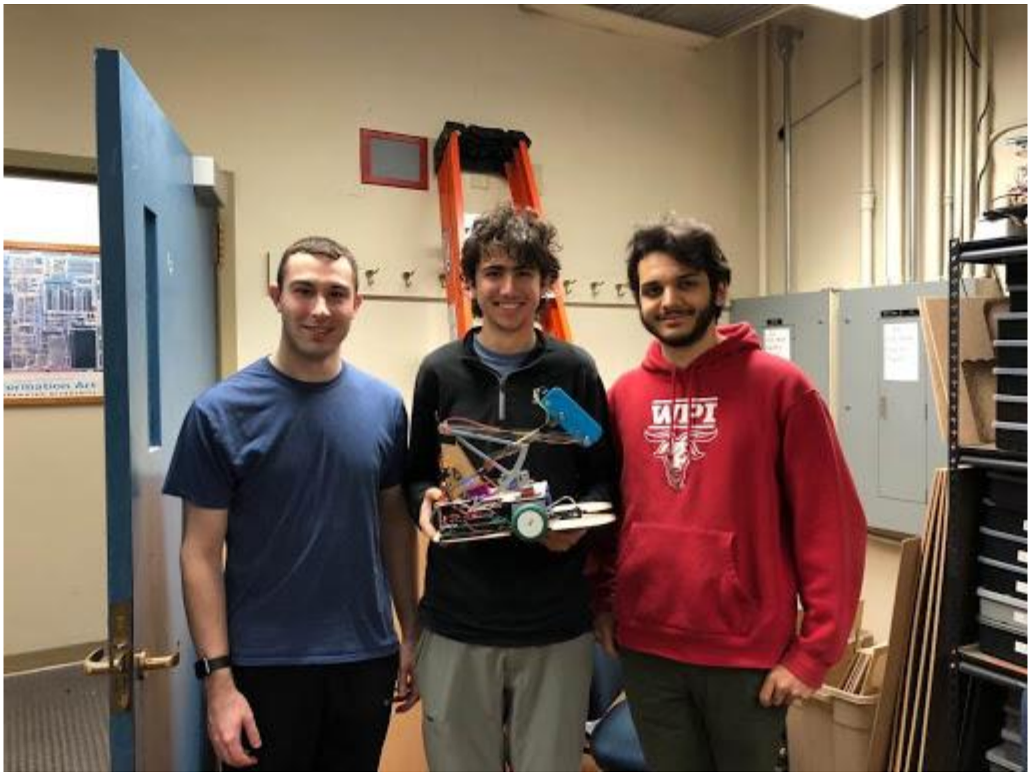
RBE 2001 C'18

Final Project Report



"Robert"

Team 18

Nicholas Johnson, Michael Abadjiev, Peter Guglielmino

Abstract

This project was the capstone project for RBE 2001: Unified Robotics. The primary goal of this project was to apply the knowledge acquired in class to the real world. Our knowledge of sensors, actuators, and software design were all necessary to design, build and program a robot to accomplish the tasks set forth in the problem statement. We were able to successfully make a robot that acted as a service robot for a simulated nuclear reactor. While not perfect, the resulting robot was robust and consistent, and was able to remove fuel rods from the nuclear reactors and replace them with new ones, all while communicating with a reactor control system via Bluetooth to determine its' next steps.

Table of Contents

List of Figures

Intro

Background

For this project, an Arduino Mega with a WPI robotics shield, along with motors and sensors from the RBE 2001 lab kits were used. Three DC motors and a 90g continuous rotation servo were used for actuators. A custom line follower circuit, motor rotation encoders, as well as multiple mechanical switches make up the sensory equipment on the robot.

Goals

The goal of this project is to design, build and program a robot to service a simulated nuclear reactor. The robot should be able to communicate with the reactor control system (RCS) via Bluetooth to determine which reactor cores require refueling, as well as which storage and supply tubes are empty or full. It will be able to remove a fuel rod from the reactor, place it in an empty storage tube, retrieve a new fuel rod from a supply tube, and finally place the new rod in the reactor. It will also communicate with the RCS during these operations to provide a heartbeat message, as well as a radiation warning while transporting fuel rods and status messages upon completion of a task.

Methodology

Mechanical Design

Driving base

The driving base of the robot was designed with simplicity and ease of use in mind. A simple rectangular base was designed with an alignment notch in the front of the robot for alignment with reactor tubes and the alignment posts at each storage/supply location. Two motors were positioned centrally on the robot, with a direct drive from the motor gearbox output to the wheels. This, coupled with a single rear omni wheel produced a base that could easily be driven around. The motor positioning also placed the turning center directly in the center of the robot for increased maneuverability.

Four-bar lifting mechanism

The main lifting mechanism on the robot was a four-bar lift located on the second floor. The four-bar was designed using a SOLIDWORKS sketch that had the required positions of the follower link determined.

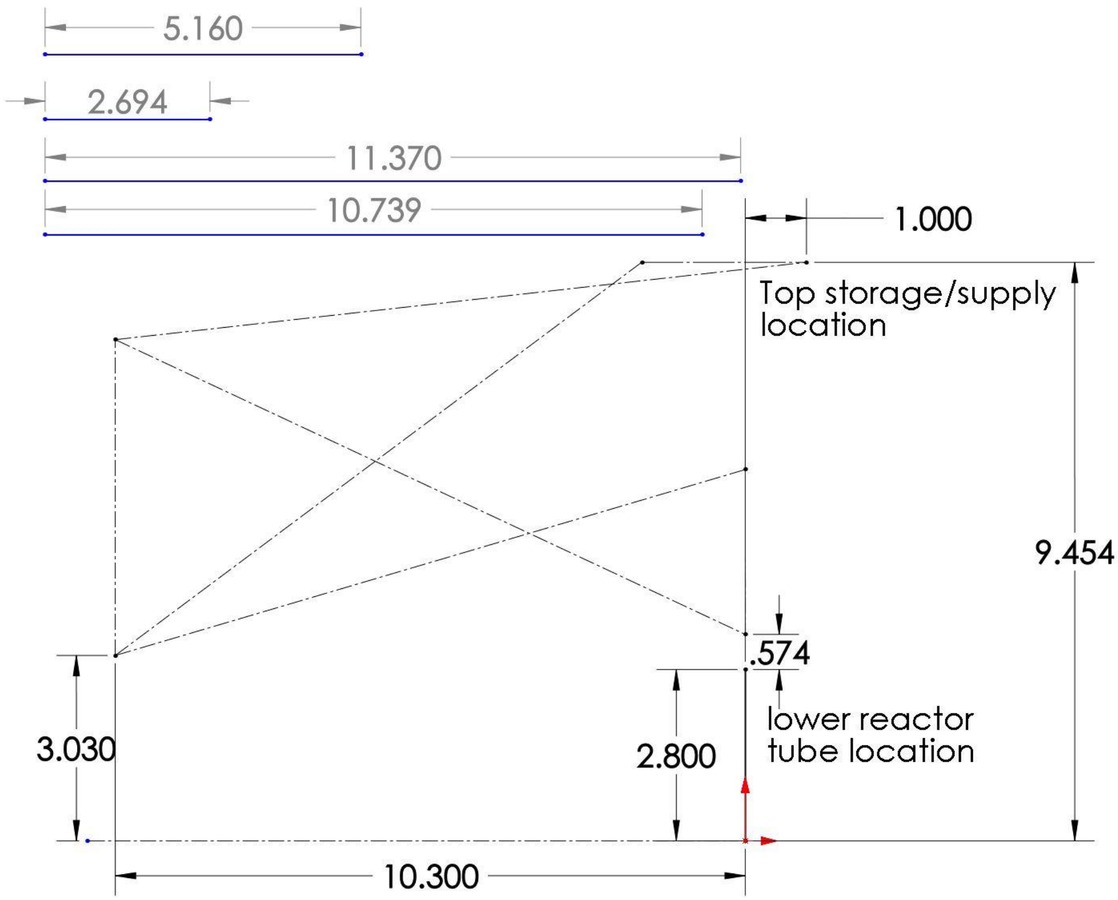


Figure 1: Four-bar Dimension Analysis

As can be seen in Figure 1, certain values were predetermined, such as the height of the ground link, the distance to the alignment notch, and the distances from the reactor and storage/supply tubes. For the ground link positions, the height was determined from the height of second floor from the ground, and the horizontal distance was maximized while keeping the overall robot length reasonable. The distances to the reactor and storage/supply tubes were determined based on the intake design to ensure consistent intake/outtake of fuel rods. The lift was driven by a DC motor with a 1:50 gearbox, and then a 12:300 gear reduction, allowing for smooth, easily controllable positioning.

Intake mechanism

The intake mechanism was designed to allow the robot to draw the fuel rods smoothly out of the reactors, and then dispense rods equally precisely. In order to do this, parallel rollers driving rubber bands were used. These rollers were driven by a 9mg continuous rotation servo directly attached to them. A switch mounted on the end of intake provided information on when a fuel rod had been fully collected or deposited. A problem with using rubber bands is that they would ride up against the sides of the roller if it was a U-shape. This was solved by instead creating a convex roller so that the rubber band would settle centered on the roller due to the tension that is created, as shown in Figure 2. This meant, however, that external alignment would be needed as the fuel rod could not sit in the groove of the roller.

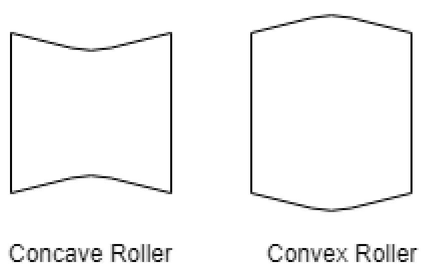


Figure 2: Roller Diagram

Electrical Design

Several sensors were used to provide input to the robot. These included limit switches, infrared light sensors, and wheel encoders. The front limit switch was used to detect when the robot arrived at a rod location, while the up and down limit switches detected when the arm reached the up and down limits of its travel. The intake limit switch informed the robot when it was holding a fuel rod as well as when it had successfully picked one up. Three infrared light sensors were used together to create a line tracker and detector. These allowed the robot to follow the lines and know where on the field it was by keeping track of how many lines it had crossed.

The motors selected for the drive and arm were the Pololu 37Dx70L mm with 64 CPR Encoder motors with a 50:1 gearbox. These were selected as they were readily available at the campus bookstore and met our power requirements for the drive system and lift arm. In addition, they also have built in encoders for feedback control. To drive the Pololu motors, the Pololu MC33926 motor controllers were used. These allow easy use of the Pololu motors while keeping the complexity of wiring a H-bridge off the robot.

Software Design

To run the robot on an Arduino while completing the complex tasks required a state machine as the basis of the software design. One of the most important items for the robot to stay up to date on were the incoming Bluetooth packets being sent by the RCS. Whenever a new loop was entered the robot checked for any new packets to handle. If a new packet was found, then the robot looked at the information. If it was data about fuel rod storage or fuel rod supply that data was then stored for later use. If the data was a stop or resume command, then the robot complied with that command. After handling incoming communications, the robot then focused on outgoing communications. If any data such as a heartbeat packet, radiation alert, or status message needed to be sent then that was taken care of in this section of the program.

The next section of the program was determining the goals to execute. The program looks at the status of the reactor tubes and depending on if they needed to be emptied of a spent rod or filled with a new rod it would look at the state of the supply and storage tubes to see where it could pick-up/drop-off the rods and then plot a course. As the code cannot be blocking, which would interfere with the Bluetooth updates and sensor updates, it was built as a state machine with each loop through checking on the status of the move and either continuing if it hadn't finished or executing a different move if completed.

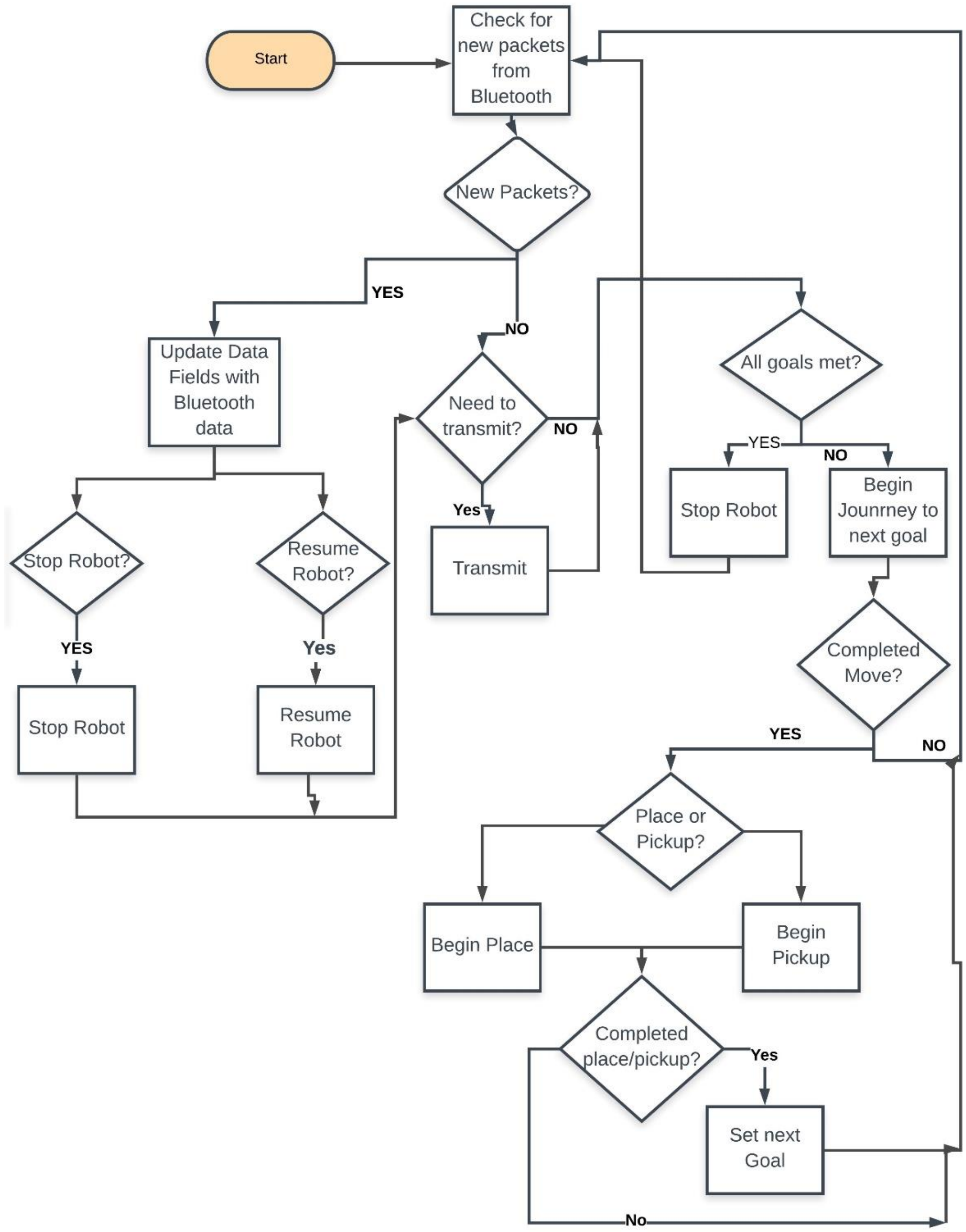


Figure 3: Programming Flowchart

Analysis

Drive

The first step in the analysis was creating the driving base, the Pololu 50:1 motors were used, with the following motor data at 7.4 Volts.

Speed (RPM)	Torque (N-m)	Torque (in-lbf)	Current (A)	Pout (W)
0	0.74	6.56	3.10	0.00
12	0.67	5.91	2.82	0.87
25	0.59	5.25	2.54	1.54
37	0.52	4.59	2.26	2.02
50	0.44	3.94	1.98	2.31
62	0.37	3.28	1.71	2.41
74	0.30	2.63	1.43	2.31
87	0.22	1.97	1.15	2.02
99	0.15	1.31	0.87	1.54
112	0.07	0.66	0.59	0.87
124	0.00	0.00	0.31	0.00

Figure 4: Pololu Motor Data

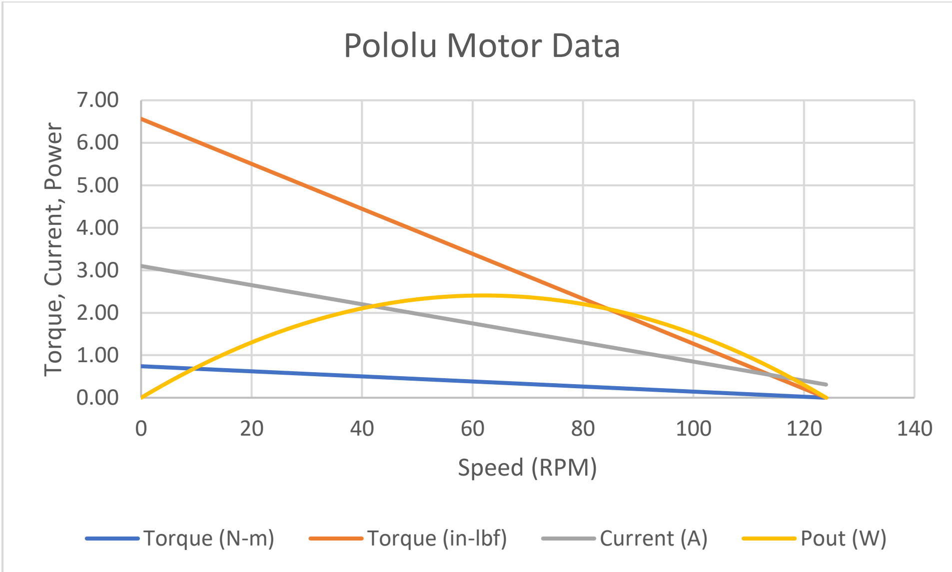


Figure 5: Pololu Motor Curve

From the data table in Figure 4 the free running speed is 120 revolutions per minute, which when using a 2.75-inch wheel creates a top speed of 17.85 inches per second, and an operating current of 0.31 Amps.

$$124\frac{rev}{minute} * \frac{1\ minute}{60\ seconds} * \frac{2\pi\ rad}{rev} = 12.99\frac{rad}{sec}$$

$$v = r * \omega$$

$$v = \frac{2.75}{2} inches * 12.99\frac{rad}{sec} = 17.85\frac{inches}{sec}$$

For purposes of current analysis, the maximum current draw from the motor when operating is needed, which happens when the robot is pushing against a wall. Figure 5 is a free body diagram of a wheel.

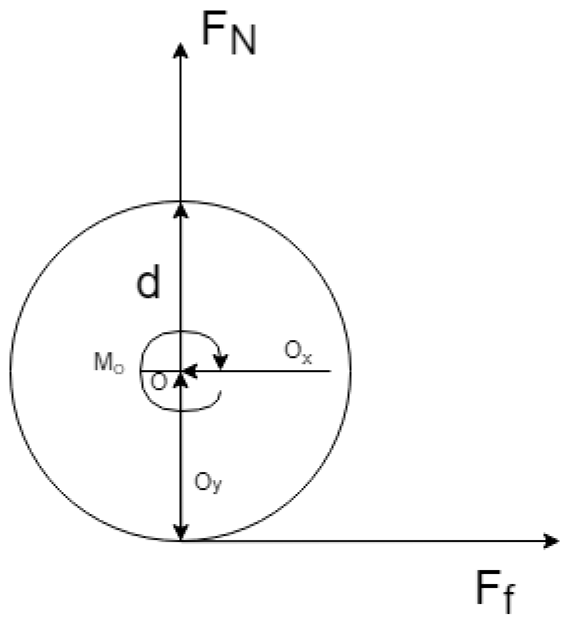


Figure 5: FBD of Wheel

With a robot weight of 8.3 pounds and coefficient of friction of 1, the maximum torque that can be exerted by the 2.75-inch wheel is 11.4 inch-pounds before slipping.

$$F_f = 8.3 * 1 = 8.3 \; pounds$$

$$M_O = 8.3 * \frac{2.75}{2} = 11.4125 \; inch \; pounds$$

Looking at the motor data, the maximum output torque that can be generated is 6.56 inch-pounds, which means that the motor will stall before the wheels slip, and the stall current of the motor is 3.1 Amps.

Sensors

The only requirement for the electrical design was wiring up the limit switches, as the other components were externally manufactured. Because the switches want to detect when they are hit by a component, a pull-down resistor is used to ensure a default off state as shown in Figure 6, with a resistor value of 1k Ohm being used.

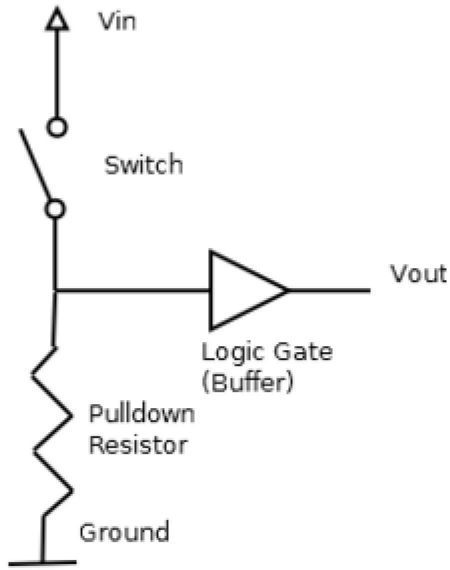
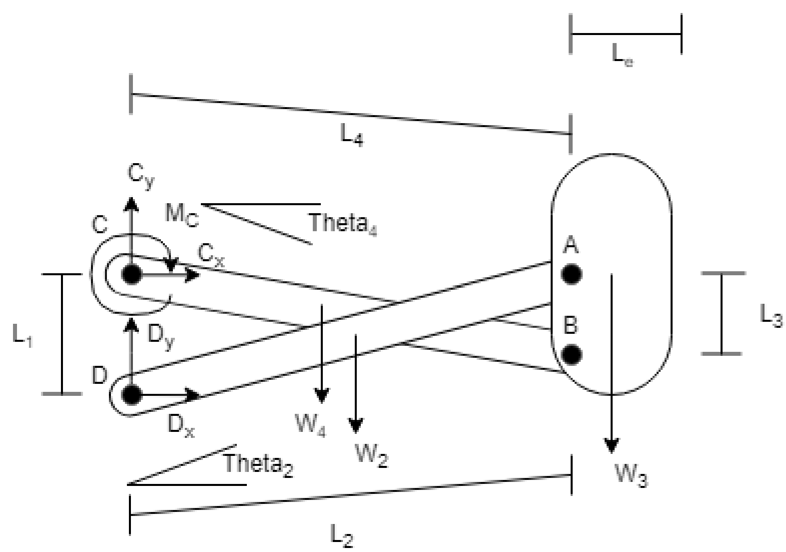


Figure 6: Pull-Down Resistor

Four-bar

A forces analysis was done on the four-bar; the following free body diagrams in figures 7, 8, and 9 were created with the corresponding equations of equilibrium.

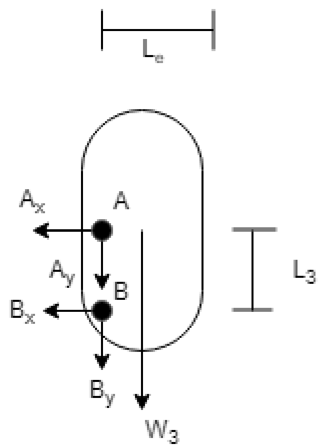


$$0 = M_C - C_x \cdot L_1 - W_3 \cdot \left(L_4 \cdot \cos(\mathrm{Theta}_4) + \frac{L_e}{2}\right) - W_2 \cdot \frac{L_2}{2} \cdot \cos(\mathrm{Theta}_2) - W_4 \cdot \frac{L_4}{2} \cdot \cos(\mathrm{Theta}_4) \qquad \Sigma M_D := 0$$

$$0 = C_x + D_x \qquad \Sigma F_x := 0$$

$$0 = C_y + D_y - W_2 - W_3 - W_4 \qquad \Sigma F_y := 0$$

Figure 7: FBD of the Link 2, 3, 4 System

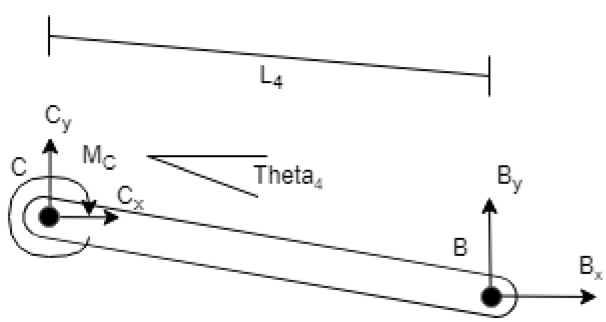


$$0 = B_x \cdot L_3 - W_3 \cdot \frac{L_e}{2} \qquad \Sigma M_A := 0$$

$$0 = A_x + B_x \qquad \Sigma F_x := 0$$

$$0 = A_y + B_y - W_3 \qquad \Sigma F_y := 0$$

Figure 8: FBD of Link 3



$0 = -W_4 \cdot \frac{L_4}{2} \cdot \cos(\text{Theta}_4) - B_x \cdot L_4 \cdot \sin(\text{Theta}_4) - B_y \cdot L_4 \cdot \cos(\text{Theta}_4) + M_C$	$\Sigma M_C := 0$
$0 = -C_x - B_x$	$\Sigma F_x := 0$
$0 = -C_y - B_y$	$\Sigma F_y := 0$

Figure 9: FBD of Link 4

Solving these equations with the calculated values gives these results:

$L_2 := 10.897\text{in}$ $\quad L_3 := 2.694\text{in}$ $\quad L_4 := 11.370\text{in}$ $\quad L_1 := 5.160\text{in}$ $\quad L_e := 2\text{in}$

$W_2 := .1\text{lbf}$ $\quad W_3 := .47\text{lbf}$ $\quad W_4 := .1\text{lbf}$

$\text{Theta}_2 := 15.1\text{deg}$ $\quad \text{Theta}_4 := 22.7\text{deg}$

Figure 10: Given Values

$SA_x = -0.17\,\text{lbf}$

$SA_y = 0.06\,\text{lbf}$

$SB_x = 0.17\,\text{lbf}$

$SB_y = 0.41\,\text{lbf}$

$SC_x = -0.17\,\text{lbf}$

$SC_y = -0.41\,\text{lbf}$

$SD_x = 0.17\,\text{lbf}$

$SD_y = 1.08\,\text{lbf}$

$\boxed{SM_C = 5.55\,\text{in}\cdot\text{lbf}}$

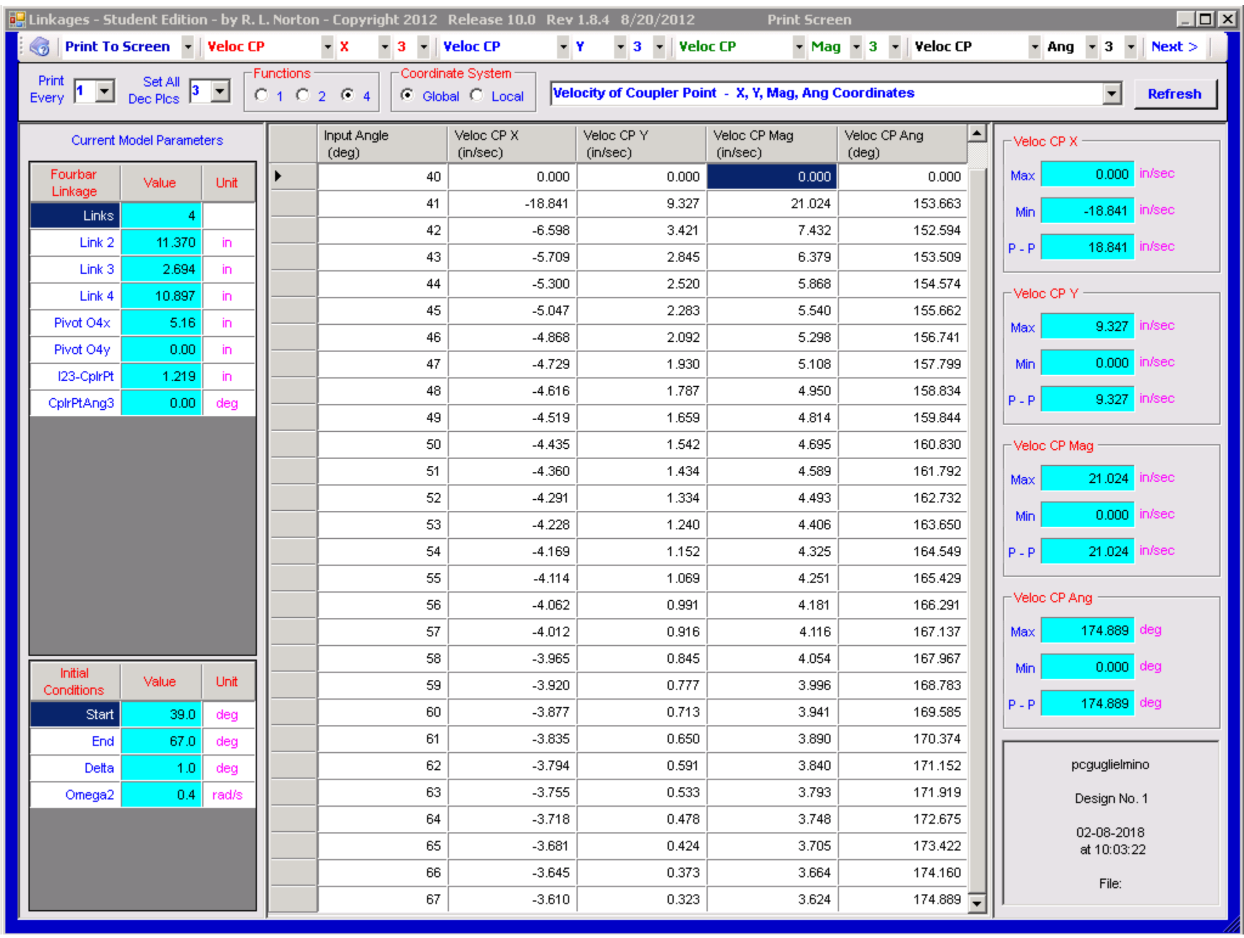
Figure 11: Solved FBD Equations

Looking at the geometry and weights of the four-bar, and assuming a 5 second lift time, the required output power of the motor is 0.08 Watts

$$P = \frac{.6\ lbs * \big(9.454 - (0.574 + 2.8)\big)inches}{5\ sec} = 0.7296\ \frac{inch - pounds}{second}$$

$$0.7296\ \frac{inch - pounds}{second} = 0.08\ Watts$$

Because of the required power, the motor can run within almost its entire operating range, so the only design consideration that needs to be made is the operating speed, and using the Norton Linkage software, an acceptable speed reduction comes from a 12:320 gear reduction.



Input Angle (deg)	Veloc CP X (in/sec)	Veloc CP Y (in/sec)	Veloc CP Mag (in/sec)	Veloc CP Ang (deg)
40	0.000	0.000	0.000	0.000
41	-18.841	9.327	21.024	153.663
42	-6.598	3.421	7.432	152.594
43	-5.709	2.845	6.379	153.509
44	-5.300	2.520	5.868	154.574
45	-5.047	2.283	5.540	155.662
46	-4.868	2.092	5.298	156.741
47	-4.729	1.930	5.108	157.799
48	-4.616	1.787	4.950	158.834
49	-4.519	1.659	4.814	159.844
50	-4.435	1.542	4.695	160.830
51	-4.360	1.434	4.589	161.792
52	-4.291	1.334	4.493	162.732
53	-4.228	1.240	4.406	163.650
54	-4.169	1.152	4.325	164.549
55	-4.114	1.069	4.251	165.429
56	-4.062	0.991	4.181	166.291
57	-4.012	0.916	4.116	167.137
58	-3.965	0.845	4.054	167.967
59	-3.920	0.777	3.996	168.783
60	-3.877	0.713	3.941	169.585
61	-3.835	0.650	3.890	170.374
62	-3.794	0.591	3.840	171.152
63	-3.755	0.533	3.793	171.919
64	-3.718	0.478	3.748	172.675
65	-3.681	0.424	3.705	173.422
66	-3.645	0.373	3.664	174.160
67	-3.610	0.323	3.624	174.889

Figure 12: Link Speed Analysis

$$\frac{N_{in}}{N_{out}} = \frac{T_{in}}{T_{out}}$$

$$\frac{12}{320} = \frac{T_{in}}{5.55\ inch\ pounds}$$

$$T_{in} = 0.21\ inch\ pounds$$

Viewing the motor data at 0.21 inch-pounds of torque, 0.3 Watts of power is output with a 0.41 Amp current requirement.

Intake

For the intake servo, a rubber band acts as a belt to drive the intake, the following figure is a FBD for the servo output roller

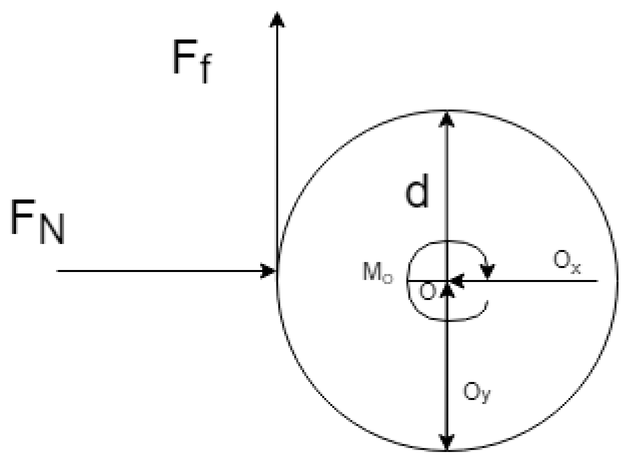


Figure 13: Servo Output FBD

The force of friction is the external torque that drives the motor performance, to find the coefficient of friction between the rubber band and the PLA plastic that makes up the roller on the servo, a test was done pulling a .45-kilogram weight along a piece of PLA, and measuring the force required to overcome the force of friction.

The measured force was 3 Newtons, which means that the force of friction acting on the object was also 3 Newtons.

$$\Sigma F = 0 \quad F_f = \mu F_N \qquad F_N = 9.8 * weight$$

$$\mu = \frac{F_f}{F_N}$$

$$\mu = \frac{3}{9.8 * .45} = .68$$

The force acting on the roller given the bands displacement was measured to be 6 Newtons, which can be used with the determined coefficient of friction and the above free body diagram to find the external moment on the servo, knowing that the diameter of the roller is .37 inches.

$$F_{f-roller} = .68 * 6 = 4.08\ N = 0.917\ lbf$$

$$M_O = 0.917 * \frac{.37}{2} = 0.34\ inch\ pounds$$

Using the moment that was found and the following motor data for the servo the servo is operating in the optimal range of its operation, with an output power of .32 Watts and a current requirement of .32 Amps.

Speed (RPM)	Torque (N-m)	Torque (in-lbf)	Current (A)	Pout (W)
0	0.18	1.56	0.65	0.00
10	0.16	1.41	0.61	0.17
20	0.14	1.25	0.56	0.30
30	0.12	1.09	0.52	0.39
40	0.11	0.94	0.48	0.44
50	0.09	0.78	0.44	0.46
60	0.07	0.63	0.39	0.44
70	0.05	0.47	0.35	0.39

80	0.04	0.31	0.31	0.30
90	0.02	0.16	0.26	0.17
100	0.00	0.00	0.22	0.00

Figure 14: Servo Motor Data

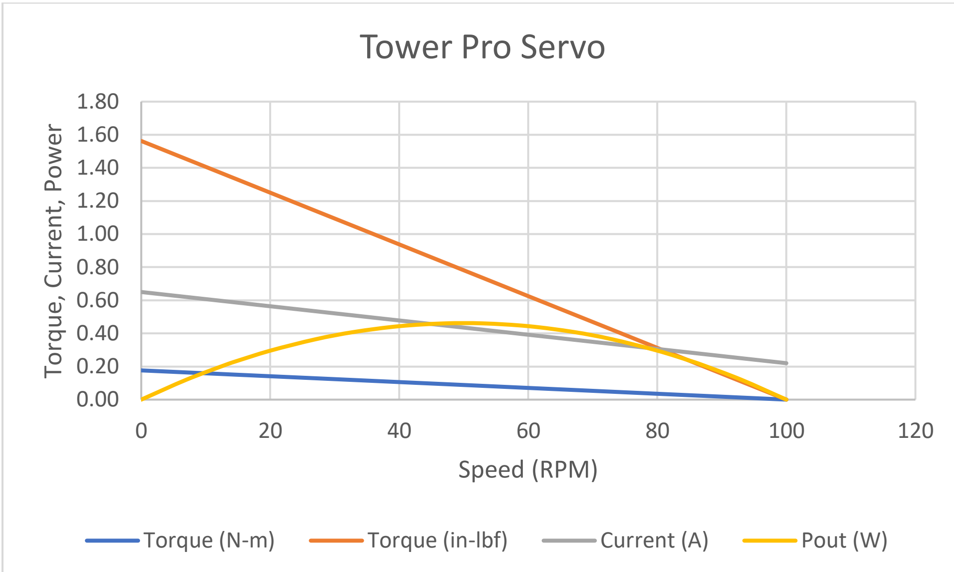


Figure 15: Servo Motor Curve

Current Analysis

The total current draw of the robot at maximum load is from the following:

· 2 Pololu motors at stall from the drive, 3.1 Amps each

· 1 Pololu motor running from the arm, this motor will not be considered at stall because of the controls put in place, 0.31 Amps

· 1 Servo at operating current, 0.32 Amps

· 4 limit switches, however they operate at almost 0 current because of the high resistance of the microcontroller

· 2 Encoders at 10 mA each, data found from the motor website

· 3 Line sensors operating at 3 mA each, data measured with multimeter

This totals to a maximum current of 6.96 Amps, which is within the Arduino shield fuse limit of 7.5 Amps

Results and Discussion

Mechanical

Drive

The driving base worked successfully; it was small enough to easily navigate the field, while being large enough to provide plenty of space to mount all of the necessary hardware. One problem encountered had to do with the variability in the field; sometimes the base would not be able to align itself with the reactors. The two reactor bases were not of equal height, and as a result the base would sometimes bump into the stand before it could reach the reactor. This was easily rectified by adjusting the height of the rear omni wheel to tilt the robot upwards.

Four-bar

The four-bar lift worked as intended, successfully moving the intake between the two desired positions. It was sturdy enough to lift the intake without breaking, and the 12:330 gear ratio allowed for very smooth, easily controllable operation. One thing that could have been improved on the four-bar was accuracy; it often slipped off to one side, causing some alignment issues with the fuel rods. This was counteracted by an intake design that allowed for misalignment, but a more accurate four-bar would have been helpful as well. This could have been accomplished by adding some sort of alignment path to the four-bar, such as a “V” shape at the bottom for it to line up with.

Intake

The final intake of the robot worked as intended, however there were still flaws in the design that needed to be fixed before the demonstration of the robot. For example, while the rollers were easily capable of grabbing the rod and pulling it, there were issues with aligning the rod into the roller, because there was a limited space that the roller would make full contact with the fuel rod. Because of this, extra material was added to the walls of the intake to aid the rod travelling into the mechanism. Small triangular aligners were also added to the sides of the intake to help the rod align properly for intake when being pulled from the tubes, because there were problems with the rod getting wedged in the wrong location because of the inaccuracies of the four-bar. The limit switch that was added to the back of the intake was very useful as it ensured that the fuel rod made it all the way into the mechanism.

Sensors

To better guarantee repeatable success for the robot, additional sensors could be used. One would be a gyroscope to assist with driving. This would allow for more accurate straight lines, as well as more accurate turning. Another useful sensor for the robot would be an ultrasonic sensor. This would be useful for determining distance to a wall to assist in the approach to a fuel rod supply or storage location. Having more sensors usually allows for more success, but with having multiple sensors combing their data, even higher success could occur with data from each sensor assisting the others to allow redundancy.

Software

The Software worked quite well, but the main improvement to be made would be coding efficiency. There is probably a better way to plan the path of the robot from one location to another without setting up a switch statement with many case statements for each path like what was done. This was not attempted for a couple of reasons. The first was to have the robot take the most efficient path taken every time. The second was to complete the path planning program in the given time frame. The simpler yet quite customizable way of having many case statements was chosen, but could have been improved with a more code efficient method which would also help with code readability.

Conclusion

This project successfully demonstrated our team's understanding of the course material. We were able to design, build, and program a robot to service a simulated nuclear reactor facility. The robot made use of a four-bar lifting mechanism, along with a parallel roller intake design, on top of a two-wheel drive base to accurately and repeatable replace fuel rods in both reactor cores. It communicated reliably via Bluetooth with the reactor control system to determine which reactors needed refueling, as well as the locations of new fuel rods and availability of storage tubes. The robot utilized a state machine to iterate through tasks until it had completed each assigned goal.

Comments

Nicholas

This robotics class was quite stressful as they will all be due to the WPI time constraints. It did help to really evaluate how to design the robot, and redesign due to unforeseen problems quite quickly. Some of the elements taught earlier in the course seemed quite rushed due to the fact that there are only so many weeks. It seemed overwhelming at first but having additional time to work on the robot in the second half of the term was invaluable.

Michael

I found this class to be quite challenging, but in a very interesting and engaging way. The project did a good job of tying most of what we were learning in class to a real world application and was an interesting design challenge. My main complaint about the project was the inconsistency between the SOLIDWORKS model of the field and the real thing, this was to be expected, however was an issue because the real field was not set up in time for much testing or measuring to be done in advance while designing.

Peter

I enjoyed this class, coming from 1001 in B-term, I thought that this class was a good extension of what was learned previously. I felt that the labs were well structured and helpful, my main complaint is that on the quizzes, since I do not have a laptop, I could not use Mathcad or the other software that it seems was expected to be able to use on the quizzes, which put me at a time disadvantage, because I had to do all the calculations by hand.

Bibliography

"Addicore SG90 9g Mini Servo." *Www.addicore.com*, www.addicore.com/Addicore-SG90-Mini-Servo-p/113.htm.

Arduino Playground - PullUpDownResistor, playground.arduino.cc/CommonTopics/PullUpDownResistor.

"Line Tracker (3-Pack)." *VEX Robotics*, 28 Feb. 2018, www.vexrobotics.com/276-2154.html.

"MathCAD 15.0." MathSoft, 2018.

Norton Associates Engineering - Linkage Design, designofmachinery.com/Linkage/.

"Online Diagram Software & Visual Solution." *Lucidchart*, 2 Mar. 2018, www.lucidchart.com/.

"Pololu Robotics and Electronics." *Pololu Robotics & Electronics*, www.pololu.com/.

Appendix

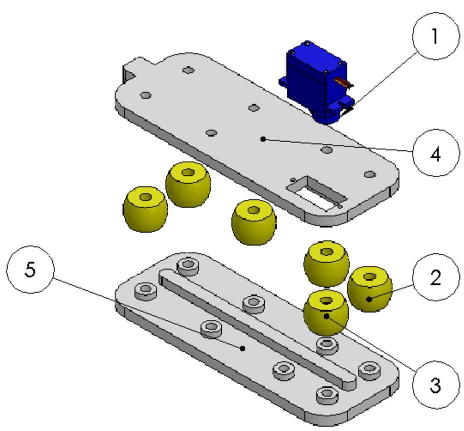
Robot Code

Attached Team18FinalRobot.pdf file

Bill of Materials

Intake

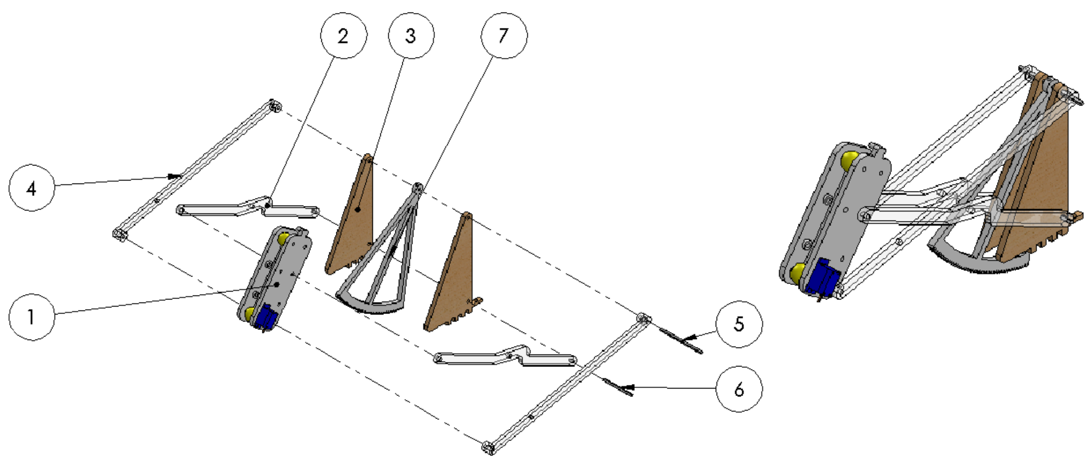
ITEM NO.	PART NAME	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	TOWER PRO 9G MICRO SERVO SG90	1	$5.00	$5.00	N/A	0.198	0.20
2	UNDRIVEN ROLLER	5	$0.10	$0.50	PLA	0.01	0.05
3	DRIVEN ROLLER	1	$0.01	$0.01	PLA	0.01	0.01
4	DRIVEN GRABBER FRAME	1	$1.00	$1.00	PLA	0.10	0.10
5	UNDRIVEN GRABBER FRAME	1	$1.00	$1.00	PLA	0.11	0.11
				ASSEMBLY COST $7.51			ASSEMBLY WEIGHT 0.47 (LBS)



TEAM 18	SCALE: 1:2	GRABBER BOM	3/1/2018

Four-Bar

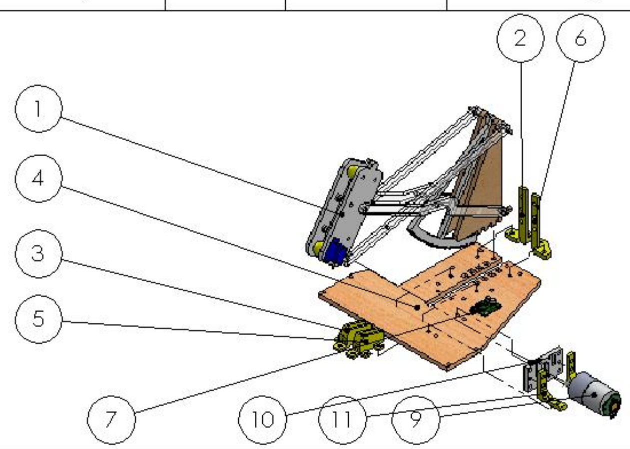
ITEM NO.	PART NUMBER	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	INTAKE FINAL	1	$7.51	$7.51	N/A	0.47	0.47
2	LINK 4	2	$0.50	$1.00	ACRYLIC	0.06	0.12
3	LINK 1	2	$0.20	$0.40	BIRCH	0.02	0.04
4	LINK 4	2	$0.50	$1.00	ACRYLIC	0.05	0.10
5	LONG SHAFT	1	$0.01	$0.01	STEEL	0.00	0.00
6	SHORT SHAFT	1	$0.01	$0.01	STEEL	0.00	0.00
7	GEAR	1	$0.45	$0.45	PLA	0.02	0.02
				ASSEMBLY COST $10.38			ASSEMBLY WEIGHT 0.75 (LBS)



TEAM 18	SCALE: 1:6	FOUR-BAR BOM	3/1/2018

Second Floor

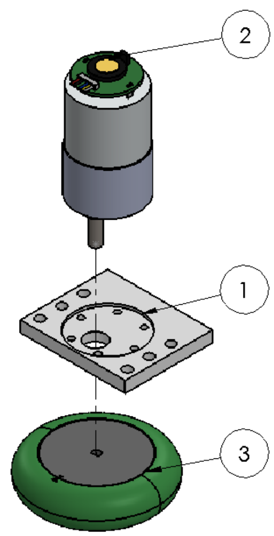
ITEM NO.	PART NUMBER	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	FOURBAR	1	$10.38	$10.38	N/A	0.75	0.75
2	RIGHT BACK SUPPORT	1	$0.20	$0.20	PLA	0.02	0.02
3	RIGHT FRONT SUPPORT	1	$0.10	$0.10	PLA	0.02	0.02
4	SECOND FLOOR	1	$0.70	$0.70	BIRCH	0.02	0.02
5	LEFT FRONT SUPPORT	1	$0.10	$0.10	PLA	0.02	0.02
6	SUPPORT BACK LEFT	1	$0.20	$0.20	PLA	0.02	0.02
7	MOTOR CONTROLLER	1	$17.95	$17.95	N/A	0.00875	0.01
8	PINION GEAR	1	$0.10	$0.10	PLA	0.00332092	0.00
9	POLOLU MOTOR	1	$39.95	$39.95	N/A	0.496	0.50
10	ADJUSTABLE MOUNT	1	$0.10	$0.10	PLA	0.03	0.03
11	MOTOR MOUNT SUPPORT	2	$0.15	$0.30	PLA	0.01	0.02
				ASSEMBLY COST $70.08			ASSEMBLY WEIGHT 1.41 (LBS)

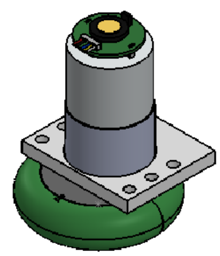


TEAM 18	SCALE: 1:6	SECOND FLOOR BOM	3/1/2018

Motor Mount

ITEM NO.	PART NUMBER	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	MOTOR MOUNT	1	$0.15	$0.15	PLA	0.05	0.05
2	POLOLU MOTOR	1	$39.95	$39.95	N/A	0.496	0.496
3	VEX WHEEL	1	$2.50	$2.50	N/A	0.110	0.11
				ASSEMBLY COST $42.6			ASSEMBLY WEIGHT 0.656 (LBS)

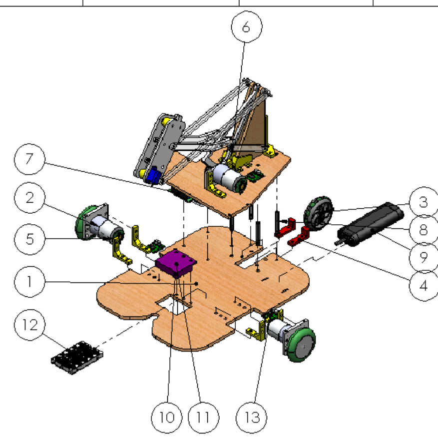




TEAM 18	SCALE: 1:2	MOTOR MOUNT	3/1/2018

Final

ITEM NO.	PART NUMBER	QTY.	UNIT COST	EXT. COST	MATERIAL	UNIT WEIGHT (LBS)	EXT. WEIGHT (LBS)
1	BASE	1	$1.15	$1.15	BIRCH	0.57	0.57
2	MOTOR MOUNT	2	$42.60	$85.20	N/A	0.656	1.31
3	OMNI WHEEL	1	$9.95	$9.95	N/A	0.095	0.10
4	REAR WHEEL MOUNT	2	$0.05	$0.10	PLA	0.01	0.02
5	MOTOR MOUNT SUPPORT	4	$0.15	$0.60	PLA	0.01	0.04
6	SECOND FLOOR	1	$70.08	$70.08	N/A	1.41	1.41
7	ARDUINO	1	$50.00	$50.00	N/A	1.23	1.23
8	STANDOFF	4	$0.01	$0.04	ALUMINUM	0.00	0.00
9	BATTERY	1	$11.99	$11.99	N/A	0.75	0.75
10	LINE TRACKER	1	$39.99	$39.99	N/A	0.04	0.04
11	LINE TRACKER MOUNT	1	$0.10	$0.10	PLA	0.08	0.08
12	BREAD BOARD	1	$2.50	$2.50	N/A	0.2	0.20
13	MOTOR CONTROLLER	2	$17.95	$35.90	N/A	0.00875	0.02
				ASSEMBLY COST $307.6			ASSEMBLY WEIGHT 5.77 (LBS)



TEAM 18	FINAL BOM	SCALE: 1:6	3/1/2018

Contribution

Peter Guglielmino 33%

Michael Abadjiev 33%

Nicholas Johnson 33%