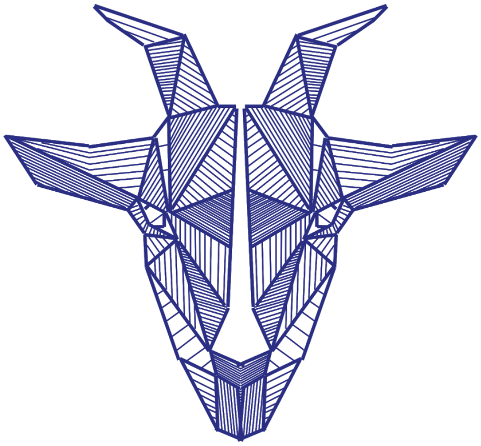
Pluto



An Extensive Analysis

Date Submitted: 03/04/2016

Design Team 8
Section C02
RBE 2001 Final Project Robot
Worcester Polytechnic Institute

Member	Signature	Contribution
Clayton “Rhododendron” Dembski		(33+⅓)%
Junbong “Edward” Jang		(33+⅓)%
Floris “Grashof?” van Rossum	FvRossum	(33+⅓)%

Abstract

The purpose of the final project is to overcome the challenge of replacing the nuclear fuel rod with a robot, which will complete the task quicker and with less consequence than humans. A field is set up with two nuclear reactors, a set of storage tube and a set of supply tubes. The robot must navigate to each reactor, remove the fuel, place it in storage, then use the supply tubes to obtain and replace the fuel in the reactor within ten minutes. The group needs to design and assemble the mechanical, and electrical parts of the robot and program them to work coherently. Also, the implementation of bluetooth communication with the reactor control is necessary to get an update on the field condition and send the robot's status to reactor control. After seven weeks of dedicated efforts and love, Team 8's robot has successfully overcame the challenge with a solid bluetooth communication.

Table of Contents

List of Figures:

1. Introduction

The use of robotics in situations otherwise life threatening for humans has become increasingly prevalent in modern day society. Natural disasters --earthquakes, tsunamis, hurricanes-- have created situations of chaos, unfit to navigate and dangerous to be in. Yet they are also being used in situations of controlled consequence. Nuclear fuel rods, those depleted of plutonium, uranium, and other radioactive elements and those not, are safely contained in a controlled environment. If the rods are to be moved, humans must go to extensive, time consuming lengths to protect themselves.

By having the robot that is immune to nuclear radiation perform tasks for humans, many human lives can live safely, while still benefitting from the power generated by the nuclear energy. With this in mind, the nuclear fuel rod replacement prototype robot has been requested by professor Bertozzi. The team 8 designs the frame of the robot out of plywood, actuate parts with Vex motors, and feel the external world through sensors such as line follower and quadrature encoder, and potentiometer. The frames of the robot are designed in 3D software called SolidWorks and the parts are later laser-cutted for assembly. Many lines of code are written in object oriented way so that the robot has objects for controlling motors and sensors, and communicating through bluetooth.

2. Methodology

Problem Statement

A problem is presented where a robot must be designed, built and tested to move depleted and fresh nuclear fuel rods to and from various storage containers. Navigation between storage containers is enhanced by a system of lines that could be used for line following. This line system can be seen in Figure 1. There are two ways to check the occupancy of the spent and fresh rod storage area, LEDs next to the storage container or a bluetooth signal. The robot is to start either at tube A or tube B and grab the rod set in a vertical position. The rod is to be placed horizontally, about 6 inches above the vertical position, in the spent rod storage area. The new rod must then be grabbed horizontally before finally being placed down vertically in the reactor. The storage and supply rods are randomly placed at the beginning of the trial. Red LED lights next to the rod and a bluetooth signal, encoded in binary, both indicate these positions.

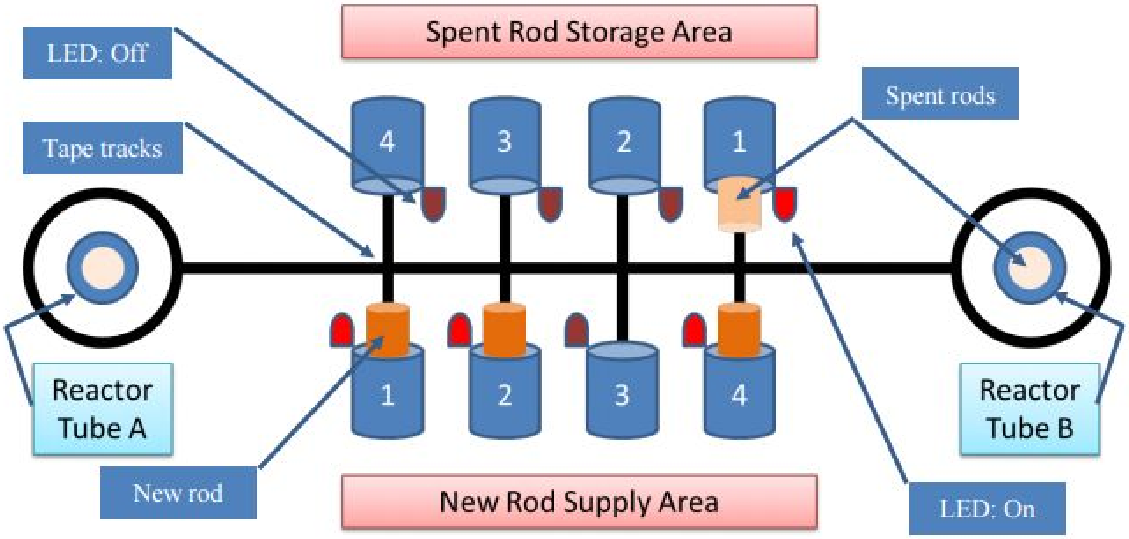


Figure 1: A diagram of the field

Because this problem was presented as a game various point values are assigned to completing different tasks. The point values for different tasks can be observed in Appendix A. Worth noting is that utilizing bluetooth to communicate with the field is worth a great amount of points, namely up to 45.

It is important for a group intending to complete the challenge to completely analyze the problem at hand in order to make well thought out decisions which form the basis of a solid design. The problem statement must be fully understood. That means all the documentation of the game must be read before moving on to the next step which is formulating design decisions.

Early Design Decisions

Team 8 had an early meeting to make early design decisions that would dictate the design of the robot and the code. This session was used to create the first concepts our robot would be based on.

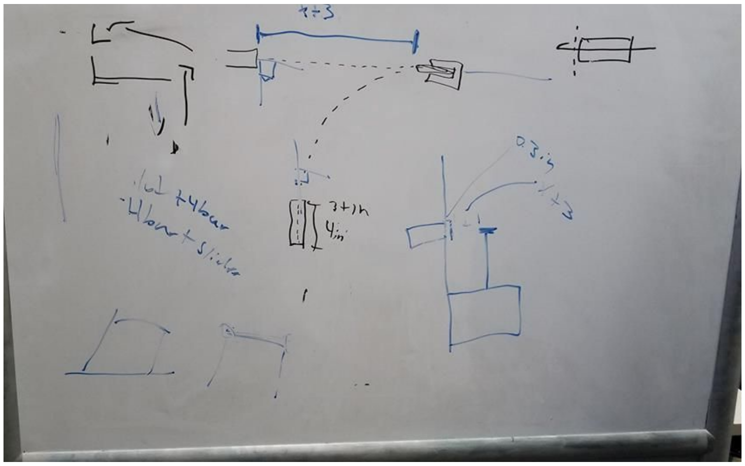


Figure 2: The notes of the first design session

Below are some of the design decisions that resulted from the session.

- Have two drive wheels
- Have a turning center around those two drive wheels
- Use a passive alignment mechanism
- Use the line follower to navigate the system of lines
- Communicate with the field using bluetooth messages
- Make a fourbar with hard stops
- Use a limit switch to detect arrival at a container
- Use encoders to measure orientation during or after a maneuver.
- Do not use the supplied grabber

We choose to use bluetooth over LED sensing due to its high point value. Because our group was made up of 2 computer science double majors, we believe that we could get the bluetooth fully functioning. This would act as a bit of a safety net, as, if the robot did not function to its full capabilities, but the bluetooth did, we would still get the assigned points from said bluetooth.

Because the robot had to receive the rod in both the vertical and horizontal positions, we needed to choose which method would be best to do so. Examining the past designs, the two that seemed most common were the linear slider and the fourbar. Although the linear slider seemed in general more accurate once assembled. We decided it would be too hard to tweak if there was any error in the design, and it would not have the power we needed to pick up our created grabber mechanism. The grabber mechanism was too error prone. It had no alignment system, and, if it grabbed a rod skewed, there would be no way of passive recovery. We wanted a form of sucking mechanism with a passive alignment mechanism. In this way, we would not need to worry about the positioning of the rod, as long as the fourbar was in the correct place.

With these decisions our team moved on to the next step and started creating designs for each sub assembly.

Preliminary Design

The team worked on various different sub assemblies at the same time. Almost all sub assemblies had an initial design before they were finalized, even the code.

The Drivetrain

In the Computer Assisted Design software SolidWorks, the drivetrain design was created. It was fairly thorough so most features in the initial design (Figure 3) led to the final design. This is because the design did not encounter any serious problems. There were, however, a few problems with the design. First, the gearbox was implemented right with the wheels. This generally causes problems as the wheels may conflict with the gears and will be, overall, less stable. In addition, the passive alignment mechanism was too small to align with the storage and supply wall. Finally, the front wall extended too far to allow the fourbar to pass: the top would be hit before passing to the alignment system.

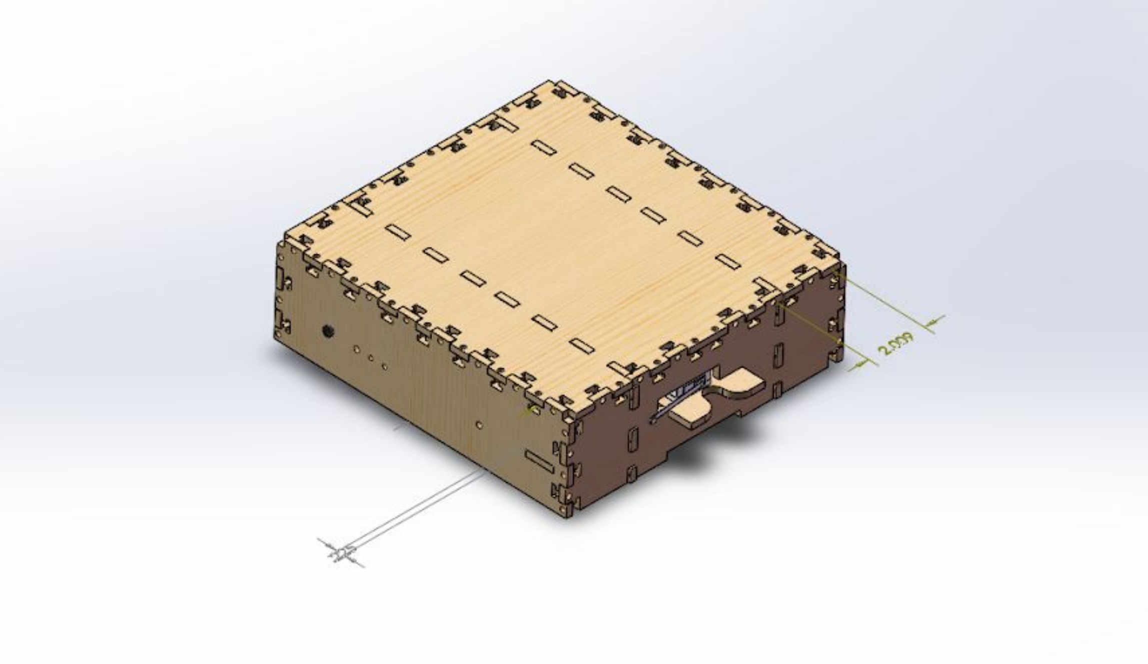


Figure 3: The first design model of the drivetrain

The Fourbar:

The fourbar was created based on the dimensions in the CAD (Computer Aided Design) model of the field [2]. The measurement were taken from this file as the field was not yet set up. With these measurements, SolidWorks was further used to create preliminary sketches of the fourbar (Figure 4). As one can see below, there are two small grey boxes, these signify the containers that store the nuclear fuel rods. The grabber mechanism, the blue box, has to align with those containers. The method of perpendicular bisectors was used to determine the optimal fourbar path with the desired endpoints.

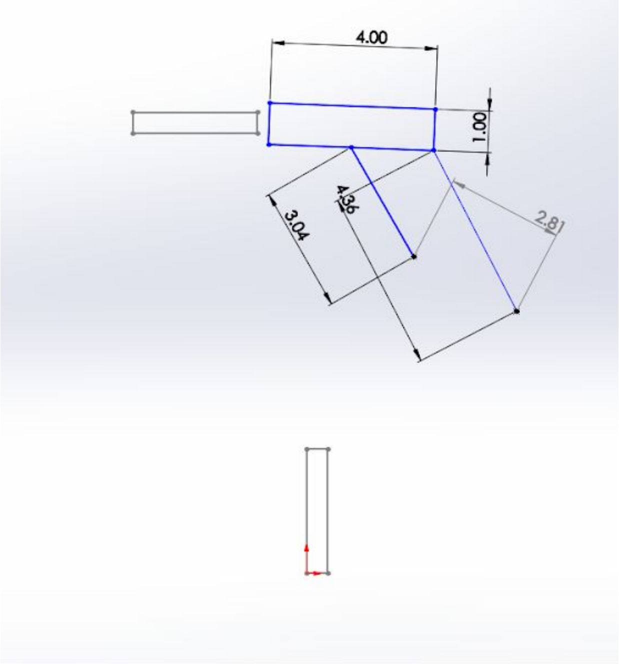


Figure 4: The first sketch of the fourbar mechanism

It took many different sketches and tries until a favorable fourbar linkage positions and lengths were found. The picture above was one of the first successful sketches. Not only did it align with the top and bottom, the path followed straight up from the lower position and straight back from the upper, allowing us, the controllers, to depend less on accurate PID control. An early analysis of the fourbar was not performed until the final design was being close to completed. Later on, MathCad and Norton Linkage Software were used to perform the force and velocity analysis to predict and understand its behavior.

Grabber:

Our group identified the need for a grabber/sucking mechanism rather than the given pincer mechanism. The group found that the way the rod could be retrieved and deposited in a linear fashion was preferable. This was found out soon after the early design session. Like the other sub-assemblies SolidWorks was used to create an early model (Figure 5). A funnel on the end was designed to help position the rod into the mechanism, and a limit switch was to be placed on the end to tell when the fuel was in position. Finally, a rubber band, or equivalent, was to be used as the main sucking mechanism.

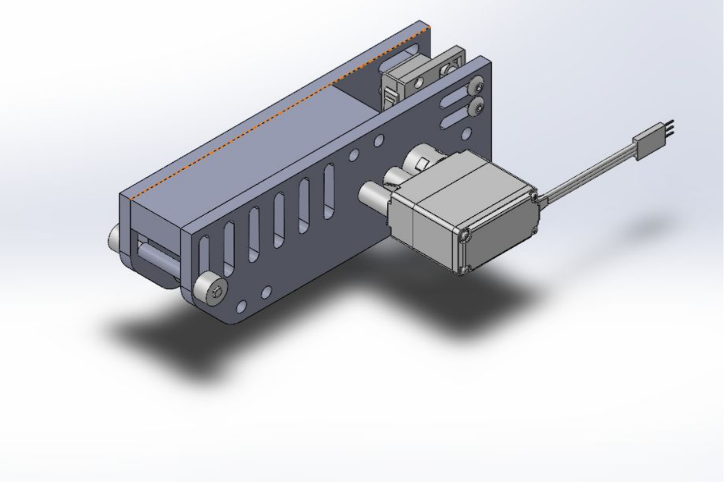


Figure 5: The initial grabber assembly

Sensors:

There were some sensors that are essential for each of our sub assemblies to function together correctly. First, as mentioned one limit switch was needed to detect when the rod is fully inside the grabber and a second to detect when the robot reached the tube. Second, one encoder that count how much the wheel has turned so that we can accurately turn 90 or 180 degrees. Third, line follower sensors mounted about one inch in front of the robot's virtual turning center to follow the black line. Last but not least, potentiometer on the shaft of the output gear of the fourbar to set the fourbar in two defined positions, upper and lower.

Coding:

Although it was difficult to begin coding when the was robot not even close to being finished, plans can be made. A map was drawn out on a whiteboard and the team started thinking of the path the robot would have to eventually take (Figure 6). This early brainstorming sessions identified numerous findings that would later on be used in the generation of the state machine function in the final code.

Due to the sucking mechanism discussed, we were able to eliminate grabbing multiple rods at once. This pushed our group to make the code so that the robot would travel from reactor to storage to supply and back again twice. We had decided the general path of the robot (figure 6). We also decided to try to implement object oriented programming to largest extent we could. In this way, multiple members could work on the code together at once. One could work on the main state machine function, while the others could give the header names and create the objects needed for functionality.

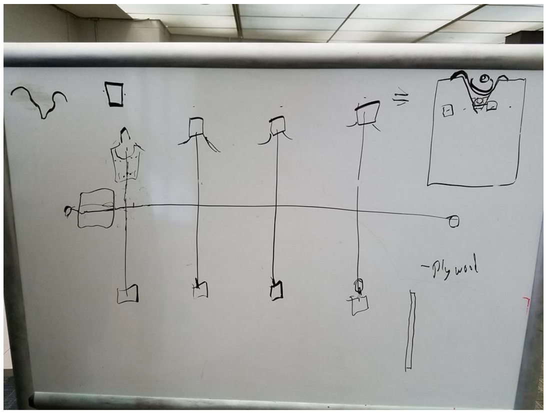


Figure 6: A drawing of the field

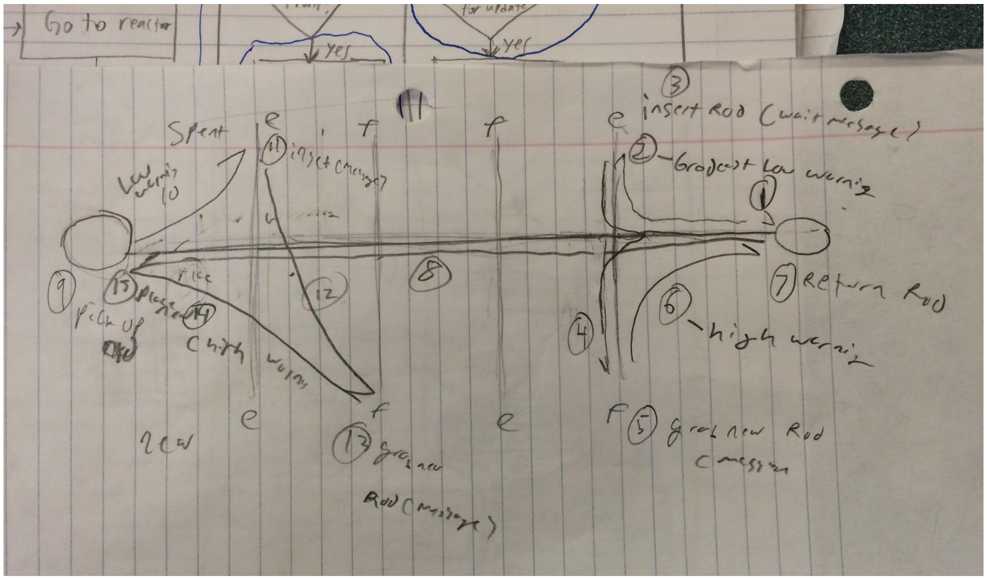


Figure 7: A quick sketch of the robot's path through the course

Early Testing

As soon as the preliminary design is done, numerous improvements can be made to the models while manufacturing is still ongoing. For our group, the drivetrain was assembled first and could be tested on the actual field. This testing session led to improvements to the drivetrain, such as the line follower mount. The purpose of early testing is to quickly identify problems in the initial design, because the quicker problems are noticed, the sooner a solution is brought up and implemented. After our early drive testing, we had discovered two things. The new passive alignment mechanism on the front was too small: the fourbar would hit the storage plates and not align with the storage or supply walls. We had a tolerance of about an inch of width in order to align correctly with the reactor tube. We also needed to move back the position of the line follower, so that we could have more play before the robot would notice the line. Otherwise, it would jot back and forth with little productive forwards movement. A picture below shows the drivetrain during the early testing phase.

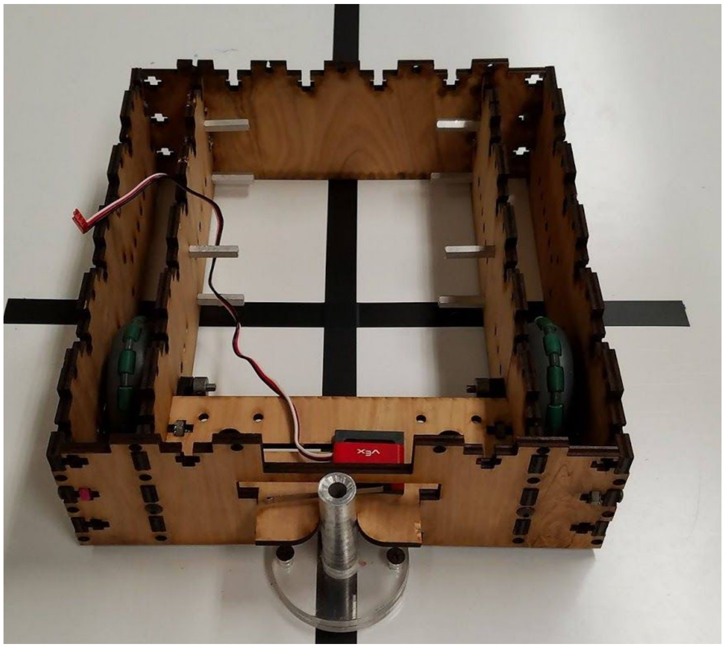


Figure 8: The robot in the early testing phase
**Note the first iteration of the lineup plate*

We were able to power the motors for the wheels to see speed and turning ability of the robot. The group was worried, initially, that the 5 to 1 gear ratio would slow the robot too much and the ⅛ an inch that the wheels extended from the bottom of the robot would be too low for proper movement. Later on it would appear that the 5:1 gear ratio worked fine, and the low clearance allowed the robot to read the lines exceptionally well.

Aside from testing on the actual field, the robot was placed on the field in SolidWorks (Figure 9). From this, numerous improvements were made that were very significant. Errors in measurement or calculation could quickly be identified and resolved while never building the robot.

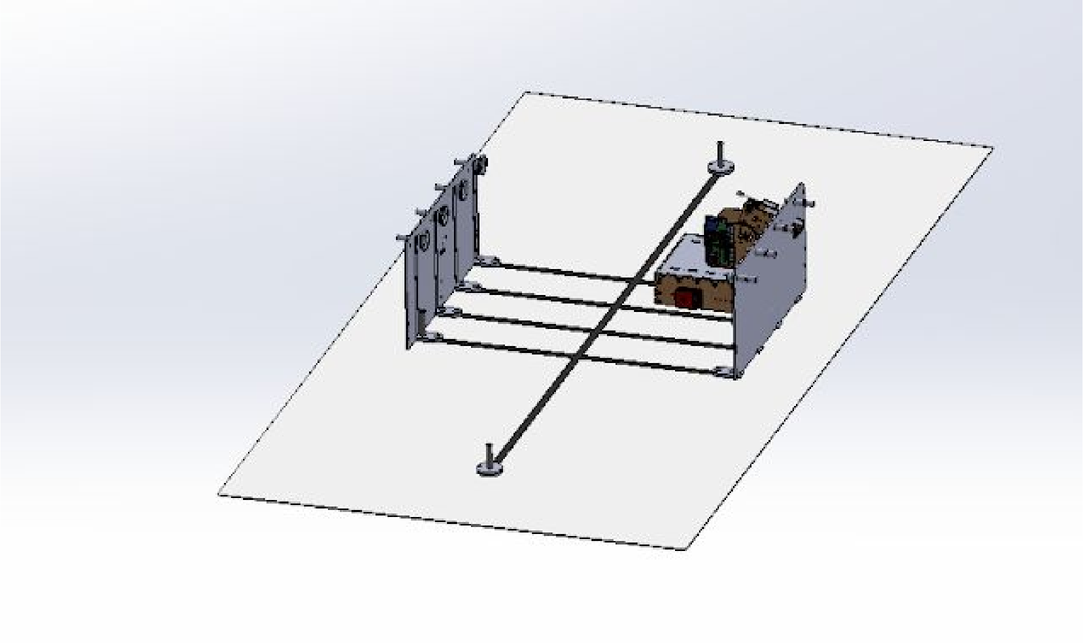


Figure 9: The robot in SolidWorks on a model of the field

5. Results

Final Testing and Design:

The Drivetrain

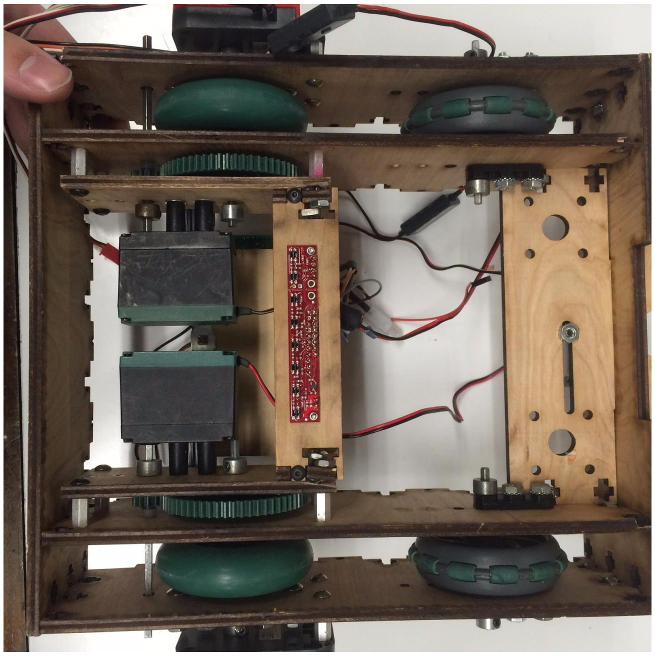


Figure 10: The final drivetrain

After testing the drivetrain was done. The gear ratio, the lineup plate and the line following were working up to appropriate standards and no further improvements had to be made. Because the drivetrain was built so early on, many errors were resolved early in the process. The placement of the line follower can be distinctly seen slightly in front of the motors. A gear box set behind the wheels was implemented, in comparison to colliding with the wheels themselves.

The Fourbar

Finding a working fourbar turned out to be a challenge. In order to pass by such a design challenge, our group solved the problem in steps. As seen previously, SolidWorks was used to create sketches of the fourbar. These were then used to create actual models of the linkages. The linkages are then used to create an assembly which is able to move. There is a picture of fourbar below (Figure 11). The path of the linkage was tested on the SolidWorks model of the field to determine if it was correct. Because of eventual errors in manufacturing or other unexpected results, the decision was made to make slots in the fourbar plate to allow for these events. The slots are seen below in Figure 11. This allows the fourbar links to move up and down, thus changing the path of the fourbar. This turned out to be crucial as it allowed the group to fix the fourbar after a miscommunication. Accommodating for errors similar to these is essential and preventative measures similar to these should always be implemented before completing the final design.

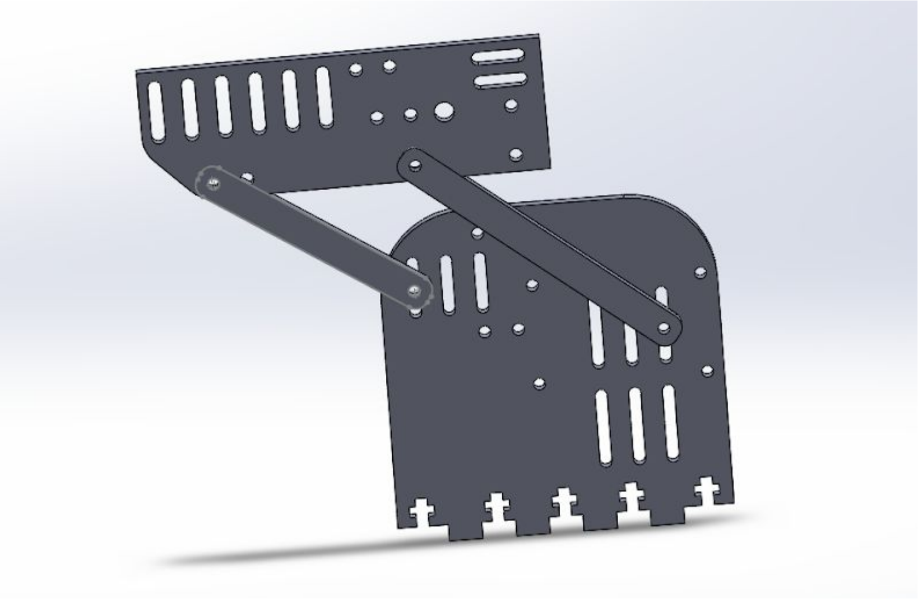


Figure 11: Sliding slots on the fourbar plate

Although many designs and sketches of the fourbar were made in addition to a laser-cut wooden fourbar, another one was made before it was complete. The final iteration of the fourbar pictured below, had holes instead of the previously mentioned slots to attach the fourbar linkages. This decision was made because our group did not want the fourbar path to be able to change by accident. Creating a defined path for the fourbar was possible now that mistakes in the previous manufactured fourbar were found and fixed.

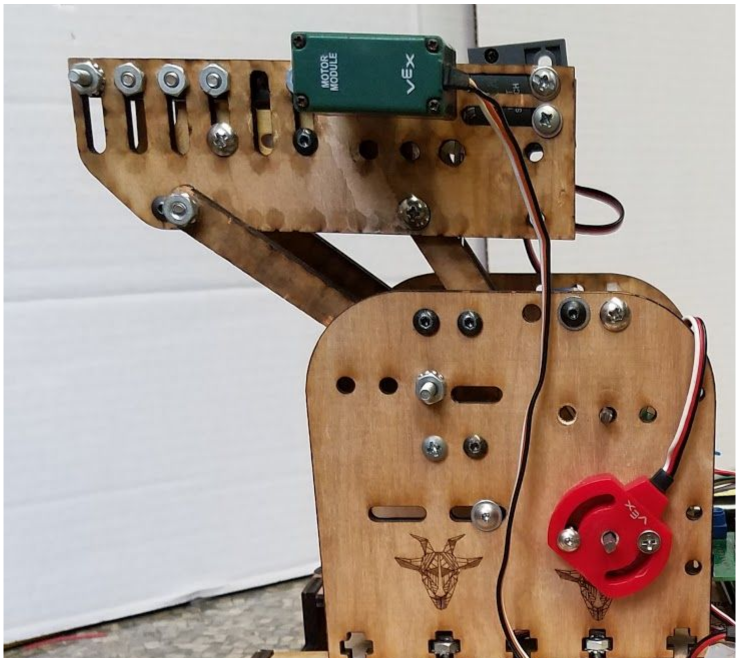


Figure 12: The final fourbar

The Grabber

The final iteration of the grabber did not have a significant improvement over the initial design. This is because the group did not have an early testing phase for the grabber assembly. This meant that few improvements were made to the grabber. The final grabber, pictured below (Figure 12), did have a number of improvements over the initial concept.

In the picture below on the left a gold 3D printed mount for the nuclear rod can be seen. This is because the grabber assembly that was made according to the design did not house the nuclear fuel rod well. So a 3D printed mount was made using SolidWorks and consequently 3D printed. In the SolidWorks design the grabber initially had the motor and the rubber band on the bottom. This was not found to be possible since the rubber band and axle underneath it would collide with the nuclear fuel rod in an attempt to grab it: the open side would contact the rod, pushing the rod away. The fourbar was physically unable to come down around the rod. By moving the motor and sucker to the top, and the “catcher” to the bottom, the edge of the sucker could rotate down onto the rod and press the rod into the system instead of outside of the system (figure 13)

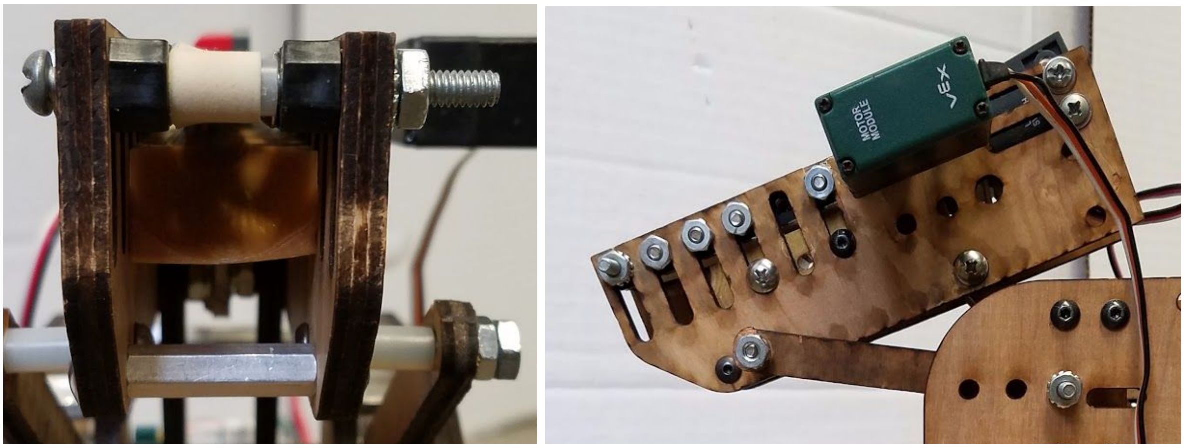


Figure 13: The final iteration of the grabber assembly

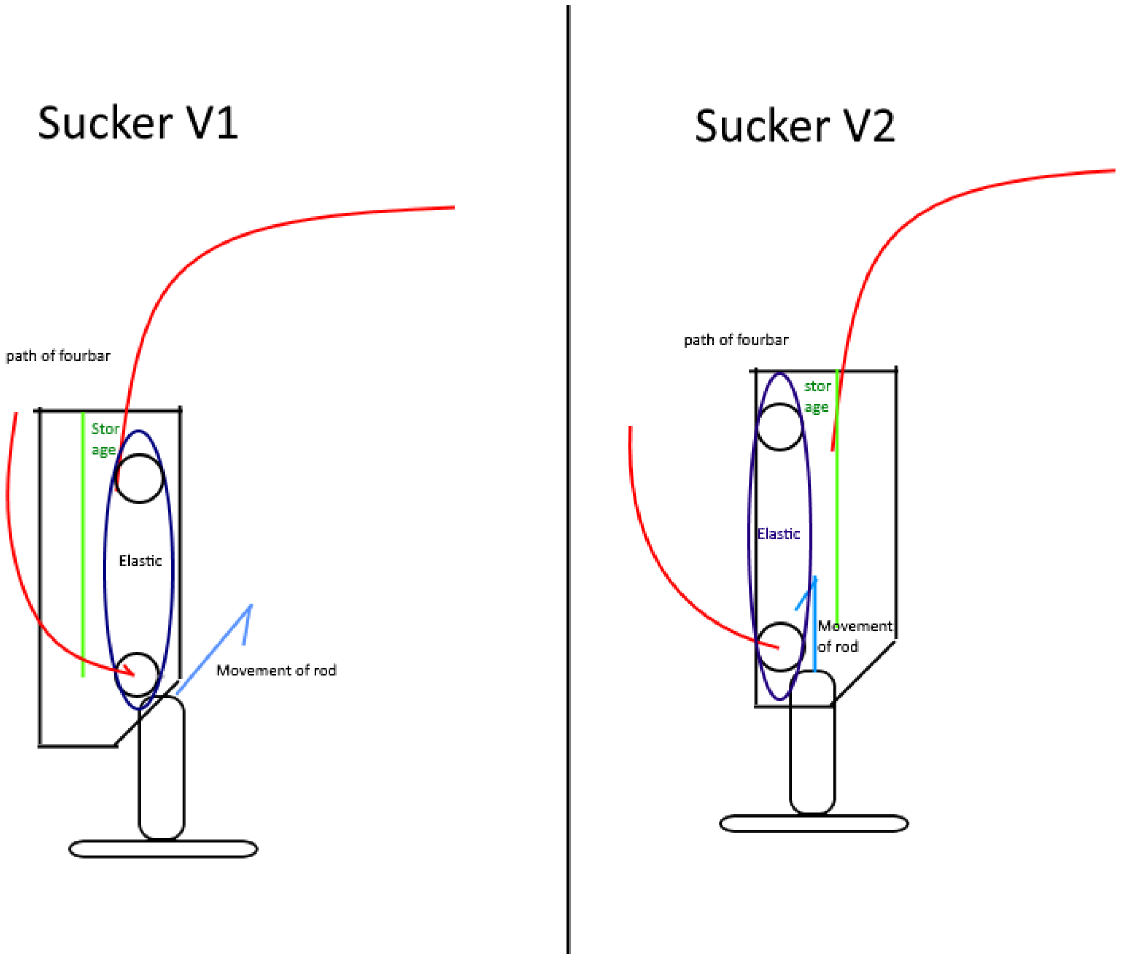
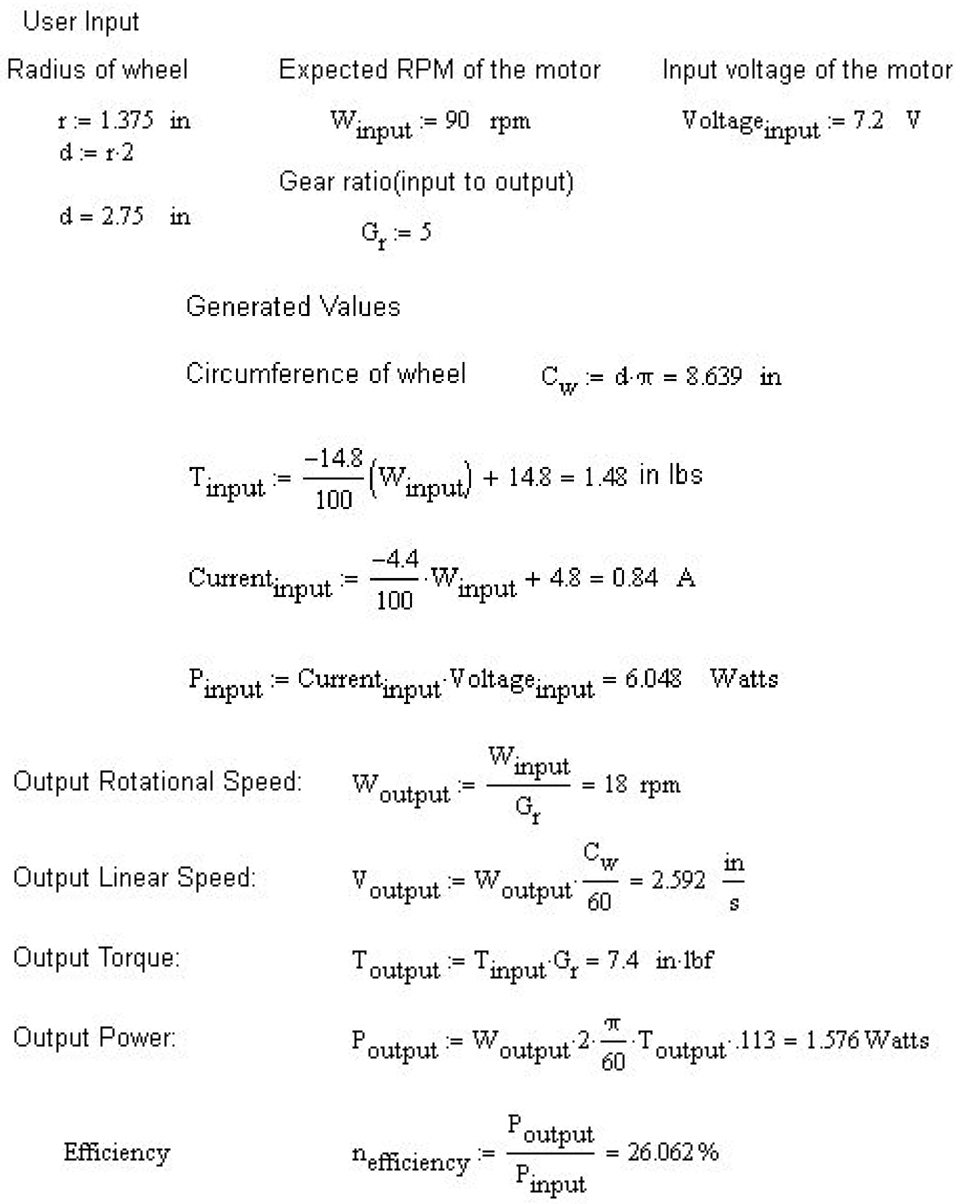


Figure 14: Drawing of initial grabber error

Driveline Analysis

For the initial drivetrain design, we designed equations in mathcad so that when we plug in gear ratio with expected input motor RPM, and radius of the wheel, the output linear speed will be calculated. After plugging in the various gear ratio, we agreed upon that gear ratio of 5 which results in linear output speed of about 2.6 in/sec is an optimal speed since it is not too fast nor too slow. Also, we found other values such as output torque, power and efficiency of the driveline system to understand and predict the expected performance of the driveline with a gear ratio of 5.

Below is a picture of the performed calculations. As you can see in the figure below, these findings influenced the gear ratio of the final design.



User Input

Radius of wheel

$r := 1.375\ \text{in}$

$d := r \cdot 2$

$d = 2.75\ \text{in}$

Expected RPM of the motor

$W_{input} := 90\ \text{rpm}$

Gear ratio(input to output)

$G_r := 5$

Input voltage of the motor

$Voltage_{input} := 7.2\ \text{V}$

Generated Values

Circumference of wheel $C_w := d \cdot \pi = 8.639\ \text{in}$

$$T_{input} := \frac{-14.8}{100}\left(W_{input}\right) + 14.8 = 1.48\ \text{in lbs}$$

$$Current_{input} := \frac{-4.4}{100} \cdot W_{input} + 4.8 = 0.84\ \text{A}$$

$$P_{input} := Current_{input} \cdot Voltage_{input} = 6.048\ \text{Watts}$$

Output Rotational Speed: $W_{output} := \frac{W_{input}}{G_r} = 18\ \text{rpm}$

Output Linear Speed: $V_{output} := W_{output} \cdot \frac{C_w}{60} = 2.592\ \frac{\text{in}}{\text{s}}$

Output Torque: $T_{output} := T_{input} \cdot G_r = 7.4\ \text{in} \cdot \text{lbf}$

Output Power: $P_{output} := W_{output} \cdot 2 \cdot \frac{\pi}{60} \cdot T_{output} \cdot .113 = 1.576\ \text{Watts}$

Efficiency $n_{efficiency} := \frac{P_{output}}{P_{input}} = 26.062\,\%$

Figure 15: The final drivetrain calculations

Fourbar Analysis

The mathcad equations for a fourbar from Homework#4 were modified to match the group's fourbar design and reaction forces in the x and y directions were obtained on link A, B, C and D. However, the most useful information we got is SM2 which is the torque required for the crank motor to move up or down the fourbar mechanism. According to the force analysis of the fourbar, the maximum output torque required is about 3 in-lbf. This means that the motor will run without any problem even if we used gear ratio of 1 since motor's stall torque is 14.8 in-lbf. However, we wanted the fourbar mechanism to move quite slowly so that it is more stable and easier for us to fine tune the lower and upper position of the fourbar using potentiometer. In both of our fourbar and driveline design, we wanted the movement to be slow since we have ten minutes to perform a task that requires precision.

$$SA_x = 0.15\,\text{lbf}$$

$$SA_y = 0.62\,\text{lbf}$$

$$SB_x = -0.15\,\text{lbf}$$

$$SB_y = -0.14\,\text{lbf}$$

$$SC_x = -0.15\,\text{lbf}$$

$$SC_y = -0.11\,\text{lbf}$$

$$SD_x = 0.15\,\text{lbf}$$

$$SD_y = 0.65\,\text{lbf}$$

$$SM_2 = 2.99\,\text{in}\cdot\text{lbf}$$

Four-Bar's Output Speed, Torque and Power consumption

$$\text{Torque}_{\text{output}} := SM_2 = 2.99\,\text{in}\cdot\text{lbf}$$

$$\text{Torque}_{\text{input}} := \frac{\text{Torque}_{\text{output}}}{5} = 0.6\,\text{in}\cdot\text{lbf}$$

$$\text{Speed}_{\text{input}} := \frac{-100}{14.8\,\text{in}\cdot\text{lbf}}\cdot\text{Torque}_{\text{input}} + 100 = 95.96 \quad \text{rpm}$$

$$\text{Current}_{\text{input}} := .4\text{A} + \frac{(4.8 - 0.4)\text{A}}{14.8\,\text{in}\cdot\text{lbf}}\cdot\text{Torque}_{\text{input}} = 0.58\,\text{A}$$

$$\text{Speed}_{\text{output}} := \frac{\text{Speed}_{\text{input}}}{5} = 19.19 \quad \text{rpm}$$

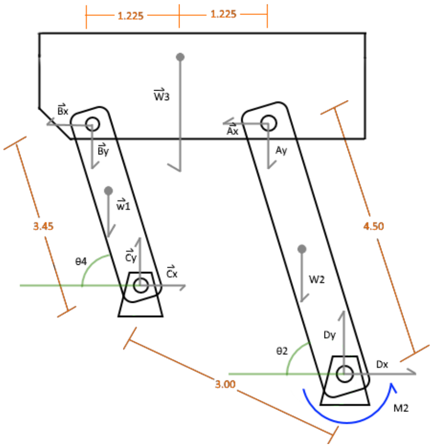
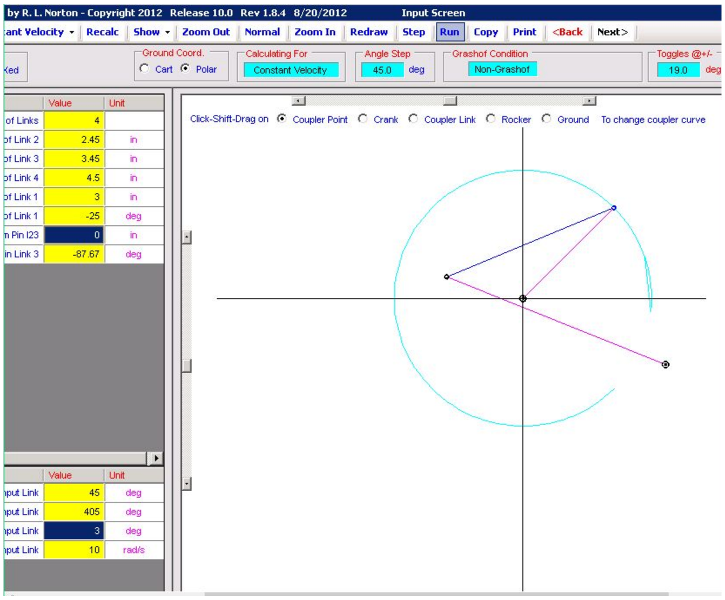


Figure 16: The force analysis of the fourbar mechanism

Norton linkage software was used to trace the path of the fourbar and find the speed at specific points in the path. Below screenshots (Figure 16) can be found of the fourbar trace as well as graphs representing the speed at specific angles.


by R. L. Norton - Copyright 2012 Release 10.0 Rev 1.8.4 8/20/2012 Input Screen

Ground Coord. ○ Cart ● Polar

Calculating For: Constant Velocity

Angle Step: 45.0 deg

Grashof Condition: Non-Grashof

Toggles @+/-: 19.0 deg

	Value	Unit
of Links	4	
of Link 2	2.45	in
of Link 3	3.45	in
of Link 4	4.5	in
of Link 1	3	in
of Link 1	-25	deg
n Pin I23	0	in
in Link 3	-87.67	deg

Click-Shift-Drag on ● Coupler Point ○ Crank ○ Coupler Link ○ Rocker ○ Ground To change coupler curve

	Value	Unit
nput Link	45	deg
nput Link	405	deg
nput Link	3	deg
nput Link	10	rad/s

Figure 17: The path of the final fourbar in Norton's Linkages Software

As can be seen in Figure 16 above, the fourbar is non-grashof. This can be seen by the incomplete circle in the fourbar trace as well as in the graphs below (Figure 17). The software produces unexpected and incorrect results at angles the fourbar should not be able to reach. The graph appears to move backwards in angles 350 - 375deg. This unexpected behavior was very interesting to see. The only angles the fourbar was moving between was 45-70 deg.

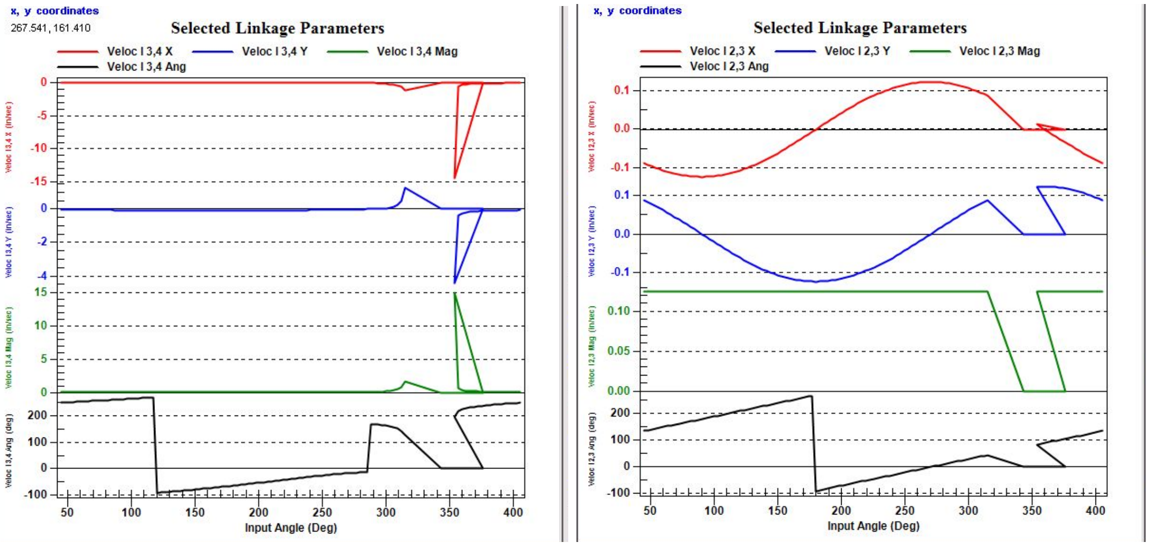


Figure 18: The velocity graphs for the fourbar

Calculations were performed to show that the fourbar is indeed non-grashof:

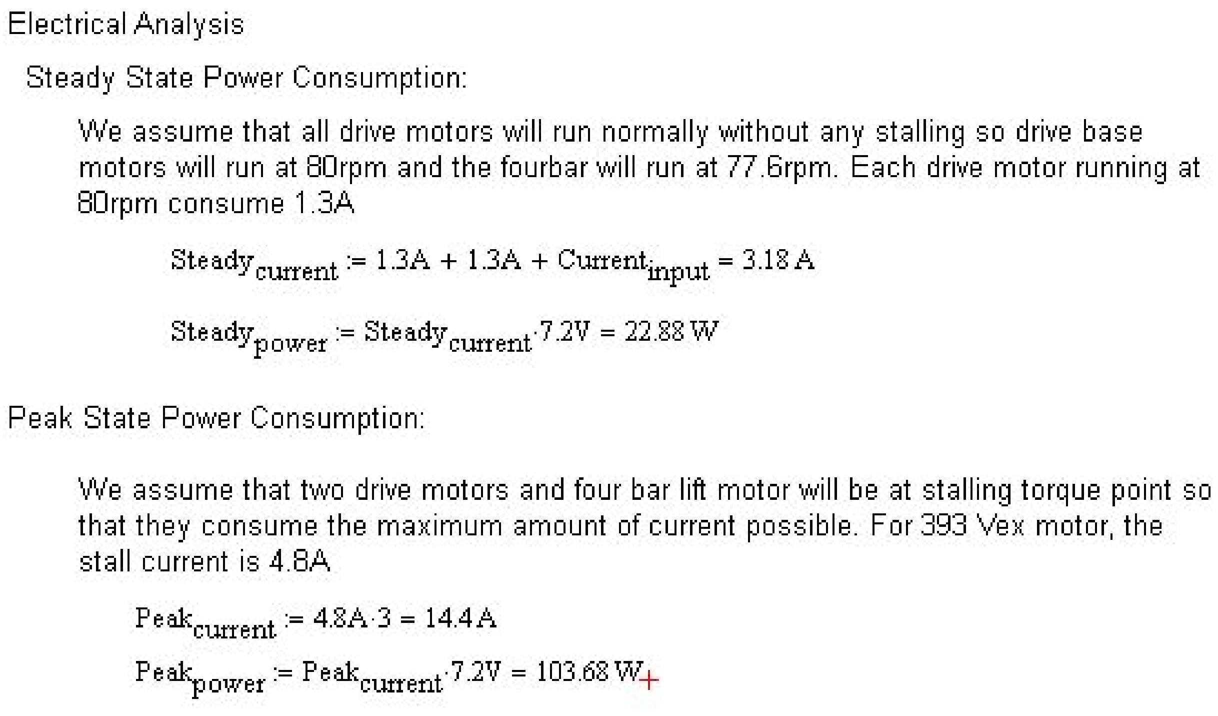
$$In\ order\ for\ a\ fourbar\ to\ be\ Grashof:\ S + L \leq P + Q$$

$$S = 2.45in\ \ L = 4.5in\ \ P = 3.45in\ \ Q = 3in$$

$$2.45 + 4.5 = 6.95in \quad 3.45 + 3 = 6.45in \quad 6.95 < 6.45\ (which\ is\ false)$$

Thus the grabber is not grashof and is not able to complete a full revolution. However for this challenge, the fourbar that is non-grashof works.

Electrical Analysis



Electrical Analysis

Steady State Power Consumption:

We assume that all drive motors will run normally without any stalling so drive base motors will run at 80rpm and the fourbar will run at 77.6rpm. Each drive motor running at 80rpm consume 1.3A

$$Steady_{current} := 1.3A + 1.3A + Current_{input} = 3.18\,A$$

$$Steady_{power} := Steady_{current} \cdot 7.2V = 22.88\,W$$

Peak State Power Consumption:

We assume that two drive motors and four bar lift motor will be at stalling torque point so that they consume the maximum amount of current possible. For 393 Vex motor, the stall current is 4.8A

$$Peak_{current} := 4.8A \cdot 3 = 14.4\,A$$

$$Peak_{power} := Peak_{current} \cdot 7.2V = 103.68\,W$$

Figure 19: The electrical analysis of the robot

Final design of the code

We drew a state machine diagram to know what helper functions we would need and program the state machine function methodically. The state machine is divided into 8 steps. The first step is to retrieve the old rod from the reactor. Second step is to deliver the old rod to the deposit station. The third step is to eject the old rod, move back and turn 180 degrees to get ready for next step. The fourth step is to go to the new rod station. The fifth step is to suck in the new rod, move back and turn 180 degrees for next step. The sixth step is to go back to the reactor. The seventh step is to eject new rod to the reactor, move back, and turn 180 degrees for next step. The eighth step decides whether to run the same sequence one more time for another reactor.

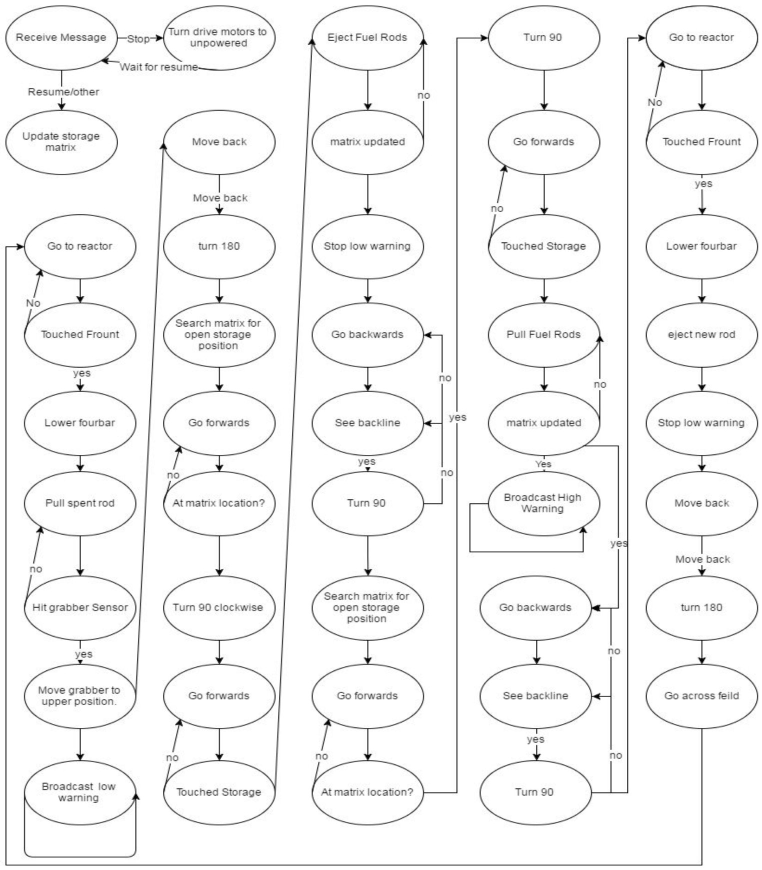


Figure 20: State Machine Diagram

While the main loop and the state machine function is in the ReactorControl.ino file, there are helper functions called by main loop inside five different classes for controlling the motors, collecting and processing values from sensors, and communicating with bluetooth. BTComms and Messages classes are for bluetooth communication. The classes generates messages that will send a low or high warning to the field and continuously updates the positions of the storage and supply rods. It also allows the robot to receive stop and resume messages from the field, which override the motors. LineFollower class enable the robot to follow the black line using line follower sensors and go to certain destination. A position matrix was used to tell where the robot currently is in comparison to where the robot should be, and computes the necessary actions to get from point a to point b. BaseEncoder class utilizes quadrature encoder to make the robot turn 90 or 180 degrees. The encoder values would need to be reset before the robot could rotate again. The FourbarMechanism class actuates the crank motor with potentiometer to set it at either upper, or lower positions using PID control. By designing the code in this object oriented manner, the code became more comprehensive and programmers could separately implement different classes and integrate them later in a State Machine function.

BtComms	Messages	LineFollower	BaseEncoder	Fourbar Mechanism
writeMessageWithData(unsigned char b1, unsigned char b2, unsigned char b3, unsigned char b4)	Read (bool *stopped)	precisionDrive (int, Servo, Servo)	turnDegrees (double)	void moveFourbar(int, Servo, int);
	LowWarning()	turnNinety (int, Servo, Servo)	resetEncoder()	Double pidCompute (int, int);
	HighWarning()	Position matrix functions		
	decodeAndUpdate()			

Figure 21: Object Oriented Design table

6. Discussion

In this subsection of the report, numerous results and their meanings can be found. From those meanings, conclusions and future improvements were identified.

Driveline

The driveline was found to be unpredictable with changing variables. Occasionally when the robot turned 90 or 180 degrees, it overturned or underturned. These variations were caused by change in weight of the robot or the voltage of the battery. At first, the robot without fourbar mechanism on top or extra weight near VTC (Virtual Turning Center) was slipping significantly when turning and would trail off the line. However, when we put the weight near the vtc, it affected the tractive force on the wheel and made the turning much more reliable and accurate. In the future more calculations should be performed to estimate the required tractive force in order to turn without slipping. Additionally, after testing different code numerous times, as the battery voltage decreased, the motor power and encoder's sensitivity decreased and turned less accurately. Thus, we always tried to use a fully charged battery. The afternoon before the demonstration, we were working with, what we presumed was a full battery, yet it turned out that the battery was undercharged. This increases encoder's sensitivity in the robot, thus forcing the robot to overturn during our final presentation. Thanks to the lineup plate, the robot managed to catch itself when aligning to the container. However, it sometimes overturned and lost the line, so it had to be assisted to continue with the operation.

Lining up the base to a specific location in the correct orientation for the fourbar to align up with the container tubes proved to be challenging. The line follower and the lineup plate were enough to get the robot to the right position but it would often not be in the correct orientation. Improvements such as a passive mechanism that ensures that the robot is in the correct orientation would have been helpful.

Fourbar

The four bar of the robot was not perfectly aligned with tubes so it required many changes. The mechanism was shifted slightly towards the right, yet, when it came time to align with the supply and storage, we were shifted either to the left or to the right. Much of the two weeks before the final performance was spent on cading the final four bar, attempting to get something that would be absolutely perfect. If we had spent another month on it, the positioning still would not have been satisfactory. It was not until we were able to test the four bar that we were able to understand the problems and begin to fix it. It would be advantageous to create a preliminary design of the robot, model, and then manufacture it in order to test it to the extent of its abilities, before going back to redesign. In this way, we could tell more rigorously what we needed to take into account: the play of the four bar arms and the exact height for the back two links of the bar. Although this was already done, as we manufactured two fourbars, it is something that should be tested much earlier in the process. Aside from this, a shortage of parts meant that the proper spacers that were required for the fourbar were not found. As a result, the offset of the fourbar was not easily fixed. Additionally, problems in offset could also have been fixed by anticipating this, and creating a design solution that allows the fourbar and the grabber to be adjusted left and right.

Grabber

The grabber required certain amount of friction to properly suck in the reactor rod. As a result, if the rubber band on the grabber wasn’t touching the rod close enough, grabber could not suck in or push out the rod. This meant a constant readjusting of the 3D printed part to allow for sufficient pressure to manipulate the rod. Following this, when the robot went to place the fuel rod in storage, the sucking mechanism seemed to be a setback. Because the rod was aligned perfectly straight, yet the fourbar was shifted to one side, the sucker had problems pushing out the rod. The rod would need to be pushed diagonally in order to move into the storage, yet this was impossible due to the mechanical design. With the standard grabber, because the rod was grabbed at the end, there was sufficient torque and play on the rod that, when inserted, the rod would shift diagonally due to the normal force of the input cone and slide into the storage hole. The rubber band needed to be replaced often since it wore down and broke. Not all of the edges on the 3d printed part, nor the addition to the limit switch were rounded. The rubber band would slowly chafe against these parts until the damage was irreparable. Additionally a large amount of force had to be applied to the rod in order to insert it into the container. Sometimes when the rod was sucked in, the rod did not properly contact the touch sensor at the end of the grabber so the state machine never executed the next step. As a result, small metal fragment was attached to the limit switch to decrease the chance of this happening. This metal fragment improved the accuracy of the limit switch but had the consequence, as mentioned before, of lowering the lifespan of the rubber band.

7. Conclusion

In a mechanical sense many lessons could be learned. Many problems encountered throughout the final project can be traced back to a flawed and improperly planned out schedule. It is virtually impossible to eradicate all errors in the design, instead errors should be taken into consideration when designing and manufacturing. Although there was room for error, such as the slots in the four bar plate, or the ability to re-make and redesign numerous parts of the robot, more time should be set aside for solving them. Whenever the group encountered mistakes it everything back significantly. Because of delays in the design and manufacturing the rest of the robot, such as the electrical and programming, was delayed as well. A more detailed schedule with room for mistakes and delays as the ones described should be made for the next challenge. Because of the delays less testing was performed than was required. For example the fourbar and the grabber should have been tested at the same time the drivetrain was done in order to make improvements. Relatively, the team performed exceptionally well. The challenge was completed a day before the due date, an exceptional feat. Aside from the scheduling problems, the mechanical solution to the problem was sound. The drivetrain was extremely rigid and well-built. The grabber was a good solution to deposit and retrieve the nuclear rod from the container. The lineup plate worked extremely well and saved the robot numerous times from being misaligned with a desired position. However the four bar was not built as well. While it traced the correct path it was not as rigid and could be offset from the desired location. This was in part due to the shortage of parts and the inability to find the correct spacers. Still this problem could have been foreseen and the four bar should have been designed better with this problem in mind. Overall the robot was mechanically sound, however time should be allocated better and mistakes and delays should be expected.

Electrically, we would use a different sensor to measure the number of degrees the robot has turned. Rather than using quadrature encoder, a gyroscope should have been used to accurately measure the rotation of the robot. When we ran the same code that turns 90 degrees, sometimes the robot overturned or underturned, causing the robot to go totally off the line. This also would have reduced unnecessary overhead in the code because we had to set the initial value of Encoder, and reset the encoder value every time we want to use it again in the program. The Pololu QTR-8 line follower sensors require calibration every time we reboot the robot so it was very cumbersome but it was very reliable and predictable.The placement of the fourbar potentiometer could have been more optimal. The sensor struggled to measure the small amount of rotation that was required at certain points in the routine. In the future a specific axle with a gear reduction to allow for less rotation (to not break the potentiometer) but still allow for a large amount of rotation for the encoder to measure.

Program-wise, the structure and hierarchy of the code could have been designed in a more sophisticated and neat way. For example, we should have made the LowerBody class that takes LineFollower class and BaseEncoder class and UpperBody class that takes FourbarMechanism class and grabber class. Lastly, the ReactorControl main function would utilize objects of the class LowerBody, UpperBody, Messages and BTComms classes. Also, with more time, we would have fine tuned PID control more for turning 90/180 degrees and lowering and lifting the four bar mechanism so that they behave more reliably throughout multiple tests, instead of relying on forces that mildly stall the four bar on the mechanical stops for short periods of time.

Overall there is a very extensive list of lessons that can be taken away from this challenging experience. In all areas improvements can be made and the mistakes the group made should not be made in the future.

8. Comments

We liked the fact that information about final project was given out from the very early beginning of the term so that each team can have enough time to prepare for final demonstration. we learned how to analyze the robot to see whether the possible design of the robot is feasible. Also, we learned how to utilize and program with the external sensor other than vex sensors by referencing the library and example codes. we improved the skills in constructing object oriented code for the robot, programming PID controls, and debugging the robot methodically. However, it would be helpful if someone upload documents containing correct information about final projects, especially bluetooth. Also, it would be helpful if SA/TAs checked to see if the reactor control field needed maintenance everyday for the two weeks before the presentation.

Overall near the due date of the final project the equipment and parts in the lab were in a bad state. The most challenging part of building the robot could be the fact that finding parts was extremely difficult. Problems occurred when a specific part was required but no longer available. It was extremely hard to find the tools that one needed to build the robot.

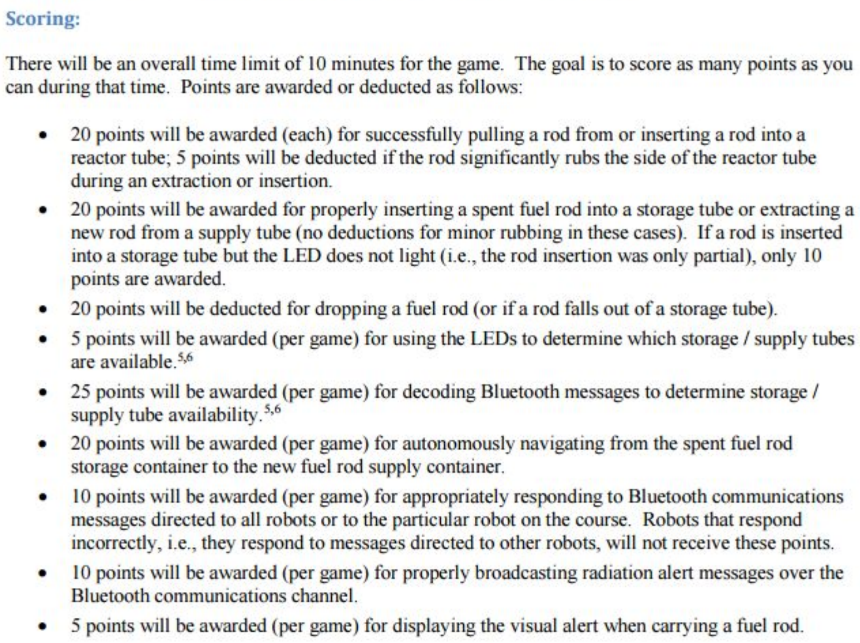
9. Bibliography

[1]https://www.dropbox.com/sh/uslanjjr9n81ksy/AAA1Cd_jGIjHp-hE7v1Ym45ga?dl=0
[2]https://canvas.wpi.edu/courses/2855/pages/force-analysis-review-i-ken-stafford-lecture?module_item_id=54517
[3]https://canvas.wpi.edu/courses/2855/pages/force-analysis-review-ii-ken-stafford-lecture?module_item_id=54518
[4]https://canvas.wpi.edu/courses/2855/pages/lectures-2-and-3-kinematics-fundamentals-i-and-ii?module_item_id=39187
[5]https://canvas.wpi.edu/courses/2855/pages/lecture-4-position-analysis?module_item_id=39188
[6]https://canvas.wpi.edu/courses/2855/pages/lecture-8-arduino-programming-i?module_item_id=39192
[7]https://canvas.wpi.edu/courses/2855/pages/lecture-12-bluetooth-and-packet-protocols?module_item_id=39196
[8]https://canvas.wpi.edu/courses/2855/pages/lecture-13-pid-control?module_item_id=39197
[9]https://canvas.wpi.edu/courses/2855/pages/lecture-16-state-machine-programming?module_item_id=39201
[10]https://canvas.wpi.edu/courses/2855/pages/lectures-17-and-18-more-arduino-programming-ins-and-outs-of-isr-programming-and-c-slash-c++-libraries?module_item_id=39202
[11]https://canvas.wpi.edu/courses/2855/pages/final-project-page?module_item_id=39288
[12]https://canvas.wpi.edu/courses/2855/pages/final-project-bluetooth-info
[13]https://github.com/WPIRoboticsEngineering/ReactorControl
[14]https://play.google.com/store/apps/details?id=edu.wpi.rbe2001.field
[15]https://en.wikipedia.org/wiki/PID_controller
[16]https://www.arduino.cc/en/Reference/HomePage

10. Appendices

Appendix A: Given Resources

[1]. Canvas: Final Project Assignment Page: RBE 2001 C 17 Final Project PDF, Point Table

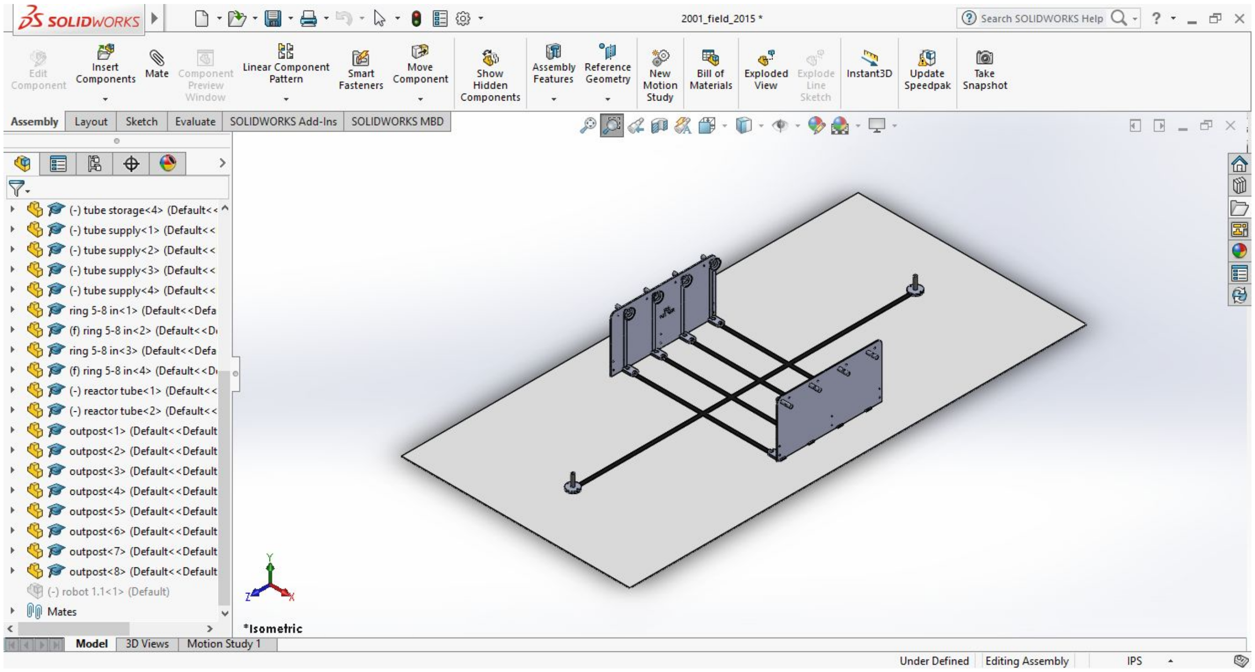


Scoring:

There will be an overall time limit of 10 minutes for the game. The goal is to score as many points as you can during that time. Points are awarded or deducted as follows:

- 20 points will be awarded (each) for successfully pulling a rod from or inserting a rod into a reactor tube; 5 points will be deducted if the rod significantly rubs the side of the reactor tube during an extraction or insertion.
- 20 points will be awarded for properly inserting a spent fuel rod into a storage tube or extracting a new rod from a supply tube (no deductions for minor rubbing in these cases). If a rod is inserted into a storage tube but the LED does not light (i.e., the rod insertion was only partial), only 10 points are awarded.
- 20 points will be deducted for dropping a fuel rod (or if a rod falls out of a storage tube).
- 5 points will be awarded (per game) for using the LEDs to determine which storage / supply tubes are available.[5,6]
- 25 points will be awarded (per game) for decoding Bluetooth messages to determine storage / supply tube availability.[5,6]
- 20 points will be awarded (per game) for autonomously navigating from the spent fuel rod storage container to the new fuel rod supply container.
- 10 points will be awarded (per game) for appropriately responding to Bluetooth communications messages directed to all robots or to the particular robot on the course. Robots that respond incorrectly, i.e., they respond to messages directed to other robots, will not receive these points.
- 10 points will be awarded (per game) for properly broadcasting radiation alert messages over the Bluetooth communications channel.
- 5 points will be awarded (per game) for displaying the visual alert when carrying a fuel rod.

[2]. Canvas: Final Project Assignment Page: 2001_field_2015_rev1.zip



(-) tube storage<4> (Default<<
(-) tube supply<1> (Default<<
(-) tube supply<2> (Default<<
(-) tube supply<3> (Default<<
(-) tube supply<4> (Default<<
ring 5-8 in<1> (Default<<Defa
(f) ring 5-8 in<2> (Default<<D
ring 5-8 in<3> (Default<<Defa
(f) ring 5-8 in<4> (Default<<D
(-) reactor tube<1> (Default<<
(-) reactor tube<2> (Default<<
outpost<1> (Default<<Default
outpost<2> (Default<<Default
outpost<3> (Default<<Default
outpost<4> (Default<<Default
outpost<5> (Default<<Default
outpost<6> (Default<<Default
outpost<7> (Default<<Default
outpost<8> (Default<<Default
(-) robot 1.1<1> (Default)
Mates