WORCESTER POLYTECHNIC INSTITUTE
ROBOTICS ENGINEERING PROGRAM

Robotic Replacement of Solar Collector Panels

Team 13

Submitted By: Connor Craigie, Nathan Stallings, Victoria Thornton

Date Submitted: 3/1/2019

Course Name: Unified Robotics I

Course Professors: Bertozzi and Miller

Lab Section: RBE 2001-C02

Abstract

The goal of this project was to design and develop a robot capable of picking up and placing solar panels on a roof of 25 degrees and 45 degrees. it should also be able to place and pickup plates from the staging area, which is two inches from the ground. The robot must be able to pick up two plates of different weight. An additional goal for our team was to perform both sides of the field in one attempt. We were provided with a drive train, and we performed calculations to determine that in the case of a collision, the wheels of the robot will slip rather than stall the motor. We also had to design a four-bar capable of maneuvering to certain locations. We performed a four-bar linkage synthesis using perpendicular bisectors, and a free body diagram system to solve for the output torque of 20.24 in-lbf. Using the datasheet for our motor, we solved for a lifting speed of about 3 seconds for our four bar to translate 130 degrees. We also designed a unique gripper that was capable of opening in both directions and it used a toggle point to keep closed on the plate. The combination of the four-bar and gripper allowed us to place both types of plates on both roof angles. The designing of the robot was very successful, however, the programming and the performance of the robot was not as successful. Our robot was capable of completing some of the tasks, but the lack of time prevented us from completing the whole challenge with consistency.

Table of Contents

List of Figures

I. Introduction

The problem we were tasked to solve was creating a robot with the capabilities of replacing solar collector panels on a planet with two suns. The robot must be capable of autonomously picking up a panel from the staging area and placing it on the empty slot on the solar farm roof. It must also be able to pick up the solar panel already on the roof and place at the staging area. Since this planet has two suns, there is another side of the field with a solar farm roof at a different angle. The robot must have the ability to repeat the above process on both sides of the field. This means the four-bar must be set up to reach the required angles of 25 and 45 degrees. Furthermore, although not necessary, we designed our robot to complete the whole operation in one run, meaning the robot must be capable of relocating to the other side of the field. Due to uncertainties in the exact location of the solar panels, the robot must have the ability to communicate with the field to determine which spots on the roof contain the solar panels. Furthermore, the robot must wait for the confirmation from the field before it places or picks up any panels from the roof. In order to complete this task successfully, we needed to design, and manufacture a functioning four-bar and gripper. The gripper and four bar system must be strong enough to move the solar panels. We were provided with a drive chassis and line follower, which allowed us to incorporate them into our final design. We also had to develop code capable of performing the desired tasks. We developed a PID to ensure that the motors achieved the desired setpoints and wrote autonomous code allowing the robot to solve the problem.

II. Methodology

Our team prioritized creating functional geometry. We wanted to assure that plate placement would be correct in all three positions. Therefore, our design is centered around a two-dimensional sketch which defines the movement of the four-bar. This was done by setting up a sample solar farm roof that had resting panels at 25 degrees, 45 degrees, and a sample staging location with a panel parallel to the ground two inches high. We then placed a drawing of our gripper at the three locations. An important aspect of our design is that our coupler does not line up for 25-degrees. It is set to about 27 degrees. This was a design choice that allows us to easily place the plate down at the 25-degree angle without worrying about hitting the roof or struggling to line up the plate. We are still capable of picking up the plate due to our unique gripper design which opens from both the top and bottom. We found the desired placements of our four-bar by creating the perpendicular bisectors from the ends of the gripper. We were able to calculate the link lengths from this information, but due to their location, we had to design bends in the links to prevent a collision.

The next step in the design process was centered around the four-bar. We wanted to easily control the location of the coupler with our four-bar. We decided the best way to do this was to gear the four-bar for torque rather than speed. We started this process by finding the required torque to move the four-bar when a plate was being held in the gripper. We picked this configuration because it is the heaviest the four-bar system will ever be. We created free body diagrams for each link and used SolidWorks to approximate the center of gravity and material weight. From this information, we used Mathcad to solve a system of equations derived from the forces of each link acting on each other. We then solved the system of equations for the input torque. With this information, we decided that we wanted our four-bar to move 130 degrees in about 3 seconds, providing us with a good amount of control over the coupler position. We also performed a four-bar component speed analysis. The four-bar component speed was calculated in SolidWorks. SolidWorks internally computes the linear velocities over time of our key points in the linkage. To manually calculate these values, you must determine the geometric position of the instantaneous center and calculate linear velocities of the coupler from this point. SolidWorks plots the magnitude of these linear velocities as the instantaneous center is changed infinitely over time.

The last major component of the design was the Gripper. Our team figured that a gripper which actuates gripping surfaces in two opposite directions would provide much more reliability and produce a larger margin of positioning error. Even if the plate was placed in the gripper at an inconsistent angle, the range of the gripper would correct for this error and reorient the plate when undergoing a closing operation. The gripper uses a set of three four-bars to complete its range of motion. To assure the least load is placed on the motor, the gripper returns to a toggle point when holding the weight of the plate. With this geometry, the motor which actuates the gripper requires no extra torque to hold the plate in position. Due to the toggle point, the force produced by the weight of the plate is applied directly on the axle of the motor and therefore produces no torque on the motor axle.

A drive chassis and motors were provided for us to use, which we decided to do. Because of this, the calculations and design related to the drive train are minimal. However, we still performed an analysis on the wheels to make sure if we collided with a wall, we would not stall the motors. We did this by finding the diameter of the wheels and looking up the stall-torque of the provided motors. We considered the provided gear ratio to determine whether it would be a slip or stall condition. The motors we used where Pololu Motors and they were provided for the class. We performed analyses to ensure that they meet the correct requirements and would provide the necessary torque for our robot design.

Once we had a gear ratio determined, we had to design and analyze them to ensure they would not break under the force applied to them. Our Gears were designed with Common Wealth Robotics from Bowler Studio. This allowed us to easily make function generated gears for our robot. Since we already knew the gear size from the four-bar calculations, the gears were generated, and we could measure the tooth thickness and width. We wanted to make sure that our gears would not break under the applied force, so we calculated the factor of safety for both sized gears to ensure that neither would break.

The last step in the process of building our robot was writing the code for it. For the code, we began from the template that was provided for us by Kevin Harrington. This base code included RBE2001Code13.ino, DriveChassis.h, DriveChassis.cpp, RBEPID.h, RBEPID.cpp, StudentsRobot.h, StudentsRobot.cpp, and config.h. The StudentsRobot class came with initialization of motors 1 through 3 and a servo and also 7 states for the main state machine. In lab 3, we added code for the functions distanceToWheelAngle(float distance), chassisRotationToWheelDistance(float angle), driveForward(float mmDistanceFromCurrent, int msDuration), and turnDegrees(float degreesToRotateBase, int msDuration). For the function distanceToWheelAngle we used the equation (2*pi*r / (360*distance))^-1 to calculate a delta in wheel angle to get to a specific distance as seen in (Figure 2.1). For the function chasissRotationToWheelDistance we used the equation (2*pi*r* (angle/360)) to calculate the arch length distance for the robot to travel based on the wheel track as seen in (Figure 2.1).

distanceToWheelAngle function

```
/**
   Compute a delta in wheel angle to traverse a specific distance

   arc length  = 2*  π*  R*  (C/360)

   C  is the central angle of the arc in degrees
   R  is the radius of the arc
   π  is Pi

   @param distance a distance for this wheel to travel in MM
   @return the wheel angle delta in degrees
*/
float DrivingChassis::distanceToWheelAngle(float distance) {
  float R = 25.4; //our wheel has a diameter of 1.9 inches or 48.26 mm
  float C; //declares varable for central angle
  float pi = 3.14159; // had to make a variable for Pi because the arduino function PI was not working.
  C = (2 * pi * R) / (360 * distance); //calcualtes the inverse of the central angle
  C = 1 / C; // calculates the central angle by inverting the above calcualtions
  return C;
}
```

Figure 2.1: A snipping of the distance to wheel angle function

chassisRotationToWheelDistance function

```
/**
   Compute the arch length distance the wheel needs to travel through to rotate the base
   through a given number of degrees.

   arc length = 2*  π*  R*  (C/360)

   C  is the central angle of the arc in degrees
   R  is the radius of the arc
   π  is Pi

   @param angle is the angle the base should be rotated by
   @return is the linear distance the wheel needs to travel given the this CHassis's wheel track
*/
float DrivingChassis::chassisRotationToWheelDistance(float angle) {
  float R = 285; //our chassis has a radius of 142.5 mm
  float d; //declares varable for the needed distance
  float pi = 3.14159; // had to make a variable for Pi because the arduino function PI was not working.
  d = 2 * pi * R * (angle / 360); // calculates the linear distance needed to achieve a certain wheel base degree
  return d;
}
```

Figure 2.2: A snipping of the chassis rotation to wheel distance fucntion

after filling in the bodies of these two functions to calculate conversions, we then used them inside of the functions driveForward and turnDegrees so that we could simply type in a value for degrees and a duration when we wanted to use these functions. For the bodies of driveForward and turnDegrees, we used the two conversion functions to calculate the setpoint, and then we used the startInterpolationDegrees function to get to that setpoint as seen in (Figure 2.2).

Once we started testing the robot on the field, we found that we need to add a few more functions to the DriveChassis class. Two of those functions were oneWheelRotateClockwise(float degreesToRotateBase, int msDuration) and oneWheelRotateCounterClockwise(float degreesToRotateBase, int msDuration). In the turnDegrees function, the robot makes a turn using both of the wheels with one wheel going in the direction opposite the other one at the same speed. After testing out a routine to back up from the staging platform stop at the first black horizontal line, back up 240 mm, and then make a 90 degree turn, we found that at the completion of the 90 degree turn the back casters of the robot would be off of the table. The two new functions changed this 90 degree turn to pivoting around just one wheel instead which resulted in a decrease in our turn radius and allowed our robot to stay on the table after making a 90 degree turn at that point. We also added a driveBackwards function as well so that we wouldn't have to put a negative value into driveFoward function in order to drive backwards. The last changes for the DriveChassis class included adding the functions followLine(String direction) and stopAtLine(). The function followLine reads input from the QTR Reflectance Array, checks the values of 2 sensors performs 1 of 3 functions each time it is called based on the values of the sensors. After testing on the field, we found the threshold for the black line to be above 3000. In the function followLine the first thing it does is run the setVelocityDegreesPerSecond function to start driving forwards or backwards depending on the direction input. Next, if the left sensor value reads black, the robot turns to the right, if the rightSensor sensor value reads black, the robot turns to the left. If both of the sensors read black then the function sets a boolean value, onHorizontalLine to true, which allows the state machine to know when to transition to the next state. Along with changes to DriveChassis we also made changes to the RBEPID class in lab 3. We tested the deadband values for the motors and completed the PID controller by adding equations to calculate the integral and derivative terms. After running tests in the field controller jar file, we were able to optimize the gains for our PID controller. We put these optimized gains into the RBEPID class.

The next step in the process included creating two classes to cover the other two subsystems of the robot, the Lift and the Gripper. The Lift class includes methods such as lift25 and lift45 to get the lift to preset positions for the 25 and 45 degree sides of the solar farm. The Gripper class includes methods to open and close the gripper.

After the class structure was set up, the final steps included editing the state machine. To be able pick up and place plates on both sides of the solar farm based on field input, we used 13 states. Through testing, we found that for functions like startInterpolationDegrees we would have to wait a time for that function to finish before moving on to the next state in the state machine. To do this, we used three variables, pastTime, currentTime, and waitTime to tell how much time has passed between a state transition. We also found it helpful to use the overrideCurrentPosition() function to reset the current motor position to 0 when transitioning between states as well. The first state of the state machine is StartupRobot where the line sensors are calibrated before moving on to the CollectFieldInformation state. In the CollectFieldInfomation state, the robot waits for input from the field controller app, and chooses which state to move on to next based on the information provided. After testing, we decided to have two separate programs, one for the 25 degree side and one for the 45 degree side.

III. Analysis

The first step of making our four-bar was determining the link lengths and position on the base of our robot. This was accomplished by using the method of perpendicular bisectors. Once we created the solar panel roof and placed our gripper to the positions we wanted, we drew lines connecting the lower connection points of the gripper, which acts as our coupler (See Figure 3.1). We repeated this process for the upper connection points. This results in the creation of four lines, two of which connect the upper connection point of the coupler, and two that connect to the lower connection point of the coupler. We then go to the center point of these lines and draw perpendicular lines. Where the two perpendicular lines coming from the lower coupler connection points cross is the location of one link, and where the two other lines intersect is the location of the other link. From this information, we can determine the lengths of our links and where on the base of the robot they must be located.

Perpendicular Bisectors

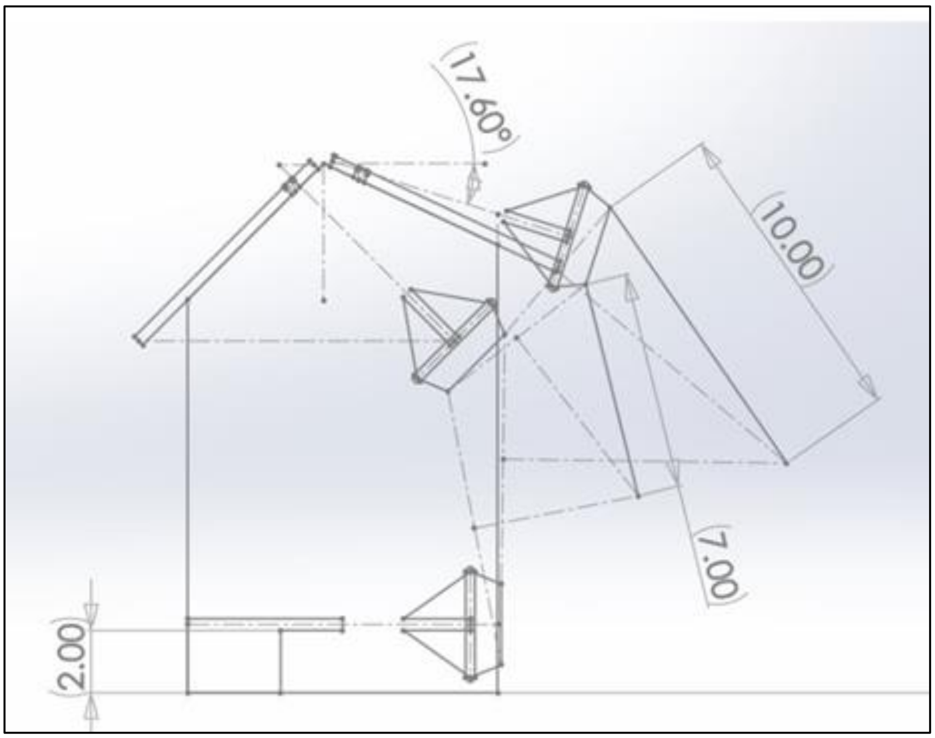


Figure 3.1: This figure shows the process of perpendicular bisectors and demonstrates how we set up our 2D sketch to perform a linkage synthesis.

The next step involved performing an analysis for the speeds of the four-bar's components. This was done using SolidWorks, and the resulting graph is displayed in figure 3.2.

Four-Bar Component Speed

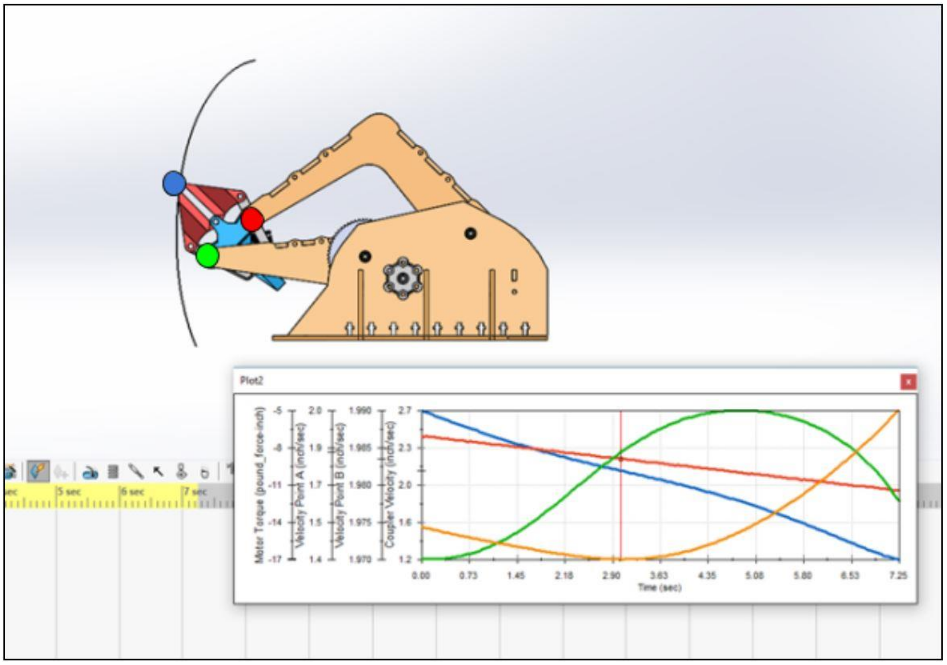


Figure 3.2: The figure above depicts an overlaid graph of point velocities measured at different locations within the four-bar linkage

The orange line represents the transmission output torque required to provide equilibrium in the system and therefore remain moving at a constant velocity. Due to the orientation of the torque, the value is displayed as negative. The largest required magnitude of torque exists just after 2.9 seconds and is graphically depicted by the vertical red line. This point in time is also physically depicted by the position of the linkage above. This knowledge of physical displacement was crucial to our team's decisions moving forward with free body diagrams. It was clear the instance with the highest magnitude of torque existed when the driving link was horizontal to the ground. This max magnitude is graphically depicted to be 17-inch pounds of torque. Note that this value does not include the weight applied by the steel plate when in the gripper. The blue graph represents the linear velocity of the end effector of the gripper over time, the red graph represents the linear velocity of point B and the green graph represents the linear velocity of point A attached to the driving link.

A free body diagram was created for each component of the four-bar. A system of equations was created and solved to determine certain aspects of the four-bar.

FBD of Link A

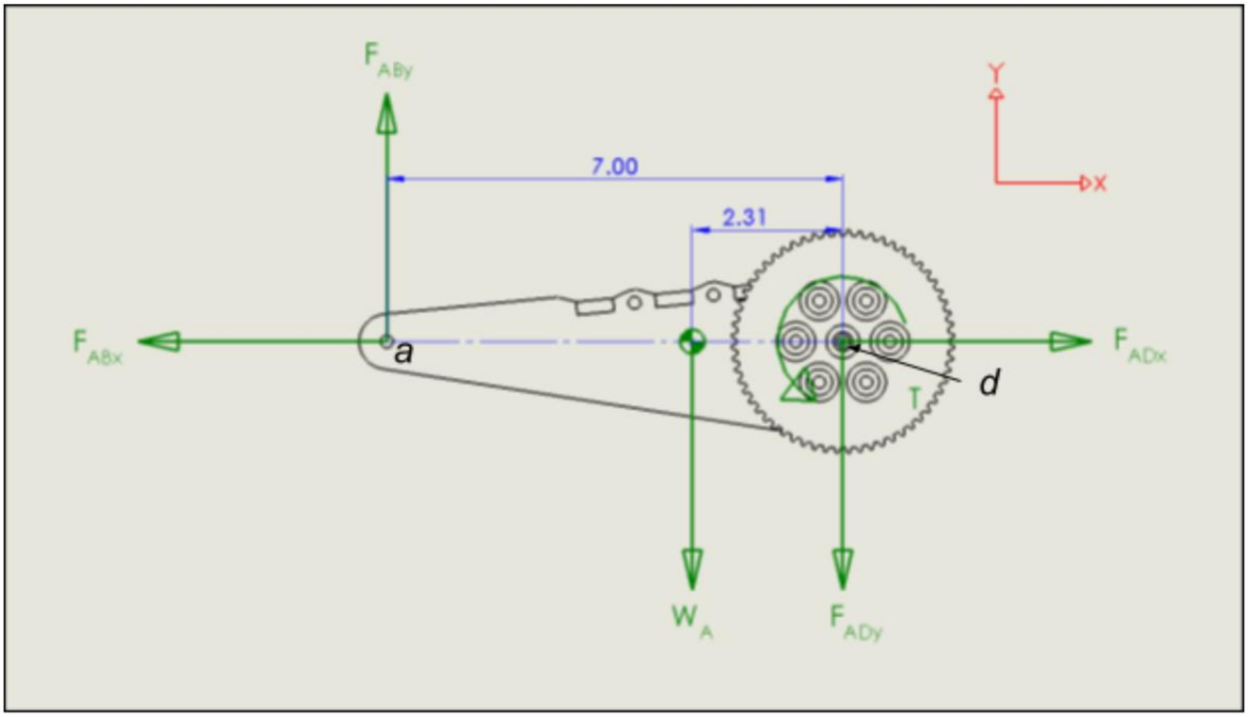


Figure 3.3: The free body diagram depicted above represents the forces enacted on link A.

In figure 3.3, point *a* is considered to be a free spinning join. For the purpose of simplification, we will assume all free spinning joints exist as frictionless and therefore only have enacting forces in the x and y directions. Point d has an acting torsional force that we would like to solve for. This variable will be represented as T and we have written it in the positive direction (counterclockwise about the z-axis). The final enacting force on link A is its weight force which exists -2.31 inches in the x-direction from point *d*.

Summation of Equations for Link A

LINK A: length ---> 7in

Knowns

$W_A := .25\text{lbf}$ $\quad L_B := 7\text{in}$

$d_A := 2.31\text{in}$

Equations of Equilibrium: LINK A

$$0 = W_A \cdot d_A - F_{ABy} \cdot L_B + T_A$$

$$0 = F_{ADx} - F_{ABx}$$

$$0 = F_{ABy} - W_A - F_{ADy}$$

Figure 3.4: A system of equations relating all the forces acting on Link A.

With the variable in figure 3.4, we can write a system of three equations with four unknowns. The length of the link is defined by LB as 7 inches. The distance to the CG is defined by dA as 2.31 inches. The weight of link A is defined as WA to be 0.25 pounds. With these displacements and force orientations, we can create a summation of moments. We do this about point d to negate the linear forces acting on those points.

Our second free body diagram represents link B (Figure 3.5). This system is comprised of two free spinning joints. Therefore, it will have four linear forces exerted on points *a* and *b*. This system also has the introduction of a fifth linear force in the form of the steel plate being manipulated by the gripper.

FBD of Link B

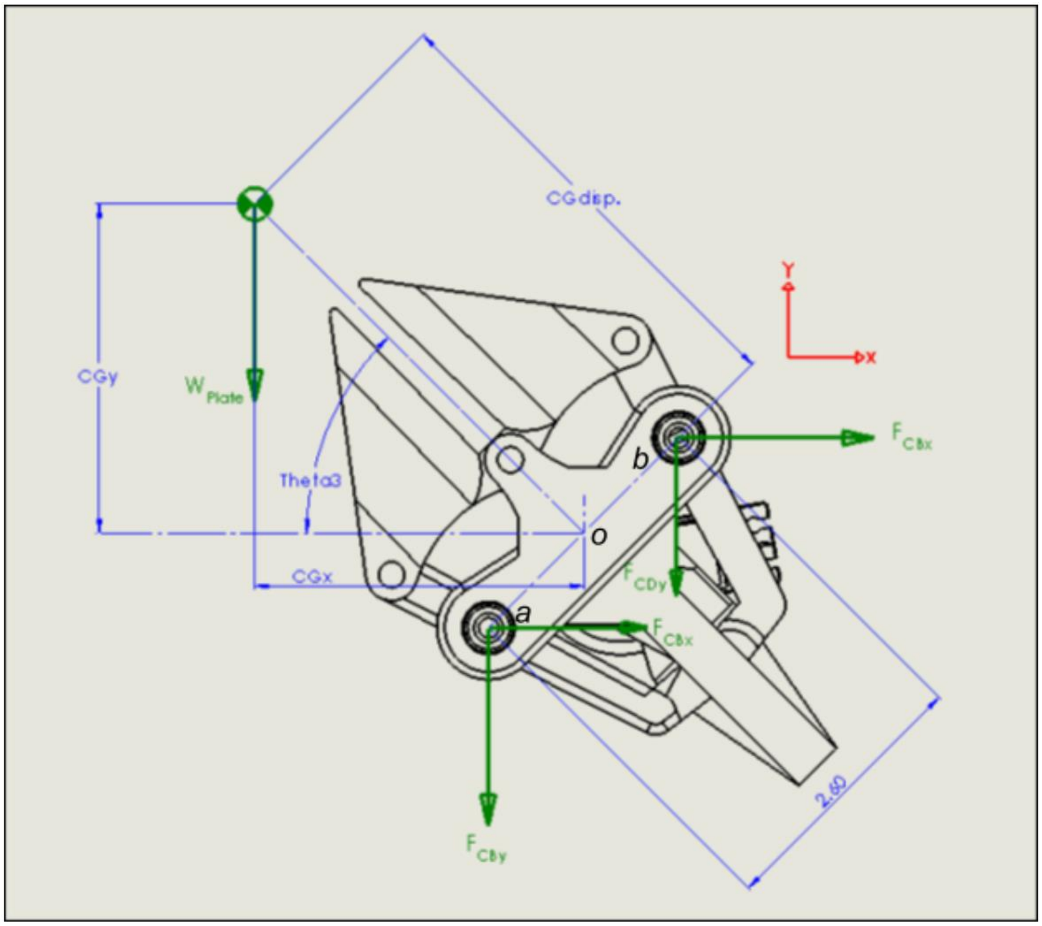


Figure 3.5: The free body diagram depicted above represents the forces enacted on link B.

Summation of Equations for Link B

LINK B: Length ---> 2.6in

Knowns

$theta_3 := 45.282deg$

$L_{BB} := 1.3in \cdot cos(theta_3) = 0.915 \cdot in$

$H_B := 1.3in \cdot sin(theta_3) = 0.924 \cdot in$

$CG_x := 2.17in = 2.17 \cdot in$

$CG_y := 2.16in$

$CG_{disp} := \left(CG_x^{\,2} + CG_y^{\,2}\right)^{.5} = 3.062 \cdot in$

$W_B := 3.02lbf$

Equations of Equilibrium: LINK B

$0 = W_B \cdot CG_{disp} + F_{ABy} \cdot (L_B) + F_{ABx} \cdot (H_B) - F_{CBx} \cdot (H_B) - F_{CBy} \cdot (L_{BB})$

$0 = F_{CBx} + F_{ABx}$

$0 = -W_B - F_{ABy} - F_{CBy}$

Figure 3.6: A system of equations relating all the forces acting on Link B.

Link B was offset by nearly a 45-degree angle; therefore, points a and b exist outside of the same linear plane. The summation of moments was made about a central point O between points a and b. This means that our summation of moments will include a summative term for all five forces applied in the system. LBB represents the x component displacement of point b from the origin point O. Similarly, HB represents the y component displacement of point b from the origin point O. The origin exists exactly between points a and b, therefore we can assume the displacements calculated for point b are the same for point a. Using this symmetry, we will re-use these displacements as coefficients when calculating the moments produced by forces which exist at joint a.

FBD of Link C

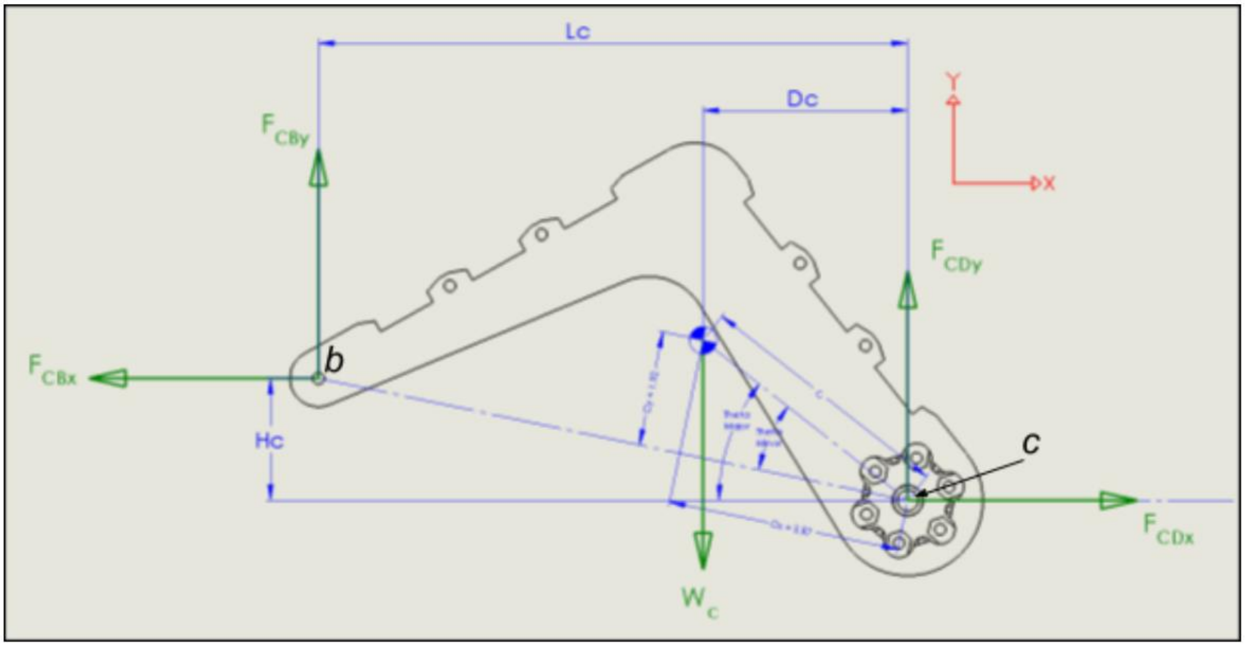


Figure 3.7: The free body diagram depicted above represents the forces enacted on link C.

The free body diagram of Link C exists at an odd angle in relation to the x y coordinate system. This link also has a significantly displaced center of mass from line bc. To relate the displacement of the CG to the x y coordinate system we calculate the linear distance c to the CG. Using the linear distance c and theta major we can calculate the x and y component displacement of the CG from point c. Once these constants are solved, we can then produce an accurate summation of moments.

Summation of Equations for Link C

LINK C: Length ---> 10in

Knowns

$C_{cx} := 3.87in \qquad C_{cy} := 1.92in$

$C_c := \left[1 \cdot \left(C_{cx}{}^2 + C_{cy}{}^2\right)\right]^{.5} = 4.32 \cdot in$

$theta_4 := 175.5deg$

$theta_{Minor} := atan\left(\frac{C_{cy}}{C_{cx}}\right) = 26.387 \cdot deg$

$theta_{Major} := 180deg - \left(theta_4 - theta_{Minor}\right) = 30.887 \cdot deg$

$W_C := .38lbf$

$d_C := C_c \cdot cos\left(theta_{Major}\right) = 3.707 \cdot in$

$L_c := 10in \cdot cos\left(180deg - theta_4\right) = 9.969 \cdot in$

$H_c := 10in \cdot sin\left(180deg - theta_4\right) = 0.785 \cdot in$

Equations of Equilibrium: LINK C

$0 = W_C \cdot \left(d_C\right) - F_{CBy} \cdot \left(L_c\right) + F_{CBx} \cdot \left(H_c\right)$

$0 = F_{CDx} - F_{CBx}$

$0 = F_{CBy} + F_{CDy} - W_C$

Figure 3.8: A system of equations relating all the forces acting on Link C.

Theta 4 relates the rotational orientation of line bc to the x-axis. From here we have manipulated the given geometry of the CG to produce its x and y component displacement from point c. Both points b and c exist as free spinning joints, therefore they will each have two component forces acting on them. We have summed the moments about point c to negate the linear forces acting on that point.

The last step for the four-bar analysis was calculating the four-bar speed and power requirements. The first step to this process is finding the amperage drawn by the motor. In order to do this, it is important to note that current has a linear relationship with torque. We can, therefore, set up an equation using f(x) $= \frac{Rise}{Run}x + b$, where x is the variable changing with time, torque, and b is the y-intercept. In terms of torque and amperage, the equation is A $= (Astall - Anoload * \%T + Anoload$. In this equation, %T is equal to x/run. This means that it is equivalent to the applied torque divided by the stall torque. The applied torque is given by multiplying the required torque found in our free-body diagram by the calculated gear ratio of our four-bar system. This will give us the output torque given by the four-bar. With this information, we can find the percent of torque used with the equation %T $= \frac{T_{applied}}{T_{stall}} * 100$. The stall amperage and the no-load amperage are provided by the datasheet on the motor. Plugging these values along with the calculated percent torque into the above equation will give us the amperage drawn by the motor. With this information, calculating the power is simple. The equation for power is P=V*I, where V is voltage and I is the amperage. The Voltage the motor is run at is given by the datasheet. Multiplying this value with the amperage drawn by the motors will give us the power requirements for the four-bar.

Four-Bar Power

Robot Lift Current

Note : Current is linearly related to the percentage of stall toruqe

Due to the fact that this motor is not a frictionless system, the no load condition of the motor will still draw a base amount of current. From this base current there is a linear increase in current untill it reaches the stall current rated by the manufacturer.

As the %Torque is increased, so does the amerage draw of the motor.

Therefore, we can determine that $A = (A_{Stall} - A_{No\_Load})(\%T)+(A_{No\_Load})$

We calculate the T applied by the motor by multiplying it by the constant of our set gear raito

Given

$A_{stall} := 5A$ $\quad$ $A_{NLoad} := .3A$

$T_{req} := 20.24 in \cdot lbf$ $\quad$ $T_{stall} := 10.625 in \cdot lbf$

$$T_{applied} := T_{req} \cdot \frac{1}{25} = 0.81 \cdot in \cdot lbf$$

$$\%T_{used} := \frac{(T_{applied})100}{T_{stall}} = 7.62$$

$$A_{draw} := (A_{stall} - A_{NLoad})\left(\frac{\%T_{used}}{100}\right) + (A_{NLoad}) = 0.658 A$$

Power Consumption

Note : The motors run at full 12V

$V_{run} := 12V$

$$P := V_{run} \cdot A_{draw} = 7.898 W$$

Figure 3.9: This shows the process for calculating the four-bar power requirements

In order to calculate the speed our four-bar will move it, we needed to set some desired constraints to its performance. We knew from our CAD that the four-bar will have about 130 degrees of freedom. A normal motor curve shows that a motor is most efficient around 20% of the stall torque (see appendix figure 9.1). However, we wanted a large amount of control over our four-bar for good precision, so we decided to run our motor at 7.5% of the stall torque, which is still efficient for the motor. With this information, we can find our desired gear ratio. The equation for gear ratio when considering torque is $Ratio = \frac{Tin}{Tout}$. Tout was calculated during our four-bar free body diagram analysis. Tin is the input torque provided, which can be found by multiplying the stall torque by the percent of stall torque we will run the motors at. Once we have the output torque and the input torque, the gear ratio can be found. The next step is converting this percent torque into a rotational speed output by the motor. When calculating this, it is helpful to recognize that Speed and Torque have a linear relationship. As torque increases, it causes the speed to decrease. This means that we can find the rotational speed of the motor by multiplying the no-load speed with 1-%stall torque. With the rotational speed output by the motor, we can find how long it will take for our four-bar to move the 130 degrees. The first step is to find the output speed the gear ratio will provide. This is done using the equation $Speed\ out = \frac{rotational\ speed}{gear\ ratio}$. Using the equations above, we can find the gear ratio and angular speed. Once the output speed is found, the lift time can be calculated without a unit conversion by using $time = \frac{angular\ displacement}{speed\ out}$, where the angular displacement of our lift is 130 degrees and the speed out was calculated with the above equations. This will give us the time it takes the four-bar to transgress the desired angular range.

Lifting Speed Calculations

Lifting Speed Calculation

A basic DC motor is known to want to operate near 20% or less of its total stall torque. This is the most effident mode of operation for the motor, and inturn reduces the heat produced by the system.

It is also better to have a slow moving linkage system to allow for more precise movement of the end affector.

Therefore : We will set the gearing of the lifter to result in a motor output of 7.5% total torque during its period of highest power consumption

Given

$\%T := 7.5 \cdot 10^{-2}$

$Spd_{Motor} := 200rpm$

$Spd_{Load} := (Spd_{Motor}) \cdot (1 - \%T) = 185 \cdot rpm$

Required Gear Ratio : This will be calculated with the given constant of percent torque.

$T_{Stall} := 170ozf \cdot in$

$T_{in} := T_{Stall} \cdot \%T = 0.797 \cdot in \cdot lbf$

$T_{out} := 20.24in \cdot lbf$

$Ratio_{Exact} := \frac{T_{out}}{T_{in}} = 25.399$

$Ratio_{Est} := 25$

Speed Calculations Based of of the determined ratio

$Spd_{out} := \frac{Spd_{Load}}{25} = 7.4 \cdot rpm$

$Theta_{disp} := 130deg$

$Lift_{Time} := \frac{Theta_{disp}}{Spd_{out}} = 2.928\,s$

Figure 3.10: This shows the process for calculating the four-bar speed

We also performed an analysis on the provided drive chassis to ensure it would meet all our requirements. Since there was no design process included in this step, all we had to do was determine the Linear speed and Power requirements of the motor with the given transmission. The gear ratio was a 3:1, and the wheels had a diameter of 1.9 inches. The first step to finding linear speed is finding the rotational speed. This is found by multiplying the motor RPM by the gear ratio of the transmission. Looking at the spec sheet for the motors, we can determine that the rotations per minute output by the motor. Once this is multiplied by the gear ratio, which is Gear driving/ Gear Driven, the rotational speed needs to be converted to linear speed. This can be done by multiplying the rotational speed with the circumference of the wheel. The circumference of the wheel can be found using the equation C=Pi*2r or C=Pi*d, where C is the circumference, r is the radius, and d is the diameter. We can also calculate the power required by one of these motors. The electrical power equation is P=I*V, where P is power, I is current, and V is voltage. Both voltage and no-load amperage were provided by the spec sheet for the motor.

Motor Power Calculations

Motor Drive Calculations

$$Wheel_{OD} := 1.9in$$

$$Gear_{Driving} := 16$$

$$Gear_{Driven} := 48$$

Motor Information:

Gearbox Ratio ---------------------- 50:1
No Load Speed ------------- 200 RPM
Stall Torque ------------------- 170oz in
Stall Current ---------------- 5A (@12v)

Given

$$Motor_{RPM} := 200rpm$$

$$Drive_{RotSpd} := (Motor_{RPM})\cdot\left(\frac{Gear_{Driving}}{Gear_{Driven}}\right) = 66.667\cdot rpm$$

$$Wheel_{Circ} := 3.14159\cdot Wheel_{OD} = 5.969\cdot in$$

$$Drive_{LinSpd} := Drive_{RotSpd}\cdot(Wheel_{Circ}) = 2.5\times10^{3}\cdot\frac{in}{min}$$

Power Requirement

$$A_{NoLoad} := .3A$$

$$V_{in} := 12V$$

$$P := V_{in}\cdot A_{NoLoad} = 3.6W$$

Figure 3.11: This shows the process for analyzing the drive train power

Since the drive train was provided, we had to ensure that if the robot collided with a wall, the wheels would be at a slip condition, rather than a stall condition. A stall condition would be bad for the motors and we would have to redesign the provided wheels. In order to determine the condition, we started by determining the torque the motor would stall at. This equation is found by taking the given stall torque and multiplying by the drivetrain ratio. The equation can be found in figure 3.12. For the wheels to slip, the stall torque from the motor must be able to overcome the frictional force. The friction force can be found with the equation $Ff = \mu * W$, where W is the weight of the robot and μ is the coefficient of friction. The weight of our robot came out to be 8.75 lbs. By setting up a sum of moments around the wheel, we can determine what will happen upon collision. The sum of moments equation for our wheel is $\Sigma M = 0 = T - (r * Ff)$, Where T is the stall torque, r is the radius. And Ff is the force of friction. By solving for the Friction force equation, we can determine if the motor will slip or stall upon collision.

Stall Torque

$$T_{in} := 170 \mathrm{ozf} \cdot \mathrm{in}$$

$$T_{Stall} := T_{in} \cdot \left(\frac{Gear_{Driven}}{Gear_{Driving}} \right) = 31.875 \cdot \mathrm{lbf} \cdot \mathrm{in}$$

Figure 3.12: This figure shows how to find the stall torque.

The next step was to analyses our designed gears to ensure they would be able to perform to our expectations. The restraining factor for a gear is most commonly a shear stress force acting on the tooth. This will be the first place a gear breaks. Shear stress is defined as the component of stress coplanar with a material cross section [1]. From this information, it can be inferred that to find the shear stress force is equal to the force applied over an area, resulting in the equation $\tau = \frac{F}{A}$, where F is a force and A is the area. The gear tooth is connected by a rectangular cross-section to the dedendum circle. Since our teeth were by a computer program, we were able to take the face width and thickness of the gear allowing is to calculate the cross-sectional area with Area=Width*Thickness. The next aspect of the shear stress equation is calculating the force acting on the gear tooth. Since we know the specs of our motors, we know the torque provided by them. Using a modified version of the torque equation T=F*d, where T is torque, F is a force, and d is distance, we can solve for force. Since the shaft of the motor is located at the center of the gear, the distance will be the distance from the center of the gear to the pitch circle. This was a measurable distance for us since our gears were already generated. With this information, we could successfully calculate the force acting on one gear, but since the two gears are directly driving each other, the forces acting on one gear equal the forces acting on the other gear. The only difference will be the cross-sectional area of the gear teeth. Once each aspect of the shear stress equation is found, we can calculate the shear stress from both the driven and driving gears. The next step is to find the factor of safety to ensure our teeth will not break. The factor of safety equation is $F.O.S = \frac{Shear\ Max}{Shear\ Stress}$. The Shear max is a given property based on an object's material. Homework 5 gave us a Shear max of 2500 psi for a 3d printed plastic. Since we planned on 3d printing our gears, we deemed this an acceptable value.

Gear Tooth Analysis

Shear Stress= F/A   Ultimate Stress =2500 psi   FoS= ultimate stress/ working stress

Area= thickness* width   Force=Torque/radius

The force is caulcuated using the output torque of the gearbox. It is caulcuated using the 60 tooth gear, which is the driven gear. It is important to note that the force acting on both gears will be the same.

60 Tooth Gear

Torque

$t_{out} := 20.24 \text{in} \cdot \text{lbf}$

Shear Ultimate Stress

$S_{max} := 2500 \text{psi}$

Thickness

$t_{60} := .09 \text{in}$

Width

$w_{60} := 0.6 \text{in}$

Diameter

$d_{60} := 3.34 \text{in}$

$$F_{gear} := \frac{t_{out}}{\left(\frac{d_{60}}{2}\right)} = 12.12 \text{lbf} \qquad A_{60} := t_{60} \cdot w_{60} = 0.054 \text{in}^2$$

$$stress_{60} := \frac{F_{gear}}{A_{60}} = 224.44 \text{psi} \qquad Fos_{60} := \frac{S_{max}}{stress_{60}} = 11.139$$

12 Tooth Gear

Diameter

$d_{12} := 0.587 \text{in}$

Thickness

$t_{12} := .08 \text{in}$

Width

$w_{12} := 0.6 \text{in}$

$$A_{12} := t_{12} \cdot w_{12} = 0.048 \text{in}^2$$

$$stress_{12} := \frac{F_{gear}}{A_{12}} = 252.495 \text{psi} \qquad Fos_{12} := \frac{S_{max}}{stress_{12}} = 9.901$$

Figure 3.13: The above figure shows the process for analyzing our gears.

We calculated the total power and amperage requirements of all the motors on our robot. This was done using addition. We added the amperage draw from the four-bar motor with the amperage draw from each wheel driving motor to find the total amperage draw. It is important to remember that there are two motors driving the wheels, so the amperage drawn by the wheel motor needs to be multiplied by two. This also applies to the power required by the motors driving the wheels, which added to the power required by the four-bar motor, will give us total power.

Total Power and Current

The Wheel Drive Amperage and Power Requirments are as follows
Important Note: This is for one motor, not both

$$A_{drive} := .3A$$

$$P_{drive} := 3.6W$$

The Four-Bar Amperage and Power Requirments are as follows

$$A_{lift} := 0.658A$$

$$P_{lift} := 7.898W$$

The Total Amperage and Power Requirments are as follows

$$A_T := A_{lift} + \left(2 \cdot A_{drive}\right) = 1.258\,A$$

$$P_T := 2P_{drive} + P_{lift} = 15.098\,W$$

Figure 3.14: This shows our calculations for total power and current between all motors.

IV. Results

Final Linkage Placement

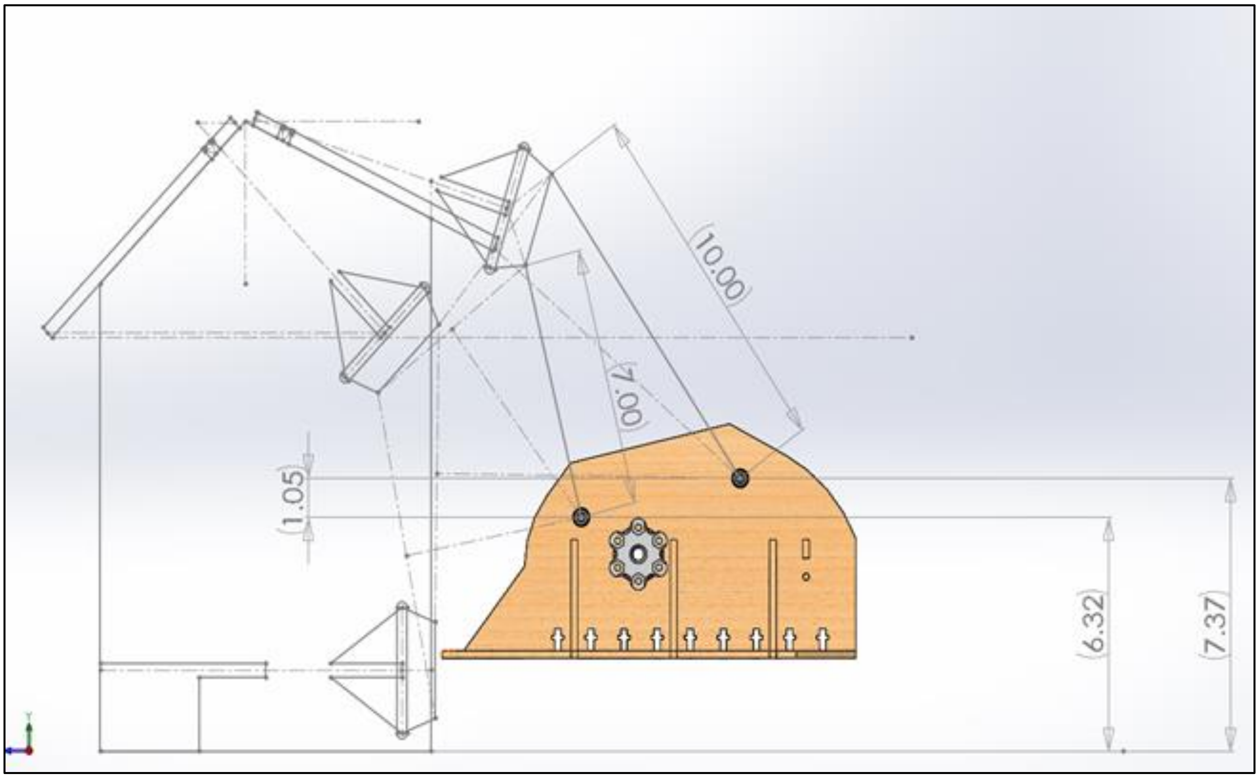


Figure 4.1: This shows our final placement and lengths for the four-bar links.

From our SolidWorks sketch, it can be seen that the link lengths we need are 10 inches and 7 inches. This will create a four-bar that reaches our desired positions, and this information allows us to find the positions relative to our base bot for a successful four-bar. Taking this into account, we were able to finish the SolidWorks model for our robot.

Final CAD Model

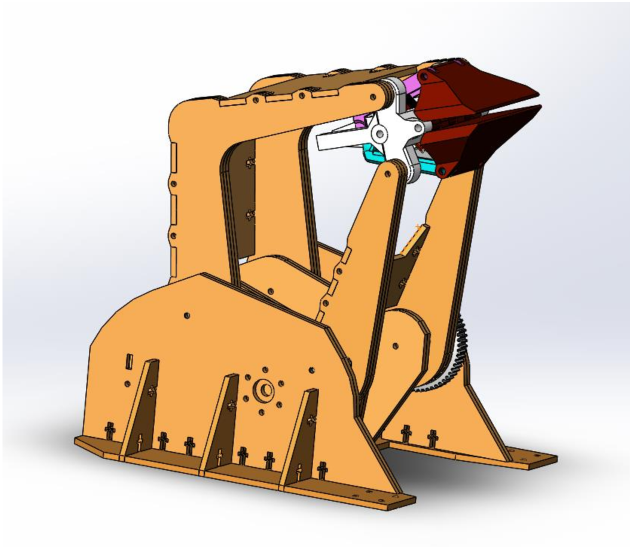
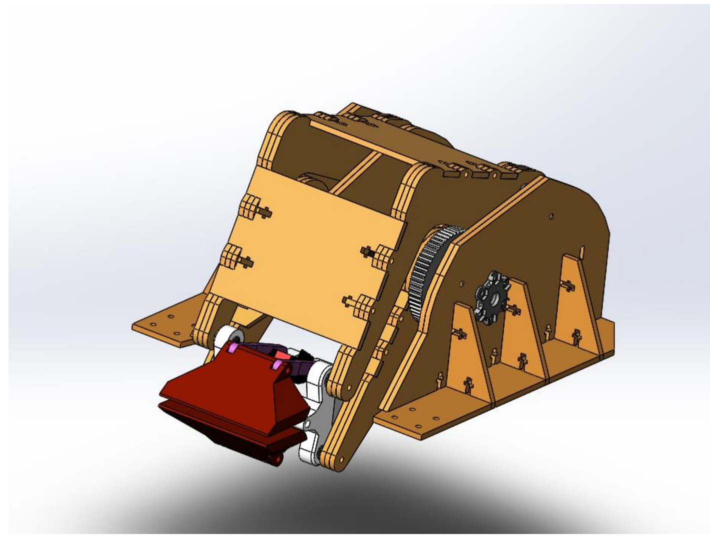
 

Figure 4.2: This is our final solid works model for our robot

Using SolidWorks, we have determined the maximum velocity in inches per second of the coupler's points. Point A reaches a maximum linear velocity of 2 inches per second, point B reaches a maximum velocity of 1.9 inches per second, and the end effector of the coupler reaches a maximum of 1.7 inches per second. It was also discovered that with the additional weight of the plate in our gripper the maximum torque experienced by the motor was a magnitude of 20.24-inch pounds of torque. This value is displayed in our four-bar solutions above. The value is negative due to the orientation of our drawn input torque variable. It is true that the torque would need to be applied in the opposite direction for the system to exist in equilibrium.

FBD Force Analysis

Solutions:

$$\begin{pmatrix} SF_{ADx} \\ SF_{ADy} \\ SF_{ABx} \\ SF_{ABy} \\ SF_{CBx} \\ SF_{CBy} \\ SF_{CDx} \\ SF_{CDy} \\ ST_{A} \end{pmatrix} := \mathrm{Find}(F_{ADx}, F_{ADy}, F_{ABx}, F_{ABy}, F_{CBx}, F_{CBy}, F_{CDx}, F_{CDy}, T_{A})$$

$$SF_{ADx} = 4.477 \cdot lbf$$
$$SF_{ADy} = -3.059 \cdot lbf$$
$$SF_{ABx} = 4.477 \cdot lbf$$
$$SF_{ABy} = -2.809 \cdot lbf$$
$$SF_{CBx} = -4.477 \cdot lbf$$
$$SF_{CBy} = -0.211 \cdot lbf$$
$$SF_{CDx} = -4.477 \cdot lbf$$
$$SF_{CDy} = 0.591 \cdot lbf$$
$$ST_{A} = -20.241\ in \cdot lbf$$

Figure 4.3: This shows our final calculations for the FBD of the four-bar

With this information, we can calculate the robot lift current. By performing a free body diagram analysis on our four-bar we were able to calculate your needed output torque (See Figure 4.3). Taking in the stall torque provided by the motor, we were able to calculate the percent torque we needed to run the motor at. Our calculated value was 7.62%, which we then used to find the total amperage drawn by this motor. using the method discussed in the analysis, we calculated an amperage draw of 0.658A. Multiplying this by the 12 volts the motor is run at gives as a power requirement of 7.898W (See Figure 3.9).

We then calculated our four-bar lifting speed. By running our four-bar motor at 7.5% of stall torque, we calculated a necessary gear ratio of 25:1. This is geared to torque to allow us to have more precision over our coupler link. We found that at 7.5% torque, our motor will be spinning at 185 RPM. Taking into consideration our 25:1 gear ratio, the four-bar output speed will be 7.4 RPM. Our desired range of freedom for the lift is 130 degrees, which means it will take our four-bar 2.93 seconds to move the full 130 degrees (See Figure 3.10).

We then completed the analysis on the motors powering our wheels. This information is important to understand the movement of our base robot. Our calculated linear speed is 250 inches a minute. Each motor will require a power of 3.6W, which is important because we must make sure we supply enough power to drive the chassis (See Figure 3.11). We then had to make sure our motor would not stall upon collison. Solving the equation for Tstall gives us a value of

31.9 in-lbf. This means that the wheel can exert a torque of 31.9 before stalling the motor. Taking the equation $\Sigma M = 0 = T - (r * Ff)$, and plugging in our known values leaves us with $\Sigma M = 0 = 31.9 - (.95 * \mu * 8.75)$. From this we calculate μ to equal 3.83. This means for friction to produce enough force to stall the motor, the coefficient of friction of the surface of the field must be 3.83. Since this is not the case for our field, it is safe to assume that the motor will slip rather than stall (See Figure 3.12). Considering all the motors, we can find the total amperage and power requirements for our robot. Our total power requirement is 15.098W and our total amperage requirement is 1.258A (See Figure 3.14).

In order to ensure our gears can handle the force, we completed the equations to find our Factor of Safety. When completing the equations to find the shear stress acting on our gear teeth, we found that for the 60-tooth gear, our shear stress will be 224.44 psi, resulting in a factor of safety of 11.139. The 12-tooth gear, which has a smaller cross-sectional area has a shear stress of 252.495 psi, resulting in a factor of safety of 9.901. From this information, we know that the gears are designed in such a way where the breaking of teeth is very unlikely, however, if it does occur the 12-tooth gear will fail before the 60-tooth gear (See Figure 3.13).

V. Discussion

The link lengths that allowed us to hit our desired angles of 45%, 27%, and 0% located two inches above the ground worked out to be 7 and 10 inches. These lengths made it easy to design the links, however another problem occurred. Due to their necessary location on the base bot, the two links would collide with each other when trying to pick up the plate from the starting location. This problem was easily solved by creating a curve in our 10-inch link which allowed for the desired positions to be reached without collision. Although we could change the location of the links on the base bot by changing the lengths of the links, keeping them longer allows for our base bot to be small and fit under the solar panel roof to perform the whole operation in one run, which was one of our goals. Using perpendicular bisectors to find the location of the links is the most efficient way to find out where our connection points need to be and using SolidWorks to perform this process was very efficient as it allowed for us to change variables and angles to optimize our four-bar.

With the information on our four-bar analysis, we can properly calculate the necessary gear ratio to functionally manipulate the plate to its proper orientations on the field. With this in mind we can calculate a safe operating torque to run the motor at. From here we can understand how much power the system will output and how much amperage will be drawn from the battery under constant operation. When we calculated a gear ratio of 25 to 1, we decided to perform this task in two stages with a 5:1 ratio in each stage. This would give us an overall gear ratio of 25:1 without having a large size difference between our gears. This large gear ratio gave us good control over the coupler as it allowed for an output speed of 7.4 rpm, which means it would take us about 3 seconds to move 130 degrees. This gives us plenty of precision to reach our needed angles of 27 and 45 degrees. We can lower the output speed by increasing the gear ratio. This can be done by lowering the percent stall torque we are using, but the lower the stall torque is, the less efficient the motor gets. We decided that it was best to run the motor at around 7.5% stall torque because it remains efficient and sets the output speed to a controllable value.

Our team did not have much control over the design of the drive transmission as the gear ratio and motors where provided for us. A linear speed of 250 inches per minute is a good speed for our robot because it was slow enough for our line tracker to pick up the line, but quick enough where we didn't have to worry about not finishing the challenge in under ten minutes. A power requirement of 3.6W per motor means we will not struggle providing enough power to the drive motors allowing our robot to maneuver successfully. Our calculations told us that in the case of a collision, our wheels will slip rather than stalling the motor. This is the ideal situation because it prevents any damage from occurring to the motor. Ensuring that the wheels would slip upon collision meant that we did not have to redesign them, and we can use the ones supplied. Changing the diameter of wheel will increase the force of friction. This value can also be increased by using a material with a higher coefficient of friction. Doing the opposite of these actions would decrease the force of friction. This was not necessary for us, as the motors combined with the gear ratio produce an adequate amount of torque.

A power output of 7.89 W means that we can supply enough power to the system with the provided power supply. Since the motors run at 12 volts, the only way to change the power required is to change the amperage draw of the motor, which has a linear relationship with percent torque used. The overall power requirement for the robot is 15.1W and the amperage required is 1.23A. These values are good because they are easily supplied by the 8 AA batteries, we use to power our robot.

Using the Commonwealth Robotics software to generate our gears was a very efficient way to do it. It creates a CAD model of our two gears which messed properly and worked very well for our robot. Our factor of safety calculations suggest that each gear has a large factor of safety which pretty much guarantees there will not be any teeth failures. If there is one, it will occur on the 12-tooth gear, which is much smaller and the ideal gear to break because it was cheaper and a quicker print on the 3d printer. If we needed to decrease the shear stress, we can increase the cross-sectional area of the gear tooth. We can do this by making the gear thicker or changing the width of the gear teeth.

VI. Conclusion

Overall, we were happy with our robot. Although it could not complete the entire challenge, it was successful in many areas. The design of our robot meets all of our needs and wants. The gripper was successful in holding both the plastic and metal plate, and the four-bar could reach all the angles we need it to. We successfully made a PID that worked for all three Pololu motors and reached the desired setpoint at a quick pace. We had a successful line follower when driving backward and could successfully communicate with the field controller. Our code allowed us to pick up a plate from the staging area and move to both position 1 and 2 for both sides of the field. Although not perfectly consistent, it could place the plate for both angles, wait for the field controller's approval, and release. Further work for us includes programming the robot to back up from one position and move to pick up the plate at the other position. It should then move the plate to the staging area. We would like the robot to be able to perform this on both sides of the field while being more consistent. if this was successful, we would attempt to do both sides of the field in one run. For future projects, it would be wise to start programming earlier, allowing for more time to test. This experience taught us a lot on the design, manufacture, and programming of a successful robot.

VII. Comments

Although the final project did not go as smoothly as we would have liked, there were certain aspects of the final project that we enjoyed. Being provided with a drive train and line follower was very helpful and saved us time when designing the robot. We also appreciate the way the labs were set up. It was very helpful having the labs build on each other, with each one focusing on a major aspect of the final project. There are also a few things we wish worked better. Our team and many other teams seemed to have a hard time coding and operating the servo to control the gripper. A short lecture or lab section on how to set up the code for this would have been helpful and saved us a few hours of coding. Furthermore, the interaction between the controller and the code could have been clearer. It was hard to understand exactly how it worked and getting it to properly work with our robot was a struggle.

VIII. Bibliography

Bertozzi, N., & Miller, B. (n.d.). *RBE 2001 C19*. Lecture presented at WPI Robotics Program in Massachusetts, Worcester.

[1] Shear stress. (2019, February 11). Retrieved from https://en.wikipedia.org/wiki/Shear_stress

[2] CIM Motor Curve | scatter chart made by Googolydox | plotly. (n.d.). Retrieved from https://plot.ly/~googolydox/27/cim-motor-curve.embed

IX. Appendices

Motor Curve

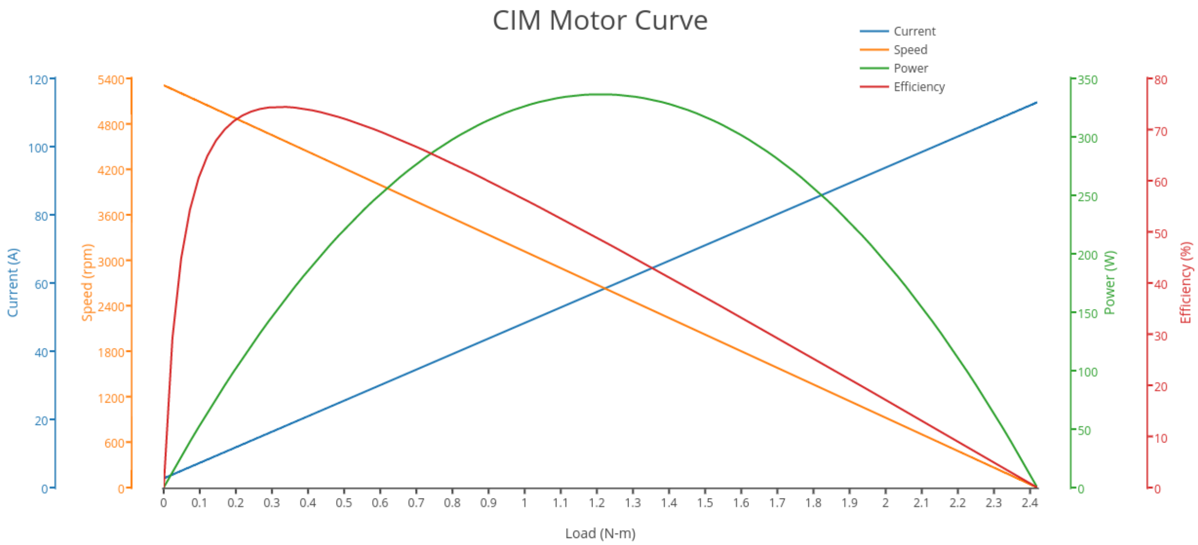
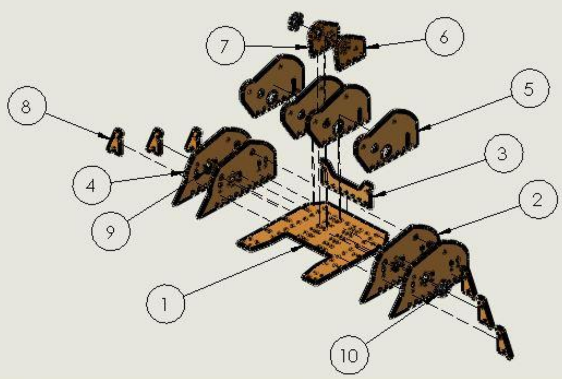


Figure 9.1: This represents an example motor curve. This is not a curve for our motors but represents a general representation of the relationships between Current, Speed, Power, and Efficiency. [2]

Bill of Materials for the Base Assembly

ITEM NO.	PART NAME	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	BASE PLATE	1	$0.93	$0.93	PLYWOOD	0.44	0.44
2	SIDE UPRIGHTS	2	$0.44	$0.88	PLYWOOD	0.208	0.42
3	BACK PANEL SUPPORT	1	$0.13	$0.13	PLYWOOD	0.06	0.06
4	SIDE UPRIGHTS WITH BEARING INSERT	2	$0.44	$0.88	PLYWOOD	0.21	0.42
5	INNER SIDE UPRIGHTS	4	$0.34	$1.36	PLYWOOD	0.16	0.64
6	INNER MOTOR MOUNT UPRIGHTS	1	$0.11	$0.11	PLYWOOD	0.05	0.05
7	INNER MOTOR MOUNT UPRIGHTS FOR BEARING	1	$0.11	$0.11	PLYWOOD	0.05	0.05
8	OUTTER ANGLE SUPPORT	6	$0.02	$0.12	PLYWOOD	0.015	0.09
9	STEEL BEARING	4	$0.53	$2.12	ALLOY STEEL	0.01	0.04
10	BEARING INSERT	3	$0.23	$0.69	ABS	0.01	0.03
			ASSEMBLY COST $3.28				ASSEMBLY WEIGHT 2.24 (LBS)

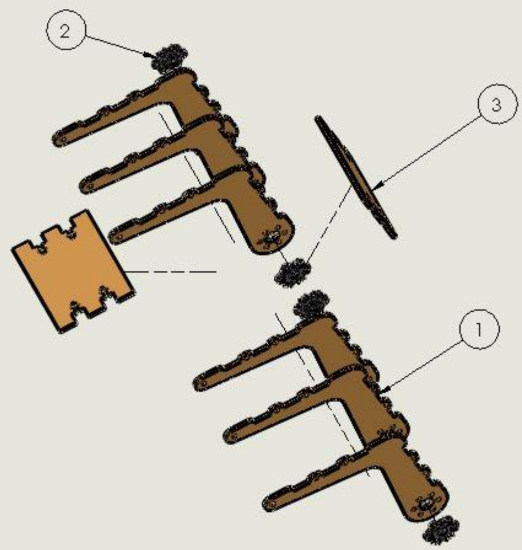


TEAM 13	SCALE: 1:12	BASE ASSEMBLY	3/01/2019

Figure 9.2: This figure is the Bill of Material for the base structure of our robot

Bill of Materials for the Driven Truss Assembly

ITEM NO.	PART NAME	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	DRIVEN LINK	6	$0.19	$1.14	PLYWOOD	0.09	0.54
2	BEARING INSERT	4	$0.23	$0.92	ABS	0.01	0.04
3	TRUSS SPAR	2	$0.21	$0.42	PLYWOOD	0.10	0.20
				ASSEMBLY COST $2.48			ASSEMBLY WEIGHT 0.78 (LBS)

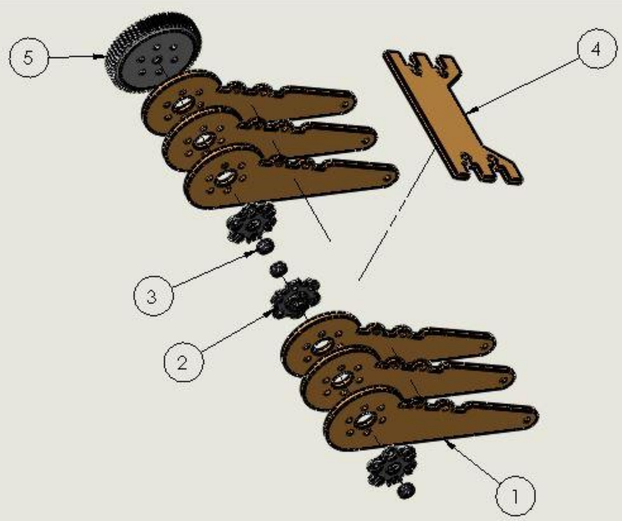


TEAM 13	SCALE: 1:6	DRIVEN TRUSS ASSEMBLY	3/01/2019

Figure 9.3: This figure is the Bill of Material for the driven truss assembly of our robot

Bill of Materials for the Driving Truss Assembly

ITEM NO.	PART NAME	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	DRIVING LINK	6	$0.13	$0.78	PLYWOOD	0.06	0.36
2	BEARING INSERT	4	$0.23	$0.92	ABS	0.01	0.04
3	STEEL BEARING	4	$0.53	$2.12	ALLOY STEEL	0.01	0.04
4	TRUSS SPAR SMALL	1	$0.11	$0.11	PLYWOOD	0.05	0.05
5	60 TOOTH GEAR	1	$4.08	$4.08	ABS	0.18	0.18
				ASSEMBLY COST $8.01			ASSEMBLY WEIGHT 0.67 (LBS)

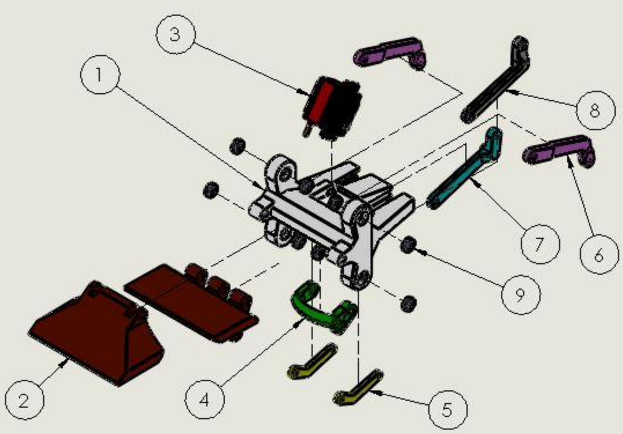


TEAM 13	SCALE: 1:4	DRIVING TRUSS ASSEMBLY	3/01/2019

Figure 9.4: This figure is the Bill of Material for the driving truss assembly of our robot

Bill of Materials for the Gripper Assembly

ITEM NO.	PART NAME	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. MASS (LBS)
1	GRIPPER GROUND LINK	1	$8.39	$8.39	ABS	0.37	0.37
2	CLAW	2	$6.80	$13.60	ABS	0.30	0.60
3	SERVO PDI 6621MG	1	$12.23	$12.23	MIXED MATERIAL	0.14	0.14
4	CLAW DRIVING LINK	1	$0.23	$0.23	ABS	0.01	0.01
5	1.9IN LINK	2	$0.11	$0.22	ABS	0.00	0.00
6	3.25 INCH LINK	2	$0.45	$0.90	ABS	0.02	0.04
7	3.25 INCH BOTTOM LINK	1	$0.45	$0.45	ABS	0.02	0.02
8	3.25 MIRRORED BOTTOM LINK	1	$0.45	$0.45	ABS	0.02	0.02
9	STEEL BEARING	8	$0.53	$4.24	ALLOY STEEL	0.01	0.08
				ASSEMBLY COST $40.71			ASSEMBLY WEIGHT 1.28 (LBS)

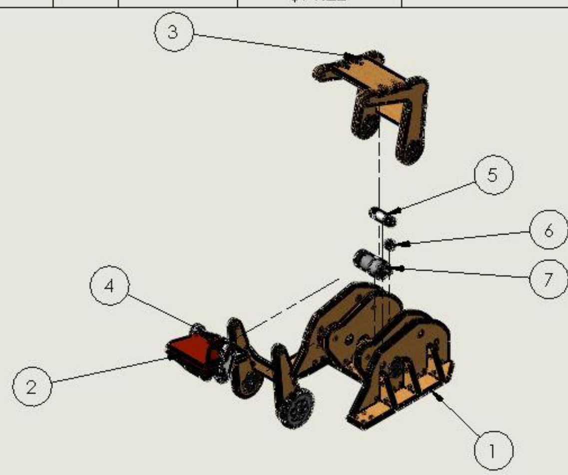


TEAM 13	SCALE: 1:4	GRIPPER ASSEMBLY	3/01/2019

Figure 9.6: This figure is the Bill of Material for the gripper assembly of our robot

Bill of Materials for the Full Robot Assembly

ITEM NO.	PART NAME	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	BASE ASSEMBLY	1	$3.28	$3.28	MIXED MATERIAL	2.24	2.24
2	GRIPPER ASSEMBLY	1	$40.71	$40.71	MIXED MATERIAL	1.28	1.28
3	DRIVEN TRUSS ASSEMBLY	1	$2.48	$2.48	MIXED MATERIAL	0.78	0.78
4	DRIVING TRUSS ASSEMBLY	1	$8.01	$8.01	MIXED MATERIAL	0.67	0.67
5	12 TOOTH TRANSITION STAGE	1	$0.68	$0.68	ABS	0.03	0.03
6	12 TOOTH GEAR	1	$0.11	$0.11	ABS	0.00	0.00
7	Gearmotor_37D_100.stp	1	$35.95	$35.95	MIXED MATERIAL	.50	0.50
				ASSEMBLY COST $91.22			ASSEMBLY WEIGHT 5.5(LBS)



TEAM 13	SCALE: 1:10	ROBOT ASSEMBLY	3/01/2019

Figure 9.5: This figure is the Bill of Material for the full robot assembly

Team Contribution

	RBE 2001 Lab Work	Final Project Work
Connor Craigie	33.3%	33.3%
Nathan Stallings	33.3%	33.3%
Victoria Thornton	33.3%	33.3%

We were all happy with the work distribution and each member played a crucial role in successfully completing RBE 2001.