Figure 5 shown below is an example of how the centipede would look like.

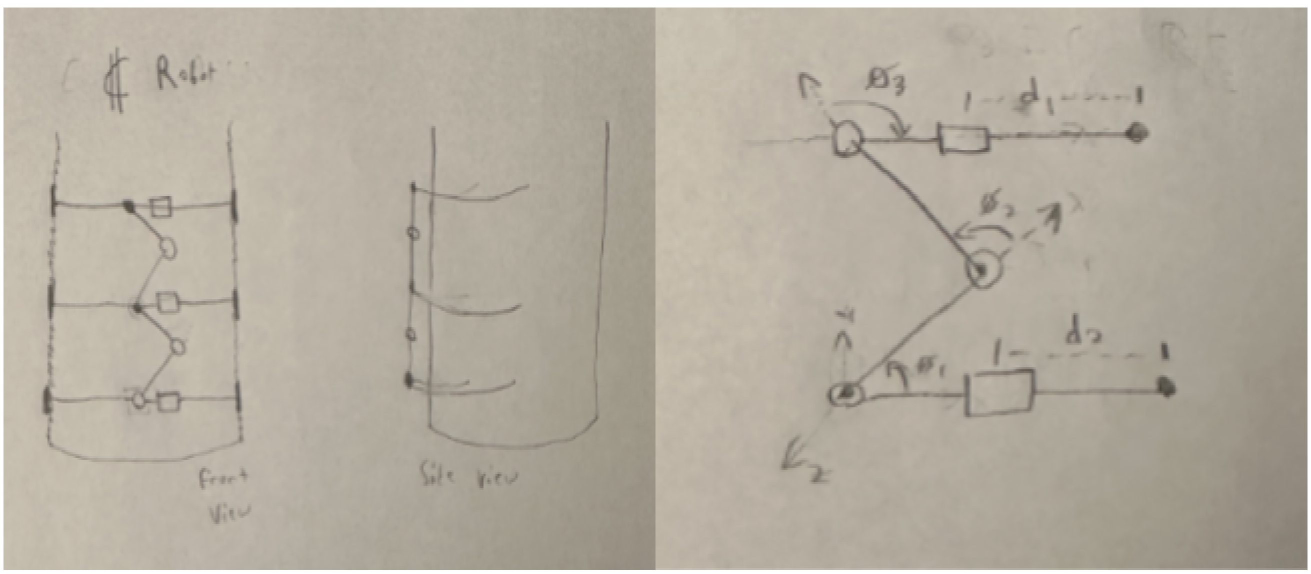


Figure 5: *Centipede (Concept #2)*

The link arm robot design (concept #3) will contain some sort of linkage to link the wheels. The wheels chosen are omni wheels because of the versatility of being able to move vertically and rotational. There would be motors for each wheel, the linkages would be controlled by two motors on either side. The DOF will be roughly 6 in total, which is quite low as well in terms of budget. This design is an improvement compared to the centipede design where it needs to be in an ideal tree. The link arm bot will be able to rotate around the tree to find a position where there are no obstructions, however, it needs 180 degrees of distance around the tree. The drawback with this design would be a scenario where there are two branches exactly 180 degrees apart or multiple branches within the diameter of the tree. Figure 6 shown below is how the design might look.

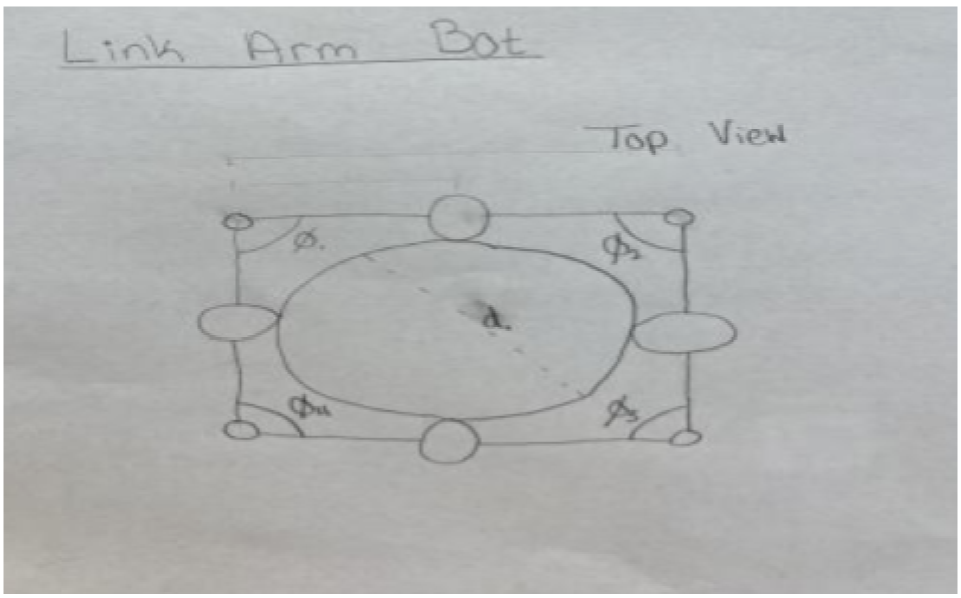


Figure 6: *Link Arm Bot (Concept #3)*

With the unfeasible idea colored red, the leftover ideas can be seen in Table 6 below. Each constraint is weighted based on importance with feasibility at the greatest and each design evaluated per constraint, with a higher value being better for the design. Utilizing the weighted values below, this leaves the monkey-based robot (AAA) and the centipede robot (¢) in the lead with both having around 180 points total. With the monkey robot being more capable at the downside of increased cost, the group is confident that it can realize one of these designs based on the budget awarded to the project. If the budget does not allow for such an expensive system, then the centipede robot should be capable of realizing the group's mission of generating a tree-climbing robot.

Table 6: Weighted Selection Matrix

Abrev	Affordability	Feasibility	Capacity	Flexibility	Branch Angle	Total
Weighting	7.5	10	3	3	5	
AAA	2	7	6	9	10	180
¢	7	8	8	2	3	177.5
LAB	8	7	8	1	1	162

Preliminary and Detail Design

Based out of our concept development and selection weighted matrix, the articulated arm ape bot had the highest score. However, this design had an anticipated cost beyond our assumed budget. Therefore, the team has decided to move forward on the centipede bot design as the project to be developed and prototyped. If additional funding is acquired, then the group believes expanding the centipede robot's capabilities will lead to the most optimal result. The following preliminary and detailed design was performed on this chosen centipede bot design.

System Architecture

Overview of System Architecture

The general design includes a set of four claws each spring tensioned to grasping, all set up in a vertical line. These claws are connected via paired revolute joints which allows for linear traversal. As seen in Figure 7 below, the robot cycles through a function that allows it to move up the tree. Effectively releasing two claws moving those two grippers upwards while the other two are bracing the position. Those two grippers will then grasp the tree, allowing the other two claws to release. This pattern is then repeated to climb up the whole tree.

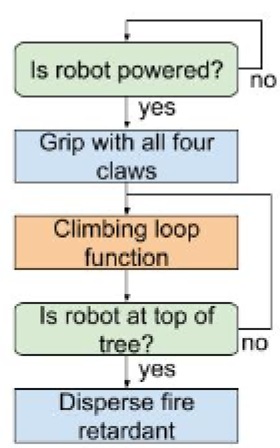


Figure 7: Overview of System Architecture

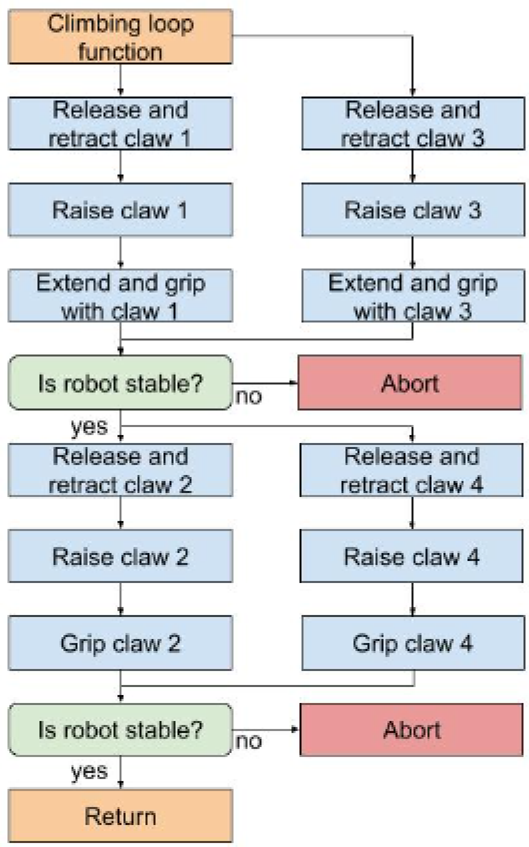


Figure 8: Climbing Subfunction

Hardware Architecture

For the hardware architecture, there are two main sections, the physical movement subsystem and the sensor subsystem. All joints are actuated by rotational motion, servo motors.

The prismatic joints are simply rotational motion translated to linear by rack pinion. The rotational acting servos will be used for the arm joints on the robot which bring the robot up throughout the gait cycle. The prismatic acting servos drive rack and pinions which are used for the claw itself and the horizontal actuation into the tree for the claws. All of these servos have potentiometers that serve as a feedback loop to the motor controller. In terms of sensing, there is not all that much that will be used in our first prototype. The primary ones being a method to scan the three to understand the topology and a sensor that will check how far from the tree each separate claw is. This will be done with a one-directional LIDAR and will serve as feedback to the prismatic joints controlling the distance of each claw from the tree.

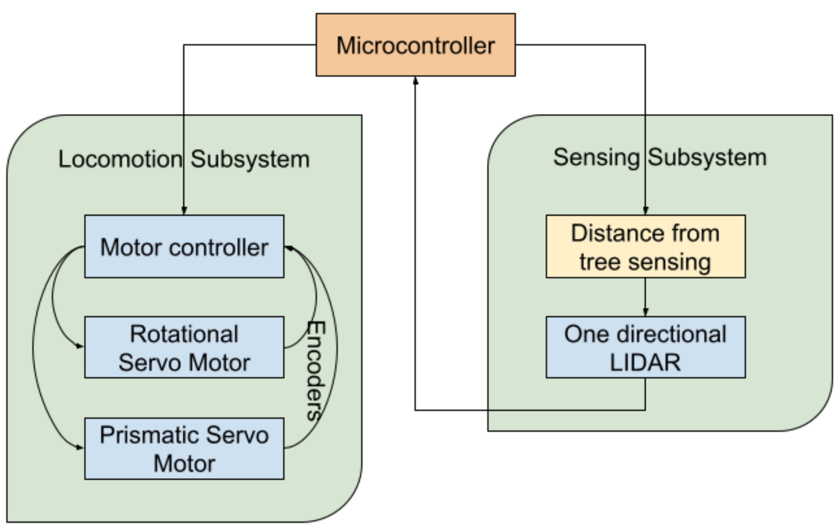


Figure 9: Hardware Architecture

Software Architecture

Our robot will be utilizing a Raspberry Pi as the lead controller and an Arduino board as the follower. The Raspberry Pi will be taking sensor inputs, and then using this info to organize a path for the robot. It will calculate the trajectory required for the joints of each limb of the robot to successfully reach its next planned waypoint. This information is then sent to the Arduino board which is responsible for controlling the servo motors.

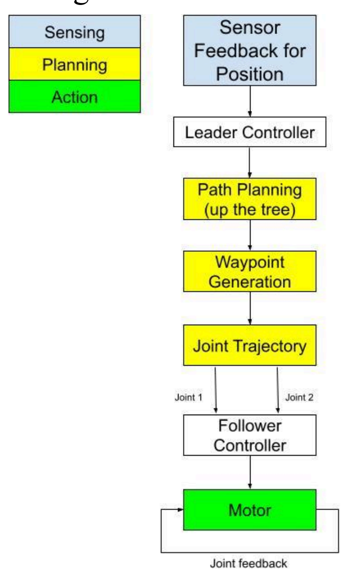


Figure 10: Software Architecture

Security Architecture

As the final design of this project would result in a swarm of robots, security would be important so that the network cannot get hijacked. As the current prototype has minimal to no remote capabilities, the maximum required security will be physically storing the robot when not in use to prevent tampering. Further iterations of this project will have to consider the security risk of having multiple robots interacting with each other and keeping their communications secure.

Communication Architecture

For communication, the main example would be the communication from the leader to the follower, in this case from a Raspberry Pi to the Arduino. The Raspberry Pi effectively tells the Arduino exactly what functions to perform. Then the Arduino performs the actions, such as rotating specific motors at certain amounts of degrees or other similar processes

System Design

The system will behave as a series of two gait cycles, where two limbs move at the same time per cycle. Once the robot is on the tree, it will idle until it has had time to assess its environment before acting. At this stage, if the robot finds itself at the desired height, the robot will stop climbing the tree and begin fire fighting operations. If the robot needs to climb higher, then the robot begins path planning by determining a target it wishes to move to, then releasing two grippers from the tree, and then moving the grippers to the new position. The robot will then scan the tree to find an optimal position to grip, once it does it will grip on the tree with its gripper. A stability check comes after gripping the new position. If that test fails, it repeats the path planning. If stability has been satisfied, then it will begin to ascend again until the desired height is reached. Upon reaching the desired height, the robot will be triggered to release oxygen-depriving chemicals, in order to suppress a forest fire.

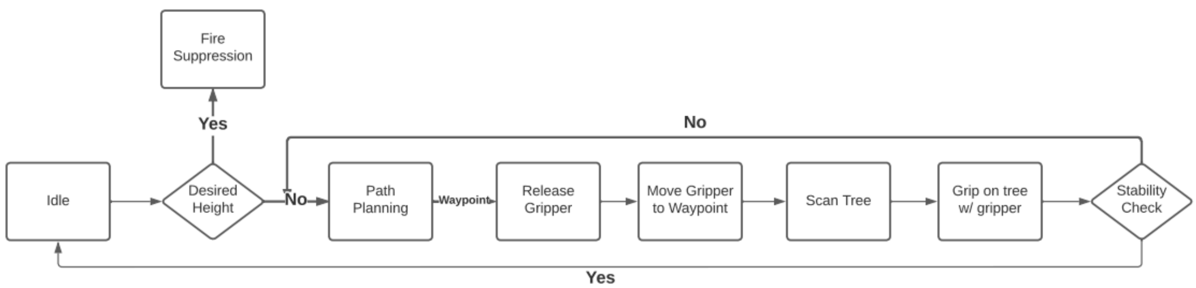


Figure 11: System Design

The system will operate mainly on volatile storage, as power loss is not expected during the robots operation. Large data storage is not needed here, so information can be stored locally on the leader and follower controllers. Additionally, the robot will store digital values from its sensors. If analog sensors are used, then an analog to digital converter will be needed. All of this sensor information will be stored in local volatile storage. Furthermore, the binaries for the programs themselves will be stored in non volatile storage so that the robot can lose power without needing to reflash the robot.

The robot will be fully autonomous, so there will be a minimal user interface for this robot. The user will be prompted to enter the desired height the robot should climb, and what types of fire fighting procedures should be conducted at the top. From there, the robot will automatically climb the tree, and provide diagnostic information in a terminal during its ascent to update the user on its progress.

Mechanical Design

The team’s chosen design was modeled with SOLIDWORKS modeling software in order to develop a more tangible proof of concept and mechanical motion. The modeling was first performed on the mechanical claw ends that will provide the gripping force for the adhesion to the tree. This model can be seen below.

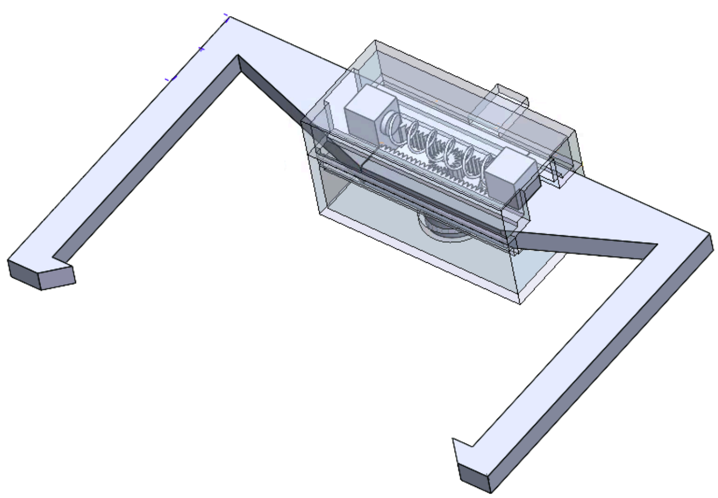


Figure 12: Mechanical Claw Model

As seen in the image above, the claw has a neutral state of being compressed due to the mechanical spring applying a spring force compressing the claws when there is no force applied by the motor. This force will be used to grip the tree between the two teeth of the gripper. When the motor applies rotation to the gear centered in the rack and pinion, the spring is expanded and the claws open allowing for the release of grip. This motion can be seen analyzed in a motion analysis performed on SolidWorks in the following link:

Link to Motion Analysis of Claw

The claw is then attached to another rack and pinion system acting as a prismatic joint to allow for a linear motion behind the claws, which permits the robot to reach an effective point for grip on the surface of the tree it is to climb. The prismatic joint is controlled entirely by the dc motor attached to its gear and does not require a linear spring like the claw. The prismatic joint and claw can be seen assembled as intended in the figure of the model below:

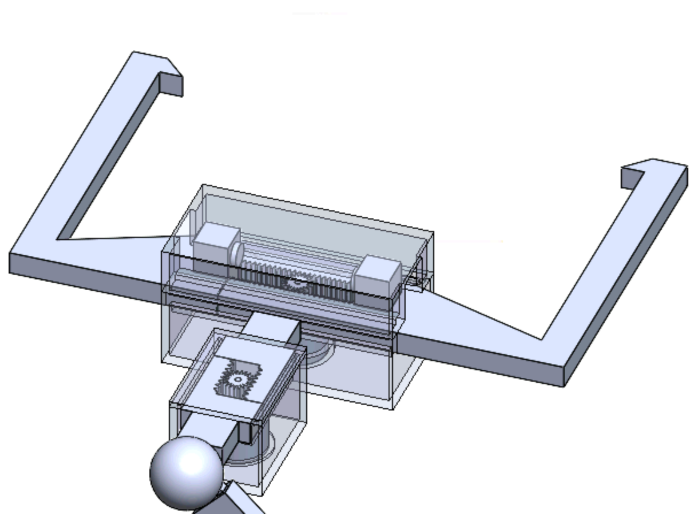


Figure 13: Mechanical Claw and Prismatic Joint Connection

The intended revolute joints in the robot were then modeled as spherical joints in SOLIDWORKS and attached to basic linkages in order to represent the overall design of the articulated manipulators connecting the four claws called for in this design. The intended electrical hardware housing was then modeled in the overall assembly to give a better image of the overall design intent. Additionally, in order to better represent the way this robot will hold to a tree, a cylindrical surface was modeled in the assembly for reference. The fully assembled concept design for the Arboreal Ascender can be seen here:

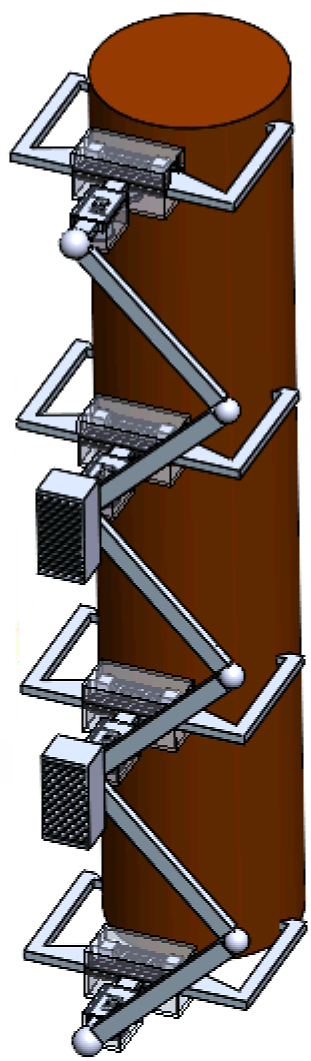


Figure 14: The Arboreal Ascender Full Assembly

For the production of the Arboreal Ascender, many components can be produced through additive or 3D manufacturing, which is beneficial due to the low price for these printed pieces. These pieces will utilize standard PLA filament as the polymer for printing, which will comprise the housings for the electronic hardware and rack and pinion systems on the robot. For the mechanical pieces, the rack and pinion pieces selected for use are made of hardened acrylic. For the arm and claw pieces, they will be cut from cold-rolled aluminum sheets. Additionally, the mechanical springs used in the compression of the claw at its resting state will be made of an extruded steel characterized by the appropriate spring constant value. These selected materials will provide the necessary strength and stress resistance to meet the systems requirements without risk of failure. As well, the material processing methods being implemented in creating these parts will provide the material strength in the system to retain an optimal factor of safety in design. A thorough list of the proposed components can be found in a complete bill of materials found in the appendix.

Electronics/Electro-Mechanical Design

The general layout of our electronic design for our robot consists of a power source, which feeds into both our Raspberry Pi and our power board, while the Raspberry Pi will feed into the Follower Controller, which will then utilize the power board to connect our servo motors. As noted in the bill of materials, we will be utilizing 15 servo motors. To run for 12 hours at 0.5 amps, they each would require a flow of 6 amp hours from the power source. This would indicate that our power source would need to be able to supply 90 amp hours (Ah) in order to be sufficient to supply the energy required for all of our motors to function. This is assuming that all of our motors are functioning the entire time, which ideally will never be the case. This will ensure that the robot will always have enough power to function, even in the worst-case scenario. However, for our proof of concept and prototype, the group intends to utilize direct power cables to ensure the robot is supplied with sufficient power.

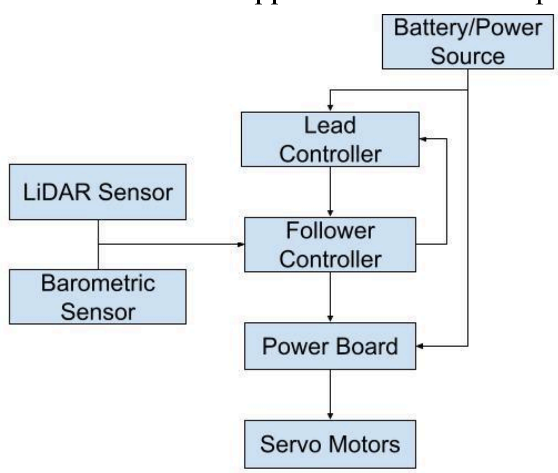


Figure 15: Electronic Analysis

In order for our robot to be able to ascend properly, it will utilize a barometric sensor to ensure that its altitude is increasing, and a LiDAR sensor to find the ideal tree for climbing and to maintain proper climbing distance from the tree surface. Along with this, the grippers will utilize dc hobby servo motors at each joint in order to traverse.

Software Component Design

As previously stated, our robot will be utilizing two controllers, a Raspberry Pi as the lead controller and an Arduino board as the follower. The Raspberry Pi will utilize the Robot Operating System (ROS) and the Arduino board will utilize the native Arduino API. The Raspberry Pi will be responsible for taking all sensor inputs which are originally fed into the

Arduino board, and then using this info to organize a path for the robot. To do so, it will create a set of waypoints for the robot to follow in its path and will utilize the various sensors and joint feedback to ensure that it reaches its waypoints with minimal error. Once the path is evaluated, the Raspberry Pi will calculate the trajectory required for the joints of each limb of the robot for it to successfully reach its next planned waypoint. Finally, the required info is sent to the Arduino board which will then control the servo motors, which are responsible for the locomotion of the robot.

For the robot to successfully climb the tree, it must be able to scan the tree's surface and understand its curvature. The path planning will determine specific waypoints along the tree surface the robot wishes to grip onto. Once the robot reaches this point, it needs to grip a certain span of the tree surface to remain statically stable. To do so, the robot needs to scan the tree's topology.

In order for our robot to be able to find a sufficient spot on the tree to reach for, it will make use of a waypoint-finding method. In this method, our robot will get a scan of the tree surface ahead of it with the LiDAR sensor, and then utilize its data points to estimate the curve of the surface. Once this has been found, the controller will then find the derivative of this curve and compare it to the curve of our ideal standards for climbing. A higher derivative would indicate a flatter and more ideal surface. If the conditions are met, the robot will then rearrange itself to move to the found waypoint, and begin the climbing loop function (Figure 7). If the conditions are not met, however, the robot will abort the method and continue scanning until it finds another path that is ideal for climbing.

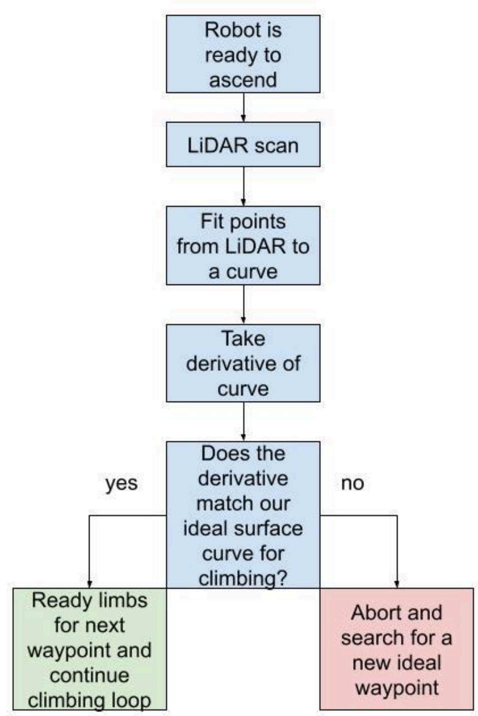


Figure 16: LiDAR Waypoint-Finding Method

Once the robot has understood the tree topology, it will need a way to ensure that it grips the tree at a certain distance to ensure a proper angle between the tree surface and the gripper. To do so, another algorithm will be used that will also make use of the LiDAR sensor, which will allow the robot to estimate how far the sensor is from the surface of the tree as it climbs. It would then compare that to the value required for the grippers to be able to reach their next waypoint to continue the climb. If the robot were to notice that the geography of the tree changes too drastically to a point where it can't continue the climb, it would abort the climbing loop and begin descending to then find a new suitable tree.

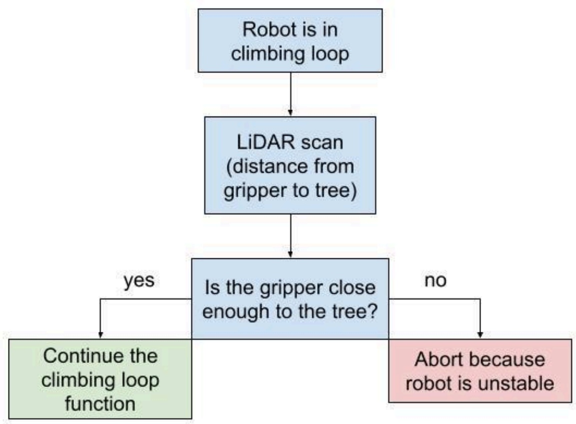


Figure 17: LiDAR Ascension Method

Along with the LiDAR ascension method, we will also be utilizing a barometric sensor to ensure that the robot is ascending upward properly. In order for this to work properly, the barometric sensor will need to first take a reading before attaching itself to the tree, then as it climbs it will take a reading every few cycles of the climbing loop and compare the current reading to the most recent one taken before it. If the current reading is lower than the past reading, this indicates that the pressure has dropped, which means that the robot has ascended to a higher altitude than it was during the previous reading.

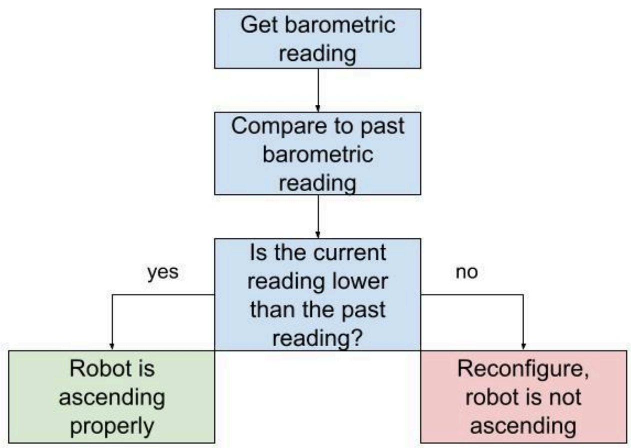


Figure 18: Barometric Ascension Method

Design Evaluation Methods

Analysis Cases and Results

An initial topical analysis was performed on the overall design for the centipede bot. Considering the original design with three grippers, this analysis showed that the gait cycle consisted of three fundamental steps of one claw releasing its grip, retracting from the surface, ascending to the next point, and gripping onto that point. To remove one of those steps, our topical analysis showed that by adding a fourth claw a full step in the gait cycle can be removed. With the fourth claw, the gait cycle now consists of two fundamental steps of claws one and three moving in unison and claws two and four moving in unison, both pairs performing the necessary release, ascension, and grip to allow for the robot to climb vertically up the tree.

The main concern for this design is its capability to remain attached to the tree. The group is targeting a stable open loop gait for the robot, which requires the robot to remain statically stable throughout its gait cycle. To determine the forces that are required to achieve static stability, a thorough force balance was conducted.

To begin, first the gripper design was analyzed to determine how it can remain fixed to a curved surface. In the figure below, a horizontal cross section of the tree is shown below at the point where one claw is grasping the tree. Each tooth presses against opposite sides of the tree, and squeeze the tree. This action generates the friction needed to fight gravity and prevent slipping down the tree (which are not shown in the diagram below, as they act perpendicular to the cross section below). However, this action also generates a normal force reaction that risks pushing the robot off of the tree. This force is also combated by the friction force. The points at which the teeth make contact with the tree can be modeled as slopes with a certain angle from the horizontal. The span of the tree being gripped is therefore twice this angle. For example, if theta is 90 in the diagram below the robot is grabbing across a 180 degree span of the tree, and there is no normal force reaction in the y direction.

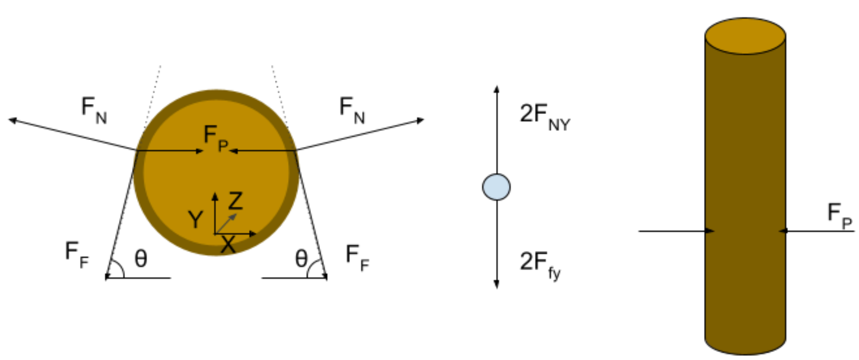


Figure 19: Free Body Diagrams of Robot Based on Tree

Using the diagram above, a force analysis was conducted. The system is assumed to be symmetric about the y axis, so the forces acting in the x direction cancel out. This results in some portion of the normal force acting to push the gripper off of the tree, and some portion of the resulting friction acting to combat this force to achieve static stability. Assuming no other forces are at play, this leads to equation 3 which relates the coefficient of friction of the surfaces to the angle of contact. For the contact between plastic and metal, the coefficient of friction is estimated to be 0.5 (Department of Physics and Astronomy at Douglas College, and OpenStax). Making this substitution leads to equation four below, where theta must be between 90 and 65 degrees. This offers a 25 degree margin on either side of the tree where the gripper can make contact and remain stable.

$$\textit{Force balance in Y direction:}$$

$$F_f * sin(\theta) = F_N * cos(\theta) \quad (1)$$

$$F_N = \mu * F_N \quad (2)$$

$$\textit{Combining equation 1 and 2:}$$

$$\mu \geq \frac{1}{tan(\theta)} \geq cot(\theta) \quad (3)$$

$$\textit{Assuming } \mu = 0.5:$$

$$65 \leq \theta \leq 90 \quad (4)$$

The previous analysis only considers one gripper and its requirements to remain attached to the tree. The friction force above needs to prevent the robot from "popping off" of the surface of the tree, but friction is also needed to prevent slipping down the tree in the z direction. This force is needed as a result of gravity acting on the robot parallel to the tree's surface. To calculate this component of the friction force, a free-body diagram of the whole robot is needed. In the diagram below, the robot is assumed to only have two grippers (labeled B and D) attached to the tree surface, while grippers A and C have been released and are being actuated up the tree. During this process, grippers B and D are modeled as pinned connections to the tree, which is assumed to be grounded. This means that each gripper has two reaction forces. The weight of the robot is assumed to act on the center of gravity of the robot, which is offset from the tree and is halfway between the two hardware boxes shown in the mechanical analysis earlier. The free body diagram is shown below. It is important to note that the system is again symmetric across the yz plane, so all forces perpendicular to the yz plane will cancel out.

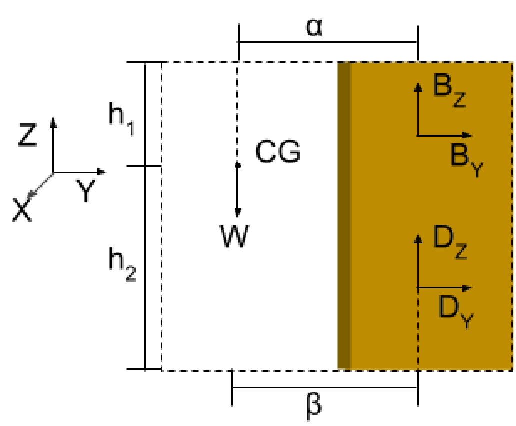


Figure 20: Free Body Diagram for Both Forces into Wall

To solve for the two reaction forces at each pinned connection, force and moment balance equations are used. The force balance in the y and z directions are shown below, along with the moment balance along the center of gravity and gripper B. The center of gravity is assumed to be in the center of the robot. The distance along the y-axis between the grippers B and D and the center of gravity are defined as α and β respectively the distance along the z-axis between the center of gravity and grippers B and D are defined as $h_1$and h_2 respectively.

$$\sum F_z = B_z + D_z - W \quad (5)$$

$$\sum F_y = B_y + D_y \quad (6)$$

$$\sum M_{cg_x} = \alpha B_z + \beta D_z + h_2 Dy - h_1 * B_y \quad (7)$$

$$\sum M_{B_x} = \alpha W + (\beta - \alpha) D_z + (h_1 + h_2) D_y \quad (8)$$

These four equations above were then transitioned into a system of linear differential equations in the form $Ax = b$. This system is shown below in the first figure. After solving the system, the solution is shown below. It is important to note that the system is rank deficient, so there are infinite solutions. This comes from the fact that if $\alpha = \beta$ then the two forces are parallel and collinear, and could be modeled as a single force in the z-direction. How this force breaks down into the two reaction forces is undefined here. This leads to the results below, where depending on the arbitrary value q the portion of the reaction forces in the z direction is divided between the two pinned connections. For this analysis, it will be assumed that this force acts all on one gripper to simulate a worst-case scenario.

0	1	0	1		By		W
1	0	1	0		Bz		0
0	0	h1+ h2	b-a	*	Dy	=	-W*a
-h1	a	h2	b		Dz		0

By		(W*a)/(h1 + h2)	
Bz		W- q	
Dy	=	-(W*a)/(h1 + h2)	
Dz		q	Let 0<q<W

Figure 21: Reaction Force Solution

These two forces need to be supplied by the friction and normal forces as well. The result of these calculations lead to the applied forces shown in the figure below. The top gripper B must be analyzed further, as friction needs to provide the By and Bz forces.

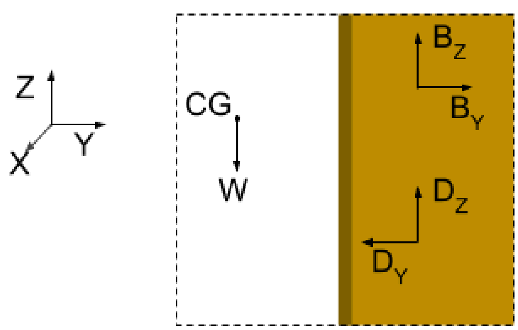


Figure 22: Free Body Diagram for Forces in Opposite Directions

Since there are additional forces that need to be counteracted by friction, equation 3 is no longer sufficient to achieve stability. Friction needs to combat the normal reaction, and provide additional force at joint B to counteract the moment caused by gravity and the joint A. This static analysis is combined with the analysis above to create a more accurate result below. This inequality relates the coefficient of friction of the surface contact with the contact angle and force being applied by the gripper.

$$\mu \geq cot(\theta) \pm \frac{\alpha W cos(\theta)}{P(h_1+h_2)}$$

Impact on System Design

The analysis above impacted the system design in various ways. The decision to shift to four grippers was made to simplify the analysis and break the gait cycle into two mirrored steps. This helps simplify the analysis as only one step needs to be analyzed and the results can be mirrored to the second step. Another important impact was splitting the hardware subsystems between two boxes on the second and fourth grippers. This helps reduce the movement of the center of gravity of the robot during each step, and again creates similar movements between the two major steps.

The gripper analysis helped drive the mechanical design of the robot. Various factors can be adjusted to decrease the span of the tree that it must grasp to remain stable. By having a smaller required span, the robot will have a larger margin of safety and will be more likely to achieve static stability. Increasing the step length, reducing the weight, and keeping the robot close to the surface of the tree can all have this effect. Squeezing the tree harder can also help, however this runs the risk of increasing the pressure at the contact point too high and making the linear coulomb friction assumption break down. Further experimental testing will be done to find the optimal values for these factors.

Alpha Prototype: Plan & Budget

The team's prototype planning involves the utilization of a Gantt chart to ensure the adherence to set deadlines. Figure 23 illustrates that the duration of milestone 3 spans approximately 6 weeks, and describes how the team plans to use that time. Commencing in early January, the team initiated the planning phase, specifically determining which components to order based on the CAD models generated previously. The total budget of the project includes the $700 from Stevens Institute of Technology, as well as the additional $2000 from L3 Harris. This additional funding facilitates the acquisition of high-quality parts, swapping from the planned plastic (PLA) parts to metal for most parts. The plan for the alpha prototype is to test the claw assembly, as this is the most important aspect to test.

Milestone 3 places a primary emphasis on achieving an alpha prototype for the claw design and subsequently progressing to develop a prototype for the entire arm. By prioritizing the prototyping of the claws for the alpha prototype, the team aims to manage expenses within the allocated budget, recognizing the fact that a total of four of these claws will be needed for the complete prototype. When formulating the budget, considerations need to extend beyond the claw design and articulated manipulator systems to encompass the cost considerations of electrical hardware as well, including an Arduino Uno board and an NVIDIA Jetson Nano kit.

In discussions within the team, there was a deliberate evaluation of vendors to expedite the robot's production. By focusing on certain vendors such as DigiKey, McMaster-Carr, and Amazon, the group can optimize shipping times and expedite production.

This strategic vendor selection is geared towards optimizing the production timeline and ensuring the timely progress of the prototype development. The comprehensive planning approach, as illustrated by the Gantt chart and budget allocation, reflects the team's commitment to meeting project milestones efficiently and effectively.

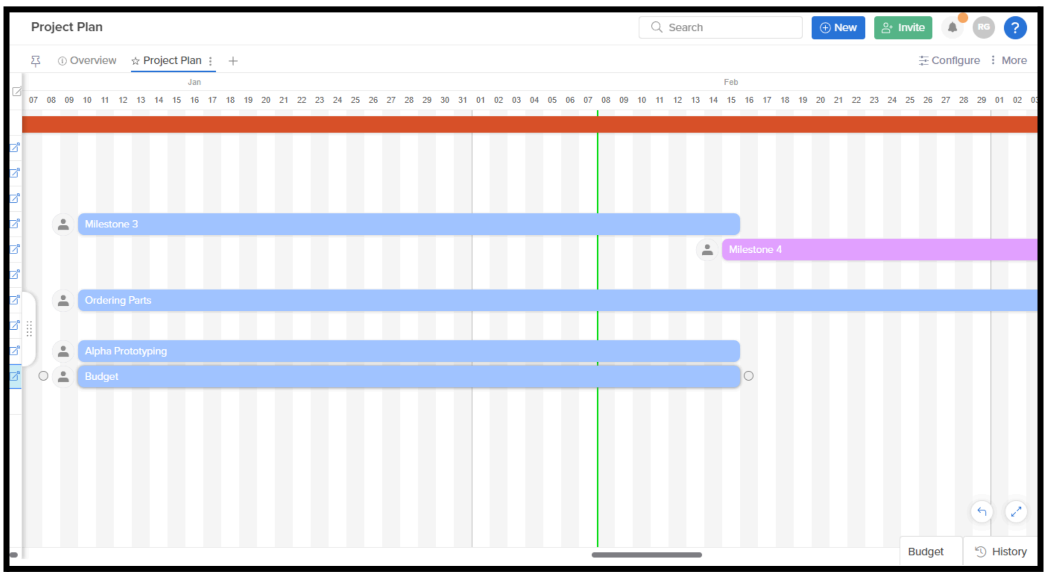


Figure 23*: Gantt Chart*

The spreadsheet in Figure 24 details the required orders for the claw design, amounting to a total cost of $1077.20 for all four claws. Notably, the additional comments section highlights a cost-saving measure by cutting the 2 ft long rack into smaller sections, allowing for its use in creating at least two claws. After covering the expenses related to the claw design, there is a remaining budget of $1622.80.

The spreadsheet also factors in the costs associated with electrical hardware, recognizing its significance in influencing the design of the overall arm. Specifically, the one Nano kit and four Uno boards contribute to a total cost of $216.96. Taking into account both electrical and mechanical equipment, the final amount totals $1,294.16. This leaves Arboreal Ascenders with a remaining budget of $1405.84, which can be allocated for the development of the articulated manipulator.

This financial breakdown reflects a thorough and strategic approach to budgeting, ensuring that funds are allocated judiciously for various components while leaving room for subsequent project phases. The team's consideration of both immediate needs, such as the claw design, and future requirements, such as the articulated manipulator, will allow for the successful prototyping of the entire robot.

Product Name / Description	Part Number	Vendor / Source URL	Quantity	unit Price	Price	Additional Comments
Extension Springs with Hook Ends	3114T74	https://www.mcmaster.com/produ	1	$ 12,00	$ 12,00	
Extension Springs with Loop Ends	8594N124	https://www.mcmaster.com/produ	2	$ 5,51	$ 11,02	
Positional Rotation DC Motor Servomotor	CN0023	https://www.digikey.com/en/prod	1	$ 11,11	$ 11,11	
37mm Spur Gear Motor w/ Encoder	2685-MOT-I-81566	MOT-I-81566-E ISL Products Internat	1	$ 59,20	$ 59,20	
NVIDIA Jetson Nano Kit	945-13450-0000-	Amazon.com: NVIDIA Jetson Nano K	1	$ 149,00	$ 149,00	Just need 1
Elegoo UNO R3	EL-CB-001	Amazon.com: Elegoo UNO R3 Placa	1	$ 16,99	$ 16,99	Need 4 for each claw
Linear Motion Shafts	6061K101	Linear Motion Shaft, 1566 Carbon St	4	$ 6,57	$ 26,28	
Metal Gears and Gear Racks—20° Pressure Angle	6832K62	https://www.mcmaster.com/produ	3	$ 22,52	$ 67,56	
Metal Gear Rack - 20 Degree Pressure Angle	7854K15	https://www.mcmaster.com/produ	1	$ 110,00	$ 110,00	This is for two claws
Set Screw Shaft Collars	6432K12	Set Screw Shaft Collar, for 1/4" Dian	4	$ 1,57	$ 6,28	
D-Profile Rotary Shaft	8632T139	D-Profile Rotary Shaft, 1045 Carbon	1	$ 10,86	$ 10,86	
C3 EMQ Premium Sealed Radial Ball Bearing 10 Pack	R4 2RS	Amazon.com: QBBC R4-2RS (1/4"x5/	1	$ 9,99	$ 9,99	
				Total before shipping	**$ 490,29**	
				Total shipping	**$ 17,18**	
				Total	**$ 507,47**	
				Total for each claws w/o racks	**$ 214,30**	
				Total for claws w/o racks	**$ 857,20**	
				Total for claws w/ racks	**$ 1.077,20**	
				Funding (Stevens & L3Harris)	**$ 2.700,00**	
				Budget left over (after claws)	**$ 1.622,80**	
				Nano kit & Uno boards	**$ 216,96**	
				Total for claws and electrical equipment	**$ 1.294,16**	
				Budget left over (after claws & electrical boards)	**$ 1.405,84**	

Figure 24: *Spreadsheet of budget for claw design*

Testing and Evaluation: Plan & Results

The current focus for analysis in the alpha prototype centers around the claw subsystem, which constitutes one quarter of the entire system. This targeted examination aims to validate the successful operation of our closed-loop system. The chosen claw design below is intentionally straightforward, featuring a geometry that induces high stress concentrations at its edges. While acknowledging the anticipated drawbacks arising from this concentrated stress such as increased deformation, the team views this initial design as a pivotal step. The successful functionality of this basic claw design serves as a crucial validation step, and should lead to subsequent improvements in the design.

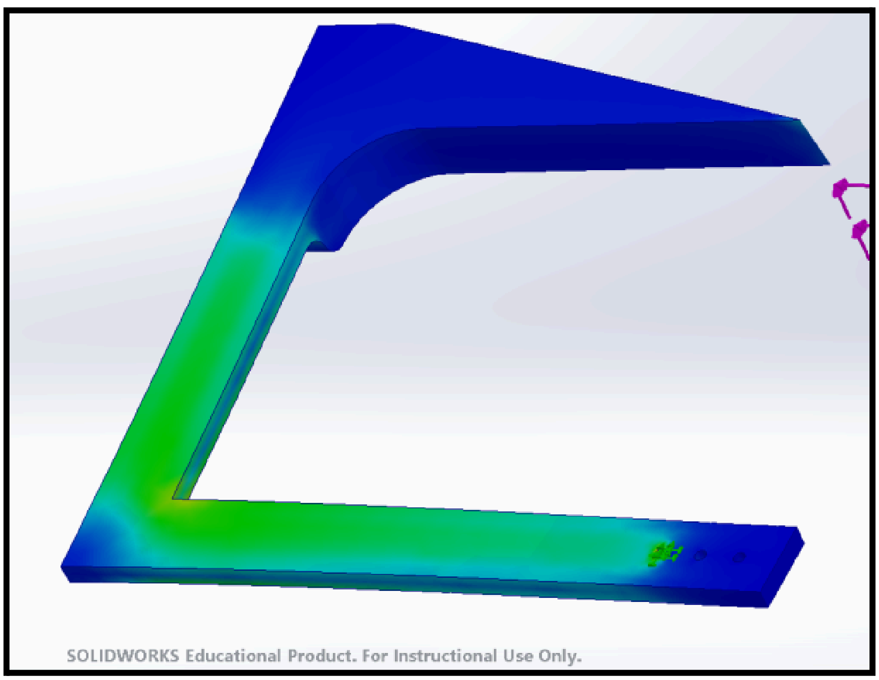


Figure 25*: Fundamental Design*

As depicted in Figure 25 above, the team's primary design underwent a stress analysis in SOLIDWORKS to assess its ability to withstand the stress generated during tree clamping. This analytic approach ensures that the claw can perform as intended under real-world conditions. Due to the time invested in acquiring both mechanical and electrical components, the team has prioritized the development and analysis of the fundamental design above. The outcomes of this analysis will provide valuable insights into the feasibility and performance of the current claw design, serving as a foundation for iterative improvements and optimizations in subsequent phases of the project.

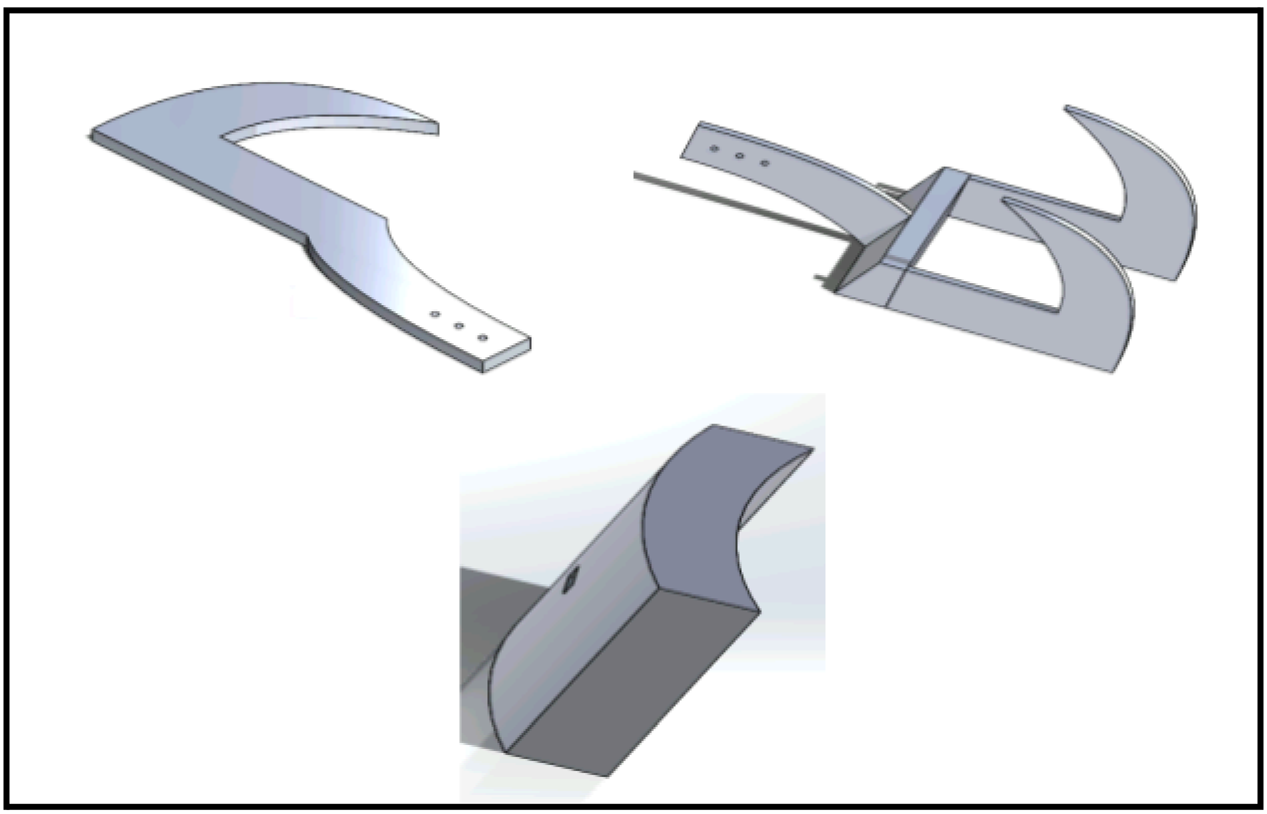


Figure 26*: Alternative Claw Designs*

Figure 26 (as seen above) presents various design styles that the team contemplated for the claw. Departing from the pointy edge of the main design, the alternative designs aim to reduce stress on the claw. The design on the left introduces two curved arcs strategically placed near the screw holes and the clamping area. This configuration is intended to distribute stress more evenly. The design on the right takes a similar approach but features three claw-like elements, providing two points of contact on each side. The theory behind this design is to increase surface area contact, subsequently reducing stress and pressure during tree clamping. A larger surface area not only minimizes stress but also enhances the probability of the claw adhering to the tree.

Below these two designs is a paddle design, introducing potential challenges related to sticking to the tree. To address this issue, a solution is proposed: the paddle will be composed of two different materials. The part in contact with the tree will have weaker mechanical properties, allowing for deformation to improve adherence, while the claw connected to the paddle will be made of a material capable of handling stress after the paddle adheres to the surface. These alternative claw designs represent considerations for the beta prototype. Further exploration and analysis of these designs will be conducted to assess their feasibility and performance, paving the way for improvements and refinements in later stages of the project.

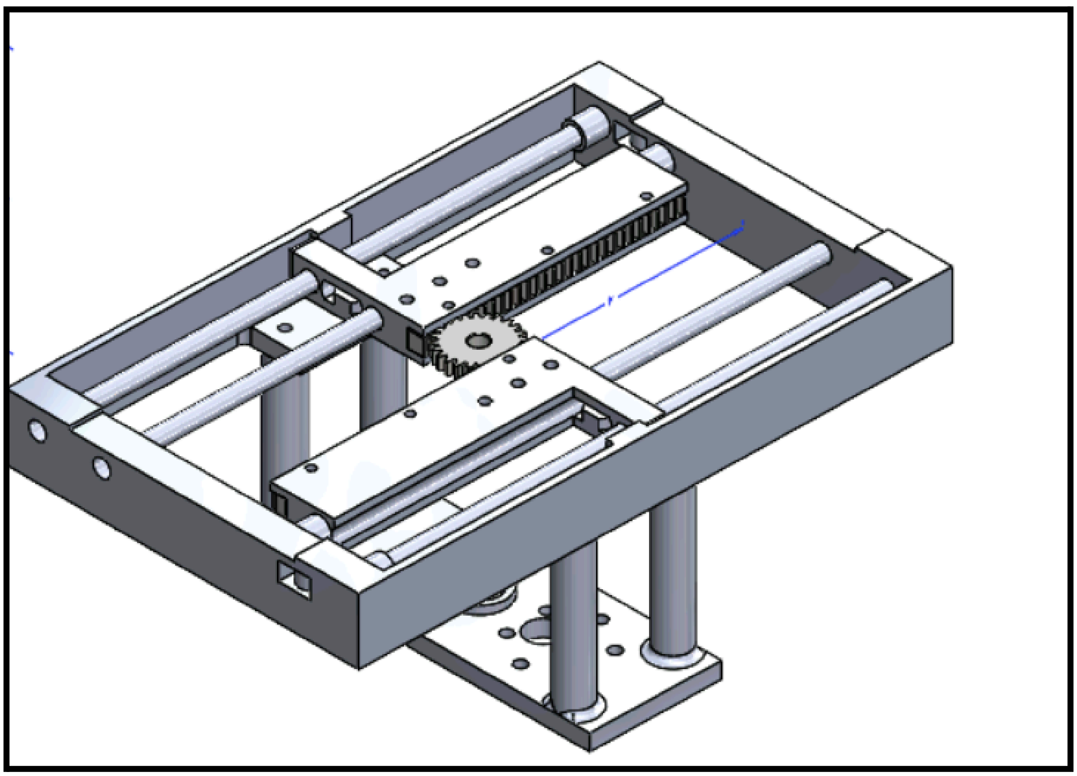


Figure 27*: SOLIDWORKS Housing Assembly*

The claws are planned to be operated through a rack and pinion mechanism, facilitating the expansion of the spring as a result of the force applied by the motor. Figure 27 illustrates the assembly, showcasing the housing and the operational configuration of the claws. This configuration represents the envisioned final design that the team aims to achieve. However, the team acknowledges that refinements may be necessary in response to any issues that may arise during the testing and development process.

The depicted assembly provides a visual representation of the intended integration of the rack and pinion mechanism within the housing, showcasing how the claw will be operated. This design aligns with the team's current vision for the final product, but a commitment to ongoing refinement underscores the team's dedication to addressing challenges and optimizing the overall performance of the system. As the project progresses, testing and iteration will play a crucial role in ensuring that the final design meets performance expectations and overcomes any potential obstacles. This iterative approach allows the team to continuously enhance the functionality and reliability of the claw mechanism.

The rack is connected to the mounting point for the claw. This rack is mounted to a larger L-shaped piece which has through holes to guide the placement of the claws and house the screws. The motor, situated at the bottom, supplies power to drive the rotation of the gear connected to the rack and pinion. Another gear meshes with it, sharing the same shaft as the pinion. Ball bearings are strategically attached to both the top and bottom ends of the axle to reduce rotational friction, providing minimal support for radial and axial loads.

The alpha prototype was constructed using purchased steel gears, screws and axles due to the fact that they must be weight bearing along with various 3D printed housing pieces and a motor. This was done to allow for rapid prototyping of the arrangement of the force bearing components as different discoveries were made during prototyping. The group could also drill new holes where needed to accommodate additional components being added. The motor was powered using a custom cable connected to the dc power supply in the makerspace to ensure the motor is strong enough to fully open the assembly.

During the assembly of the alpha prototype, a practical test was conducted in which the group utilized a segment of an ideally shaped tree to validate the results, with an emphasis on evaluating the fundamental design. However, the outcomes revealed a significant flaw in the design, as illustrated in Figure 28 below. The entire sub-assembly failed to stay perpendicular to the tree, exhibiting a downward tilt. This is due to the area of contact of the claw being too small to provide a moment response, acting more like a pinned connection than a fixed connection.

Figure 28*: First alpha prototype*

The rest of the robot will weigh the robot further, exaggerating this lack of stability. In the initial testing phase, the team had intended to attach a bucket as the payload to evaluate the model's performance. However, it was recognized that applying a load to the claw would increase the torque on the system, potentially leading to unclamping from the contact surface.

This insight highlights the need to carefully assess the system's ability to handle varying loads and moments. The team acknowledges the potential challenges associated with payloads and torque, and this understanding will guide further refinements and improvements in the design. It emphasizes the importance of developing a claw mechanism that not only accounts for its primary function but also considers the system's stability under different operational conditions, including the introduction of external loads.

While this outcome did not invalidate the housing, rack and pinion, and spring components, it underscored the importance of addressing the issue of surface area contact. The results emphasize the need for an increased surface area contact to ensure the stability and effectiveness of the claw mechanism during operation. This realization serves as valuable feedback for the team, guiding future iterations and refinements to enhance the overall performance and reliability of the system.

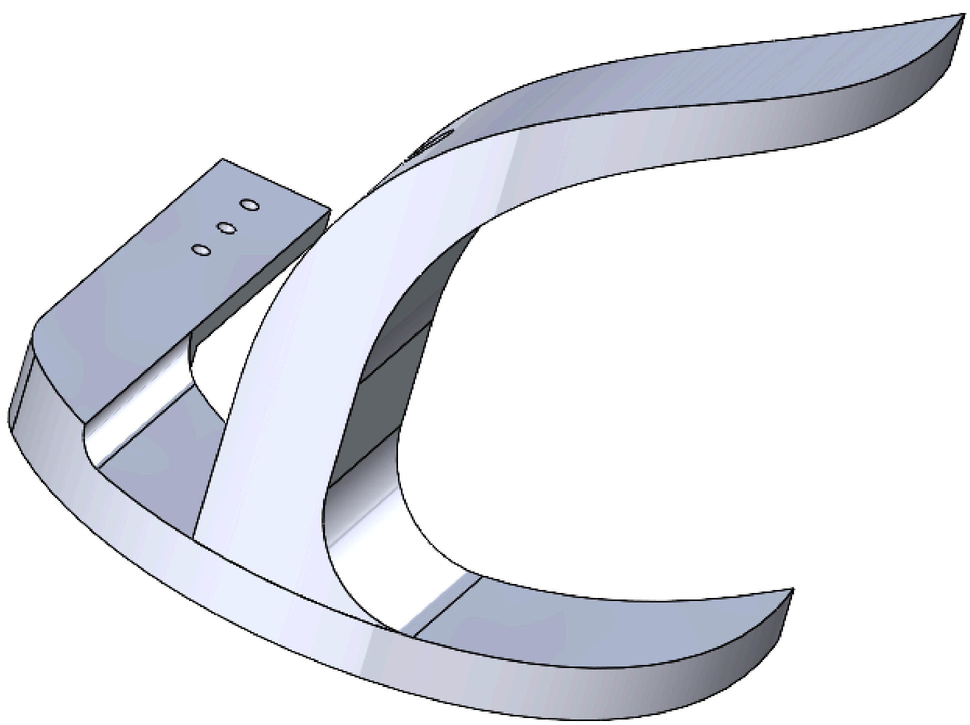


Figure 29: *New dual claw design model*

The team has introduced a completely new claw design above, showcasing two points of contact on each side of the tree. The claws are connected by screws, linking the gray and red PLAs. Figure 30 shown below illustrates this innovative design, although there are noted dimension offset issues. Instead of the initially planned 4-point contact, the design currently concentrates on a 3-point contact configuration. The team also tested this design with the motor to ensure it can open and close properly. This test was recorded and is available in the link below.

https://drive.google.com/file/d/1uHFy3LysFyEvcIRgqnISFzNHhCeTv-M5/view?usp=sharing

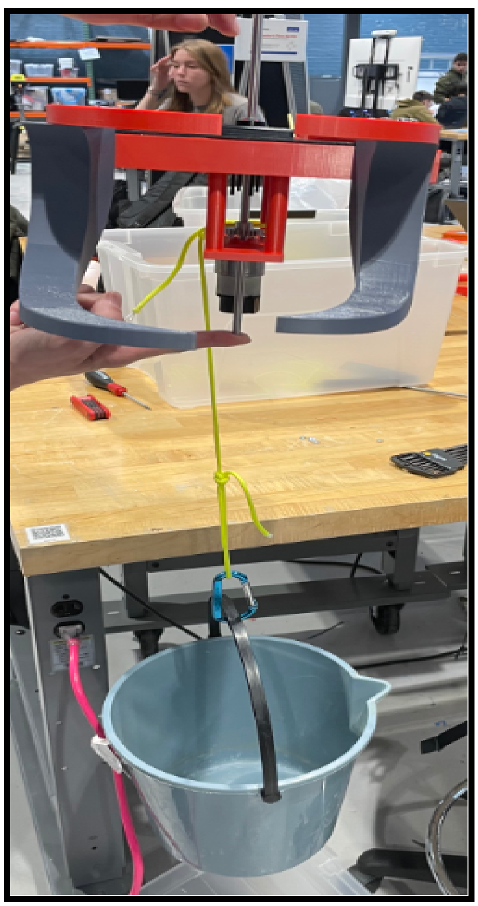


Figure 30: *Fully updated claw design with bucket*

This newly developed claw design is set to undergo further examination and refinement to address the dimension issues and potentially enhance the number of points of contact. The bucket in the illustration serves as a representation of the payload for the entire system. The payload is expected to support approximately 9 lbs of weight, which equates to roughly 9 water bottles at 1.1 lbs per bottle.

By addressing dimension issues and exploring additional points of contact, it will aim to enhance the performance and reliability of the claw mechanism. As the design evolves, further testing and adjustments will be crucial to ensuring that the system meets the specified requirements and can effectively handle the intended payload.

In Figure 31, an updated version of the alpha prototype is presented, showcasing a claw design with a relatively small tilt compared to the initial version. The team's analysis has demonstrated success, as the claw remained stable enough to hold the payload while connected to the tree. This positive outcome validates the effectiveness of the modifications made to the design.

Figure 31*: Current final alpha prototype with payload*

Moving forward, the team plans to consider an increase in the factor of safety through higher gear ratios to further enhance the stability and reliability of the closed-loop system, as the motor begins to struggle when the claw is fully open. By adjusting and reinforcing elements in the design, the team aims to optimize the performance of the claw mechanism, ensuring its success in handling both the intended payload and various real-world conditions.

The next steps for the team involve iterative testing of various claw designs to optimize stress handling, enhance tree contact, and ensure the system remains perpendicular to the tree by reducing deformation of the claw. Following the refinement of the claw mechanism, the focus will shift to the assembly of the arm that connects each claw to the other, a critical component for the complete autonomous robot. Considering the remaining budget, careful planning and considerations will guide the procurement of parts for the arm assembly.

Simultaneously, attention will be directed towards the electrical and software aspects, constituting the brains of the design. These components are crucial for executing the intended gait cycle, contributing to the overall functionality of the autonomous robot.

With one quarter of the mechanism completed during the beta prototyping phase, the team anticipates the need to replicate the process three more times to construct the entire system. This iterative approach allows the team to learn from each phase, refine the design, and work towards an optimal solution. Expanding the Gantt chart will be essential for managing goals within the given time constraints. Additionally, for the upcoming prototyping phase, the team plans to order from vendors that may be slightly more costly. This decision is made with the intention of allowing more time for the fabrication of each physical model, emphasizing the importance of thorough testing and refinement in achieving the desired outcomes.

In summary, the team's strategy involves a systematic and iterative approach, addressing different components of the autonomous robot to achieve an optimized and functional end product. Thorough planning, testing, and adaptation to feedback will drive the project forward toward its goals.

Beta Prototype

Mechanical Design

The team made significant modifications to the mechanical design, including reducing the housing volume, adjusting the curvature of the claws, introducing a spine component, and implementing a wrist mechanism. Reducing the housing volume was done to lower material costs, decrease weight, and increase the maximum travel of the claw tips. Figure 32 provides a side-by-side comparison of the initial housing design and the team's latest iteration, highlighting these changes.

Figure 31: Old Design vs New Design

The design incorporates claw curvature to ensure contact that can grip onto the tree past its max diameter.. Compared to the previous design, the claws are thicker, enhancing durability under stress and lowering failure risks. Figure 32 shows the claws making flush contact with the tree housing, introducing acceptable amounts of deformation as compared to the previous design. This robust design effectively establishes a stable connection with the tree and successfully resists loading from the payload's weight and gravity to ensure stability and reliability.

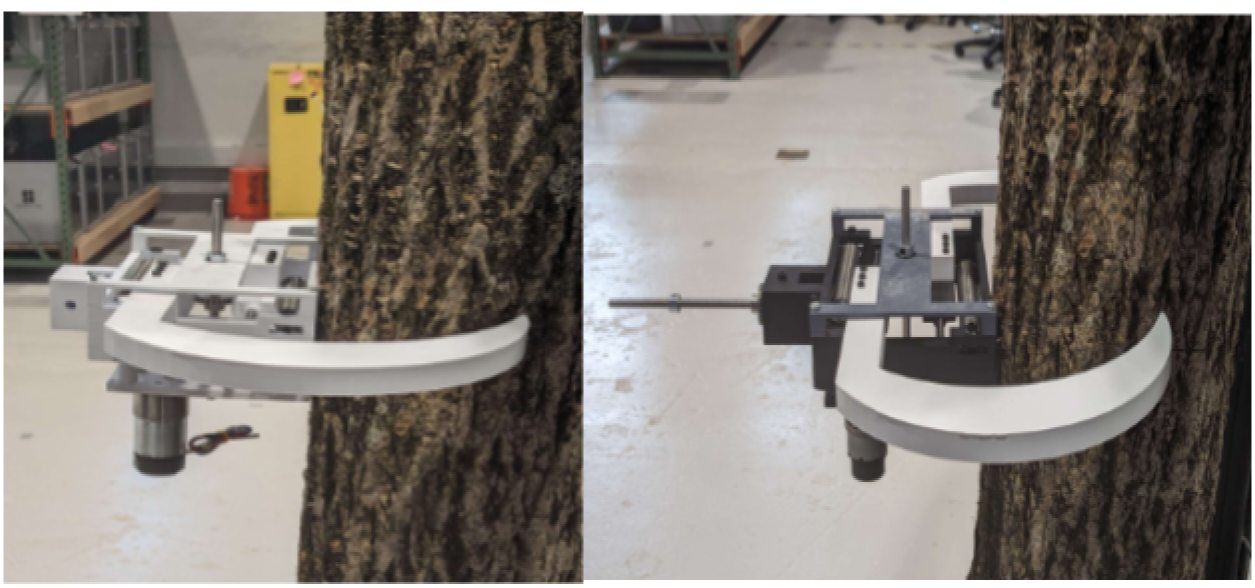


Figure 32: Side view of the new claw designs.

At the rear of the housing, two interlocking gears will facilitate the expansion and contraction of the robot's body, akin to the movement of a caterpillar as it adjusts its body. The team has concluded that transitioning from a four-claw mechanism to a simpler two-claw mechanism is feasible. The body of the robot is shown in Figure 33, it will have a linear position when fully expanded and it will compress up to an angle of 30 degrees.

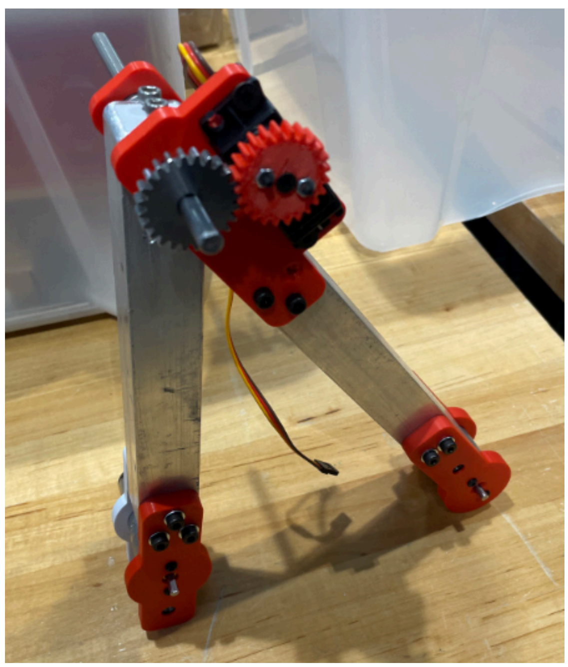


Figure 33: Spine of the robot

Figure 34 displays the body structure designed to maintain the robot's integrity. The two-step mechanism will operate by having one claw unclamp from the tree while the other remains clamped. This action will cause the body to either compress or expand, depending on which claw is in the clamped or unclamped position. This alternating clamping and unclamping of the claws will enable controlled movement and adjustment of the robot's position and grip on the tree.

Figure 33: Full Mechanical Assembly of Beta Prototype

Electronic/ElectroMechanical Design

The team has significantly reduced the number of motors required for the robot design based on previous iterations and proof of concept. Initially, the concept called for over 20 motors due to the high number of degrees of freedom in the wrist mechanism. The wrist was originally intended to function with prismatic, roll control, and pitch control capabilities. However, after refining the design, the focus shifted primarily to roll control with minimal pitch control.

Additionally, transitioning to a two-claw mechanism also contributed to the reduction in the number of required motors. As a result, the current design only needs five dc/Servo motors for the robot to operate effectively. This reduction in motor count streamlines the design, reduces complexity, and likely contributes to improved efficiency and reliability of the overall system.

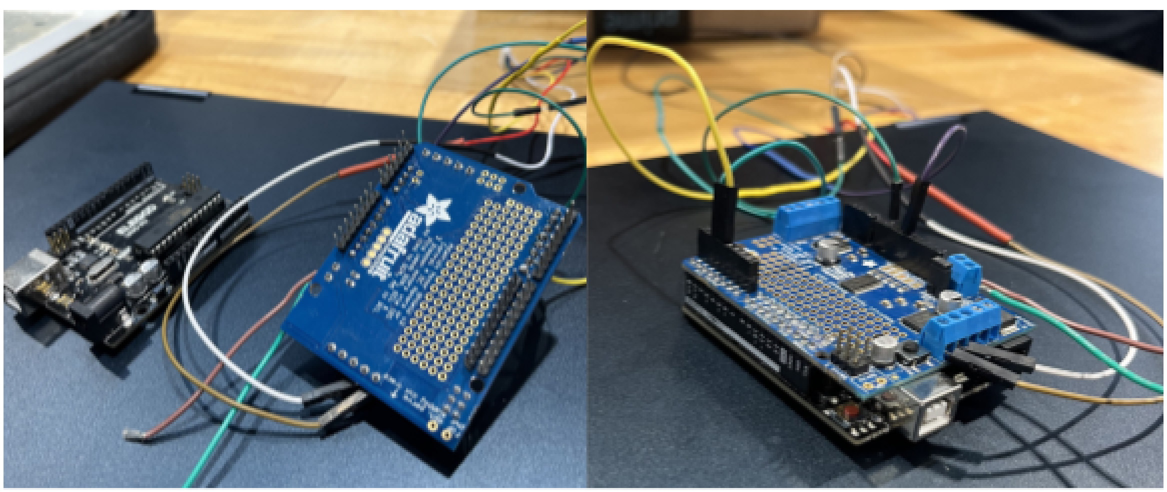


Figure 34: Arduino and Motor Shield

Prior to the realization that the robot would require less motors, the group had invested in a motor shield as seen in Figure 34, in order to efficiently run all of the motors. Despite this change, the motor shield is still necessary to ensure that all of our motors are receiving the proper power needed to run, along with the added benefit of better organizing them in the circuit, rather than having an excess of loose wires in the system. The motor shield isolates the servo power supply from the arduino, allowing the group to utilize current. The servo power supply is currently a 6V 5A supply capable of running all 5 motors at once.

In order to properly power the circuit, the robot will be utilizing both the usb input on the arduino board, along with a power supply. The motors are capable of running solely off the power supplied from the arduino board at 5V, but the external power supply allows the motors to run at full capacity without limiting their current draw. This will be necessary for the robot to support its weight while completing its climb as quickly and efficiently as possible.

The current build of the robot relies on 4 dc motors with encoders and one servo, which are all driven with the motor shield shown above. Later on, two of these dc motors will be swapped for servo motors so that the robotic arm enabling the robot to scale the tree is entirely operated by servo motors, with the 2 dc motors remaining opening and closing the claws. At that time, the system will be split across two driver boards, one dedicated to the dc motors and one dedicated to the servo motors.

Software Design

Before work could begin towards coding the robot in its entirety, the group first had to create a program to control the position of the dc motors to allow them to act similar to a RC servo motor. In order to do this, the group designed a PI controller for each motor that takes the position of the motor based on its encoder, and guides the motor to a set position using control actions. A portion of this action is proportional to the current error, and a portion of this control action is proportional to the sum of all previous errors. The controller begins by requesting the encoder count generated by the motor encoders from the motor to the arduino board. Then, it determines the error between its current orientation and the desired orientation. The current error gets used in two ways, being multiplied by kp to determine the proportional action and being summed with previous errors and then multiplied by ki to determine the integral action. The sum of the proportional and integral actions then becomes the controller's speed command that gets sent back to the motor. The two constants, kp and ki, were tuned through trial and error for each motor to achieve a low settling time with limited oscillations.

With the PI control systems tuned for all of the dc motors, the team then tackled the challenge of moving the robot with precision up the tree. To actuate the arm, the team actuated the arm to a sequence of different positions over time to simulate velocity based control. For each gait cycle, the robot first releases one of its claws. It then determines a list of waypoints that the released claw will follow to establish a linear trajectory that ends 5 inches above its starting point, before the claw then closes back onto the tree. To move from waypoint to waypoint, the group used a resolved rates algorithm to determine the necessary velocity to move the claw along the path while keeping the claw horizontal with the ground. The redundancy resolution problem was then solved in such a way to keep the norm of the joint speed vector as low as possible to reduce vibrations in the system.

Since the claw mechanisms are actuated using dc motors that lack encoders, a limit switch is used instead to determine if the claw is fully open and another control loop is used to keep the claw open as wide as possible during movement. This system ensures that the claw remains open wide enough to allow the system the necessary margin of error needed when traveling along its path. Furthermore, an inertial measurement unit will be added to each claw to allow the system to detect when the claw rotates and the system is capable of taking action to counteract these movements. This is especially important for the claw that remains fixed to the tree during movement, as it may shift slightly as its loading changes as a result of the other claw moving. Being able to adjust for these shifts during movement is essential, and mimics the leg slipping problem walking robots also face.

System-Level Design

The gait cycle of the Arboreal Ascender robot at the system level remains largely unchanged, as depicted in Figure 35. The notable difference is the reduction in the number of claws and the body length was reduced by half.

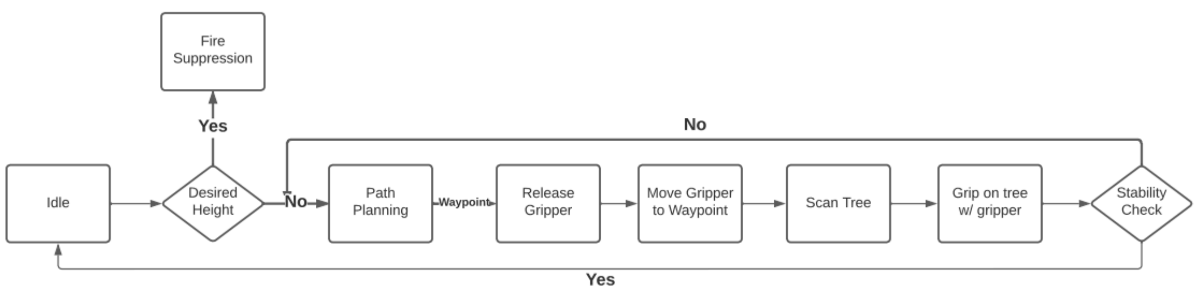


Figure 35: Gait Cycle remains the same

Regarding the overall system design, the wrist mechanism underwent a reduction in one degree of freedom. Initially, the wrist mechanism included a prismatic joint, roll control, and pitch control. However, the team realized that the prismatic joint would not be beneficial due to the vertical and linear nature of the path along the tree. This reduced the need for extensive pitch control in the wrist and claw mechanism to maintain perpendicularity to the tree. Removing the prismatic joint not only simplifies the system by reducing degrees of freedom but also cuts down on complexity, cost, and time required to perform the gait cycle. This optimization streamlines the design without compromising functionality, aligning the system more efficiently with its operational requirements. The gait cycle will focus heavily on the fundamental movements of the robot to serve as a foundation for the tree climbing robot.

Fabrication, Assembly and Manufacturing Processes

The fabrication, assembly, and manufacturing processes for the Arboreal Ascender robot involved multiple iterations of 3D printing using PLA material for the claws, housing, wrist, and body components. The body comprises a seven-inch long aluminum rectangular beam, which underwent processes such as hole drilling, sanding, and cutting in a machine shop to achieve the desired specifications. Various components such as gears, motors, and d-shaft collars were sourced from suppliers like Mc-MasterCARR, Amazon, and DigiKey. This process enabled the full assembly of the robot, integrating the 3D printed parts with the machined aluminum body and the purchased mechanical components. The fabrication process included designing the parts in SOLIDWORKS, followed by 3D printing them using PLA material. Post-printing processes such as sanding or polishing have been performed to ensure smooth surfaces and proper fitment during assembly. The machined aluminum beam was fabricated according to precise measurements and specifications to accommodate the assembled components effectively. Overall, this comprehensive approach involving additive manufacturing (3D printing), traditional machining, and component sourcing facilitated the successful fabrication and assembly of the Arboreal Ascender robot.

High-Performing Testing and Evaluation

The testing phase for the robot was simplified to focus on the functionality of the two-claw mechanism. This setup requires the implementation of five PI controllers, corresponding to the five dc/servo motors utilized in the design. However, due to limitations in the availability of components, the testing process faced challenges. Delays in receiving ordered products, such as Arduino boards, dc connectors, and power supplies, were experienced, partly due to delays from suppliers and receivers in the Mechanical Engineering (ME) Department. To work around these constraints, the team utilized resources available in the makerspace, such as high power dc power supplies, to perform basic validation tests. These tests aimed to evaluate the mechanical and electrical functionalities of the robot, although under suboptimal conditions due to the absence of some components. Despite the setbacks, the simplified testing approach allowed the team to gain insights into the core functionalities of the robot's mechanism and its electrical system. This phase likely served as an initial validation step, with more comprehensive testing planned once all necessary components are available. A video/photo of the wrist roll control of the two-claw mechanism is shown below.

Claw Spinning
Claw Pivoting

Final Prototype

Mechanical Design

The mechanical design has undergone several adjustments since the last milestone. Initially, the team employed a two claw-tip design which provided excellent grip on the tree. However, issues arose during the climbing process when the robot's lower half encountered difficulties. It would gradually slip off the tree as the lower section moved upward or compressed the robot's spine. To address this challenge, the claw design is being reconfigured to better support the robot's weight and ensure optimal contact with the tree. This involved adding extra claw tips that are interconnected with the original ones but extended downwards to match the height of the housing. This adjustment provides additional stability, as the housing conforms to the curvature of the tree. Figure 36 illustrates the updated claw design.

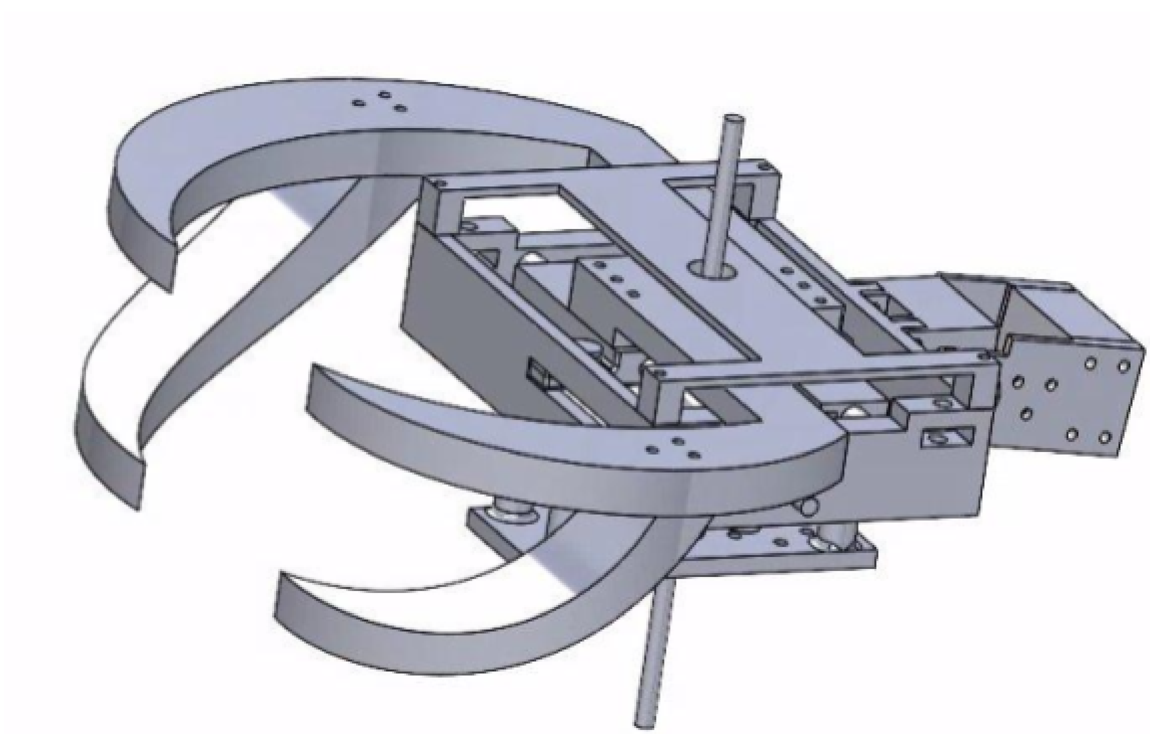


Figure 36: CAD Model of Final Claw Design

The final design has the claws attached to the spine of the robot through D-shaped rotary axles. These axles were integrated into the spine through 3D printed mounting adaptors. The initial iteration of this adaptor utilized the strength of the PLA filament to rigidly fix the axles to the claws when paired with set screws. However, after testing, it was found that the PLA was not sufficiently strong enough to support the total torque experienced by the D-shaped axel. Therefore, we redesigned the adaptor to fit milled aluminum inserts, which proved to be strong enough to withstand the maximum possible torque experienced in each of these points of connection.

The claws, rigidly fixed to rotary axles for roll, were then attached to the spine through printed mounts with fixed rotary bearings to permit the rotation of the claws with relation to the spine of the robot. These were then secured at their ideal location on the spine by simple shaft collars with set screws. The mounts designed for the attachment of the claws at the top and bottom of our final prototype are also designed to rigidly secure the driving servo motors to the spine and permit a 2:1 ratio between the rotary axel and the spine. Similarly, the middle joint of the spine was designed to rigidly attach the servo to the spine segment, created a 2:1 gear ratio, and attached the two segments of the spine at a rotational point. This fully integrated system featuring all of the described modifications above can be seen below in Figure 37.

Figure 37: Photograph of Fully Integrated Prototype in the Innovation Expo

Following the repeated iterations and testing performed on this robot and on this design project as a whole, there were many improvements and changes considered that could have improved the function of our robot. For consideration on further iterations of this entire project and design, we would suggest changing certain features to better develop a functioning proof of concept. The first of these improvements would be in the spine, which could function more efficiently as a more simplified linear system rather than our jointed system. This improvement can be justified with the argument of the work space of the robot being primarily linear and consistent. Another improvement would be the incorporation of better control systems, such as something relying on sensing data and encoder response to govern the speed and motion of the system; as was initially intended for this design. Lastly, we would suggest continual iterations to the claw tip design in order to determine the best design for tree surface and robot interaction, which would allow for both adherence and compliance.

Electronic Design

In the final electronic design of the robot prototype, the primary controller utilized is an Arduino Uno, which efficiently manages the operation of the entire system. To ensure proper voltage and amperage supply for the five motors, including three servo motors and two DC motors, a well-organized power distribution setup was implemented. The three servo motors, requiring a 6V supply at 3A each, were directly connected to a dedicated power supply unit capable of delivering the necessary current. This setup avoids overloading the Arduino Uno with motor power requirements, ensuring stable and reliable operation.

For the two DC motors, which operate at 12V and draw up to 4A each, a separate power arrangement was established. These motors were powered by a wall adapter connected to a DC motor driver through a barrel jack adaptor. This approach allows efficient control and power management for the DC motors, preventing voltage drops or instability issues that can affect motor performance. Additionally, the Arduino Uno itself was powered by a wall power supply, ensuring consistent operation without relying solely on USB power. This comprehensive power setup not only safeguards the Arduino from potential overloads but also optimizes the overall functionality and responsiveness of the robot during climbing and maneuvering tasks. The final electronic design thus demonstrates a well-engineered system that efficiently handles motor control and power distribution for successful tree traversal.

Software Design

For the final prototype of the robot, both the software and electrical aspects have undergone significant changes to accommodate the streamlined system requirements and operational efficiency. In terms of software design, the final prototype focuses on two main functions: spinal extension/compression and claw manipulation using a specific sequence of servo and motor commands. This approach simplifies the control logic, ensuring precise and coordinated movement for climbing and descending tasks. The spinal function drives the servo motors to extend or compress the robot's spine while maintaining the claws' horizontal orientation. This ensures stability during movement. The claw manipulation function controls the DC motors to open or close the claws around the tree trunk, allowing the robot to ascend or descend accordingly. By combining these functions in a sequential order, the robot performs a smooth climbing cycle, alternating between claw movement and spine adjustment as needed.

Appendices

American National Standard for Industrial Robots and Robot Systems: Safety Requirements. Robotic Industries Association, 2013.

Brooks, Malachi.“2023 North American Wildfires.” *Center for Disaster Philanthropy*, 22 Sep. 2023, disasterphilanthropy.org/disasters/2023-north-american-wildfires/#:~:text=The%20National%20Interagency%20Fire%20Center%20statistics%20show%20that%2C%20as%20of,acreage%20of%206.04%20million%20acres.

Rempfer, Kyle. “4 British Columbia Wildfire Fighters Killed in Head-on Collision, Police Say.” *The Washington Post*, WP Company, 21 Sept. 2023, www.washingtonpost.com/world/2023/09/20/firefighters-killed-british-columbia-canada-wildfires/.

“Standard for Dry Chemical Extinguishing Systems.” *NFPA 17: Standard for Dry Chemical Extinguishing Systems*, www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=17. Accessed 3 Oct. 2023.

ben_k, and Instructables. “Tree Climbing Robot.” Instructables, Instructables, 1 Nov. 2017, www.instructables.com/Tree-Climbing-Robot/.

“A Flexible Tree Climbing Robot: Treebot - Design and Implementation.” YouTube, YouTube, 4 July 2020, www.youtube.com/watch?v=wPGoWvKEOAw.

“Tree-Climbing Robot Can Harvest Coconuts Automatically | Mashable.” YouTube, YouTube, 9 Sept. 2020, www.youtube.com/watch?v=fRV4y0JslSg.

Department of Physics and Astronomy at Douglas College, and OpenStax. "5.1 Friction." *xDouglas College Physics 1107 Fall 2019 Custom Textbook*, pressbooks.bccampus.ca/introductorygeneralphysics1phys1107/chapter/5-1-friction/. Accessed 14 Nov. 2023.

BILL OF MATERIALS

Component	Part Description	Count	Manufacturer	Supplier	Description	Unit Cost	Total Cost	Status	Website
Mechanical									
1	Mechanical Arm Piece	6	N/A	Self	Custom milled component of rigid length between revolute joints	N/A	N/A	To Mill	N/A
2	Claw Ends	4	N/A	Self	Custom milled component	N/A	N/A	To Mill	N/A
3	Mechanical Spring	4	ACXESS SPRING	ACXESS SPRING	Linear compression spring	2.61	10.44	Need to Get	https://www.acxesspring.com/english/catalogsearch/advanced/result/?unit_measure=en&category=cs&cs_od%5Bfrom%5D=0.036&cs_od%5Bto%5D=9.25&cs_fl%5Bfrom%5D=0.13&cs_fl%5Bto%5D=2&cs_mt=0&form_key=4w7sCa5KhTiUzuDu
4	Rack And Pinion	8	SYLALE	Amazon	Rotational to linear system	51.78/2	207.12	Need to Get	https://www.amazon.com/SYLALE-Durable-Modulus-Precision-10x10x500mm/dp/B0CJC1MVJT/ref=sr_1_5?c=ts&keywords=Mechanical+Rack+%26+Pinion+Gears&qid=1699914325&s=industrial&sr=1-5&ts_id=16412191
5	Rack and Pinion Housing	8	N/A	Self	3D additive manufactured component	N/A	N/A	To Print	N/A
6	Electrical Housing	2	N/A	Self	3D additive manufactured component	N/A	N/A	To Print	N/A
7	Sheet of Aluminum	1	GetMetals	GetMetals	For custom milling	106.19	106.19	Need to Get	
Electrical									
8	DC Hobby Servo	15	Smarze	Amazon	Servo motors to provide rotational force to gears	18.77/10	37.54	Need to Get	https://www.amazon.com/SG90-9g-Micro-Servo-Motor/dp/B0C37CNZC8/ref=asc_df_B0C37CNZC8/?tag=hyprod-20&linkCode=df0&hvadid=673743447994&hvpos=&hvnetw=g&hvrand=13011679591138687377&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9003566&hvtargid=pla-2227359535594&psc=1
9	Arduino Uno	1	Arduino	STEVENS	To act as the follower in the software system	N/A	N/A	Need to Get	N/A
10	Raspberry Pie	1	N/A	Self	To act as the leader in the software system	N/A	N/A	Owned	N/A
11	Servo Power Board	2	goBILDA	ServoCity	Power supply for driving servo motors	12.99	25.98	Need to Get	https://www.servocity.com/8-channel-servo-power-node/
12	Lidar	1	DIYmall	Amazon	Light emitting distance sensor for depth from climbing surface	26.59	26.59	Need to Get	https://www.amazon.com/Benewake-TF-Luna-Single-Point-Ranging-Interface/dp/B086MJQSLR/ref=sr_1_2_sspa?hvadid=241917423558&hvdev=c&hvlocphy=9003566&hvnetw=g&hvqmt=e&hvrand=10115611183706482731&hvtargid=kwd-297844030301&hydadcr=24663_10400926&keywords=lidar+sensor&qid=1699916219&sr=8-2-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1
13	Barometric Sensor	1	Arduino	Amazon	Height judging sensor utilizing pressure measurement	6.99	6.99	Need to Get	https://www.amazon.com/ACEIRMC-Temperature-Barometric-Pressure-Arduino/dp/B091GWXM8D/ref=sr_1_1?crid=1FXA2XA9GA91&keywords=Barometric+sensor&qid=1699916486&sprefix=barometric+sensor%2Caps%2C87&sr=8-1
						Total Tentative Expenses:	420.85		