

Prototype Report

Tree-Climbing Robot Group

Amanda Richter
Matthew Saunders
Stephen Arnold
Matthew Meador
Robert Lewis

Tuesday, November 20, 2007
Trinity University

Senior Design
ENGR 4381
Advisor: Dr. Farzan Aminian

Abstract:

After deciding on a methodology to climb the tree, the robot was broken down into the modular subsystems of clamping, extension, rotation, and branch detection. The development of each of these systems is discussed with general schematics. A mathematical modeling code was constructed in EES based on geometry and force balance equations to accurately size the mechanical equipment and predict success of the clamping and rotation systems. Through an analysis of this program the key variables of roller diameter, distance to actuator, strength of actuator, and gear ratios were varied to find final dimensions for the design. From these results, a three dimensional representation was constructed in ProE to check for tolerances and hammer out final construction details. Detailed circuit schematics for the control systems for each module are provided and discussed as well as descriptive procedures to test each system. Finally a method to test the overall success of the robot with the mock telephone pole tree is also included.

Executive Summary

The final locomotion system consists of two separate clamps with a vertical extension between them. This system is further broken up into four major subsystems: clamping, extension, rotation, and branch detection. The clamping system consists of a linear actuator mounted between two arms, which are fixed at a shoulder joint. When the linear actuator is extended, a moment is created about each shoulder joint, causing the arms to compress and grip the tree. The extension system consists of a linear actuator mounted between the two clamps, with telescoping pipe braces on either side of the actuator, to provide additional support and stability. The rotation system is comprised of two small tread and roller systems on each set of arms. A DC motor and gear box drive a system of gears and chains, which allows the treads to rotate the robot about the tree. The detection system utilizes sonar sensors spaced around the top of the robot. These sensors allow the robot to 'see' and avoid obstacles in its path of movement.

To complete the mechanical design and determine the values of the variables required, an Engineering Equation Solver (EES) code was developed. This code modeled all of the forces present in the system, and allowed the numerous calculations to be solved simultaneously. By specifying certain values such as coefficient of friction of the tread and force applied by the linear actuators, specific dimensions for the robot were determined. Once these dimensions were determined, ProEngineer (ProE) was used to generate a three dimensional model of the entire robot. This model was extremely helpful in determining and visualizing the exact positioning of all the components, and in confirming that all of the parts would fit together properly.

The electronic design of the robot required several different control systems to be modeled with feedback to perform the specified mechanical functions. Three different motor control systems were derived for specific functions on the robot. Two of these require assembly, as only one mechanical task can be completed with a commercially available controller. Due to the large input and output demand of the system, ease and simplicity of communication is vital, and the inter-integrated circuit (I2C) bus is fully utilized in both motor controllers and the sonar array, as well as the serial peripheral interface and single bit I/O pins.

Testing of the robot will be completely bottom to top, and each system can only be tested when each of its subsystems have been successfully tested and debugged. Once each system component has been tested in this fashion (i.e. motor controllers, sonar devices, etc.), small coordinated actions can be tested. Small coordinated actions make up the majority of the subsystems for the larger logic functions, which strategically use them to accomplish larger tasks. These larger logic functions will be tested and debugged for every feasible scenario before more complex tests can be attempted. The success of the robot will be judged on its ability to climb to a preset height by a user, and return to the ground avoiding branches on both the way up and the way down. For testing purposes, the tree will be modeled by a telephone pole, with the branches simulated by dowels placed at two foot intervals about the circumference.

Table of Contents

Executive Summary	2
1. Overview Final Design	
1.1. Locomotion System	5
1.2. Clamping System	5
1.3. Extension System	6
1.4. Rotation System	7
1.5. Detection System	9
2. Detailed Mechanical Design	
2.1. Introduction	10
2.2. Key Values and Dimensions	11
2.3. Walkthrough and Technical Discussion	11
2.4. Three Dimensional Modeling	18
3. Detailed Electronic Design	
3.1. Control Logic	19
3.2. Motor Controllers	20
3.3. The Sonar Array	27
3.4. Hardware Communication	28
3.5. System Power	28
4. Testing	
4.1. Testing of Electronic Components and System Logic	29
4.2. Testing Working Criteria	32
5. Use of the Engineering Process	
5.1. Design Constraints	33
5.2. Design Process	35
5.3. Project Plan	36
5.4. Use of Engineering Tools	36
Appendix A: EES Modeling Code	38
Appendix B: EES Code Solution/ Final Dimensions	41
Appendix C: Bibliography	42
Appendix D: Project Budget	43
Appendix E: Project Schedule	45

Table of Figures

1. Final Clamping System	6
2. Final Extension System	7
3. Final Rotation System	8
4. Compressed Tread Systems for Curved Contact Surface	9
5. Positions of Sonar and Coverage Area	10
6. Clamping Schematic with EES Variables	14
7. Compressed Tread EES Variables	16
8. ProE Representation	19
9. Overall System Architecture	20
10. Vertical Linear Actuator Control	21
11. Forward Bias Relay Positions for Vertical Linear Actuator	22
12. Reverse Bias Relay Positions for the Vertical Linear Actuator	22
13. Off Relay Position for the Vertical Linear Actuator	23
14. Simple Chip Logic Control for Vertical Linear Actuator	24
15. Constant Torque Controller Utilizing the LMD 18200	25
16. Expected Current Response for Horizontal Actuator Controller Output	26
17. Test Setup	32

1. Overview Final Design

1.1 Locomotion System

The first step towards a working design for this project involved a decision of the basic locomotion the group wanted to pursue. The caterpillar design, which consists of two separate clamps with a vertical extension system in between them, was unanimously voted to be the central climbing method over a fixed wheel based robot after an evaluation through the use of design matrices. Modular subsystems were created to accomplish the main necessary functions of the overall project: clamping, extending, rotation, and branch detection. A brief explanation of the final decision regarding the subsystems may be viewed below.

1.2 Clamping System

Due to the simplicity and the cost effectiveness, the group decided that a linear actuator provided the best solution for the clamping system. The idea of using pneumatics or hydraulics was briefly considered as a form of energy transfer, but was disregarded due to the cost, construction time, and additional weight for the required supply subsystem. The arms of the clamps could be machined to the robot and fixed to free rotating shoulder joints, so that when the actuator mounted behind these joints extended, it would create a moment about each shoulder joint, causing the arms to compress and grip the tree as demonstrated by the arrows in Fig 1.

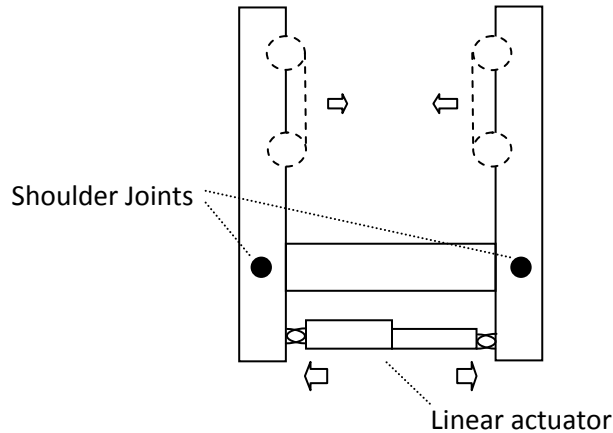


Figure 1. Final Clamping System

This actuator was sized using basic force and moment balances based on geometry and the estimated weight of the robot to ensure that the clamp would be strong enough to grip the tree. The applied force was increased to an acceptable level by lengthening the distance between the shoulder joint and the point of applied force from the actuator as well as increasing the strength of the actuator to amplify the moment. This shoulder to actuator length also governs how far back the arms can swing away from the tree, which is an important factor in the design. Additionally, since the actuators used are electrically powered, the programming for clamping and releasing will be simpler and more effective. More detail of the calculations concerning the clamping system and general mechanical sizing will be discussed later in the report.

1.3 Extension System

After weighing several alternatives involving gears and scissor lifts, the use of electric linear actuators for the basis of the extension system was deemed the best solution to accomplish the module's goals. Actuators are easy to use and can be customized to the exact needs of the project for an efficient use of funds. Since the same type of actuator was used on the clamping and extension systems, the amount of necessary programming was cut down because the same

code can be modified to accomplish both tasks. They also provide feedback to provide precision movement as well as inputs to calculate and measure the distance of travel. The final design centers on a linear actuator perpendicular between the two body sections as shown in Fig. 2.

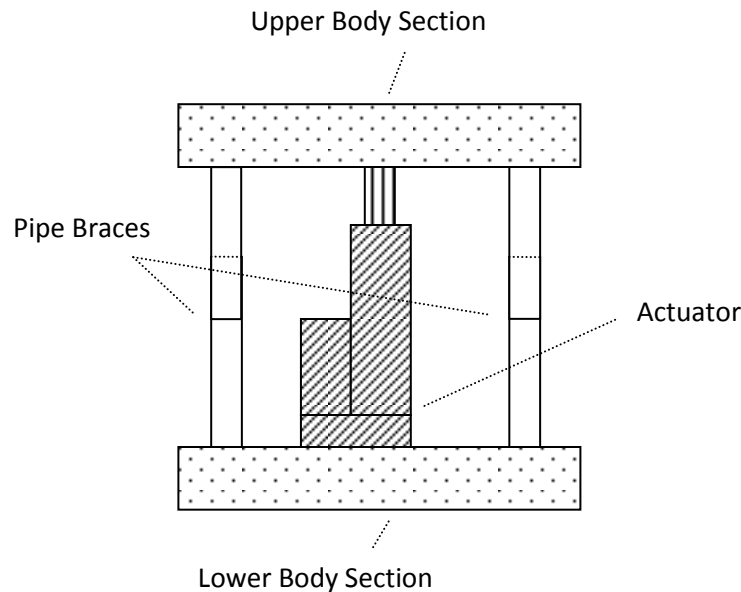


Figure 2. Final Extension System

Additional support is provided by a pipe brace on either side of the actuator comprised of a larger diameter pipe with a slightly smaller diameter pipe inside to create a sliding action.

1.4 Rotation System

The rotational module aims to allow the robot to travel around the tree to position itself to avoid branches. After numerous debates between treads and wheels, it was decided a compromise of two small tread arrangements on each set of arms would work best. As seen in Fig. 3, an arrangement of bike gears and chain powered by a DC motor and gear box drive the closer roller of the treads.

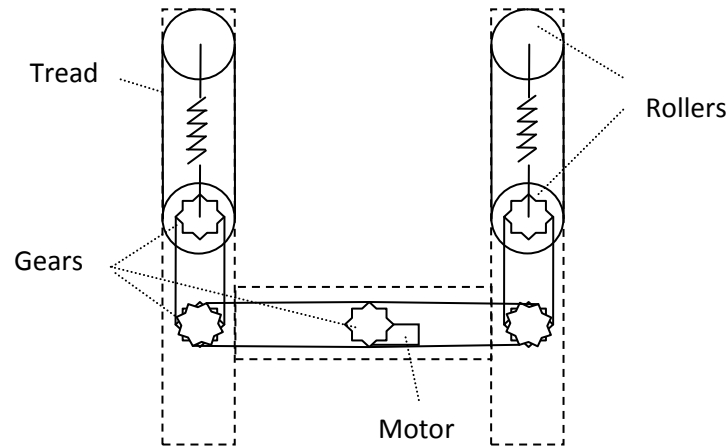


Figure 3. Final Rotation System

A gear box will be used to amplify the torque from the DC motor so that a specific motor can be used due to familiarity and available feedback options. Two bike gears are attached on the same axis to the output shaft of the gear box. Separate bike chain loops connect these two gears to one of two more gears sharing a same axis mounted on each shoulder joint. The other gear on each joint turns a secondary chain loop that powers a gear mounted on each of the closer rollers. The rotation of the roller then turns the tread to cause the robot to travel around the circumference of the tree. The intermediary gears on the shoulder joints were added to the design to eliminate the need for a device to keep the chain taut since a direct motor to roller system would require different lengths of chain as the arms were swinging to clamp or release. All of the gears have the same number of teeth to transfer torque and rpm at a one to one ratio, since a gear box is initially used to ramp up the torque output.

During the design of this system, it was decided that treads had an advantage over wheels because of the difference in contact area between the robot and the tree, but the fallback of this application is that the contact area is parabolic. To create this curved surface, the tread system is design to shape to the tree as the arms are clamped. As seen in Fig. 4, the two tread systems conform to the curvature of the tree.

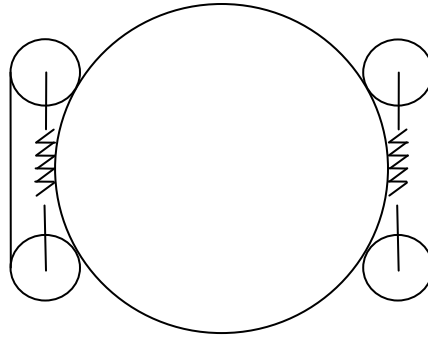


Figure 4. Compressed Tread Systems for Curved Contact Surface

The closer roller with the attached gear is on a fixed axis mounted to the arm, while the farther roller is allowed to travel parallel to the arm along a short slit as the arms clamp. As the arms compress on the tree, the normal force from the tree causes the farther roller to slide towards the closer roller since the tread has a fixed circumference. The spring between the rollers was implemented in the design to keep the rollers tight against the tread, while still allowing the tread to conform to the curvature of the tree.

1.5 Detection System

Although numerous alternatives including infrared and bumpers were considered for the task of detecting branch locations, it was determined that the use of sonar would be the best solution. This readily available technology is fairly cheap and relatively easy to use. It allows the robot to see branches before it gets to them to streamline the avoidance procedure by eliminating the need to climb slightly lower before rotating as a bumper system would require. Each sensor emits a sonar cone, which requires them to be mounted tilted away from the tree so that the edge of the cone travels parallel to the tree and for the purpose of this project can cover around 30cm of circumferential distance of the tree. The full system is comprised of six sensors positioned on the top arms and body to generate signals and detect all branches as seen in Fig. 5.

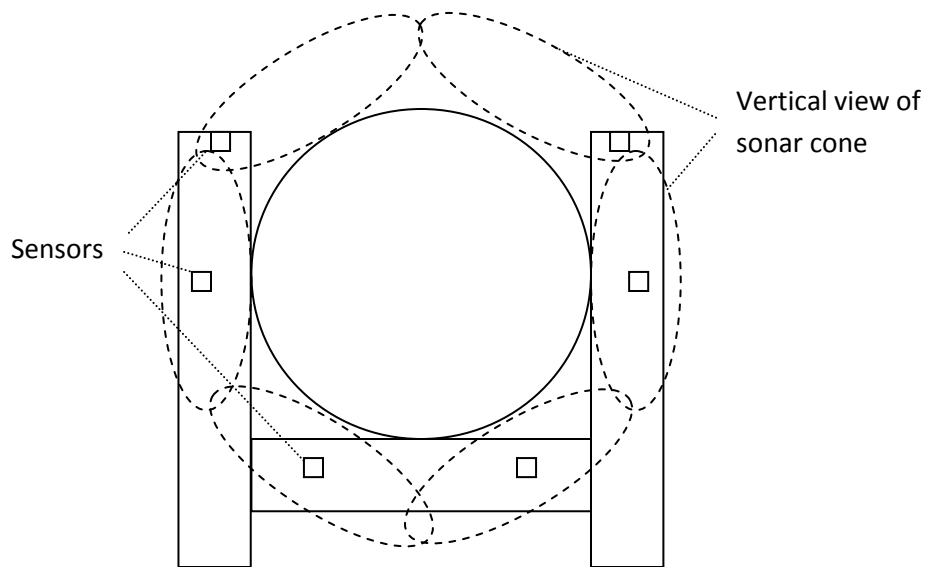


Figure 5. Positions of Sonar and Coverage Area

Some overlap of coverage area is actually desired to increase the quality of the resolution since the edges of the cone provide a weaker signal than the center. The sensors at the ends of the arms are positioned so that they overlook the area beyond the arms to ensure that if a branch is detected above the robot and it attempts to rotate around the tree, it will not hit the branch that was previously between the arms. Essentially as the robot climbs, it generates a large binary matrix and records branch positions as ‘ones’ to avoid them when backtracking down the tree.

2 Detailed Mechanical Design

2.1 Introduction

Since there are so many variables and each has a range of values to take into account for this project, it was decided that an Engineering Equation Solver (EES) code would allow for mathematical modelling of the mechanical systems to size the equipment. With this coding, it is possible to perform numerous calculations at once, and it allows sensitivity analyses to be performed to determine which scenario yields better results. The code is based on the forces and

dimensions of one body section of the robot. The full code with additional commentary and a sample solution printout may be viewed in Appendices A and B respectively.

2.2 Key Values and Dimensions

The capacity to hold the weight of the robot revolves around the strength of the linear actuator used in clamping as well as the distance from the shoulder joint to the actuator, which together generate a moment to create the clamping action. Additionally, the radius of the roller affects a wide variety of dimensions such as the distance between shoulder joints and distance between rollers as well as the torque needed to rotate the robot since the rotational friction force needed to be overcome to turn the treads is a function of the radius. This inevitably changes the required gear ratio needed to overcome the frictional force. These three key variables are the main user inputs in the code and from these, various forces are calculated.

While several variables were changed frequently, others were considered constants for various reasons. For example, when the group made a field trip to a used telephone pole distributor, it was observed that the desired 13” diameter poles were around the upper extreme of available poles, so a 12” diameter was chosen instead to ensure a more straight and circular cross-sectioned pole. Similarly, the weight of the robot was considered a constant 20lbs based on an approximation of known component weights and a safety factor of 1.25 was used to compensate for estimating.

2.3 Walkthrough and Technical Discussion

The rotational system’s treads design incorporates a high friction band of sandpaper used in belt sanders for the tread. The force needed to clamp and hold the robot is based upon the coefficient of friction between the tree and the tread. By setting the weight of the robot equal to

the force applied through the rollers, the required clamping force can be calculated, as seen in Eq. 1 below.

$$F_{climb} = \frac{W*n}{\mu} \quad (1)$$

Where, F_{climb} = Force needed by the clamping system

W = the predicted weight of the robot

n = the desired safety factor

μ = coefficient of friction between belt and pole

This calculation yielded a theoretical value for a basis of comparison to see if a given scenario worked, where any applied force greater than F_{climb} meant that the robot could grip to the tree. The applied force, $F_{applied}$, was found based on the moment generated by the extension of the actuator.

Before any calculations could be run, it was necessary to know the coefficient of friction of the tread, so an experiment was set up to find a reasonable value for this parameter. A wooden block was trimmed to be a tight fit inside the tread to provide a constant contact surface area foot print during testing and a string was attached to the block to have a single line of applied force. A force meter was then attached to this string to measure the amount of force necessary to slide the block. Various combinations of weights of known values were placed on top of the block and required applied force to move the block was recorded. Using Eq. 2, the coefficient of friction was then calculated for each run as seen in Table 1.

$$\mu = \frac{F_{applied}}{Weight\ on\ Roller} \quad (2)$$

Where, $F_{applied} = \text{Force off load cell}$

$\mu = \text{coefficient of friction}$

Table 1. Coefficient of Friction Test Results

Weight (g)	Weight (lbs)	Force (lbs)	μ
1751	3.879619	2.75	0.708833
2653.665	5.879617	4.5	0.765356
3105	6.879621	5.25	0.763123
4007.64	8.879563	6.5	0.732018
			0.742332

Although the average coefficient of friction came out to be 0.7423, the lowest value of 0.7 was used in the code as a safety precaution.

After researching available actuators, it was found that the most practical and cost competitive ones ranged in strength from 20 lb_f to 100 lb_f, with stroke lengths of 2", 4", and 6". Although the normal forces from the two rollers come in at different angle to the tree, the applied torque generated by the actuator comes out to be the same as calculating for one roller at the average roller distance as shown in Eq. 3.

$$T_{act} = (0.5) * (d_{a2} + d_{a3}) * f \quad (3)$$

Where, $T_{act} = \text{total torque applied to tree}$

$d_{a2} = \text{distance form the shoulder to first roller}$

$d_{a3} = \text{distance form the shoulder to second roller}$

$f = \text{resulting force}$

During the design of the tread system, it was desired to have the center of the tree always bisect the tread as described earlier in the general overview. The distance between the rollers depends on the circumference of the tread, and the deflection of the belt from the tree, as seen below in Eqs. 4 and 5 as well as Fig 6.

$$d_{roller} = (d_{a3} + d_{a2}) \tag{4}$$

$$c_{tread} = 21" = 2 * d_{roller} + 2 * \pi * r_{roller} \tag{5}$$

Where, c_{tread} = circumference of the tread

d_{roller} = distance between the uncompressed rollers

r_{roller} = radius of the roller

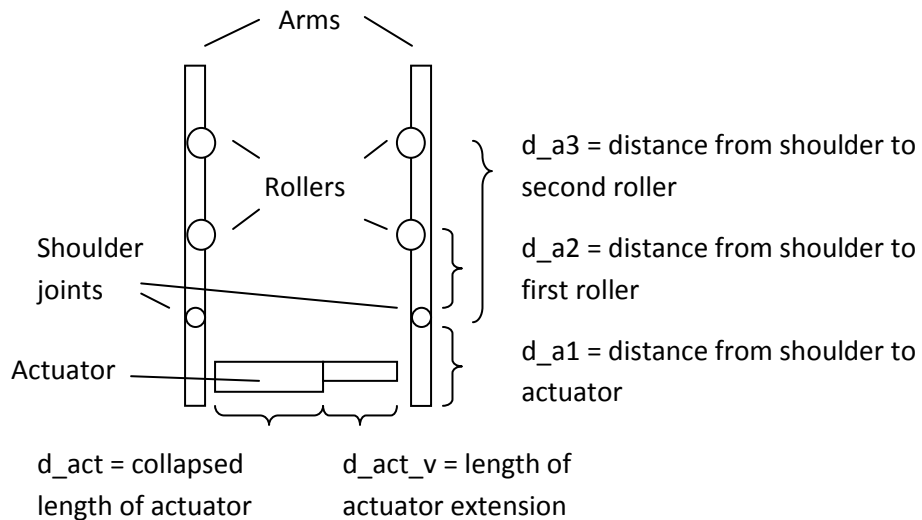


Figure 6. Clamping Schematic with EES Variables

Now, the force applied by the actuator was divided into two separate forces, represented by a point load on each roller. The force applied to the tree by the first roller will be much less than the force applied by the second roller, due to the shorter distance between the roller and the force applied by the actuator. The following equations show the relationship between torque and force for each roller.

$$T_{act} = T_{a2} + T_{a3} \quad (6)$$

$$T_{act} = F_{act} * d_{a1} \quad (7)$$

$$T_{a2} = F_{a2} * d_{a2} \quad (8)$$

$$T_{a3} = F_{a3} * d_{a3} \quad (9)$$

As shown in Eq. 10, the sum of the forces applied to the tree by each roller must add up to the climbing force required to scale the tree. If this criterion is not met, the robot will not be able to support itself.

$$F_{climb} = 2 * F_{a2} + 2 * F_{a3} \quad (10)$$

Finally, in order to compare the required climbing force and the total force applied by the clamp, it is necessary to compute the distance of each roller from the shoulder joint.

One of the constant values was the circumference of the tread, since the distance around the belt will remain the same whether it is in tension around the pole or slack, away from the tree. Equation 11 defines the relationship between the circumference of the tread with respect to the vertical compression and the diameter of the spring placed between the rollers to maintain a taut belt throughout the transition of unclamped to clamped positions. For additional clarification see Fig. 7.

$$c_{tread} = L_{arc} + L_{comp} + 2 * L_{comp2} + 2 * L_{wraparound} \quad (11)$$

Where c_{tread} = *circumference of the tread*

L_{arc} = *length of tread on the circumference of pole*

L_{comp} = *the vertical distance between rollers*

L_{comp2} = *leftover distance on back end of tread*

$L_{wraparound}$ = *portion of tread in contact with roller*

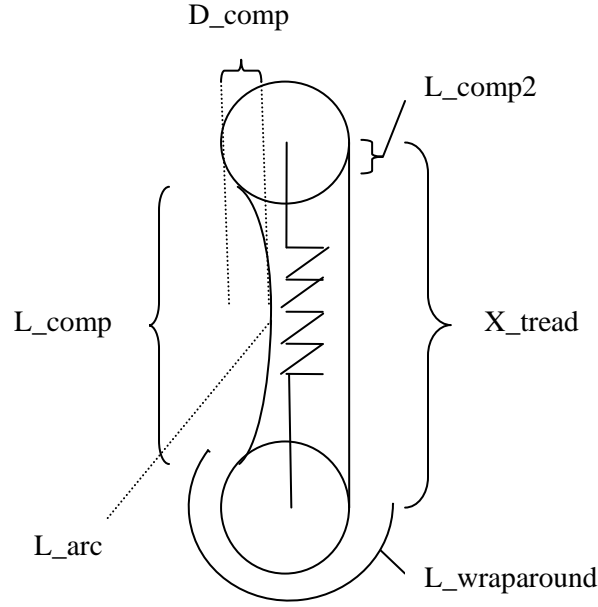


Figure 7. Compressed tread EES Variables

Next, a system of equations was formulated to compute these lengths of the tread. These equations incorporate the angles of the applied force and of the arc formed by the rollers. The vertical distance between the points of contact of the rollers has the following relationships:

$$\sin(0.5 * \theta_{comp}) = \frac{L_{comp}}{Dia_{tree}} \quad (12)$$

$$\cos(\theta_{force}) = \frac{L_{comp}}{Dia_{tree}} \quad (13)$$

Where θ_{comp} = angle of arc formed by rollers

θ_{force} = angle of force from rollers

A relationship for the arc length, which is the circumferential distance between the contact points of the rollers, is shown in Eq. 14.

$$L_{arc} = \pi * Dia_{tree} * \frac{\theta_{comp}}{360 [deg]} \quad (14)$$

While the uncompressed tread covered exactly half of each roller circumference, during compression the point at which the curved contact surface begins is slightly forces the tread to wrap around more of the

roller. This distance of the tread on the roller may be viewed as the total circumference of the roller multiplied by the ratio of the angle it covers over 360 degrees. By using geometric properties, it is possible to equate the angle the tread doesn't cover as 180 degrees minus Θ_{force} , and the resulting equation can be seen in Eq. 15.

$$L_{wraparound} = 2 * \pi * r_{roller} * (2 * \pi * r_{roller} \frac{180 - \theta_{comp}}{360 [deg]}) \quad (15)$$

Finally, the system of equations ends with a relationship to find the extra arm length that extends past L_{comp} on the back end of the tread. This relationship ties in all of the unknown dimensions and angles, as shown in Eq. 16.

$$\cos(\theta_{force}) = \frac{L_{comp2}}{r_{roller}} \quad (16)$$

The following relationships describe the compatibility of the roller distances away from the shoulder joint.

$$T_{act} = (d_{a2} + d_{a3}) * \frac{1}{2} * f_{applied} \quad (17)$$

$$d_{a3comp} = d_{a2} + L_{comp} + L_{comp2} \quad (18)$$

$$(d_{a2} + d_{a3comp}) = Dia_{tree} \quad (19)$$

EES solves all of the equations for best fit. Solving these equations by hand would be tricky, and EES provides a useful tool. However, constraints on possible angle solutions were assumed in EES to be positive, which negates any obscure solutions.

Other results include the shoulder diameter calculation. The shoulder diameter is an important dimension for construction of the robot, and this length is found using Eq. 20.

$$d_{shoulder} = r_{roller} + \frac{1}{2} * dia_{spring} + d_{gap} + Dia_{tree} \quad (20)$$

Where $d_{gap} = \text{the distance between the roller and the arm}$

In order to use a specific motor, the group needed to run a calculation for the gear ratio necessary to rotate the robot. The torque on the first roller must be greater than the resistive torque due to friction. A comparison is made below in equations 21 and 22.

$$T_{g2} = T_{g1} * \frac{n_{g2}}{n_{g1}} \quad (21)$$

$$T_{friction} = \mu * f_{applied} * r_{roller} \quad (22)$$

For the group's setup, the torque on gear 1 (T_{g1}) is equal to the torque of the motor, which is a constant 0.1085 lb_r-ft. For the robot to rotate, it is necessary for the torque on gear 2 (T_{g2}), which sits on top of the first roller, to be greater than the opposing torque of the roller. The EES code allows a sensitivity analysis to be performed in order to obtain working values for the radius of the roller and the corresponding gear ratio.

2.4 Three Dimensional Modeling

Using ProEngineer (ProE) and the final dimensions decided upon based on the results of the EES analysis, a three dimensional representation of the robot was generated as seen in Fig. 8.

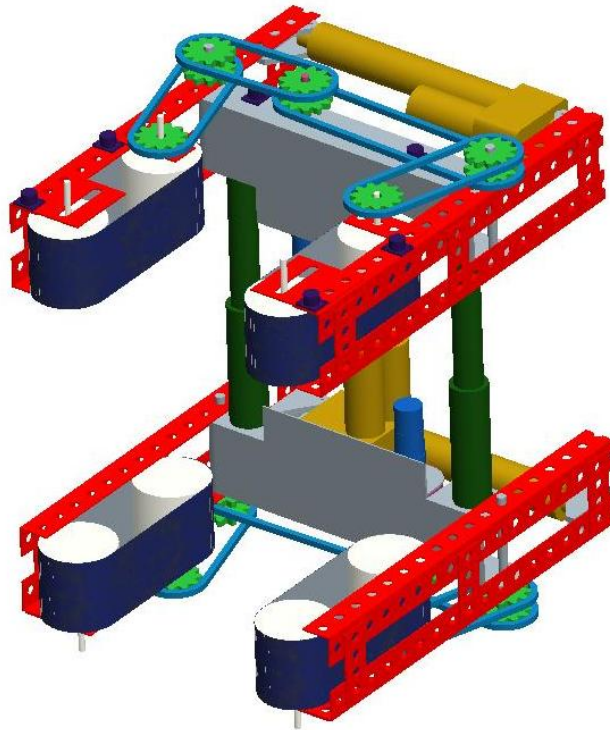


Figure 8. ProE Representation

The greatest advantage of doing this was to check for tolerances and hammer out details of positioning and mounting configurations. From this depiction construction schematics can be made for a more efficient assembly time.

3 Detailed Electronic Design

3.1 Control Logic

The system design demands the control of three linear actuators, a pair of motors that control treads, and a sonar array. To coordinate the logical functionality of these components, an EFIKA 5200B (courtesy of Genesi) will be utilized for overall system control. Given the mechanical parameters, the overall system architecture follows Fig. 9.

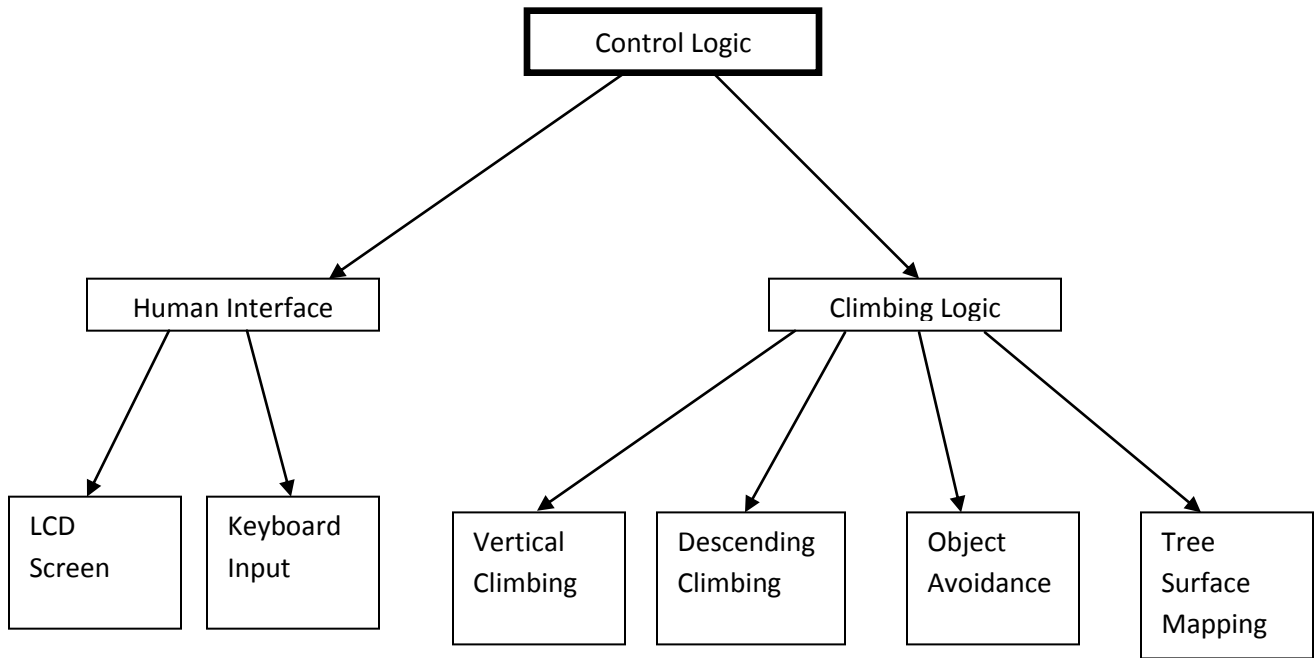


Figure 9. Overall System Architecture

Each of the climbing logic sub-functions encompass several other functions, the simplest of which interface directly with hardware controlling the motors and sonar devices. The specification of these hardware designs to assure appropriate system output and feedback follow.

3.2 Motor Controllers

The vertical linear actuator control was decided based upon the needs of the robot. Speed control of this linear actuator is not required, as the maximum speed of vertical lift is only 2” per second, switching currents will be minimal, and the control design will compensate for back electromagnetic force (EMF) from the motor. The robot weighs less than the rated force for the actuator (i.e. rated torque for the motor), therefore it can be held at full extension without harm to the actuator. Relays will be used for simplicity, and ease of digital control. The selected relays are OMIH-L with a 12VDC activation coil and will be activated by open-collector buffers with

pull up resistors to 12VDC voltage regulators. This is due to relatively large current demands for lower voltage activation (~100mA at 5VDC). Also, 5A fuses will be added in line with the motor to prevent permanent damage to system components.

Figure 10 shows a relay based control for the vertical linear actuator.

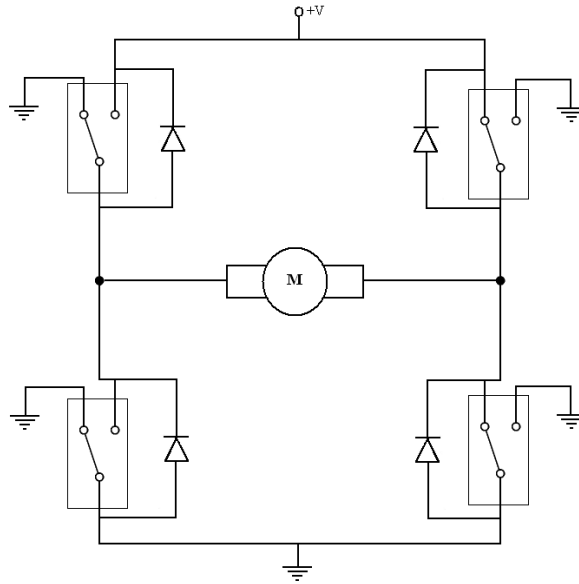


Figure 10. Vertical Linear Actuator Control

This is a simple ON/OFF control with diodes to allow for back EMF to pull current and allow for the motor to ease off before the current is switched to improve system fluidness and longevity by allowing low switching currents. Two specific relay positions work in pairs and allow for current to flow through the motor forwards and backwards.

Figure 11 shows the forward biased relay position of the vertical actuator, but this arrangement has potentially harmful positions which can ground the power supply (+V).

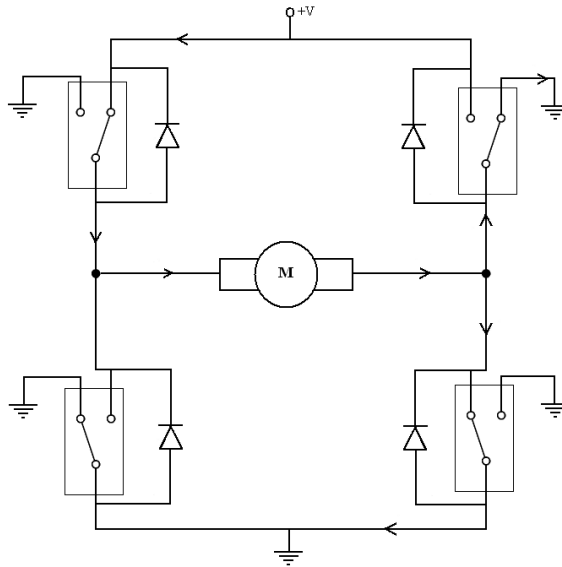


Figure 11. Forward Bias Relay Positions for Vertical Linear Actuator

There are however, two other acceptable positions as illustrated in Fig. 12 and the off position for the actuator in Fig. 13.

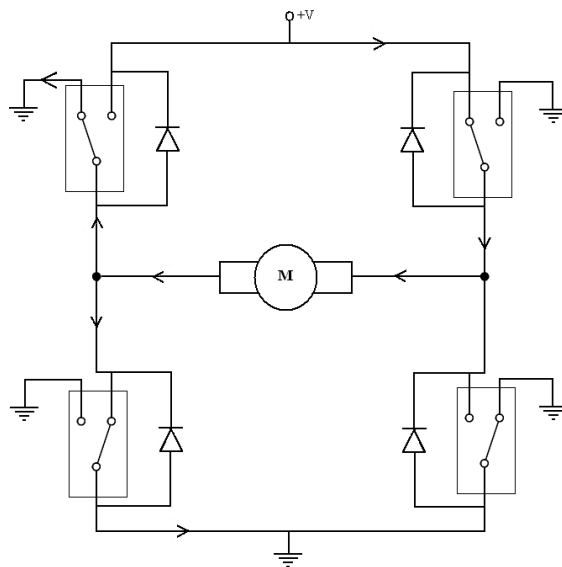


Figure 12. Reverse Bias Relay Positions for the Vertical Linear Actuator

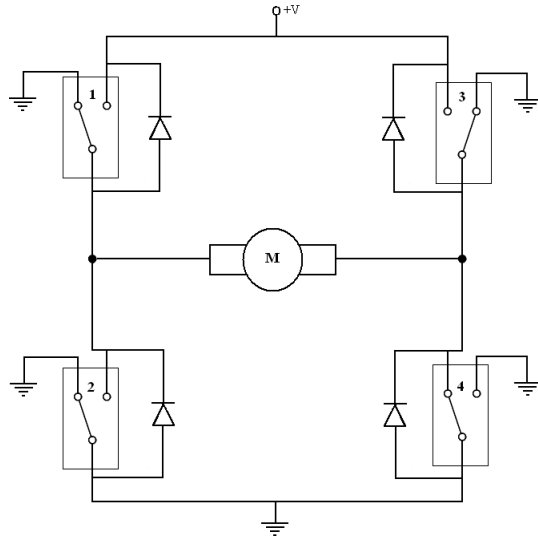


Figure 13. Off Relay Position for the Vertical Linear Actuator

To prevent destruction of the power supply by accidentally grounding it, a small logic circuit is implemented to interface with the system, only allowing for the outputs forwards, backwards, and off. This has two inputs as there are three outputs requiring two bits of information. The outputs of the logic circuit to the relays (1, 2, 3, and 4), decided by the inputs (A and B) are shown below in Eqs. 22 to 25.

$$F_1 = \overline{AB} \quad (22)$$

$$F_2 = A\overline{B} \quad (23)$$

$$F_3 = A\overline{B} \quad (24)$$

$$F_4 = \overline{A}B \quad (25)$$

Applying digital logic design, it is found that these functions can be represented by two NAND chips. By using utilizing NAND gates, OR and AND gates are constructed with two of the same integrated circuit (IC), instead of having to use two IC's of different types. This increases

manufacturability by decreasing the types of IC's required for the design. These control logic outputs, will be the inputs to the added open-collector buffers as seen in Fig. 14.

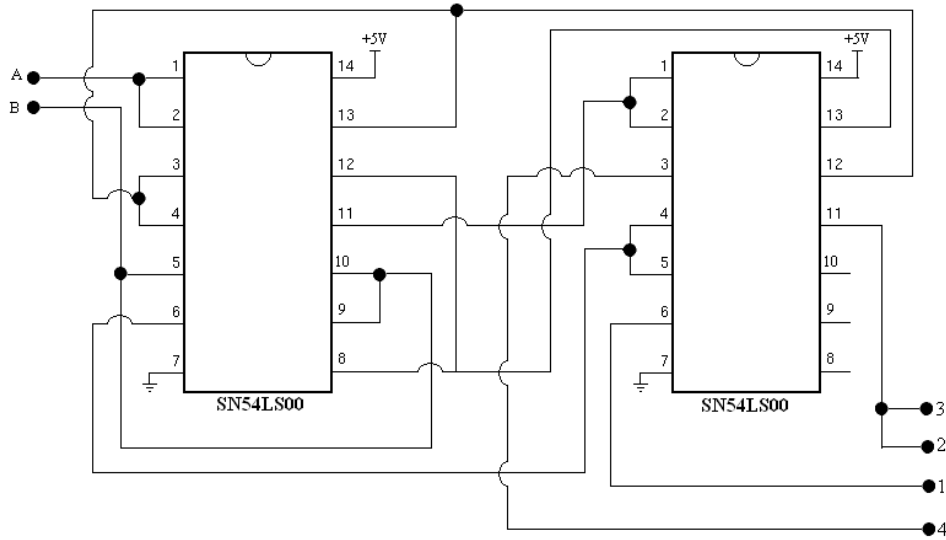


Figure 14. Simple Chip Logic Control for Vertical Linear Actuator

The two horizontal linear actuators which control the clamping mechanism need a different control methodology than an ON/OFF control, as an ON/OFF control would drive the actuators beyond the rated torque when they clamp onto the tree. All of the linear actuators come with a simple analog position feedback, which will be used to judge the current position of each actuator utilizing a quad-ADC which communicates via I2C (AD7924). A constant torque controller is taken from the application notes of an 18200 H-bridge datasheet to derive a controller as seen in Fig. 15.

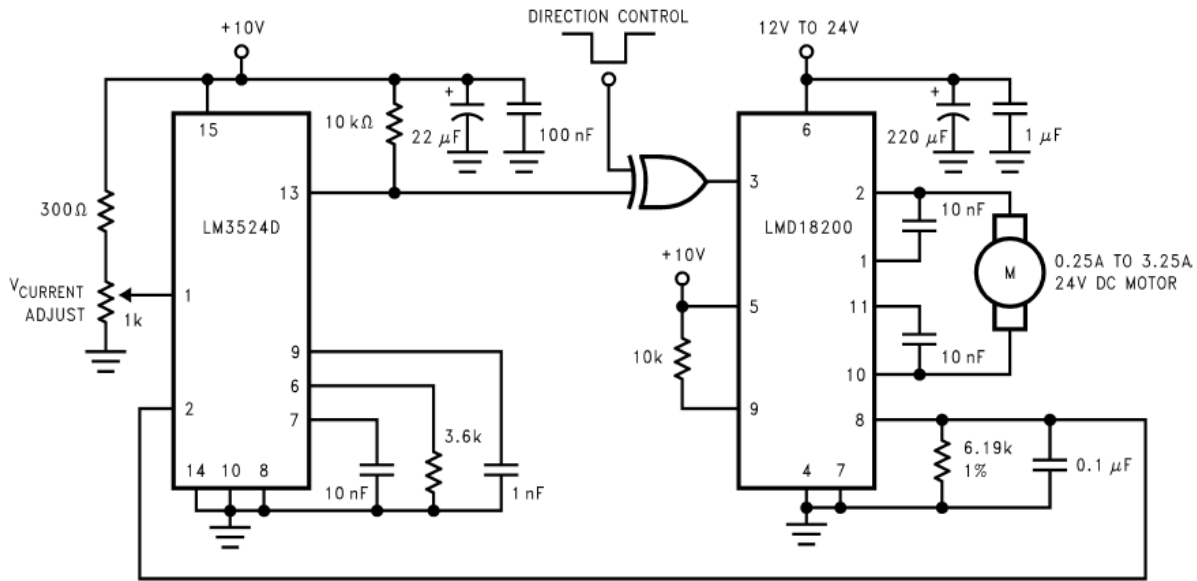


Figure 15. Constant Torque Controller Utilizing the LMD 18200 (LMD18200)

This controller uses a voltage comparator in the LM3524D to compare the current sense output of the LMD18200. The current sense output is pin 8 on the LMD18200, which is inputted to pin 2 on the LM3524D and compared to pin 1. Pin 1 is connected to a variable resistor, which effectively sets the desired torque. The variable resistor in the controller will be replaced by a digitally controlled potentiometer, as there is a multitude which can be controlled by a variety of methods including SPI, I2C, and parallel. This will result in a controller which will control torque based on the position feedback and action of the clamp with a relationship similar to that of the one in the application notes for the LMD18200 shown in Fig. 16.

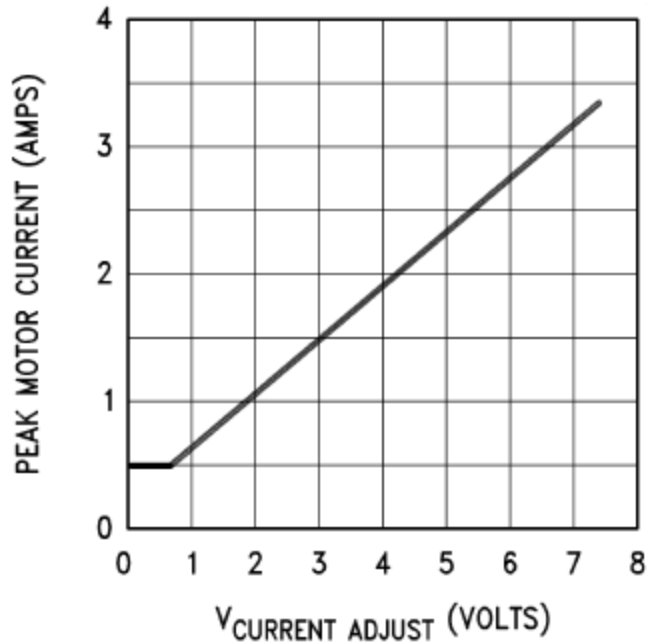


Figure 16. Expected Current Response for Horizontal Actuator Controller Output (LMD18200)

The two motors controlling the treads which propel the robot circumferentially around the tree are constrained to strict uncertainties, as positioning on the tree is the most meticulous attribute of the overall system. Without this the robot will be unable to descend the tree based on memory, and the uncertainty would cause problems as the system's position would drift. Also, the torque required to spin the robot around the tree is provided by a traditional DC motor rather than an actuator; therefore, speed control needs to be implemented. For this application, a handmade controller would be too complex, too intensive, and require a lengthy design period. Given these dilemmas, a more advanced manufactured controller is preferable to designing one from the ground up. The selected controller for the application is the MD23 by Devantech Ltd. Based on the simple I2C interface, the ability to control velocity curves, and the coupling with the EMG30 DC motor to produce an extremely accurate feedback system; the MD23 is the preferred choice and will be used to control the circumferential rotation motion.

3.3 The Sonar Array

The sonar array offers a challenge of both conformity to the other systems and simplicity, as the array must be large enough to measure around the circumference of a tree with a twelve inch diameter. Given the precedent of an I2C interface for the motor controller and the digital potentiometers, the preference for the communication of the sonar array is also I2C. The same company that produces the MD23 also produces a large selection of sonar devices for robotics applications. The SRF02 was chosen from the available sonar devices due to self tuning resonance, range, settable output (English/metric unit choices), and I2C interface. The advertised angle of sonar dispersion is 55°. Despite this information, an experiment provided dispersion closer to 45° for objects of thin cylindrical geometries close to that of small tree branches. The smallest object accurately sensed by the available SRF02 was less than 1/3” wide. The test also showed that the SRF02 can accurately sense these small cylindrical objects 20cm – 45cm away with the new measured half-angle. Based on this experiment, the number of needed sonar devices was approximated to 6 for full surface mapping.

The threshold will be set at 20cm, creating a blind region which must be tracked by the computer for object avoidance. Similarly, the maximum distance the sonar array will record is 45cm. The devices are setup so the threshold is the distance at which the ranges begin to overlap. This creates a variety of analytical solutions to the sonar responses, which fully satisfy the desired 2-dimension ranged limb tracking criterion.

3.4 Hardware Communication

To communicate effectively with the specified hardware, the computer needs to be able to communicate with I/O pins, an I2C bus, an SPI bus, RS232 serial, and USB. The RS232 and

USB communications are provided with the computer itself and can be utilized easily. The I2C bus, I/O pins, and SPI bus do not come stock with the computer, and will be provided by a Diolan USB-12. The output current of the I/O pins is extremely low, ~1.5mA. Due to this, open-collector buffers with pull up resistors will be added to each of the output pins to provide a stronger current which can power the vertical actuator relays, as well as protect the relatively expensive USB-12.

3.5 System Power

The system power supply needs to be able to effectively drive three actuators and numerous small electronic components. The actuators having similar characteristics, two having a stall current of 3A, and the vertical segment actuator having a stall current of 4A, meaning maximum possible system demands are ~10A total. The other system components such as the EFIKA draw very little power (~15W), as well as the sonar (spiking ping currents of a few hundred milliamps), and the various other small electronic components (combining to ~150mA). With this in mind, typical power usage should only be around 7A, as none of the linear actuators should approach stall currents, with spiking currents (+~20% of typical usage) of up to 8.5A. When combined with a ~45% safety factor, the chosen power supply is the Dynamite 12V 10A power supply. This allows for nearly full stall currents on all three actuators before failure, and will be coupled with a 9A fuse to prevent harmful system damage.

4 Testing

4.1 Testing of Electronic Components and System Logic

Testing methods for the electronics of the tree climbing robot will be completely bottom to top methodology. The basic subsystems consist of each motor control and individual sonar devices.

The first and primary system to be tested is the USB-12, the I2C will be tested and verified with a digital logic analyzer, as well as the SPI output, and the GPIO. The GPIO will be attached to a simple switch and LED circuit, it will be set to output all pins high, and the circuit will be attached to each pin to verify. It will then be tested to put all outputs low, and the same LED circuit will verify this. After this is achieved, each pin to be used in the design will be tested to verify correct operation, and the open-collector buffers will be added and the test will be repeated. Once this is achieved, two simple devices that are known to work will be used on the I2C and SPI provide proof of correct operation.

Each individual sonar device will be marked and tested on the same test set-up using the USB-12. The half angles will then be recorded using the base distances of the desired 20cm to 45cm robot vision range, as well as distance precision. Any individual characteristics of the sonar devices will be measured and accounted for in the final product. Once each individual sonar device is successfully tested, they will be added to the I2C bus one at a time and tested each time one is added to ensure correct operation. The sonar array will then be temporarily mounted on the robot and recordings of the outputs for objects in every 5cm by 5cm square in the desired vision range recorded. This yields a vision matrix with ~1% precision over the entire vision range which ensures that the sonar devices are capable of mapping the entire tree

circumference. Abnormalities of individual sonar devices will be compensated for as they arise before official attachment to the robot. The final matrix the tree mapping function will deliver will not give ~1% precision, as it will be based on what each sonar detects, the distance, and the triangulated coordinate of that distance when sonar overlap occurs. The final tree mapping function should ensure about ~5% to ~8% precision over the entire surface area in question.

The clamp motor controllers will be loosely constructed on a protoboard before being soldered to a permanent base. The digital potentiometers will have their resistance per ten taps measured to get resistance values for several positions independently. These resistances will then be directly compared against the output current of the total h-bridge circuit. This trend will then be compared to the same trend provided in the h-bridge application notes for validity and can then be used to approximate the initial needed values for the torque controller.

The vertical segment motor controller input logic design will be tested for correct output, and each relay will be tested individually before the entire controller is assembled. Once assembled, the controllers will be tested for correct bias in the stop, forward, and backward directions. With the horizontal tread motors, since the controller comes ready-made, only stop, break, and velocity curves will be measured for initial on tree testing.

Once each subsystem is tested satisfactorily, the sonar array will be attached to the I2C bus and one subsystem at a time will be integrated to the USB-12 and individually tested each time a subsystem is added. For the clamp action and vertical segment motor controllers, directionalities, degrees of forward and backward, and stop will be tested. These tests will occur both while the operation of another subsystems occur (the sonar devices), and when they are idle. This assures correct device designation by system buses (mainly the I2C bus) and operation.

After all the subsystems are integrated, each individual test will be repeated. The integrated system will then be ready for operational testing.

Operational testing includes entire individual actions the robot will make, and encompasses the subsystems of the overall logic. The overall logic decides what parameters to pass to the relevant subsystems. The major subsystems of the overall logic are the human interface, ascending, descending, and object avoidance functions. Each of these subsystems encompasses several coordinated actions. To assure correct operation, each individual action will be tested in a bottom to top fashion. An example of one of these simple actions is holding one of the clamps on the tree and opening the other. All of these small actions will be tested, debugged, and made to comprise small coherent individual functions. These small functions make up the lowest tier in the overall logic. After the lowest tier is debugged, the higher functions that coordinate these will be tested. Each level of programming leading to the overall logic will be tested and debugged fully before proceeding to the more dominant level. The two final tests before branch avoidance testing can begin, are the total coordination of the vertical climbing, and descend climbing functions with the tree mapping function, and the adherence of these with the parameters passed by the human interface through the control logic. These can be tested by using a 'tree' (telephone pole) free of branches, and a preset height entered via keyboard. Once this is successfully debugged, the system is ready for branch avoidance.

Branch avoidance testing is the final phase and will consist of one branch and multiple branch avoidance. An idle and stop switch will be used to protect the robot from self harm at all times. Single branch avoidance will be setting up the test tree with a single dowel inserted in a hole in a telephone pole segment. The robot will be placed in several positions at the base so the branch appears in different regions of the robots path. Once single branch avoidance is

successfully debugged, multiple branch avoidance can be attempted. Multiple dowels will be placed in the same telephone pole segment and the robot will be placed in several positions so the different vision regions are tested, as well as efficient path planning. Once completed and debugged, the robot will be ready for final demonstration.

4.2 Testing Working Criteria

After the individual components are tested to ensure that they are each working, the robot will be tested against the project's working criteria to measure the relative success of the project. A mock tree will be constructed from a 12' long, 12" diameter telephone pole as see in Fig. 17 for all testing.

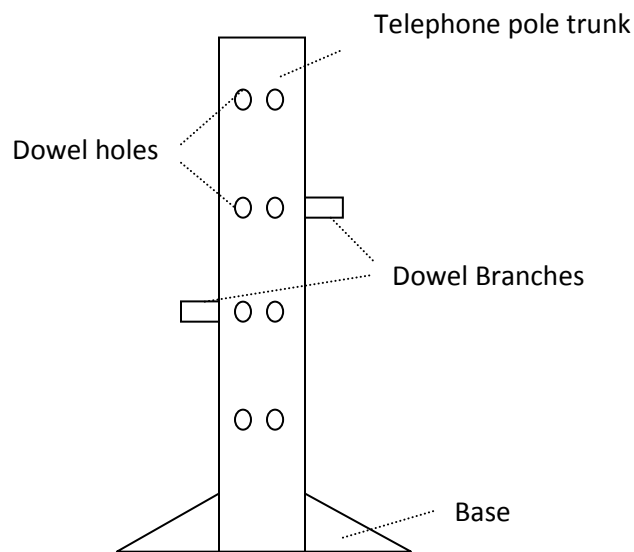


Figure 17. Test Setup

The bottom two feet will be used to attach a stand for stability while the robot is climbing. The next ten feet of climbing trunk will have six to eight holes spaced evenly around the circumference the slightly larger than 1" in diameter every two feet up the remaining length of

pole. These two inch deep holes will allow for 8” long 1” diameter dowels to be inserted to simulate various branch orientations.

By increasing the difficulty of branch locations, the maneuverability of the robot will be tested to see how practical the current robot would be at climbing real trees. This will start with having the robot avoid one branch, followed by two on different levels and circumferential locations, etc. The amount of additional weight the robot is capable of carrying will also be measured by slowly adding half pound increments to the frame and allowing it to climb two full extensions until the robot can no longer climb or begins to slide.

If there is extra time, additional testing to determine the limitations of the versatility of the robot could be performed to see if the robot could climb a wet surface by dampening the pole or to test the possible climbing diameters by adding material to thicken the pole.

5 Use of the Engineering Process

5.1 Design Constraints

Assuming that this project would lead to a commercial tree limb pruning robot for use in the logging industry, the greatest impact would relate to field of health and safety since the goal of this project is to increase the safety of loggers. The current logging industry is filled with hazards and has one of the highest work related deaths. The implementation of this robot will prevent people from having to wield a chainsaw high in the air while avoiding their safety harnesses. The operator could stay safely at the base and wait for the robot to return without ever entering harm’s way. Granted, some sort of safety precaution, such as a proximity sensor that kills the saw system of the robot, would need to be fashioned to ensure that the branches would

not be pruned if a person was in the drop zone. While there are still other major dangers involved with logging, this one task would be considerably safer.

The use of this type of robot would also have a great economic impact on the logging industry. Currently workers are not allowed to climb trees during inclement weather or trees with a high danger rating based on standard tree classifications. A robot could efficiently perform under both circumstances to increase production. The design is reliable enough that it should require little maintenance and in the long run would be cheaper than manual labor. The overall increase in safety could also cut down on liability and compensation caused by injury.

The most obvious effect of the robot on the environment would be an increase in deforestation caused by a higher productivity of the logging industry. This should be taken into account by the logging companies to replant more trees rather than simply clear cutting. While the body of the robot is constructed from sheet metal, which is easily reused, the remaining electronics cannot easily be recycled due to their complexity. Some of the computer parts also require the use of hazardous chemicals during the manufacturing process. During operation, the robot would ideally run on a battery which would have the same issue of potentially harmful chemicals and could cause pollution if damaged.

The blueprints of the robot and the subsystems were designed with manufacturability in mind. Arrangements were chosen based on the ease of assembly and which options would cut down on construction time. When comparing alternatives, the number of moving parts and the simplicity were a strong deciding factor. The simpler the design and the fewer moving parts will increase the reliability of a working system and make the robot fairly sustainable. The use of similar parts also helped make parts interchangeable and decrease the amount of programming

for the electronics. For example, since electric actuators were being used for the extension system, the same type of actuator was implemented for the clamps since the group was already familiar with the programming and necessary debugging. This ease of assembly would cause the commercial price of the robot to be relatively low, decrease necessary repairs, and increase the life of operation.

5.2 Design Process

Following the ten step engineering design process, the group is currently starting the final step. They have clearly identified the problem, decided on which criteria are important to the success of the project, researched what has been done in related fields, brainstormed possible solutions, evaluated which factors constrain the project, analyze the alternative solutions, and decided on a final design to solve the solution. Recently the group has just finished communicating and specifying the design through the use of its modeling program and schematics. This sizing of equipment has led to the ordering of parts for the implementation and construction of the actual robot.

According to the design cycle flow chart, the group is currently at the stage of building a prototype. This step precedes the inner prototype loop of testing and revising until the design meets specifications. Testing will be done individually on each subsystem/module before the systems are compiled for an easier time making modifications.

Both of these formats for the engineering design process apply directly to the project. Each step of the process has been completed progressively, with considerable time given to each step to ensure the best possible design. If the group had rushed to decisions early on, the design would be considerably different and would not accomplish the criteria as efficiently. The flow

chart also seems directed towards the developments of products because it includes the creation of a prototype. Since the project aims at creating a robot, the prototype inner revision loop is very important.

5.3 Project Plan

One of the group's first tasks of the semester was to create a project plan outlining the project's objectives, budget, and schedule. The first budget and schedule proposed were very rough estimates, based on the group's rough understanding of the project. As the semester progressed, however, the group became much more involved in the project, and came to a better understanding of the requirements. The budget and schedule naturally evolved along with this understanding to become more accurate and in depth. Microsoft Project was a useful tool in keeping track of this, since it allowed the group to make a detailed timeline, and compare its actual progress with its anticipated progress. The Project timeline is shown in Appendix E. The MMRs also forced the group to evaluate its progress each month, which helped to keep the design process moving forward.

The project objectives were set early on in the semester. When making decisions throughout the design process, the group referred back to these objectives to be sure that the final design would meet the project objectives.

5.4 Use of Engineering Tools

The group has utilized numerous tools throughout the design process thus far. EES was instrumental in analyzing the motion of the robot, as explained previously. After putting together a system of equations to model the forces and dimensions required of the robot, the group was able to vary dimensions in order to find the optimal solutions. This program was

chosen because the group was already familiar with it, after using it in other classes. The program also has the advantage of being user-friendly, while still being a very powerful and useful tool. The system of equations could have possibly been solved in Matlab rather than EES, but the programming required would have been much more complicated. Also, EES was ideal for creating parametric tables and conducting sensitivity analyses, which would have also been difficult if not impossible to achieve using another program. Matt Saunders and Rob did the majority of work in developing the EES code, while Amanda also contributed.

Another tool the group has used in the design process is ProEngineer. This has allowed the development of a 3-dimensional model of the robot, which has been very helpful. Once decisions were made about the subsystems and major parts, ProE was instrumental in deciding exactly how these parts would be put together, and in making sure that the parts are compatible with each other. ProE was used because it is readily available, and everyone is at least somewhat familiar with the program. Amanda did the majority of the modeling in ProE.

In determining the coefficient of friction of the treads, a force meter was used to measure the amount of force required to pull a loaded tread across a wooden panel, as previously described. This experimental setup was used because it was simple and its components were easily accessible. It also yielded fairly consistent results. The experiment was set up and executed by Matt Saunders, Matt Meador, and Amanda.

Appendix A: EES Modeling Code

Tree Climber Mechanical Systems

Sensitivity Variables

$$d_{a1} = 6 \text{ [in] } \textit{Distance from shoulder joint to actuator}$$

$$F_{act} = 20 \text{ [lbf] } \textit{Force of actuator}$$

$$n_{g2} = 21 \textit{Number of teeth on powered gear}$$

$$r_{roller} = 1.5 \text{ [in] } \textit{Constant radius of roller}$$

System 1: Arms

$$d_{shoulder} = 14 \textit{Distance between shoulder joints}$$

$$d_{a2}$$

Distance from shoulder joint to first roller

$$d_{a3}$$

Distance from shoulder joint to second roller

$$d_{act} = 9.4 \text{ [in] } \textit{Constant length of actuator collapsed}$$

$$d_{act,v} = 4 \text{ [in] } \textit{Stroke length}$$

$$\frac{d_{a2} + d_{a3,comp}}{2} = \frac{dia_{tree}}{2} \textit{Want rollers to bisect center of tree}$$

$$dia_{tree} = 12 \text{ [in] } \textit{Diameter of tree}$$

$$d_{a3,comp} = d_{a2} + L_{comp} + 2 \cdot L_{comp,2} \textit{Distance from shoulder to roller 2 during clamp}$$

$$x_{a3} = d_{a3} - d_{a3,comp} \textit{Distance roller 2 moves during clamp}$$

$$L_{arm} = d_{a3} + r_{roller} \textit{Min arm length to contain rollers}$$

$$T_{act} = T_{a2} + T_{a3} \textit{Torque of actuator}$$

$$T_{act} = F_{normal} \cdot d_{a1}$$

$$T_{a2} = d_{a2} \cdot F_{a2} \textit{Torque at 1st roller}$$

$$T_{a3} = d_{a3} \cdot F_{a3} \textit{Torque at 2nd roller}$$

$$F_{climb} = 2 \cdot F_{a2} + 2 \cdot F_{a3} \textit{Force required to stay on tree}$$

$$F_{climb} = weight \cdot \frac{n}{\mu}$$

$$weight = 20 \text{ [lbf]}$$

$$n = 1.25 \textit{Safety factor}$$

$\mu = 0.7$ *Coefficient of friction between tree and tread*

$$T_{act} = t_{a2.5}$$

$$t_{a2.5} = \left[\frac{d_{a2} + d_{a3}}{2} \right] \cdot f_{a2.5} \quad \text{Torque applied for point load}$$

$$f_{applied} = 2 \cdot f_{a2.5} \quad \text{Actual force applied for given act/angle/dim}$$

$$t_{clamp} = \frac{d_{act,v}}{speed} \quad \text{Total clamping time}$$

$$speed = 2 \quad [in/s] \quad \text{Extension speed}$$

$$F_{normal} = \cos[\theta] \cdot F_{act}$$

$$\theta = 0 \quad [deg]$$

System 2: Gears

$$n_{g1} = 1 \quad \text{Number of teeth on driven gear}$$

$$T_{g1} = 0.1085 \quad [lb*ft] \quad \text{Constant motor torque}$$

$$T_{g2} = \frac{n_{g2}}{n_{g1}} \cdot T_{g1} \cdot \left| 12 \cdot \frac{lb*in}{lb*ft} \right| \quad \text{Torque generated on powered gear}$$

$$rpm_{g1} = 170 \quad [rev/min] \quad \text{Constant motor rpm}$$

$$rpm_{g2} = \frac{n_{g1}}{n_{g2}} \cdot rpm_{g1} \quad \text{rpm generated on powered gear}$$

$$T_{fric} = \mu \cdot f_{applied} \cdot r_{roller} \quad \text{Rotational friction caused by clamping}$$

Angle of Deflection from tree

$$\sin[\theta_{compliment}] = \frac{d_{shoulder} - d_{act}}{2 \cdot d_{a1}}$$

$$\theta_{deflection} = 90 \quad [deg] - \theta_{compliment} \quad \text{Maximum angle arms deflect from tree}$$

Actuator Check

$$half_{actuator} = 3 / 4 \cdot 3.25 \quad \text{Maximum distance actuator can move toward tree}$$

$$act_{actual} = \sin[\theta_{compliment}] \cdot d_{a1} - half_{actuator} \quad \text{Actual distance actuator moves toward tree}$$

Other Dimensions

$$d_{gap} = 0.125 \quad [in] \quad \text{Gap between wheels and c-clamp side wall}$$

$$z = r_{roller} - 0.5 \cdot dia_{spring} \quad \text{The horizontal distance beyond tree diameter}$$

$$d_{shoulder2} = 2 \cdot r_{roller} - z + d_{gap} + dia_{tree} \quad \text{Actual shoulder pin distance}$$

System 3: Treads

Assume no slip between roller and tread

$$d_{\text{roller}} = d_{a3} - d_{a2} \quad \text{Distance between rollers}$$

$$C_{\text{tread}} = 21 \quad [\text{in}] \quad \text{Constant circumference of tread}$$

$$C_{\text{tread}} = 2 \cdot d_{\text{roller}} + r_{\text{roller}} \cdot 2 \cdot \pi$$

$$W_{\text{tread}} = 3 \quad [\text{in}] \quad \text{Width of tread}$$

$$A_{\text{tread}} = d_{\text{roller}} \cdot W_{\text{tread}} \quad \text{Contact surface area}$$

$$n_{\text{revolutions}} = 1 \quad [\text{rev}] \quad \text{Constant revolutions covered by wheel over 1 circumference}$$

$$\text{roller}_{\text{circum}} = \frac{\pi \cdot r_{\text{roller}} \cdot 2}{n_{\text{revolutions}}} \quad \text{Constant circumference of roller}$$

$$\text{rotation}_{\text{speed}} = \text{roller}_{\text{circum}} \cdot \text{rpm}_{g2} \cdot \left| 0.016666667 \cdot \frac{\text{in/s}}{\text{in/min}} \right| \quad \text{Rotational speed of entire robot}$$

$$F_{\text{tread,spring}} = k_{\text{tread}} \cdot x_{\text{tread}} \quad \text{Force of tread spring}$$

$$F_{\text{tread,spring}} = F_{a3} \quad \text{Max force, want less than this so will compress}$$

k_{tread}

Tread spring constant

$$x_{\text{tread}} = d_{\text{roller}} - [L_{\text{comp}} + 2 \cdot L_{\text{comp},2}] \quad \text{Compression of tread spring}$$

$$2 \cdot r_{\text{roller}} = \text{dia}_{\text{spring}} + 2 \cdot d_{\text{comp}}$$

$$\text{dia}_{\text{spring}} = 0.5 \quad [\text{in}] \quad \text{Diameter of spring}$$

$$L_{\text{arc}} = \pi \cdot \text{dia}_{\text{tree}} \cdot \frac{\theta_{\text{comp}}}{360 \quad [\text{deg}]} \quad \text{Parabolic contact length}$$

$$C_{\text{tread}} = L_{\text{arc}} + L_{\text{comp}} + 2 \cdot L_{\text{comp},2} + 2 \cdot L_{\text{wraparound}}$$

$$\sin [0.5 \cdot \theta_{\text{comp}}] = \frac{0.5 \cdot L_{\text{comp}}}{\frac{\text{dia}_{\text{tree}}}{2}} \quad \text{Angle of arc when compressed}$$

$$L_{\text{wraparound}} = 2 \cdot \pi \cdot r_{\text{roller}} - 2 \cdot \pi \cdot r_{\text{roller}} \cdot \left[\frac{180 \quad [\text{deg}] - \theta_{\text{force}}}{360 \quad [\text{deg}]} \right] \quad \text{Length of tread on roller when clamped}$$

$$\cos [\theta_{\text{force}}] = \frac{0.5 \cdot L_{\text{comp}}}{\frac{\text{dia}_{\text{tree}}}{2}} \quad \text{Angle of applied force}$$

$$\cos [\theta_{\text{force}}] = \frac{L_{\text{comp},2}}{r_{\text{roller}}}$$

Appendix B: EES Code Solution/ Final Dimensions

SOLUTION

Unit Settings: [kJ]/[C]/[kPa]/[kg]/[degrees]

<p>act_{actual} = -0.1375 [in]</p> <p>c_{tread} = 21 [in]</p> <p>dia_{tree} = 12 [in]</p> <p>⊙ d_{a2} = 2.682 [in]</p> <p>d_{a3,comp} = 9.318 [in]</p> <p>d_{act,v} = 4 [in]</p> <p>d_{gap} = 0.125 [in]</p> <p>d_{shoulder} = 14 [in]</p> <p>F_{a2} = 8.057 [lbf]</p> <p>F_{a3} = 9.8 [lbf]</p> <p>⊙ f_{applied} = 37.73 [lbf]</p> <p>F_{normal} = 20 [lbf]</p> <p>half_{actuator} = 2.438 [in]</p> <p>L_{arc} = 5.926 [in]</p> <p>L_{comp} = 5.688 [in]</p> <p>L_{wraparound} = 4.219 [in]</p> <p>n = 1.25 [-]</p> <p>⇒ n_{g2} = 25 [-]</p> <p>roller_{circum} = 6.283 [in/rev]</p> <p>rpm_{g1} = 170 [rev/min]</p> <p>⇒ r_{roller} = 1 [in]</p> <p>θ = 0 [deg]</p> <p>θ_{compliment} = 22.54 [deg]</p> <p>θ_{force} = 61.7 [deg]</p> <p>t_{a2.5} = 120 [lbf*in]</p> <p>T_{act} = 120 [lbf*in]</p> <p>⊙ T_{fric} = 26.41 [lbf*in]</p> <p>⊙ T_{g2} = 32.55 [lbf*in]</p> <p>w_{tread} = 3 [in]</p> <p>x_{tread} = 0.7219 [in]</p>	<p>A_{tread} = 22.08 [in²]</p> <p>dia_{spring} = 0.5 [in]</p> <p>⇒ d_{a1} = 6 [in]</p> <p>⊙ d_{a3} = 10.04 [in]</p> <p>d_{act} = 9.4 [in]</p> <p>d_{comp} = 0.75 [in]</p> <p>d_{roller} = 7.358 [in]</p> <p>⊙ d_{shoulder2} = 13.38 [in]</p> <p>f_{a2.5} = 18.87 [lbf]</p> <p>⇒ F_{act} = 20 [lbf]</p> <p>⊙ F_{climb} = 35.71 [lbf]</p> <p>F_{tread,spring} = 9.8 [lbf]</p> <p>k_{tread} = 13.58 [lbf/in]</p> <p>⊙ L_{arm} = 11.04 [in]</p> <p>L_{comp,2} = 0.474 [in]</p> <p>μ = 0.7 [-]</p> <p>n_{g1} = 1 [-]</p> <p>n_{revolutions} = 1 [rev]</p> <p>rotation_{speed} = 0.7121 [in/s]</p> <p>rpm_{g2} = 6.8 [rev/min]</p> <p>speed = 2 [in/s]</p> <p>θ_{comp} = 56.59 [deg]</p> <p>θ_{deflection} = 67.46 [deg]</p> <p>T_{a2} = 21.61 [lbf*in]</p> <p>T_{a3} = 98.39 [lbf*in]</p> <p>t_{clamp} = 2 [s]</p> <p>T_{g1} = 0.1085 [lbf*ft]</p> <p>⇒ weight = 20 [lbf]</p> <p>⊙ x_{a3} = 0.7219 [in]</p> <p>z = 0.75 [in]</p>
---	--

No unit problems were detected.

⇒ denotes important input

⊙ denotes important output

Criteria for success:

Clamping System

$$\text{if } F_{\text{applied}} > F_{\text{climb}}$$

Rotation System

$$\text{if } T_{g2} > T_{\text{fric}}$$

Appendix C: Bibliography

LMD18200 3A, 55V, H-Bridge. Datasheet, National Semiconductor, 2005.

Appendix D: Budget

Overall Total Cost
\$1,161.20

Product	Quantity	Unit Cost	Extended Cost	Date Ordered
Linear Actuator 4" Stroke - 40 lbf - 2"/sec	3	\$109.99	\$329.97	-
USB Adapter U2C-12 USB-12C/SPI/GPIO Adapter	1	\$69.00	\$69.00	11/15/2007
Sonar Sensors R287-SRf02 Devantech SRf02 Sensor	6	\$26.50	\$166.95	11/15/2007
Motor Controller MD23	1	\$187.50	\$197.25	-
Motors & Mounts EMG30 DC motor & mount	2	\$37.12	\$74.24	-
Integrated Circuit LM3524	2	\$1.98	\$3.96	-
Integrated Circuit LMD18200 H-Bridge	2	\$8.00	\$16.00	-
Integrated Circuit SN54LS00	2	\$0.54	\$1.08	-
Relays OMHI-L Relays	4	\$3.00	\$12.00	-
Integrated Circuit SN74LS03N	1	\$0.54	\$0.54	-
Circuit Board PCB	1	\$8.95	\$8.95	-

L-Bars w/Holes	6	\$8.46	\$50.76	-
Flat bars w/Holes	1	\$5.50	\$5.50	-
Sandpaper Tread	2	\$5.50	\$11.00	-
Rollers	16	\$4.00	\$64.00	-
Gears	16	-	-	-
Chain	8	-	-	-
Gearbox	2	\$30.00	\$60.00	-
Telephone Pole	1	\$36.00	\$36.00	-
Dowels	4	\$1.00	\$4.00	-
Misc.	1	\$50.00	\$50.00	-
Screws, Nuts, Hinges				

Appendix E: Timeline

