

April 2012

DESIGN AND CONSTRUCTION OF A TREE CLIMBING ROBOT

Erick T. Read

Worcester Polytechnic Institute

Justin A. Gostanian

Worcester Polytechnic Institute

Follow this and additional works at: <https://digitalcommons.wpi.edu/mqp-all>

Repository Citation

Read, E. T., & Gostanian, J. A. (2012). *DESIGN AND CONSTRUCTION OF A TREE CLIMBING ROBOT*. Retrieved from <https://digitalcommons.wpi.edu/mqp-all/3612>

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.

DESIGN AND CONSTRUCTION OF A TREE CLIMBING ROBOT

An Major Qualifying Project Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Justin Gostanian,
Computer Science Major

Erick Read,
Robotic Engineering Major

April 26, 2012

Michael A. Gennert, Project Advisor

This report is the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review.

Abstract

This Project is on the design, construction, and testing of a robot to climb trees to detect Asian Longhorn Beetle infestation. The primary goal was to design and build a robot that could successfully climb a tree. After researching existing climbing robot designs, a robot prototype was built using concepts from the existing designs. The prototype was then tested to determine the effectiveness of the design. The prototype proved to be partially successful, being capable of gripping a tree and staying on, but could not move. Though not entirely successful, the project identified many important aspects in a tree climbing robot's design.

Acknowledgements

The team would like to acknowledge the many people who contributed their efforts to helping the team construct this project and make it through the challenging research process: Professor Michael Gennert for advising the project; Benzun Pious Wisely and Ming Luo for their assistance in building the robot and providing design ideas; Dr. Clint McFarland and Donna Fernandez for providing the team with information on the Asian Longhorn Beetle; and Alfred A. Rizzi for providing the team with design details and other information on RiSE. We would also like to thank Edward Read who helped in the machining and construction of various components of the robot.

Table of Contents

Table of Figures.....	5
Authorship.....	6
Executive Summery.....	7
I. Introduction.....	9
II. Background.....	11
A. RiSE v2.....	11
B. RiSE v3.....	13
C. Kamanbaré.....	14
D. Inchworm Design.....	16
III. Design.....	17
A. Locomotion.....	18
i. Four Legs.....	18
ii. Six Legs.....	18
iii. Centipede.....	19
iv. Inchworm.....	19
v. Flight.....	20
vi. Wheels.....	22
B. Grip Method.....	23
i. Spike.....	23
ii. Bird Foot.....	23
C. Size.....	25
D. Final Design.....	26
i. Servos.....	29
ii. Microcontroller.....	30
iii. Software.....	31
IV. Results.....	34
V. Conclusion and Future Research.....	37
References	41

Table of Figures

Illustration 1: Beetle with an emergence hole.....	9
Illustration 2: RiSE v2.....	12
Illustration 3: RiSE v3.....	13
Illustration 4: Mechanical structure of Kamanbaré.....	14
Illustration 5: Kamanbaré's software architecture.....	15
Illustration 6: Treebot.....	16
Illustration 7: Inchworm design.....	20
Illustration 8: RC Helicopter.....	21
Illustration 9: Wheeled Robot Climbing a Coconut Tree.....	22
Illustration 10: A Bird Foot Gripping a Test Tree.....	24
Illustration 11: Weight Budget.....	27
Illustration 12: Torque output of Servos.....	27
Illustration 13: 4 Leg Design.....	28
Illustration 14: 6 Leg Design.....	28
Illustration 15: The Software Architecture.....	32
Illustration 16: The Conceptual GUI.....	35
Illustration 17: The Platform gripping a sample tree.....	37
Illustration 18: The working GUI tests were done in.....	40

Authorship

Justin Gostanian

II.A. RiSE v2

II.B. RiSE v3

II.C. Kamanbaré

II.D. Inchworm Design

III.D.ii. Microcontroller

III.D.iii. Software

IV. Results

V. Conclusion and Future Research

Erick Read

I. Introduction

III. Design.A.i. Four Legs

III. Design.A.ii. Six Legs

III. Design.A.iii. Centipede

III. Design.A.iv. Inchworm

III. Design.A.v. Flight

III. Design.A.vi. Wheels

III. Design.B.i. Spike

III. Design.B.i. Bird Foot

III. Design.C. Size

III. Design.D.i. Servos

Executive Summery

The purpose of this MQP was to design and build a robot capable of climbing a tree to identify signs of the Asian Longhorn Beetle (ALB). The ALB is an invasive insect species that poses a serious threat to North American hardwood industry. Our project was on the development of a robotic system that can be used by a surveyor to reduce the risk and errors faced by the current method. We met with the USDA to identify the problems that they currently faced in searching for infestations in the Worcester area. We used the information that we received from the USDA to create a list of requirements that would be used as the baseline for the construction of the robot. After determining the features needed on the robot, research into how to design and build a robot that could fulfill the requirements was done.

Before we could start the design of our robot we decided that it would be beneficial to research existing tree climbing robots. In our research we looked at many different designs to see what were the most common and effective ways people used to develop tree climbing robots. In our research we found that there were many modes of locomotion that various people had implemented in their own designs. RiSE by Boston Dynamics was primary inspiration for our design, but we also looked at alternatives. Ideas like an inchworm design and even flight were considered, but ultimately, a legged design was considered best for the requirements we were given.

Another major design decision that we had to overcome after choosing to use a legged robot was how we would attach the legs of the robots to the tree. After talking with Alfred Rizzi from Boston Dynamics and seeing the different methods that they implemented in the various RiSE projects we came up with two basic design choices. The first design was a simple spike that would be attached to the end of the leg and could be driven into the surface the robot was climbing. The second method would require the design of an entire foot that would be attached to each leg and would allow for the robot to anchor itself to where ever it placed the foot.

We also considered the size of the robot. The size dictates the design restraints of a lot of other parts. The overall height affects the length that the legs should be as well as the amount of room we have for the essential electronics. The overall size of the robot also affects the size of the servos that we have to choose to move it. We decided to make the robot nine inches long so that each leg will be able to move along a ninety degree arc and still not hit any of the other legs.

For our final design choice, we decided on a six legged design. We chose this design because in our design process, we found the most benefit for cost. A six legged design would allow for easy construction and proof of concept on a small budget. It was also one of the favored designs that we found in our research with the RiSE project.

We also planned out the software architecture. The software of the robot serves three purposes. Firstly, it must function as the control system for the robot. Secondly, it must facilitate communication between the robot and the controller. Finally, the software must display the camera output to the user. We also decided a GUI would be necessary to make the system easy to use.

We made significant progress on the requirements we set out for ourselves for this revision of the project. Since our final presentable did not have a feedback system created and implemented we did not feel comfortable calling it a robot. We did this because we felt that anything that could not take in information from its surroundings and use them to alter the platforms actions was not truly considered a robot. This did not stop us from testing the platform we created against the requirements we set out to complete at the beginning of the project. The final revision that we tested for our platform was able to hold onto a tree.

This project leaves a lot to be answered. A future MQP could stem off from any part of our current project. Further research into the mechanical design of the legs or body would allow for the final project to be more effective in completing our list of requirements. Further work in a control system would allow for the robot to react to its surrounding and allow for easier and more intuitive controls for the end user.

I. Introduction

The purpose of this Major Qualifying Project (MQP) was to design and build a robot capable of climbing a tree to identify signs of the Asian Longhorn Beetle (ALB). The ALB is an invasive insect species that poses a serious threat to North American hardwood industry. [2] In order to identify signs of an ALB infestation extensive surveys must be conducted by human climbers. This is a slow, error prone, hazardous, and expensive process. [6] The ALB and the exit hole that surveyors look for to show signs of infestation are shown in figure 1.

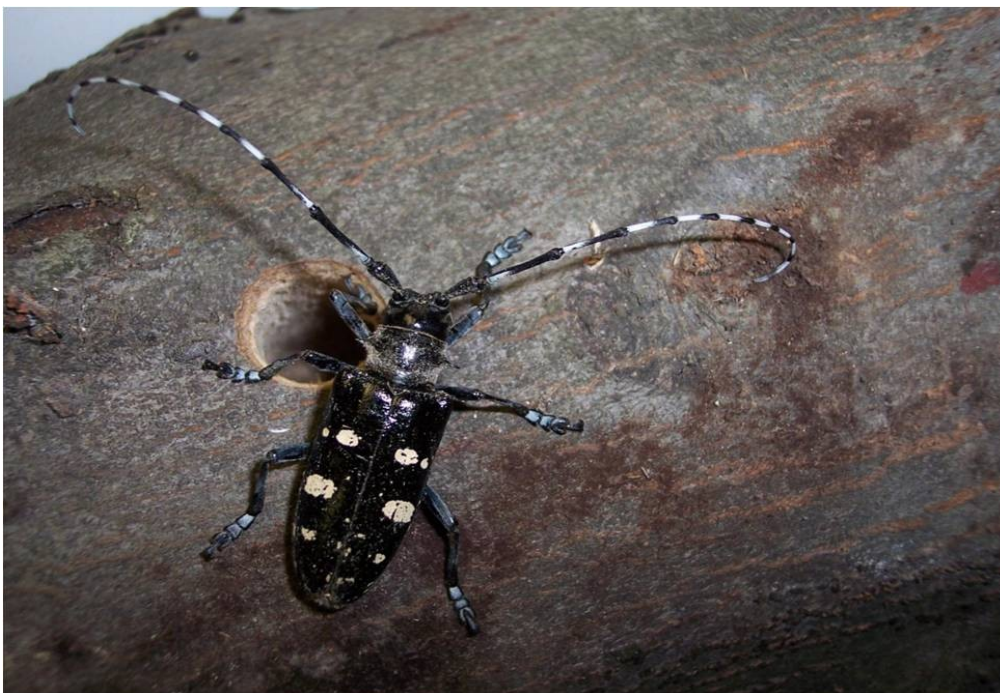


Illustration 1: Beetle with an emergence hole, Image supplied courtesy of Dr. Clint McFarland, USDA

Our project was on the development of a robotic system that can be used by a surveyor to reduce the risk and errors faced by the current method. We met with the United States Department of Agriculture (USDA) to identify the problems that they currently faced in searching for infestations in the Worcester area. We used the information that we received from the USDA to create a list of requirements that would be used as the baseline for the construction of the robot. For

instance, one of the major problems the USDA had with identifying ALB was searching trees near power lines or in other dangerous areas, such as cliff sides.

Features they requested included:

- Ease of use. They wanted use of the robot to be something that could be easily learned, so it would not slow down ALB inspection rate.
- Work safe. They wanted the system to have fail safes implemented to ensure no potential harm to USDA operators in the field.
- Minimal damage to tree. They did not want the robot to inflict any serious harm on the tree it was climbing.
- Navigation around branches. They needed to robot to be able to find its way around branches, so inspections could be complete. First signs of ALB infestation is in the canopy of the tree. [6]

After determining the features needed on the robot, research into how to design and build a robot that could fulfill the requirements was done. The primary source of information on robotic climbers was RiSE, designed by Boston Dynamics. There were two versions of RiSE that were extensively researched. The first was a general purpose climbing robot. The second was specifically designed for climbing cylindrical objects, such as telephone poles. Multiple forms of climbing robots were also researched, such as an inchworm design and RC helicopters.

After comparing various design ideas the group determined that a six legged modular design would best suit the requirements of the project. There were three major design areas that we had to focus on in the implementation of our robot: the mechanical aspect, the software aspect, and the electronics aspect. The mechanical design focused mainly on the construction of the legs and the robots ability to hold onto the tree. Size and overall weight were also issues that had to be considered before construction could be started. The proper electronics had to be determined to control the servos in the legs. A micro-controller was needed to communicate between the servo

controller for the legs and the user on the ground. The software consisted of the control system and a user interface. The control is facilitated through serial communication. For the user interface, it was decided that a GUI would be necessary to ensure that the system was easy to use.

Once the system was built, it was tested, though certain difficulties appeared in testing. The robot had difficulty lifting its arms, meaning structural changes were necessary if the robot were to ever climb. Also, without feedback, it was not possible to measure any force on the servos. This made programming the control software difficult, as all the the software was capable of doing was sending commands to move the servos in a set pattern. Despite these problems, the robot was able to securely grip a tree with four points of contact.

There is much room for improvement in this robotic system. Further work can be done on the leg design to make it possible to climb. The shape and strength of the legs could be redesigned to make the legs smaller and lighter. Changes to the power system or placement to the motors could also be experimented with. The control system needs to be changed to take feedback into account. The GUI and computer vision aspects of the project are also areas that can be improved upon.

II. Background

Before we could start the design of our robot we decided that it would be beneficial to research existing tree climbing robots. In our research we looked at many different designs to see what was the most common and effective ways people used to develop tree climbing robots. In our research we found that there were many modes of locomotion that various people had implemented in their own designs. We also researched the different ways in which these designs were able to grip their surroundings. These ideas would lay out the basic design that we would eventually follow in our own project.

A. RiSE v2

RiSE, a climbing robot developed by Boston Dynamics was one of the primary inspirations

for TreeBot. RiSE has two different completed versions, with v2 being a general purpose climbing robot, capable of climbing walls, trees, and fences. The robot uses different toe models for different surfaces.

To climb trees, it uses curved needles that penetrate the bark of the tree the robot is climbing on. The robot changes its posture to conform to the curvature of the climbing surface. A common problem that the RiSE project kept running into was the robot pitching backward. To prevent this problem, RiSE used a tail that would allow for the robot to produce a rotational moment closer to the top of the robot. This allowed the robot to continue to have the surface it was climbing in reach and would not pitch backward away from the surface it was climbing. Unfortunately, when the robots turned on a surface like a tree, the tail could not provide any aid since the curvature of the tree did not give anything for the rigid tail to push against.



Illustration 2: RiSE v2 [3]

As Figure 2 shows, RiSE v2 has six legs, each powered by a pair of electric motors. Each leg has two degrees of freedom, which Boston Dynamic's describes as “severely underactuated and must rely upon tuned compliances, built into its leg and foot structures, to be able to properly hold onto climbing surfaces.” [10] At minimum, three legs are always gripping the surface of the object

RiSE is climbing. This assures that the robot always has a secure attachment to the surface.

The robot also has an onboard computer that controls leg motion, manages communication, and services sensors.[3] The robot is controlled remotely by a laptop connected to the robot's computer by a wireless Ethernet connection. A human operator manually controls the motion of the robot.

B. RiSE v3

RiSE v3 is a legged robot designed to be specialized for dynamic, high-speed climbing of a uniformly convex cylindrical structure, such as an outdoor telephone pole. The robot has four powerful legs, each with two actuators. Each leg contains two active degrees of freedom, attached to a body with an additional central degree of freedom to change posture. The robot's centralized body degree of freedom allows the robot pitch to be adjusted during climbing. With an adjustable body, combined with the four actuated legs, the robot has kinematic freedom and range of motion to allow effective climbing of poles.[5] As can be seen in figure 3, there two degrees of freedom on each leg, oriented in the hip abductor and in the traction directions, carving a near cylindrical shape for each toe's workspace, relative to the body.

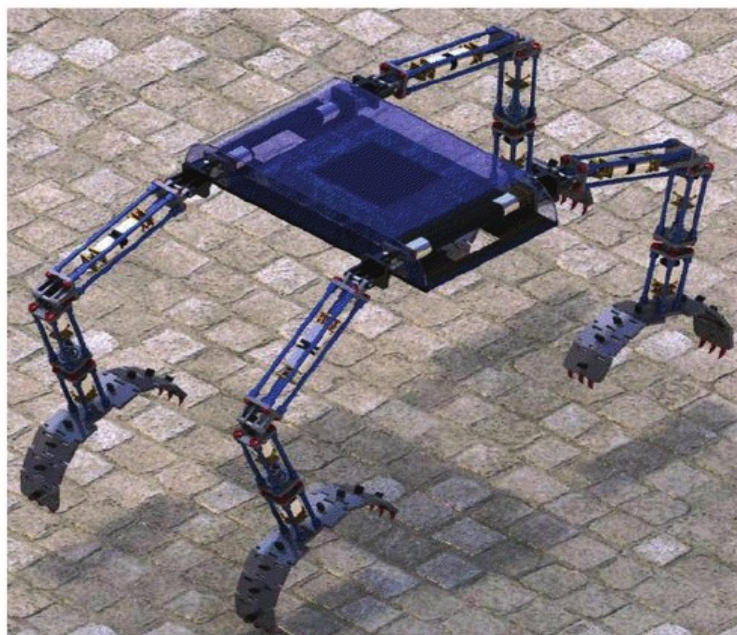


Illustration 3: RiSE v3 [5]

The prototype of RiSE v3 has few sensors other than joint proprioception (magnetic encoders), thus precluding the use of sensor-based feedback such as the force-sensitive controller. For climbing a wooden telephone pole, the robot used sharp claws that penetrate the wood. The claws used for the robot are engineered from surgical needles, and are sufficiently sharp for the task. Front claws, with the fore legs wrapping around the pole, are angled to align with the expected ground reactions forces, angled inward and slightly downward. Hind claws angle down to dig straight into the wood when generating thrust upward. This claw design ensures a strong grip, as both the front and hind claws push downward with gravity. The claws being angled inward also assist in assuring a secure grip.

C. Kamanbaré

Another example of a tree climbing robot, created by research group in Brazil, used a chameleon as a biological example. The mechanical structure, as shown in figure 4, of their platform consists of a central rigid body with four legs, identical and symmetrically distributed. Each leg comprises three links connected by two rotating joints, and it is connected to the central body by a third rotating joint. Each leg also has a paw, which is forked just as the chameleons, but with only two fingers, one on each side, for simplification purposes. The paw connects to the leg by a rotating joint, and also has another motor and reduction group that provides for its opening and



closing movements.[8]

The team also detailed the control software architecture, as shown in figure 5. [8] Their design has seven modules, each that manage different parts of the robot, but are layered and communicate with one another. The support system controls energy distribution to the platform's electronic and electro-mechanic hardware, and monitors energy consumption as well. Actuators control system is responsible for controlling the motors, and also for controlling the movements of the legs. Both of these systems are linked to the general control system, which controls all the robot's movements, sending the necessary commands to the actuators control system. There is also a communication system that is responsible for the communication interfaces existing in the platform, managing communications via Wi-fi and Bluetooth. Next, the environmental inspection system gathers data from the installed sensors, and also controls any additional hardware necessary for that purpose as well.

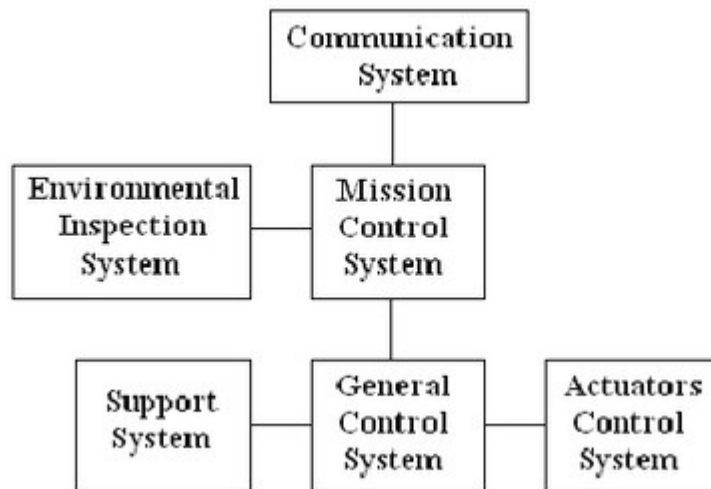


Illustration 5: Kamanbaré's software architecture [8]

Lastly, there is the mission control system. This system is the main module, the highest hierarchical level of the platform. It is responsible for receiving commands via the communications system, and for distributing them to the systems involved. It also stores information on the general status of the platform. The designers also explain the electronics architecture. The design for a

single joint take three components: a DC motor, a potentiometer and a micro switch. As the robot has four legs, they opted for distributing the control individually to each one of them. Thus, each leg control module needed four groups, namely, three for the joints, and one for controlling the opening and closing of the claw. They then developed a motor control board for this specific purpose. [8]

D. Inchworm Design

This inchworm design named Treebot, developed by a team at The Chinese University of Hong Kong, is a robot whose movement is biologically inspired by that of an inchworm. As seen in Figure 6, the gate of this “inchworm design” involves the use of a set of flexible linear actuators connected to two gripping claws. [1]



Illustration 6: Treebot [1]

While the robot firmly grips a trunk or branch with one claw, the other claw reaches and feels for a safe and secure location to grip on the tree. After finding a position, and securing a grip with the the searching claw, the claw previously used for gripping is safely released, and the process starts over.

Notably, the claw has four fingers, to ensure it can strongly grip the tree.

Treebot uses sensors that to determine a tree's shape to allow the robot to adjust its hold on the surface and navigate its way up tree trunks and over branches. Treebot possesses a certain level of autonomous climbing ability, to reduce the complexity of controlling the robot by users. The user simply requests the robot move in a general direction, and the robot's control system determines the location to move. To do this, the robot uses tactile sensors, which allow the robot to feel for the tree's shape. Using this form of feedback instead of visual feedback has three major advantages. Firstly, it allows the robot to save power. The robot runs on a small battery charge, and cameras would exhaust the charge very quickly. Secondly, processing the visual feedback is computationally expensive, meaning the robot would likely require a faster, and likely heavier and more power-consumptive, on-board computer. This form of feedback allows for much faster computation. Lastly, visual feedback's accuracy can be influenced by lighting conditions. Considering lighting conditions can vary significantly outdoors, this is a serious disadvantage.

III. Design

Since this is the first time that this project has been offered at WPI we had to determine what would be the best design for first generation project that could continued to worked on in the future. To do this we brainstormed multiple ideas that we thought could meet the requirements we set at the beginning of the project. There were five major decisions that had to be made before construction of the robot could take place. First we had to decide about the overall design of our robot before we could start design and construction of its various parts. After a choice was made for the overall design we started the design of the various components needed to complete the design. These various components were broken down into leg design, grip method, size, motors, electronics, and software.

A. Locomotion

i. Four Legs

The most basic design we saw in our research was a four legged design. Four legs were attached to the body and one leg at a time is used to climb in most cases. By using only four legs on our robot, we reduced the amount of weight and power that the robot will need to run. Compared to a six legged robot, we lose the weight of two whole legs which includes the six servos that would also be used with the robot. A four legged design also allows for the robot to have the potential to move each leg further with each step.

Using a four legged design does pose some problems though. In order to keep three legs in contact with the tree for safety reasons, only one leg is allowed to move at a time. Moving up the tree like this means that the robot will move slowly compared to similar designs with less legs. This is due to the fact that since we must keep at least three points of contact with the tree while climbing for safety reasons, thus we only have one free leg to move at a time. Since only one leg can move at a time, it takes the combined time of each leg moving to find the speed at which the robot could climb. Other designs can achieve faster movements by having more legs and being able to move multiple legs at the same time while still meeting the base safety requirements.

ii. Six Legs

A six legged design allows for the robot to keep three points of contact with a tree while also being able to move the other three legs up. This allows the walking cadence of a robot with six legs to be much faster but creates more dynamic problems. If a robot with this design only keeps three points of contact with the tree, all the legs can be moved in only two steps. Having more legs on the robot means that the robot will weigh more and take more energy to move the same distance as a four legged design. Putting more legs on a body the same size also means that the legs cannot move as far without hitting each other. By adding an extra leg to each side, we lose vertical distance that each leg could contribute in each step.

There were many design choices that we considered when we started to think about the final design requirements of the robot. Each of the designs all had problems with there basic design that we were not able to over come due to either cost, time, or design requirements.

iii. Centipede

One of the designs that we discussed early in the project was a segmented robot with two legs per segment. The design was based on that of a centipede and we thought that it was interesting because the design would allow for us to easily navigate the tree. We quickly found out that the design would not be suitable for our project though because of the amount of moving parts and size the robot would need to be quite large, and it would not be possible to create with the resources and time that we had available to us. Another issue for this design would be how to control the robots movements. With so many moving parts, it would be hard to have someone new learn to properly control the robot and navigate it up a tree with little experience.

This robot would allow for multiple points of contact with the tree while it walks. However, it requires a lot of small segmented moving parts to properly work. This was the limiting factor in our decision to not use this design. The control system that would have to be implemented to use this design would have been complicated and large to keep track of each segment of the robot and position and stress on its legs. This design would also require a large number of motors and a large power supply to keep the legs with enough power to move and grip the tree. Since one of our basic requirements for the robot was to be able to climb to the tree's canopy under its own power we decided that this design would be too taxing on the power supply to use.

iv. Inchworm

One of the designs that we looked at while researching was an inchworm design that centered around a sliding body and two claws. We like the simplicity of this design and the way that the feet of this robot could make a solid footing on the surface it was climbing. As the robot

alternates between the feet that it uses to grip the tree the telescoping body moved the body either up or down the tree. Figure 7 shows the gate of this type of robot. [11]



Illustration 7: Inchworm design [11]

Although this design allows for an easy gate, to scale a completely vertical obstacle in order to control which direction the robot moves in is very tricky. In figure 7, you can see that the team used a segmented spine to get around this issue. As each segment of the spine is pulled against at different tensions the foot being positioned can be placed in multiple positions and orientations. Although this was a design characteristic that we desired being able to see the placement of both feet and watch make sure they are set in a good position before moving on in the gate was very difficult. We also could not determine an easy way to have a user control a robot of this design. Which would not allow for easy training of personnel to use the robot.

v. Flight

Making a flying robot was briefly considered. The basis for this idea was remote-controlled (RC) helicopters. It may have been possible to build a robot similar to these toys and mount a camera to it. Figure 8 shows a sample RC helicopter flying its own power. The idea was dropped

because if we wanted to make the robot autonomous it would have taken a lot of programming to avoid the branches, and if we opted to have the robot remote-controlled the skill of the operator would have to be much higher than the skill required to use a climbing robot.



Illustration 8: RC Helicopter [9]

This design would be very difficult to use in a heavily wooded area which would mean that the design would not be an effective solution for the problem we were faced with. Although it would be the simplest and fastest way to use per-existing technology to get a camera to the top of a tree, because of the skill required to use the system and limitations of where it could effectively be used, it was decided that we could not use this design and it was removed from the list of possible solutions.

This design would allow for the fastest ascent up a tree that is in an open area. It would also be the easiest to design and test because there are many various kits that are sold and can carry additional payloads. The major issue that we ran into with this design was it would be too difficult to implement under the conditions of this project. Most RC Helicopters are fairly heavy and also require a mixed fuel source of gasoline and oil that would have to be hiked in with in addition. RC

helicopters are also hard to learn how to control and would require a lot of time to properly train surveyors how to use. The final reason why we decided that this would not be a suitable solution to our problem was that the RC helicopter would not be able to get close to branches without a high chance of crashing. Since operating close to trees and being able to avoid branches was a necessary requirement for our robot we decided that this method would not be suitable for the problem that we were trying to solve.

vi. Wheels

Wheeled robots were another design choice that we looked at that would allow for fast ascension up and down a tree. This method requires the robot to hold onto the circumference of the



Illustration 9: Wheeled Robot Climbing a Coconut Tree [4]

tree as it climbs. Using wheels, the robot can climb a tree that has no branches and a similar circumference from the base to the canopy. Figure 9 shows a robot of this type climbing a coconut tree. [4] This is the simplest design that we came across in our research and was found in use by hobbyists in many places. Although this idea is simple to implement and test, it was determined that it would not be effective for overcoming the problem we were trying to solve. In the requirements we needed to be able to climb a tree of varying widths and be able to navigate around branches.

Since this design doesn't allow us to meet either of these requirements we decided that it was not worth further pursuing this design choice. The following image shows a wheeled robot climbing a coconut tree. Robots like this are used primarily in situations where there is no need to worry about avoiding limbs or branches. Four skates are held together against the tree with a rubber band. The user on the ground tells the wheels on the skate which way to turn and the robot climbs in which ever direction it is told too.

B. Grip Method

Another major design decision that we had to overcome after choosing to use a legged robot was how we would attach the legs of the robots to the tree. After talking with Alfred Rizzi from Boston Dynamics and seeing the different methods that they implemented in the various RiSE projects we came up with two basic design choices. The first design was a simple spike that would be attached to the end of the leg and could be driven into the surface the robot was climbing. The second method would require the design of an entire foot that would be attached to each leg and would allow for the robot to anchor itself to where ever it placed the foot.

i. Spike

This method of gripping the tree uses a spike on the end of the leg to act as a way of attaching the robot to the tree. The robot uses its leg to drive a small spike into the surface of whatever obstacle it is climbing. The weight of the robot is then held by the leg and the point of contact between the spike and the surface. This idea is similar to the design that was implemented in v3 of the RiSE project. Based off of the idea of how insects and clawed creatures climb objects it is the simplest design to implement since the design doesn't require the addition of more motors or mechanisms to work.

ii. Bird Foot

This method would require a mechanism that would mimic the foot structure of birds or some reptiles to hold onto the tree. The design utilizes toes to get a sturdy grip on the tree at the center of the foot. This allows the leg to have a sturdy anchor to the tree. Being able to anchor the foot to the tree in this way would allow for the robot to move around the point of contact with more



Illustration 10: A Bird Foot Gripping a Test Tree

ease not having to worry about applying equal force from the other side of the body to hold on.

Figure 10 shows an example of a bird foot design gripping a tree. Although this method would give the robot greater stability and a firmer grip on the tree surface, it presented a large list of problems.

Because this method requires separate motors and controllers for each foot that is implemented it was decided that it would be difficult to add another part to the robot that would need its own testing. Along with that and not wanting to tax our power supply more we decided that implementing this design would cause more problems than it would solve. Because of this, we decided to use the simpler method of using a spike instead. This decision also meant that we would not have to worry about having a control system to identify how the foot is placed on the tree and if the foot was securely attached making the controls to use the robot much simpler to use.

C. Size

The size of the robot dictates the design restraints of a lot of other parts. The overall height affects the length that the legs can be as well as the amount of room we have for the essential electronics. The overall size of the robot also affects the size of the servos that we have to choose to move it. We decided to make the robot nine inches long so that each leg will be able to move along a ninety degree arc and still not hit any of the other legs. This also means that the robot will be able to move about two inches every time it completes its gate for moving.

There are many different advantages and disadvantages for having a larger or smaller robot. As a robot gets smaller it can treat the object it is climbing as more of a flat surface. Although this simplifies the problem of having a robot maneuver of a tree it creates a whole new list of problems. The size of the electrical components would require that they be special ordered and would probably require a special power rating. This would mean that the electronics use to control those motors would also have to be custom made. Because of this we had to make sure that our design was large enough that we would have access to parts that could be easily produced and used to reduce the time that it would take to get the parts and construct the robot. The other side of the spectrum is a robot more of the size of the RiSE project. The issue with robots this size is they are heavier and so need larger motors, power supplies, and more complicated electronics to run. Both of these options are outside of our scope for this project because of our limited resources and time for manufacturing.

Overall, length of the robot affects many of the aspects of how the robot will function. The length of the robot determines how close the legs have to be and for long the leg segments can be. Length also determines the surface area of the body and how much room we have to mount important hardware to the robot. Make the robot too long and it will not be able to navigate up the tree without coming into contact with some kind of obstacle. Make the robot too short and you won't have enough room on the robot to mount essential hardware and keep the center of mass close

enough to the tree. Our group worked to try and find a happy medium that could easily be obtained and tested with resources that we had available to us.

When we met with Alfred A. Rizzi at Boston Dynamics we discussed the importance of the total weight of the robot. The major issue with weight is that the heavier the robot becomes the larger motors they need to move. By increasing the size of the motor, we must also increase the amount of power that can be supplied to each motor. Since continually increasing the size of the motor and power supply would not give us useable results, instead we tried to find a compromise between the weight of the motors, the power source, and the rest of the electrical and structural parts of the robot. It was suggested that we aim for a robot that breaks its weight down into three equal sized portions in order to be successful. The motors, power source, and the rest of the robot should each weigh approximately one third of the total weight of the robot. If one part weighs more than the rest, it would be possible that the robot may not be able to function.

D. Final Design

For our final design choice, we decided on a six legged design. We chose this design because in our design process, we found the most benefit for cost. A six legged design would allow for easy construction and proof of concept on a small budget. It was also one of the favored designs that we found in our research with the RiSE project. We chose to listen to the advice of Alfred A. Rizzi when designing our robot and create a weight budget. By doing this we could predict whether our motors would be sufficient for the project. If the torque output by the servos was greater than the predicted weight of the robot then we could proceed ahead with ordering parts we knew would work. The servos that we selected were rated to produce 2.4 kg/cm of torque. As shown in figure 10, we predicted that the total weight of the robot to be approximately 1533 grams. Figure 11 shows the torque output of the servos.

Servos: 14g * 18 =	252g
Acrylic sheet:	707g
Aluminum Angle:	190g
AA Batteries: 23g * 8 =	184g
<u>Misc Elec. Comp. :</u>	<u>200g</u>
Total Weight	1533g

Illustration 11: Weight Budget

With the predetermined weights it was determined that the selected servos would be able to hold the robot to a tree with maximum distance between the claw and the joint of 14.3 cm or about 5.5". We decided to make the upper and lower leg segments out of L-channel aluminum of the same length. Using this design, we decided to make the upper and lower leg lengths both 3". This makes the distance r_{max} with the knee at a right angle to be less than the maximum distance that the servos could support.

Weight of robot: 1.533 kg

Torque of servo: 2.2 kg/cm

$$T = r * F$$

$$2.2 / 1.533 = 14.3 \text{ cm}$$

each arm segment is 7.6 cm long r_{max} is 14.3

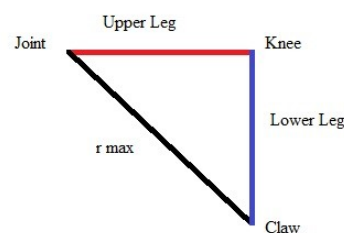


Illustration 12: Torque output of Servos

In planning the construction of the robot we plan to use 1/8" aluminum L-channel, in construction we found that 1/16" aluminum L channel would be easier to machine and would be

sufficient for the strength we needed. This also reduced the overall weight of the robot and added more room on our weight budget for future hardware. The body of the robot was a piece of 1/8" plexiglass cut into a rectangle with the outer dimensions 8" x 12". Six rectangular holes were cut

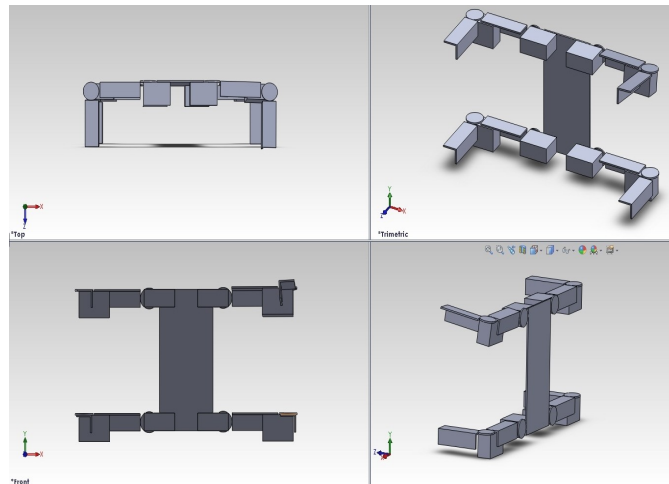


Illustration 13: 4 Leg Design

into the plexiglass for servos so that the legs could be mounted to the body. Since the servos are only bolted, this design allows for the quick removal of legs for maintenance and modularity.

The figures 13 and 14 show the designs that were created in autocad to build the robot. As we designed the robot we kept modularity in mind and created both designs to be of similar size and shape with only the amount of legs differing.

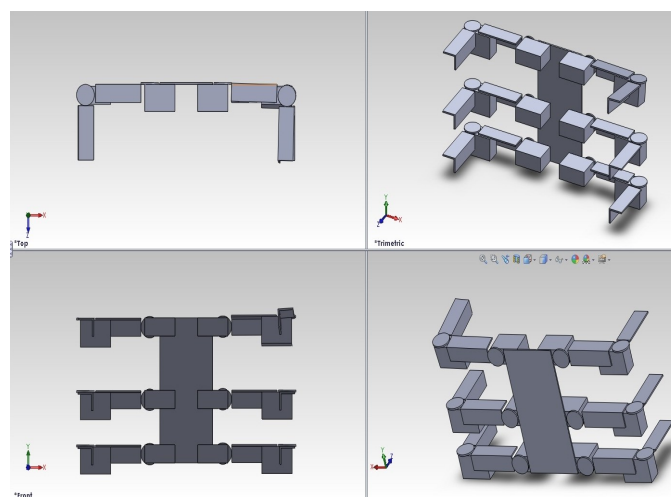


Illustration 14: 6 Leg Design

In the end we decided to use a modular design that would allow for a change between both four and six legged designs. This would allow for testing of either setup and allow for only simple actions needed to switch between the two. The reason for the choice of the legged designs over there competitors was the simpleness of the design as well as simplicity for use. We decided that it would be less complicated to teach a user how to control a four or six legged robot than it would to control a helicopter or a caterpillar.

If we chose to use a six legged design to achieve greater results in both safety and speed than a four legged design could have offered. The ability to have three points of contact with the tree and still be able to move half of the legs is a huge bonus. Although the six legged setup does have a list of disadvantages that we have to be prepared for. More legs will mean that the robot is heavier and will consume more power while running. We will also need a servo controller that is large enough to run all of the servos simultaneously.

i. Servos

We decided to use a metal gear micro servo to move the arms of the robot. Servos were chosen over motors because they gave us the ability to know there position without having to add additonal hardware. The reason behind using servos over motors was that we can start the robot knowing the position of each servo so we can continually monitor the arrangement of the leg. We decided that this would give us more control over the position the legs and know there definite position without any error that could be created by using a motor with either an encoder and a motor. The micro servo we looked at were designed to output a large amount of torque for their weight. This was needed so that the servo could be used to lift the robot without having to take up a large portion of the robots weight with motors and hardware needed to track their position. The decision to use metal gear servos was made out of the decision to not want to have to deal with plastic gears in the servos becoming stripped or warped and making the servos unusable. These servos also had much lower torque ratings and were discounted from use for these reasons.

Motors along with encoders or potentiometers were another option that was discussed.

Motors produced more torque and used the same amount of power. The use of motors require some other external device like an encoder to measure the angular distance moved which would increase the weight of the robot. The accuracy of the angle would also be determined by the control device.

Servos allowed for this issue to be removed because encoders are not a necessary requirement for our use of the servo. Another important reason why motors and encoders were not chosen was the limited number of I/O ports and power that the other on board electronics. In order to run eighteen motors and encoders there would need to be an even larger number of dedicated I/O pins, this would cause problems for our choice of microprocessor and power supply.

ii. Microcontroller

We opted on using an Arduino microcontroller for two reasons. First, because of the low cost. For a numerical comparison, the Arduino Uno board costs only about \$30. Another controller we were considering, the DyIO made by Neuron Robotics, costs \$150. Both controllers would be able to accomplish the servo control needed for this project, but Arduino could do so for a fraction of the price. The Arduino Uno was picked over other Arduino hardware because it had all that we needed, and the other microcontrollers had extra accessories that would not help us. Table 1 below shows the different microcontrollers that were considered and a cost comparison between them.

Microcontroller	Price
Arduino Uno	\$30
Arduino Nano 3.0	\$39
Netduino Plus	\$60
Arduino Mega 2560	\$65
DyIO	\$149

Table 1: Cost Comparison of Microcontrollers

Arduino was also a good decision for a microcontroller because it is easy to use from a programming perspective. Arduinio has its own IDE that comes with its own C/C++ library, which

makes many common input/output operations much easier. Arduino programs are written in C/C++. Users only need define two functions to make a runnable program. First, `setup()` – a function run once at the start of a program that can initialize settings. Second, `loop()` – a function called repeatedly until the board powers off. The Arduino library includes various functions for controlling servos that will be useful for this project.

iii. Software

The software of the robot serves three purposes. Firstly, it must function as the control system for the robot. Secondly, it must facilitate communication between the robot and the controller. Finally, the software must display the camera output to the user. Figure 15 shows the Software Architecture of the system.

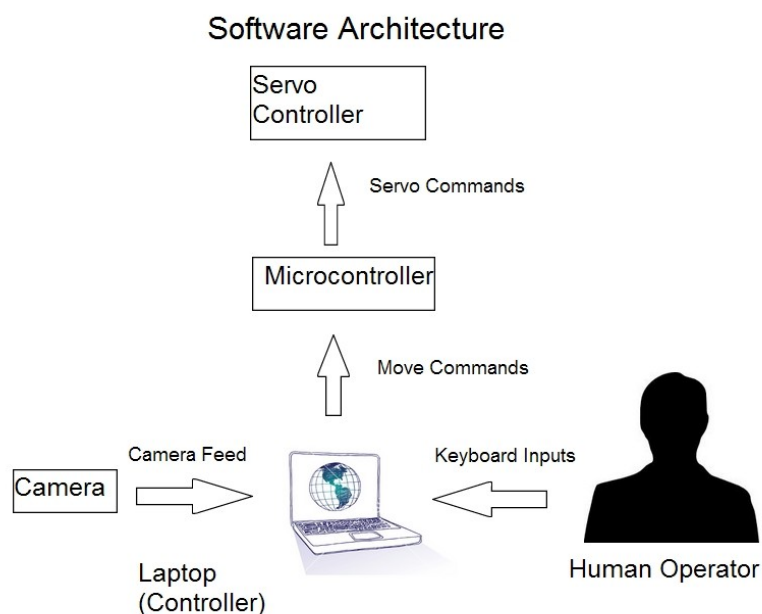


Illustration 15: The Software Architecture

The control system is designed to move the robot's servos to enable it to climb. The entire system the user's computer, a microcontroller, and a servo controller. The user gives input from

their computer, using an application that sends a request to the microcontroller to move the robot in a specified direction. The microcontroller then sends a request to the servo controller to move the servos to the specified positions. The positions based on what direction the user requests the robot to move. There is currently no feedback built into the system, so servo movements are simply predetermined patterns. This makes gripping the tree impossible in some cases.

All requests, both from the computer to microcontroller and from the microcontroller to the servo controller, and facilitated by serial communication. The physical connection between the computer and the microcontroller is made using the Arduino's USB serial port. The software connection is made using Arduino's "Serial" library, which is specifically designed for communication between the Arduino board and a computer or other devices. The function `Serial.begin(speed)`, is used to initialize a serial port, with speed being the baud rate. Typically, as is the case with this program, this function is called in the `setup()` function, which is run when the Arduino is switched on, as opposed to the `loop()` function that is run continually until the Arduino is turned off. After the serial port is initialized, `Serial.available()` is used to determine if there is data to be read from the serial port. `Serial.available()` returns the number of bytes that are available to be read, so this function is called in a conditional inside `loop()`. When available input is detected, the conditional activates. This program waits for input from the laptop controller, so once detected, it uses `Serial.read()`, which returns the first byte of incoming serial data. A single byte is all that is needed – different move commands can be distinguished by relating them to different characters. Another conditional statement is used to compare the byte to the ASCII values of characters corresponding to different move directions. After determining the direction of movement, the microcontroller sends out a signal using the `Serial.write(val)` function. `Serial.read()`, which only takes in 1 byte, `Serial.write()` can send out many. Due to a limitation on the Arduino Uno board, the signal is actually sent to both the laptop and servo controller, but the laptop simply throws away the data since it is only relevant to the servo controller.

The Arduino board is connected to servo controller using the TX and RX lines on the servo controller and to get the servo controller to move a servo, the Arduino uses the Pololu protocol “set target” to specify the servo and desired position. The Pololu must receive a message prefaced by 0xAA (or 170 in decimal) to understand it. The full format for the protocol is:

0xAA, device number, 0x04, channel number, target low bits, target high bits

As stated above, the initial 2 bytes are necessary parts of the Pololu protocol. The device number is a configuration parameter that can be changed, and exists for projects that require many servo controllers all connected to one device. In the case of this project, the device number of the servo controller was left at “12” - the default. Channel number is the number of the servo that will be moved. The target represents the pulse width to transmit in units of quarter-microseconds.

The visual system is separate from the rest of the system. A wireless webcam is attached to the front of the robot, and its video feed is sent back to the computer like any other webcam. The application developed for the robot outputs frames from this webcam using an open source library called OpenCV. OpenCV is used in the field of computer vision and allows for future projects to implement image analysis. The library contains many functions for image analysis, but for this project, only the basic functionality was used to allow the camera to work. OpenCV integrates well with Qt, the library used for the GUI.

Though it was not complete, a GUI was designed. Having a GUI was determined to be important as one of the design requirements was that it was easy to use. The GUI was to display the output from the webcam and include the various movement options. The GUI was to be designed using Qt Creator, as the latest versions of OpenCV interface easily with Qt.

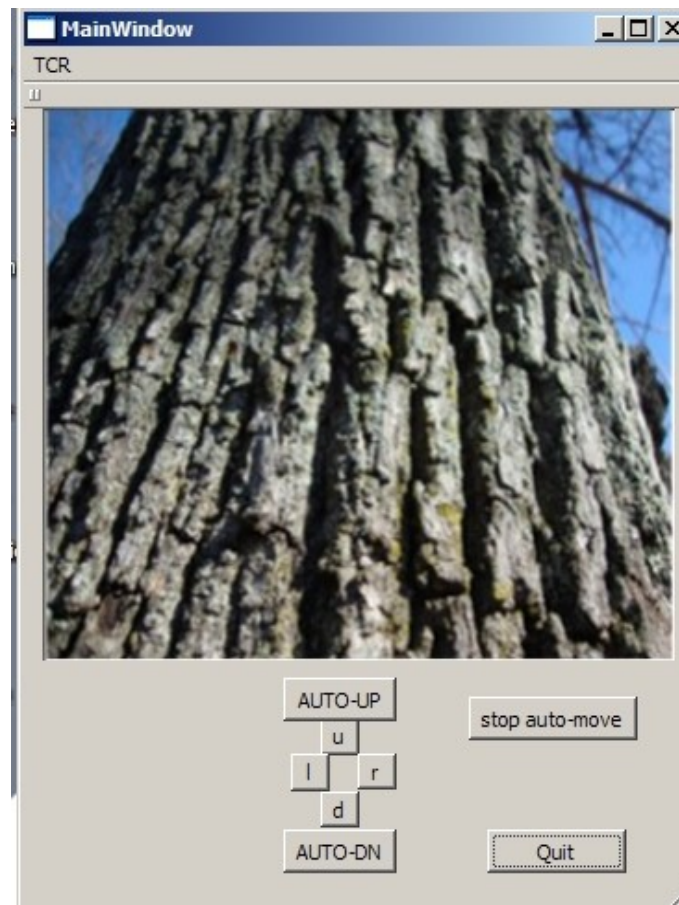


Illustration 16: The Conceptual GUI

As can be seen in figure 16, the GUI prominently displays the output from the camera in the center of the window. It was determined because the visual feedback is the most important aspect for the user, it should be the object of focus. The movement buttons are arranged in logical positioning based on direction. An additional feature is the “automove” buttons. This allows the user to have the robot constantly move in the selected direction until the “stop automove” button is pressed. This removes the trouble of having to continually enter individual move commands at the end of every single gate.

IV. Results

There were a long list of requirements that we set out to complete at the beginning of the project that were given to us by to our contacts at the USDA. This list is for the final robot that they would like to implement in their organization. The requirements that were created were put in place to make sure that the product would be able to complete the goal while not putting the surveyors or

tree's that are being surveyed in harm.

The major requirements are as follows:

- ✧ The robot must be able to carry itself and a camera to the canopy of a tree and be able to return.
- ✧ The robot must be able to be controlled by a user with little to no previous instructions and with intuitive controls.
- ✧ The robot must not deal any substantial damage to the tree that it is surveying.
- ✧ The robot must also be able to navigate around branches and other limbs.
- ✧ The robot must also have safety standards built in to reduce the chance of injury to surveyors.
- ✧ The robot must be small light and portable so that a single person can carry it to the survey sight and set it up without assistance.

These are all of the major requirements that are required for the final product to be able to complete. Since this was the first time the project was offered it would be difficult to create a product that would be able to full fill each of the requirements. we decided on a separate list of measurements to test the platform we created against. This list reflected the major list of requirements that was set out for the project but made what we set out to accomplish more manageable.

- ✧ Create a robot that can grip a tree with three feet
- ✧ Create a robot that is able to walk to simulate the gate used in climbing a tree.
- ✧ Create a basic user interface that can be implemented and used in the final design.

- ✧ Be able to get video feedback from the robot that can be used to identify signs of the ALB.

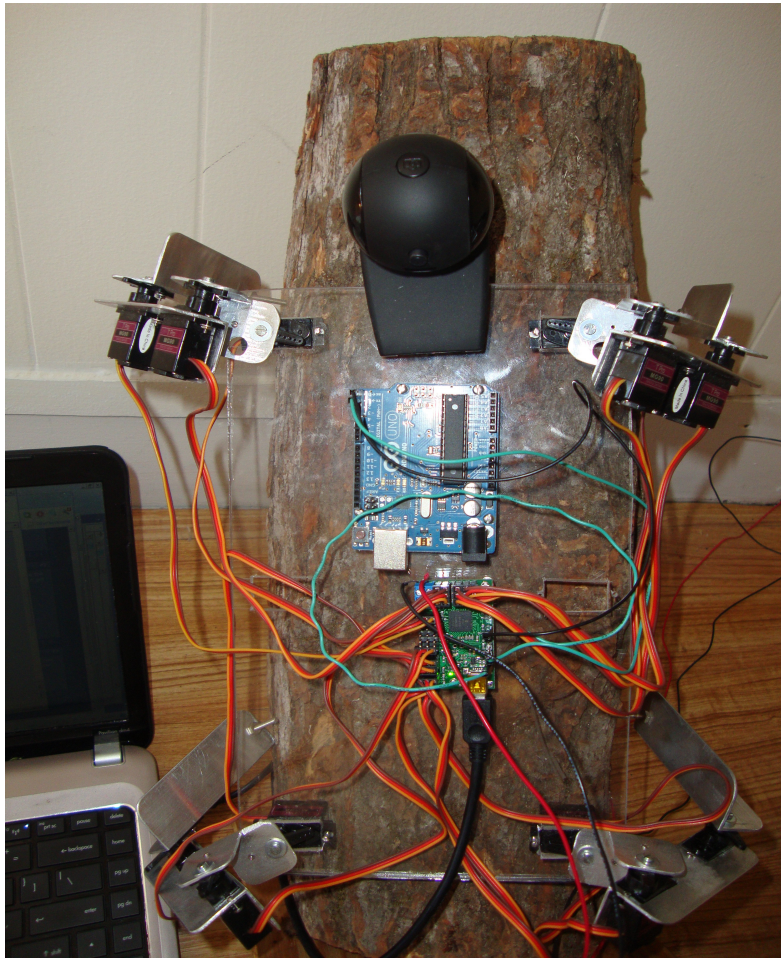


Illustration 17: The Platform gripping a sample tree

We made significant progress on the requirements we set out for ourselves for this revision of the project. Since our final presentable did not have a feedback system created and implemented we did not feel comfortable calling it a robot. We did this because we felt that anything that could not take in information from its surroundings and use them to alter the platforms actions was not truly considered a robot. This did not stop us from testing the platform we created against the requirements we set out to complete at the beginning of the project.

As seen in figure 17, the final revision that we tested for our platform was able to hold onto a tree. This proved that our design was able to hold its own weight on a tree and could be controlled to climb. Our final product did not unfortunately have much time to be extensively tested. Power consumption was much higher than originally anticipated while using servo's and required an off

board power supply capable of supplying a minimum of 8 amps of power. This number was determined from the found current draw of about 1 amp per servo while it was held under stress. It was determined that if 4 servos were holding the knee in position and another 4 servos were exerting the force needed to hold the robot all the servos together would pull around 8 amps of current total. Due to this issue we were not able to have an on board power source and so was not able to make any headway of getting a self powered robot to climb a tree. While the robot could securely grip the tree, as shown in figure 17, the robot did not have the strength to actually climb.

Regarding the requirement that “The robot must be able to be controlled by a user with little to no previous instructions and with intuitive controls,” we did not quite reach our goal, but picked a robot design that would not require significant skill on the part of the user. A basic user interface was proposed, which should ensure that the robot will not take much training to use. This would at least fulfill the revised requirement of “Create a basic user interface that can be implemented and used in the final design.”

The webcam also worked adequately and the software was able to display the output from the camera with no problems. Using OpenCV to display the visual feedback allows for a variety of improvements to be made to the software. The webcam itself is also high resolution at 1280 x 1024, allowing a user to identify signs of ALB infestation. This allows for the user to identify signs of ALB infestation.

V. Conclusion and Future Research

This project leaves a lot to be answered. A future MQP could stem off from any part of our current project. Further research into the mechanical design of the legs or body would allow for the final project to be more effective in completing our list of requirements. Further work in a control system would allow for the robot to react to its surrounding and allow for easier and more intuitive controls for the end user. Continued research into automated hole detection would also benefit the project allowing the error of identifying signs of the ALB to be reduced.

The design of the legs is something that could be explored in great detail by future projects. The shape and strength of the legs could be redesigned to make them smaller and lighter possibly allowing for the robot to climb around obstacles that our current platform could not navigate. Research on this could also look into designs that would allow for more force to be applied to the leg allowing for heavier payloads to be carried.

The placement of the motors could also prove to be an interesting project allowing the group to determine how the best way to control the leg is. It is possible a pulley system would be preferable to the servo method that we used in our first generation of the robot. Alternatively, having the motors located more centrally on the robot could have a positive effect on the design by freeing up room on the legs. An issue we ran into when trying to test the six legged design and the servos from one leg would collide with other legs. And how long would the pulley system last before stress on the cables used in the pulleys created issues with stress or stretching.

This project could also look into which is a better method of powering the legs of the robot either motors or servos. This would help determine what the best direction for future generations to go in would be. Knowing the benefits of both servos and motors for this application would allow for better controllers to be implemented. Since each method would require its own different type of control system to allow for feedback to be implemented into the robot. Currently, our robot does not have any way of sensing the stress on the servos. In the future, it will be important to the implementation of this robot that it be able to understand its surroundings. Depending on what method is used to power the legs, it will be important that future groups think about how they will receive information from the robot. After the design of the legs is completed and there is a better understanding of the power source that will be used in this project, it will critical that a good system is put in place to watch over the movements and actions of the robot.

Future groups could look into the benefits of having a segmented body or a tail to add extra stability to the robot. When we talked to Alfred Rizzi of Boston Dynamics, he suggested that there

were some benefits to using a segmented body. Due to our time restrictions, we decided to use a solid body design but a segmented body may give extra maneuverability to a robot while it climbs allowing it to place its feet and navigate around a tree with greater ease.

The GUI must also be improved. The current working GUI, as shown in figure 18, is simply a output from the camera and the commands are entered from a console.

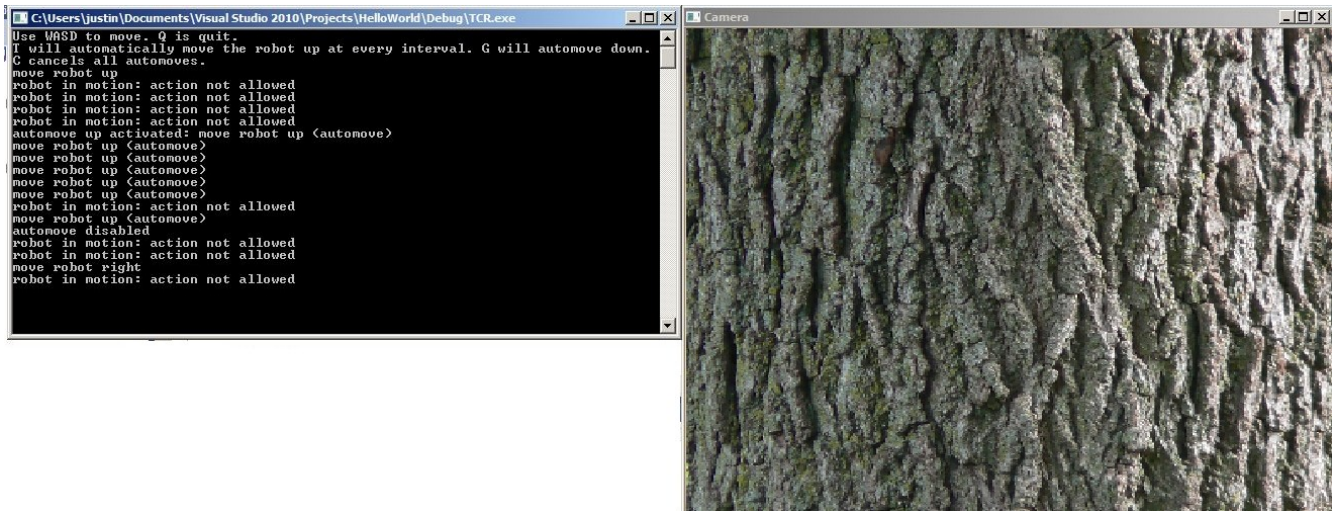


Illustration 18: The working GUI tests were done in.

The future idea for a GUI would be to design the GUI in an environment like Qt. One of the requirements of the USDA was to make the system easy to use. A console interface with text commands would likely not be considered easy to use. A GUI would allow click-able buttons, which would be much more intuitive. More features would also add to the usefulness of the GUI. The prospective users of the software would need to be asked what features would be helpful. Tests could be done on different GUI designs to determine which designs users respond best to.

Implementation of vSLAM and automated hole detection into the currently existing software would be the next step in improving the computer vision in the robot's software. VSLAM is short for “Visual Simultaneous Localization and Mapping.” It is a family of computer vision algorithms used for mapping and generating 3D information from image and distance/depth data. It has been applied to generating a dense map of world objects. [7] vSLAM was initially developed for

mapping building interiors, including rooms and hallways. In these applications, the robot is looking outward, through freespace. In this application, the robot is looking inward, toward the tree, essentially requiring an “inside-out” approach to vSLAM. Thus, the vSLAM algorithm would need to be tailored for its application in scansorial robotics for mapping the tree surface. There are additional challenges to implementing vSLAM on tree surfaces. One is the need for a dense depth map. As we detect many feature points for matching, performing accurate correspondences becomes more difficult. A tree’s curvature also provides an interesting challenge as it provides views of the same point from different angles and the common assumption of planar surfaces does not hold.

Another idea for an improvement to the Computer Vision aspect of this problem would be to implement automatic hole detection. In the current state, the interface merely displays the output from the camera. In the future, a useful feature would be to display the possibility of a hole made by the ALB. This would require an algorithm to identify the distinct shape of the hole from various possible angles.

References

- [1] Ackerman, Evan. "Treebot Learns to Autonomously Climb Trees." IEEE Spectrum. Web. May 18, 2011. <<http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/treebot-learns-to-autonomously-climb-trees>>
- [2] Aukema, J. E., B. B. Leung, K. Kovacs, C. Chivers, K. O. Britton, J. Englin, S. J. Franke, R. G. Haight, T. P. Holmes, A. M. Liebhold, D. G. McCullough, and B. Von Holle. "Economic impacts of non-native forest insects in the continental United States". *PLoS ONE* 6(9): e24587. (doi: 10.1371/journal.pone.0024587), 2011.
- [3] Boston Dynamics. "RiSE: The Amazing Climbing Robot." 2009. Web. 29 Sep. 2011. <http://www.bostondynamics.com/robot_rise.html>.
- [4] "Current and Recent Projects." Web. Apr 2012. <<http://publicweb.unimap.edu.my/~hazry/Current%20and%20Recent%20Projects.htm>>
- [5] Haynes, G. C., Alex Khripin, Goran Lynch, Jon Amory, and Aaron Saunders. "Rapid Pole Climbing with a Quadrupedal Robo." University of Pennsylvania Scholarly Commons. (2009): n. page. Web. 29 Sep. 2011. <http://repository.upenn.edu/cgi/viewcontent.cgi?article=1518&context=ese_papers&sei-redir=1#search=%22Upenn%20tree%20climbing%20robot%22>.
- [6] Liao, Zhichao, James Post, and Eric Petrin. "The Effects of Urban Tree Canopy on Residential Energy Use." 2011. Web. Apr. 2012. <http://www.wpi.edu/Pubs/E-project/Available/E-project-030411-105635/unrestricted/EEA_Final_Report.pdf>

[7]P. Paalanen, V. Kyrki, and J. Kamarainen. "Towards monocular on-line 3d reconstruction", *European Conference on Computer Vision (ECCV) Workshop on Vision in Action*, 2008.

[8] Reinaldo de Bernardi and José Jaime da Cruz (2008). A Tree-Climbing Robot Platform: Mechanical Concept, Control Software and Electronic Architectures, *Advances in Robotics, Automation and Control*, Jesus Aramburo and Antonio Ramirez Trevino (Ed.), ISBN: 978-953-7619-16-9, InTech, Available from:
http://www.intechopen.com/books/advances_in_robotics_automation_and_control/a_tree-climbing_robot_platform__mechanical_concept__control_software_and_electronic_architectures

[9] ""Nitroplanes." Web. Apr 2012. <<http://www.nitroplanes.com/rag26ccrarec.html>>

[10] "RiSE Platform." University of Pennsylvania. Web. 2007.
<<http://kodlab.seas.upenn.edu/~rise/newsite/index.php?leaf=7>>

[11] "Tree Climbing Robot." Web. Aug 2011. < <http://www.instructables.com/id/Tree-Climbing-Robot/> >