Disaster Relief Supply Delivery System Proposal - Report

Team 54

Abishek Kannappan

Trevor Ladner

Andrew Oliver

April 26, 2019

Dr. Leandra Wilson
Lead Engineer for the Harris Disaster Response Division
1025 W. NASA Boulevard
Melbourne, Florida 32919

Dr. Wilson,

The engineering team has been diligent in working to find the best possible solution for making a Global Emergency Autonomous Response System, or GEARS, that will satisfy all possible needs of the Harris Disaster Response Division. As a refresher, we were tasked with designing a prototype for a system that would be able to autonomously navigate a disaster zone. To be successful, it must be able to determine a traversable path, avoid electrical activity and high temperature sources, return a map of the path taken, deliver cargo at the target location, and present the cargo as a familiar resource to those who are needing the resources. The goal of the team was to provide the best possible version of the prototype possible by exceeding the goals outlined by Harris.

There are many aspects of the team's design that we wish to draw your attention to. The first of these features is the overall compact design of the GEARS prototype. This is a major positive aspect as it will allow for the GEARS to be able navigate safely through the terrain. When the GEARS is scaled up to the full size, the small size will allow it to more easily make its way through the disaster zone as it determines the best path towards mission completion.

Another major aspect that the team wishes to draw your attention to is the ultrasonic sensor system. The team has created a system that involves three ultrasonic sensors that are place on the front, right, and left sides to determine the distance from the walls of those crucial points. Each sensor runs continuously as it gathers data to determine possible paths there are for the GEARS to take. This system is very accurate as the data is coming from every direction to plan for the best possible path.

The final major aspect is the mapping feature of the GEARS. The GEARS has the ability to determine the distance that it travels in the forward direction and output that to a matrix. It is also able to determine the locations of where it turns and output that to the map. This allows it to successfully generate a route that displays the best way to exit the maze. In addition to this, it is able to signify the locations of the major hazards in the path and what type of hazard they are. This will allow the user to see a computerized representation of the hazard zone to allow for further success.

Some other features that the team wishes to highlight are:

- An accurate point turning system
- An auto driving correction system
- An auto cargo deposit system

These features combine to create a capable prototype for a GEARS system that can provide consistent and reliable results. The team put countless hours of work into creating the best design possible through research, prototyping, and testing with the goal of creating the best possible prototype for Harris.

An added benefit to this design is the ability for it to easily be scaled up to a full-sized prototype to be used in disaster scenarios around the world. Nearly every aspect of the design can be taken and put into the full size design.

With all of this in mind, the team hopes that you are as excited about the design as they are. The team eagerly awaits any feedback or questions that you have on the design. They wish to convey their gratitude for the opportunity to work on this project for the Harris Corporation and hope that you may return to them for any further jobs needed to be completed.

Thanks for your time and your business,
Team 54

Executive Summary

A robot that can navigate the dangerous disaster zones while avoiding damage, delivering the appropriate materials, and mapping the area has not been developed. Therefore, new robot and software designs are being developed to ensure the successful traversal and mapping of disaster areas while transporting the needed supplies.

The code for this robot design, called GEARS, featured a series of different functions compiled together in a larger function, optimizing a while loop to allow for the traversal and mapping of the maze. These functions included different turning functions, which turned the GEARS left, right, or completely around using gyroscope data to accurately turn 90 and 180 degrees, and a mapping function, which took input from the motor encoders located on the motors, determined the total distance traveled during a specific set of time, and added these values to a map. It also included methods for detecting different types of hazards, including infrared and magnetic signals. The code used these sensors to avoid hazards and prevent damage to GEARS.

The robot's structure consisted of a very compact design, using a differential drive setup for easy control and traversal. GEARS contained a number of different sensors attached to it for the purpose of traversal and detection, including the IMU, infrared sensor, three ultrasonic sensors, and a motor encoder located within the left motor. A ramp connected to a motor was attached to the back of the GEARS system for the transportation and depositing of the cargo container at the end of the maze.

During the final demonstration, GEARS was able to make 90 degree turns successfully both left and right while correcting slightly to realign the robot for an optimal later turn. It also demonstrated the ability to track and map the explored areas of the maze. It was also able to detect a magnetic source within a 6 inch radius and avoid the obstacle. The design cargo container was also able to successfully house the materials without spilling.

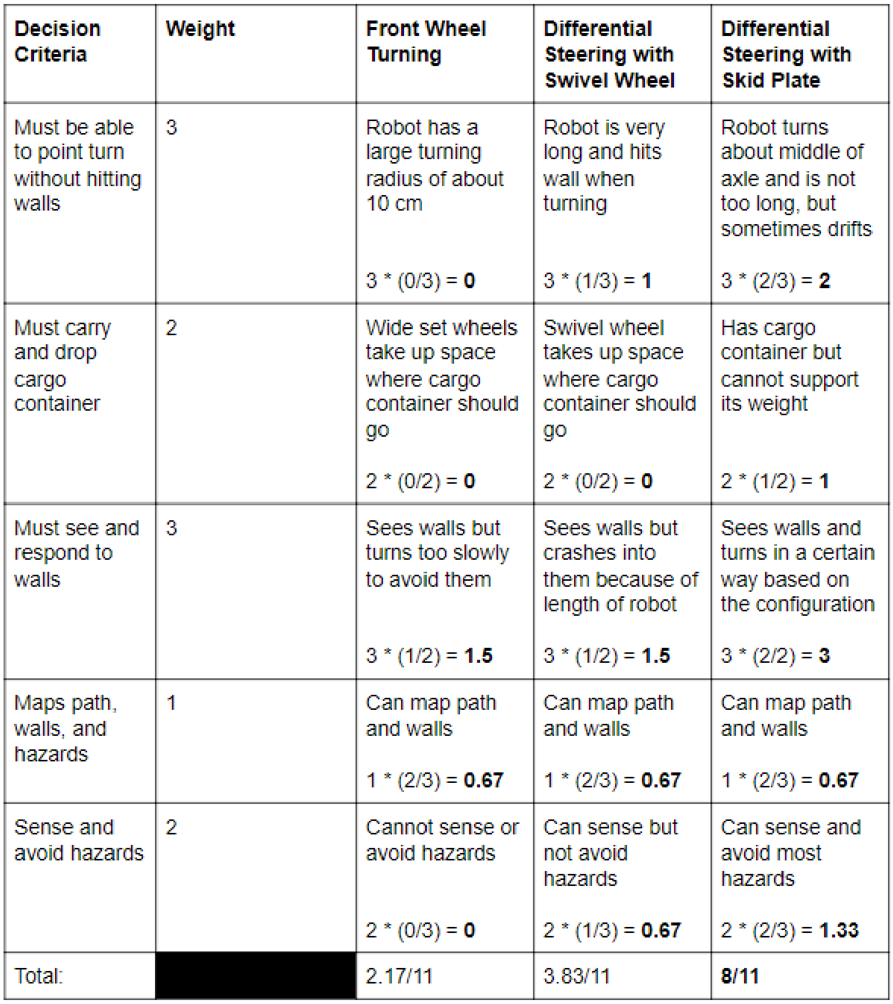
In summary, although the GEARS was unable to complete the entire demonstration course, with minor changes, a full size version could easily be scaled for a disaster situation.

Design Considerations

The final GEARS design was a product of an extensive design process encompassing decisions dealing with its hardware, its software, and the design of the cargo container. Decision-making tools, including decision matrices and data analyses, were employed to consider a wide array of factors and arrive at a suitable design.

The first step of the design process was to brainstorm ideas—specifically, for an initial physical design. Using a traditional brainstorming technique and focusing on quantity, being open to unusual ideas, and combining successful features of different ideas, three designs were thought of, built, and tested. The decision matrix below shows the three different designs considered as well as the criteria used to evaluate them.

Figure 1: Decision Matrix for Overall Design



Decision Criteria	**Weight**	**Front Wheel Turning**	**Differential Steering with Swivel Wheel**	**Differential Steering with Skid Plate**
Must be able to point turn without hitting walls	3	Robot has a large turning radius of about 10 cm 3 * (0/3) = **0**	Robot is very long and hits wall when turning 3 * (1/3) = **1**	Robot turns about middle of axle and is not too long, but sometimes drifts 3 * (2/3) = **2**
Must carry and drop cargo container	2	Wide set wheels take up space where cargo container should go 2 * (0/2) = **0**	Swivel wheel takes up space where cargo container should go 2 * (0/2) = **0**	Has cargo container but cannot support its weight 2 * (1/2) = **1**
Must see and respond to walls	3	Sees walls but turns too slowly to avoid them 3 * (1/2) = **1.5**	Sees walls but crashes into them because of length of robot 3 * (1/2) = **1.5**	Sees walls and turns in a certain way based on the configuration 3 * (2/2) = **3**
Maps path, walls, and hazards	1	Can map path and walls 1 * (2/3) = **0.67**	Can map path and walls 1 * (2/3) = **0.67**	Can map path and walls 1 * (2/3) = **0.67**
Sense and avoid hazards	2	Cannot sense or avoid hazards 2 * (0/3) = **0**	Can sense but not avoid hazards 2 * (1/3) = **0.67**	Can sense and avoid most hazards 2 * (2/3) = **1.33**
Total:		2.17/11	3.83/11	**8/11**

The first design considered was brainstormed using the prior art technique. One prior solution to the problem of turning was a tricycle, and this is what inspired the initial design. It featured rear-wheel drive but had a single wheel controlled by a motor with 360°-freedom in the front, similar to that of a tricycle. This allowed it to turn in any direction, but the turning radius was quite large, and it could not point turn accurately. The wide-set wheels in the back also took up space where the cargo should have been. Also, the ultrasonic sensor was rotating about an axis, which did not make for precise wall recognition, prompting a rework of the GEARS.

As the first turning mechanism did not work, a different solution to turning was brainstormed, once again based on a prior solution. This time, the solution considered was how cars use differential drive to turn, and it was adapted in the design to be two-wheel to be able to point turn. However, it lacked stability, so a swivel wheel was added in the back to aid with turning and to bear a load. This solved the problem of point turning, and the ultrasonic sensor configuration was changed to have one on each of the front, left, and right sides. There was also room for the IR and IMU sensors, allowing for hazard detection. However, the swivel wheel once again took up space where the cargo mechanism should have gone, so some minor changes were made to fix this.

The final design replaced the swivel wheel with a skid plate, freeing up room for a cargo holding mechanism. However, it could not support the weight of the cargo during POCs. This change fixed the final few problems encountered, and most of the issues with hazard detection and mapping were sorted out by this time. The final design had ultrasonic sensors on the front, left, and right sides, and IMU sensor in the front, and a gyroscopic sensor on the front left. Some of its unique features were its very sturdy chassis and its compact size, as well as the cargo deposition mechanism. The chassis was unique because it prevented the robot from sagging due to the weight of the battery pack or the motors, and it opened up a lot more places to connect pieces than just on the pi case itself. The compact size allowed it to point turn in corners without hitting the wall, and it made the robot lighter overall so it could move a bit faster. The cargo deposition mechanism consisted of a ramp on a motor-controlled axle that would rotate downward to allow the cargo container to slide out. It was elected to adopt this as the final design based on its performance during testing in class and in office hours.

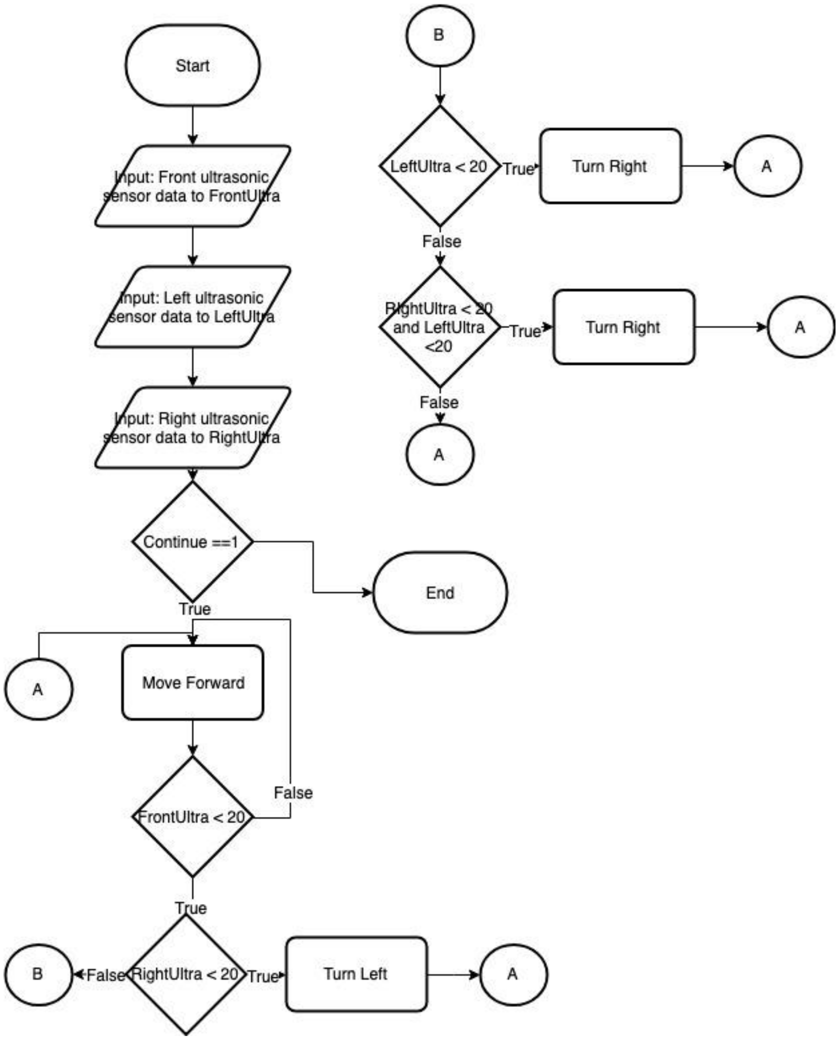
The cargo container also underwent its own design process. It was decided that a simple cube of sufficient size to hold the cargo would be suitable for our GEARS, so a cubic box with a slide-out lid was designed in Autodesk Fusion 360 and then 3D-printed. At POC 2, it was confirmed that the cargo container could indeed hold all of the cargo safely without dropping it and was of sufficient size, but the only issue was that it did not have some sort of handle to easily remove the lid. Some modifications were made to the file in Fusion 360, and the new design was 3D-printed. The final cargo container design was 2 inches by 2 inches by 2 inches with a slide-out lid with a small curved handle for easy removal.

Simultaneously, the software side of the GEARS was being worked on, and an algorithm for following walls and turning based on them was designed. It was decided based on previously known maze-following algorithms that the right-hand rule, or the idea of prioritizing right turns whenever possible, would be the easiest to understand, predict, and program. Essentially, the robot would turn right whenever going straight ahead was not possible, and when it entered a dead end or encountered a hazard, it would turn 180 degrees and turn right again. Otherwise, moving straight would have the highest priority, followed by turning right, then finally turning left. This was done using the LEGO gyroscope. The LEGO gyroscope returned degree values, allowing for easy integration. The turning mechanism consisted of first turning until the error reached a certain point, which was still used in left turn design. The final design used for right turns and 180 degree turns used proportional control, which would slow the robot down after a certain amount of degrees turned until a certain speed threshold was released, which was determined to be 1/9 the set speed, to overcome friction in the paper. The turning working within 1.5 degrees of error, which was factored into the functions, correcting to a near perfect 90 degrees.

The next important part of the GEARS robot was tracking the turning. The robot was required to do different numbers of 90 degree turns within the maze, which meant the robot needed to turn accurately almost every time. The first method that was tried was using the IMU gyroscope to track the turning of the robot. The gyroscope outputted data in degrees/second based on which ever axis the sensor turned about, of which the z axis was the desired axis for data analysis of the three. The gyroscope was subjected to a filter in order to help calibrate the values and then tested by twisting the gyroscope different directs, testing both the angular velocity, figure 7, and integrating the change in angle values. However, like the accelerometer, the IMU's gyroscope experienced an array of issues, including drift, with respect to tracking the turning of the GEARS robot. The values were very inconsistent and sporadic, making it necessary for the rotation data to come from the LEGO gyroscope.

In terms of actually sensing these walls, the ultrasonic sensors on each side were given certain bounds to never exceed. For example, if the front sensor detected something 15 cm away from it and the right sensor detected something 10 cm away from it but the left sensor detected nothing, then the robot would turn left, as the front and right sensors would have met or exceeded the bound of 15 cm, while the left sensor did not. By following this algorithm, the GEARS would complete the maze, and when all three sensors saw nothing at the exit of the maze, the motor controlling the cargo deposition mechanism would rotate the ramp and allow the cargo container to slide out in the correct orientation. A flowchart of the algorithm is shown below:

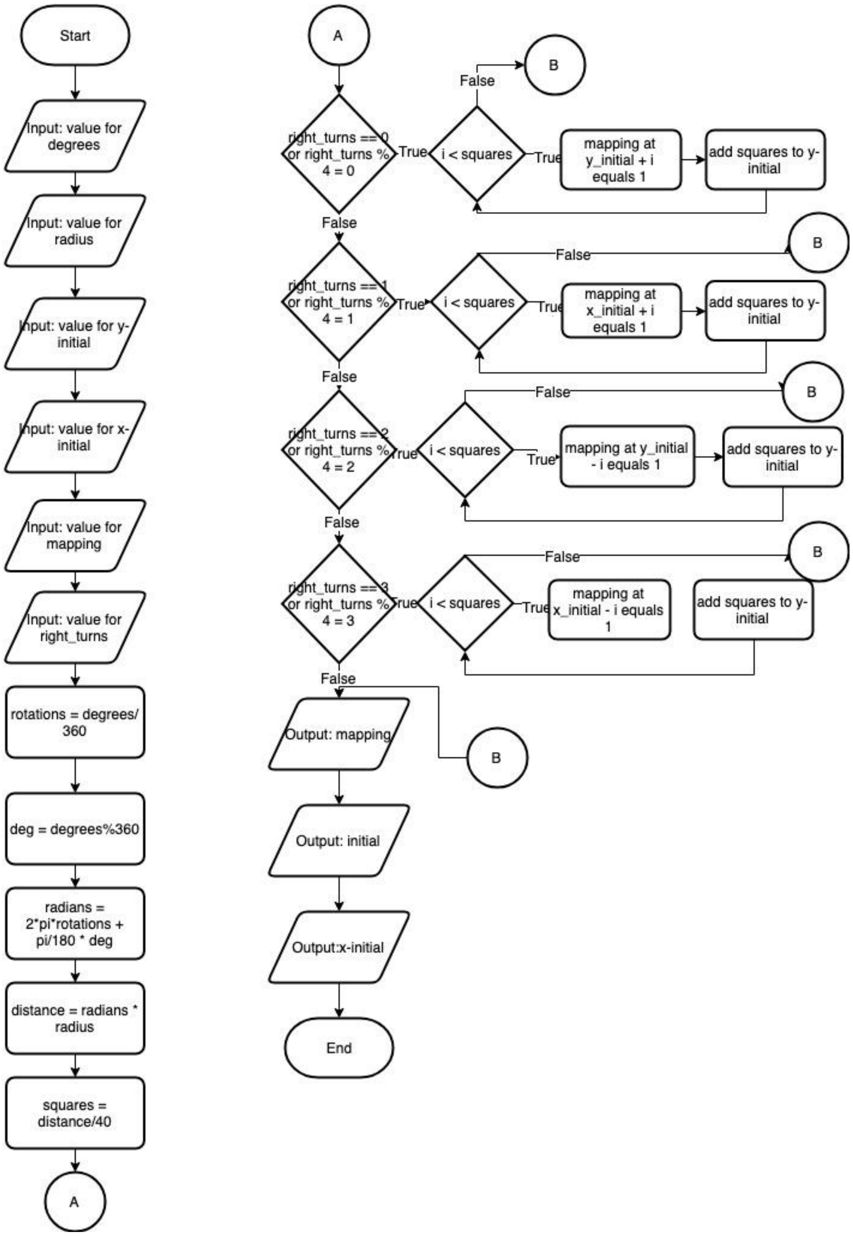
Figure 2: Flowchart for Maneuvering



When dealing with the mapping aspect of the GEARS, determining a way to track the movement of the robot was the first priority. The first method that was tried was the use of the accelerometer within the IMU. The accelerometer outputs data in g-force, meaning the data needs to be converted into m/s^2 to be useful in tracking distances. The accelerometer data was collected and filtered by movement about different axes, calibrating it to be useful. Figure 4 displays the results of data collection, moving the IMU back and forth slowly initially, and then rapidly increasing the speed. The data is very sporadic in nature, even with the filters in place. Integrating this data should have given the linear velocity and position, which should have been able to track the robot's total displacement. However, when using the filtered acceleration data, the velocity and position data were very inaccurate. The velocity data would consistently read zero or values between 1 and -1, and the change in position data would only read zero, as demonstrated in figures 5 and 6, respectively. Due to this, statistical analysis was neither possible nor necessary, as the data demonstrated the improper functionality of the accelerometer. This demonstrates the general unreliability of the IMU accelerometer, leading to the usage of the motor encoders instead.

The motor encoders return a value of a total change in degrees, based off an internal origin point within the motor. To account for this, the overall change in angle was used to calculate the distance traveled in a set amount of time. The mapping commands and sequences were organized into a single function that was calling into the overall running function, enabling the ability to call multiple times. The initial idea was create a two-dimensional array that kept track of the direction and distance. For distance, the function would take into account the radius of the wheel and the overall change in angle and use these to calculate the total linear distance for that stretch of road. For direction, it would take into account the total number of right turns to determine whether to add or subtract to the y or x direction. Each ‘square’ within the array represented a 40 cm area, which is taken into account by dividing the distance by 40. Using conditionals, it would apply changes to the map based on the direction and distance as well as the current location of the robot, as seen below:

Figure 3: Flowchart for Mapping



Lastly, designing the software for the hazard avoidance algorithms was based on the idea of working them into the overall function for maneuvering the maze. With regards to the magnetic data, the IMU was used to detect the change in magnetic readings. Ideally, the GEARS system would stay approximately 5 inches away from the hazard to avoid taking damage; therefore getting a magnetic reading for 5 inches would be one of the conditionals present. The IMU read a value of 518 uTeslas at 5 inches. Using this data, a conditional for values greater than this reading would cause the robot to turn 180 degrees away from the hazard. Avoiding the infrared beacon proved to be more difficult. An added conditional was used at this point for reading both the left and right sensors. Testing the infrared sensors gave a range of values from 155 - 210. Using the value 155 as the lower bound in the condition was decided based on the idea that the mapping was based on 40 cm increments, so a difference of 10 cm at max would not affect the mapping, and being farther away would work better than running the risk of being closer to the hazard.

Overall, the design process used to achieve the final hardware and software design, as well as the cargo container and deposition mechanism, worked well and resulted in a fairly robust GEARS with a logical mazing-following and mapping algorithm.

Figure 4: Linear Acceleration vs Number of Trials

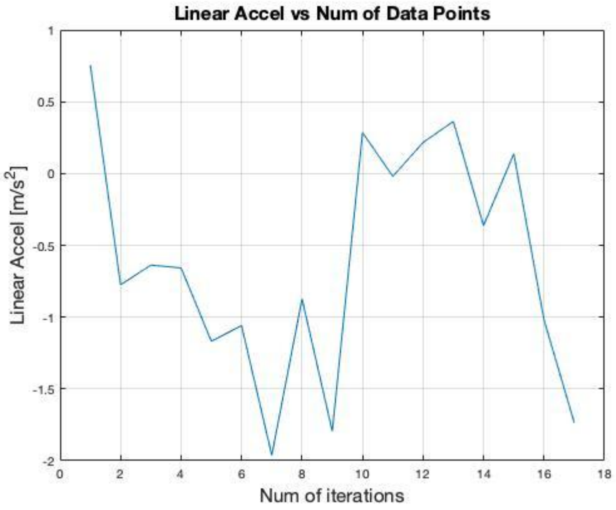


Figure 5: Linear Velocity vs Number of Trials

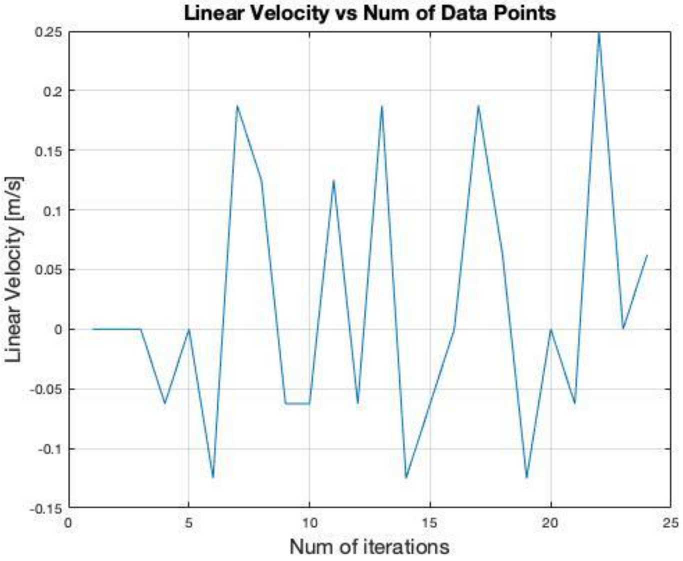


Figure 6: Position vs Number of Trials

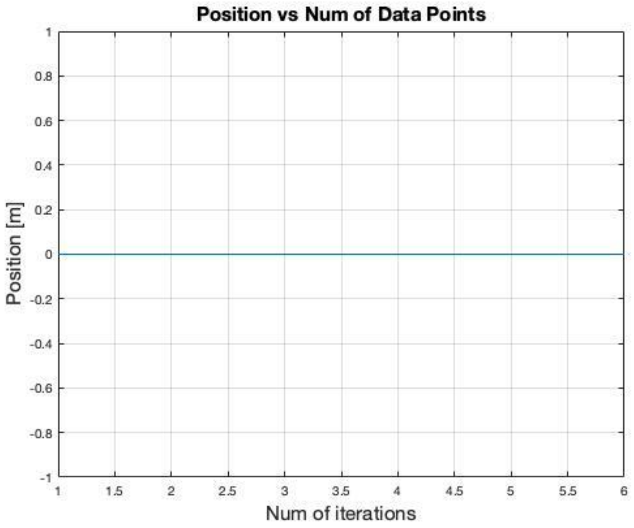
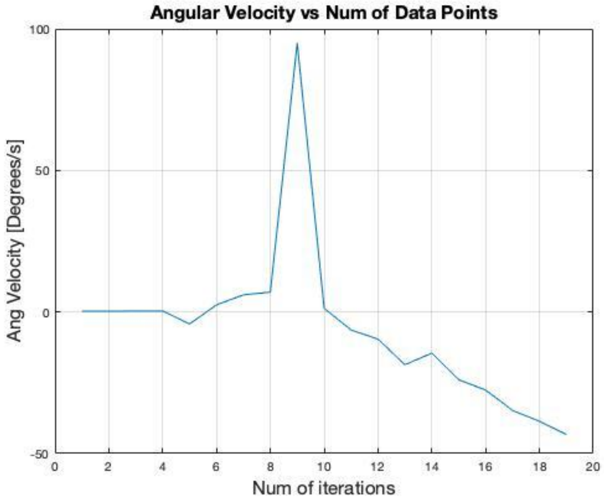


Figure 7: Angular Velocity vs Number of Trials



Data Analysis

When developing the SLAM procedure for the GEARS robot, different sensor data outputs were used to help generate the most accurate map possible with the limitations present in time, hardware, and knowledge of the tools given. The primary sensors used to develop the layout of the map and allow for the successful include the motor encoders, the LEGO and GrovePi ultrasonic sensors, the LEGO gyroscope, the GrovePi infrared sensor, and the IMU for magnet sensing capabilities. Each of these sensors served an integral role in allowing the system to work in an optimal way.

The LEGO motor encoders are sensors located within the LEGO motors that keep track of total change in the angle of the rotational part of the motor based off an initial position. Using the overall change in angle, the overall displacement of the robot could be determined by multiplying the radius of the wheels, which was approximately 3.75 cm, attached to the motors with the total change in angle to get the linear distance of the GEARS robot traveled in a specific hallway or desired timeframe. Calibrating the motor encoder was a series of trial and error. Using a ruler, the total distance the robot traveled would remain constant while manipulating the data to fit the constant. The encoder data was converted from degrees to radians and multiplied by different values ranging from 2-4.5 as seen in figure 8. While the radius of the wheel was approximately 3.75 cm, multiplying the radians by 2.75 would get an accurate representation of the distance. This could have been due to the error or unknown internal calculations associated with the encoders. For example, when the GEARS would go 30 cm, it would read a value of 31 cm, showing its relative accuracy with little systematic error. Using the data acquired statistically and experimentally, the overall change in displacement for a specific stretch of a linear path was determined and applied to map of the area. The robot essentially took these displacements and applied it to specific parts of the map based on which direction it went, determined using a counter keeping track of the total number of right turns and two overall x position and y position counters used to keep track of where the robot was in relation to the origin.

The LEGO gyroscope outputs the overall change in angle of the sensor using filters in place built into the sensor. Using this overall change in angle, the GEARS made accurate 90 degree turns in order to navigate the maze without hitting walls. One strength of the LEGO gyroscope was that the data outputted was automatically integrated and filtered giving angle values that were fairly accurate in nature, outputting data with a ratio of approximately 1:1 with respect to calculated degrees and actual degrees. When calibrating the data, a hypothesis test was performed with an alpha value of 5%. Unfortunately, the results, shown in figure 11, were that null hypothesis of turning an average of 90 degrees was rejected. Going back, this was most likely due to the systematic error present within the gyroscope outputs. This error was consistently about 1.5 degrees, so the target goal was set to 1.5 degrees less than the intended goal, which improved the results greatly, as demonstrated in figure 16, and resulted in more accurate turns. The LEGO gyro was also offset from the axis of rotation, which could partially explain the systematic error. In all, the sensor proved to be invaluable when dealing with turning the GEARS.

In order to detect and map the magnetic hazards around the maze, the IMU's magnet sensor was used. This sensor detects the intensity of magnetic fields in different directions and orientations in relation to the directions associated with the IMU in units of microTeslas. Each orientation in relation to the IMU gave dramatically different results due to the nature of the magnetic fields. When experimentally testing the sensor, the sensor was placed at different distances from the magnet to test the intensity change to discover a trend. The z and y orientations of the IMU read fairly consistent readings, with decreasing intensity follow a natural decreasing model as seen in figures 10 and 11, while the x orientation is more random and sporadic, as seen in figure 9. Because the z orientation demonstrated the most consistent behavior with varying distances, its data was analyzed and modeled for the robot. Using linearization and the method of least squares, an exponential model of the z data was developed, shown in Equation 1.

Equation 1: Exponential Model

$$intensity = 3393.8e^{-0.37266(distance)}$$

Using this equation, working backwards to determine the distance at different intensities was possible, allowing avoidance up to a certain distance. In the case of the GEARS, getting an intensity that is greater than 518 uTeslas, corresponding to 5 inches, meant that GEARS needed to turn away from the magnet. This would allow mapping of the hazard to a relatively precise positioning, contributing to the SLAM procedure. However, this method had uncertainties with relation to orientation of the robot, as different angles could possibly throw off the data. In all, this sensor proved to be very useful for the desired task despite its limitations.

In order to detect and map the infrared hazard present in the maze, the IR sensor was implemented. This sensor outputs different values in Hertz based on the intensity of the infrared source and the distance from it. Two different detecting sensors on the hardware allowed for comparison. Testing these sensors was very similar to how we tested the magnetic sensor, where the distance was varied between data collections. The primary difference had to do with the angle associated with the sensors. The more angled toward the source, the higher the intensity values. Analyzing the data was difficult, as the sensor were very sensitive and sporadic. Therefore, calibrating the sensors came down to creating a range of acceptable values based on the graphed data (figure 13). Based on the graph, each sensor demonstrated a range of approximately 155 - 285, as this was the values read by each sensor at 5 inches. Therefore, it was reasonable to assume that avoiding any intensity above 155 would ensure the safety of the robot, even if the distance was greater than 5 inches. Mapping the location of the hazard was based on the range of values, as the max distance based on the data associated with this data is 10 inches. Since the mapping method accounted for 40 cm increments, this had little effect on the map as a whole, and therefore was assumed to be within 40 cm. The sensor gave enough data with enough precision to perform the task effectively.

Lastly, in order to be able to navigate the maze effectively, the usage of the LEGO and GrovePi ultrasonic sensors was deemed necessary. These sensors emit a series of sound waves which bounce off the wall or obstacle and return to the sensor. Based off the intensity of the sound wave and the time it takes to return, it outputs a distance value in centimeters. In order to detect the different walls on different sides of the robot, three different ultrasonic sensors were used: two GrovePi ultrasonic sensors on the left and right and a LEGO ultrasonic on the front. These sensors were used when making the majority of the turning decisions, using an algorithm that would tell the robot to turn one way or another if it did not detect walls within 25 cm of the sensors. These sensors were also used to correct the movement of GEARS, moving slightly one way or another based on whether one of the side sensors read values larger than the other. Calibrating this sensor was very straightforward, as the sensors were very accurate. Testing involved moving an obstacle specific distances away from the sensors and comparing the ultrasonic sensor readings with these distances. This led to the creation of Equation 2:

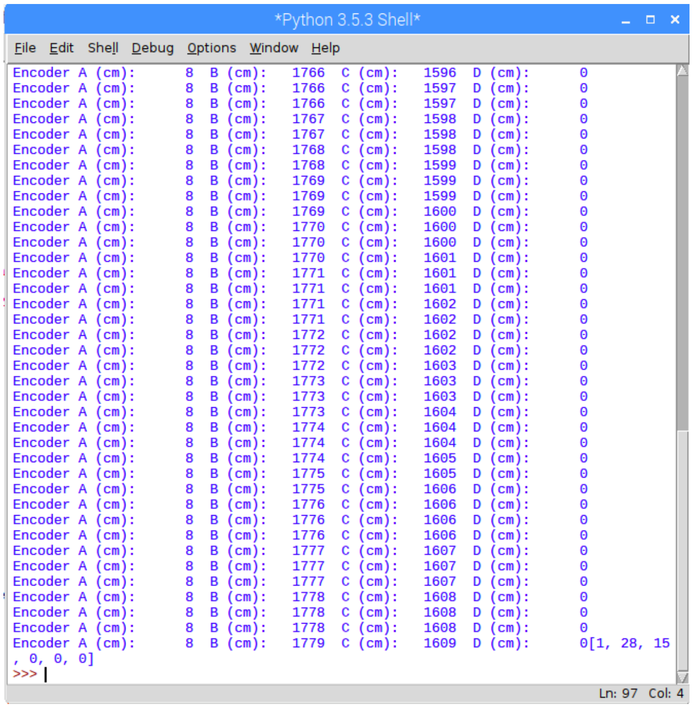
Equation 2: Linear Model

$$Distance\ away = 0.99(distance\ reading) + 0.02$$

This equation was developed using the method of least squares and demonstrates the accuracies of these sensors, with the error being +-0.1 on the LEGO ultrasonic sensor and at most 1 on the GrovePi ultrasonic (as these output integers instead of decimals), meaning it is an approximate 1:1 ratio of reading to reality. The ultrasonics were set to detect whether the it returned values greater than approximately 25 cm in order to make decisions with regards to turning, as this would mean there was no wall close. The same process was used for correction, except with a bound of 15 cm. The only real uncertainties were when the sensors hit corners or angled walls, as the sound waves scattered in undesirable ways. This demonstrates the overall usefulness and reliability of these sensors for their intended tasks.

In summary, through calibration and analysis, the five sensors enable the GEARS system to enact a SLAM system used to track the robots motion and travel. These sensors proved to be invaluable and, upon further testing, could help GEARS be an effective aid in disaster zone relief.

__Figure 8: Encoder Data__



```
Encoder A (cm):      8  B (cm):   1766  C (cm):   1596  D (cm):      0
Encoder A (cm):      8  B (cm):   1766  C (cm):   1597  D (cm):      0
Encoder A (cm):      8  B (cm):   1766  C (cm):   1597  D (cm):      0
Encoder A (cm):      8  B (cm):   1767  C (cm):   1598  D (cm):      0
Encoder A (cm):      8  B (cm):   1767  C (cm):   1598  D (cm):      0
Encoder A (cm):      8  B (cm):   1768  C (cm):   1598  D (cm):      0
Encoder A (cm):      8  B (cm):   1768  C (cm):   1599  D (cm):      0
Encoder A (cm):      8  B (cm):   1769  C (cm):   1599  D (cm):      0
Encoder A (cm):      8  B (cm):   1769  C (cm):   1599  D (cm):      0
Encoder A (cm):      8  B (cm):   1769  C (cm):   1600  D (cm):      0
Encoder A (cm):      8  B (cm):   1770  C (cm):   1600  D (cm):      0
Encoder A (cm):      8  B (cm):   1770  C (cm):   1600  D (cm):      0
Encoder A (cm):      8  B (cm):   1770  C (cm):   1601  D (cm):      0
Encoder A (cm):      8  B (cm):   1771  C (cm):   1601  D (cm):      0
Encoder A (cm):      8  B (cm):   1771  C (cm):   1601  D (cm):      0
Encoder A (cm):      8  B (cm):   1771  C (cm):   1602  D (cm):      0
Encoder A (cm):      8  B (cm):   1771  C (cm):   1602  D (cm):      0
Encoder A (cm):      8  B (cm):   1772  C (cm):   1602  D (cm):      0
Encoder A (cm):      8  B (cm):   1772  C (cm):   1602  D (cm):      0
Encoder A (cm):      8  B (cm):   1772  C (cm):   1603  D (cm):      0
Encoder A (cm):      8  B (cm):   1773  C (cm):   1603  D (cm):      0
Encoder A (cm):      8  B (cm):   1773  C (cm):   1603  D (cm):      0
Encoder A (cm):      8  B (cm):   1773  C (cm):   1604  D (cm):      0
Encoder A (cm):      8  B (cm):   1774  C (cm):   1604  D (cm):      0
Encoder A (cm):      8  B (cm):   1774  C (cm):   1604  D (cm):      0
Encoder A (cm):      8  B (cm):   1774  C (cm):   1605  D (cm):      0
Encoder A (cm):      8  B (cm):   1775  C (cm):   1605  D (cm):      0
Encoder A (cm):      8  B (cm):   1775  C (cm):   1606  D (cm):      0
Encoder A (cm):      8  B (cm):   1776  C (cm):   1606  D (cm):      0
Encoder A (cm):      8  B (cm):   1776  C (cm):   1606  D (cm):      0
Encoder A (cm):      8  B (cm):   1776  C (cm):   1606  D (cm):      0
Encoder A (cm):      8  B (cm):   1777  C (cm):   1607  D (cm):      0
Encoder A (cm):      8  B (cm):   1777  C (cm):   1607  D (cm):      0
Encoder A (cm):      8  B (cm):   1777  C (cm):   1607  D (cm):      0
Encoder A (cm):      8  B (cm):   1778  C (cm):   1608  D (cm):      0
Encoder A (cm):      8  B (cm):   1778  C (cm):   1608  D (cm):      0
Encoder A (cm):      8  B (cm):   1778  C (cm):   1608  D (cm):      0
Encoder A (cm):      8  B (cm):   1779  C (cm):   1609  D (cm):      0[1, 28, 15
, 0, 0, 0]
>>> 
```

__Figure 9: Magnet Sensor X-Axis__

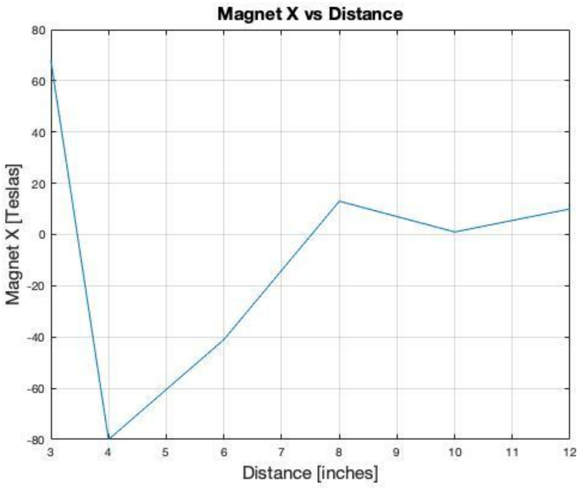


Figure 10: Magnet Sensor Y-Axis

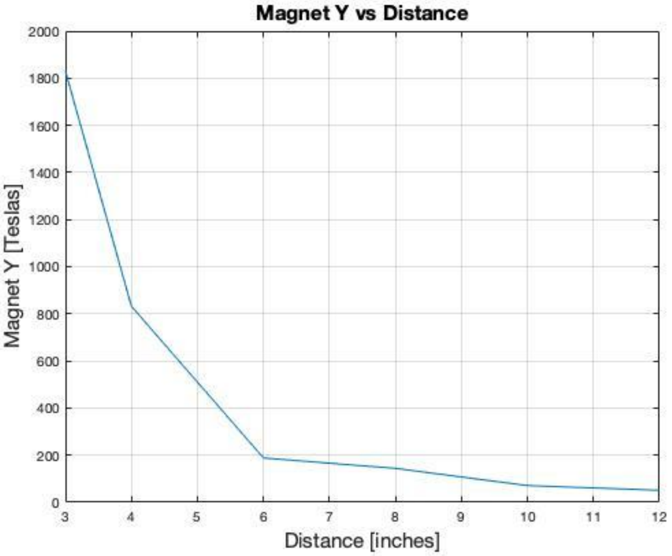


Figure 11: Magnet Sensor Z-Axis

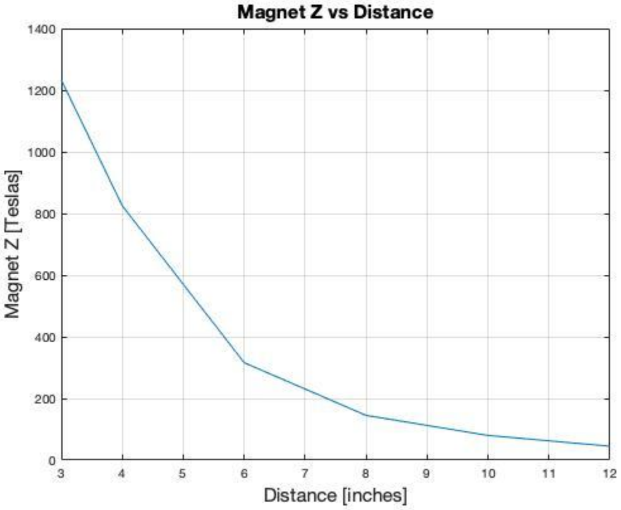


Figure 12: Exponential Model Graph Z-Axis

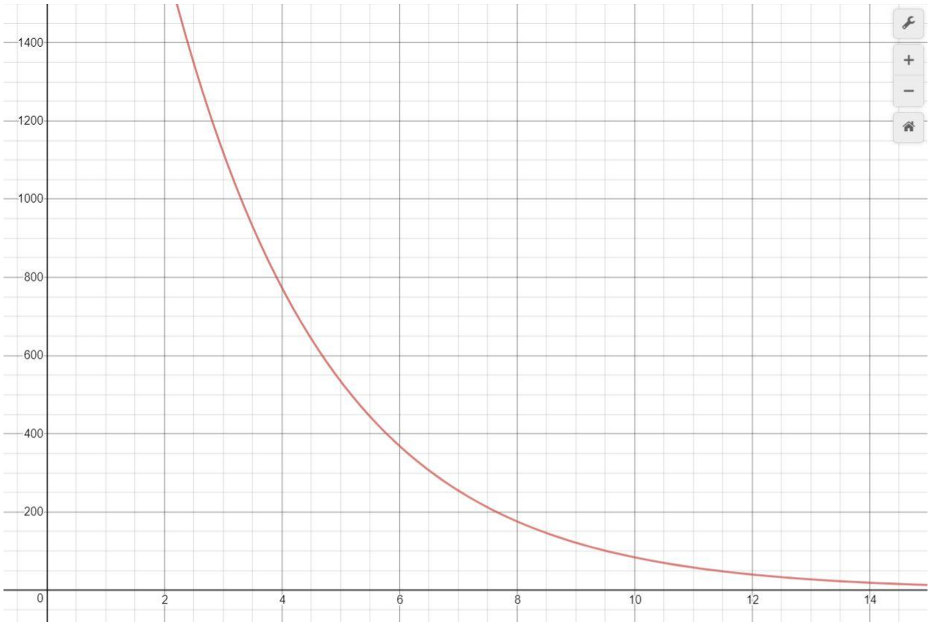


Figure 13: IR Data Graph

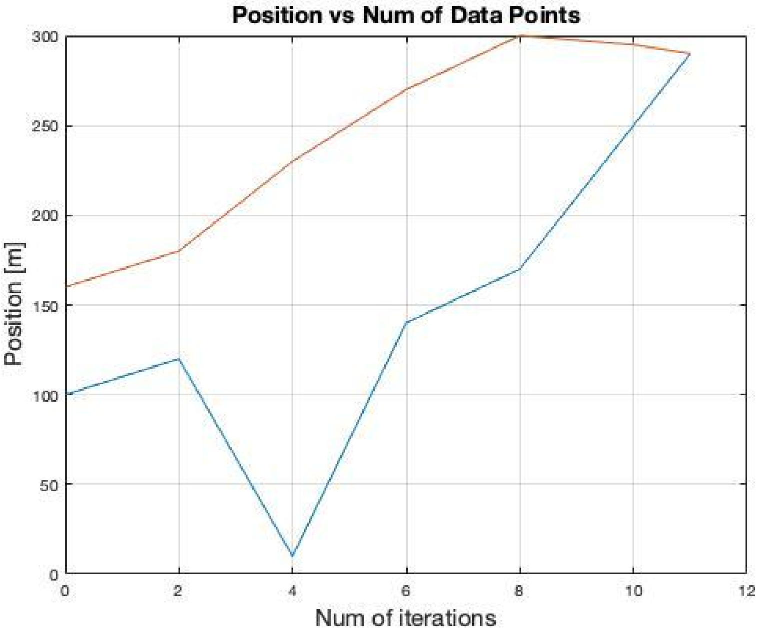
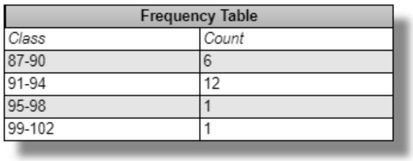
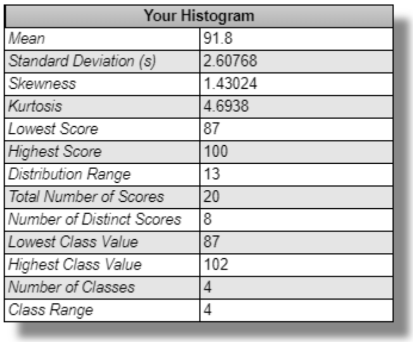


Figure 14: Hypothesis Test for Right Turning:



Frequency Table	
Class	Count
87-90	6
91-94	12
95-98	1
99-102	1



Your Histogram	
Mean	91.8
Standard Deviation (s)	2.60768
Skewness	1.43024
Kurtosis	4.6938
Lowest Score	87
Highest Score	100
Distribution Range	13
Total Number of Scores	20
Number of Distinct Scores	8
Lowest Class Value	87
Highest Class Value	102
Number of Classes	4
Class Range	4

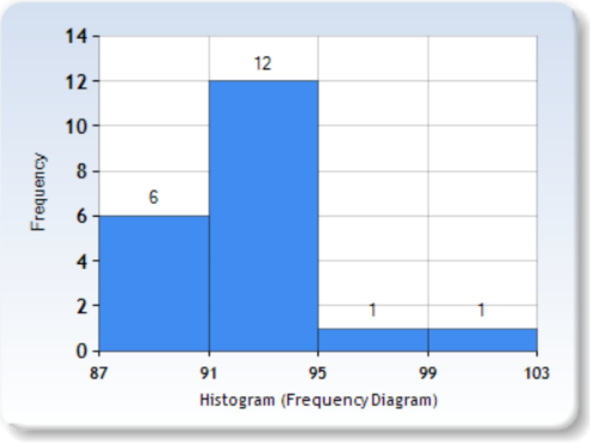
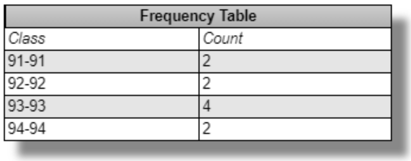
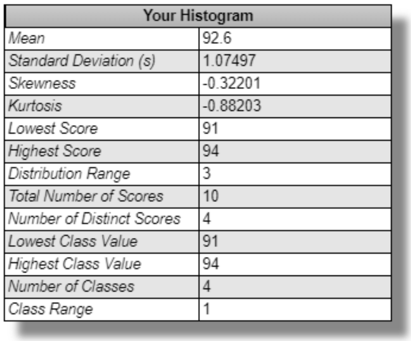


Figure 15: Hypothesis Test for Left Turning



Frequency Table	
Class	Count
91-91	2
92-92	2
93-93	4
94-94	2



Your Histogram	
Mean	92.6
Standard Deviation (s)	1.07497
Skewness	-0.32201
Kurtosis	-0.88203
Lowest Score	91
Highest Score	94
Distribution Range	3
Total Number of Scores	10
Number of Distinct Scores	4
Lowest Class Value	91
Highest Class Value	94
Number of Classes	4
Class Range	1

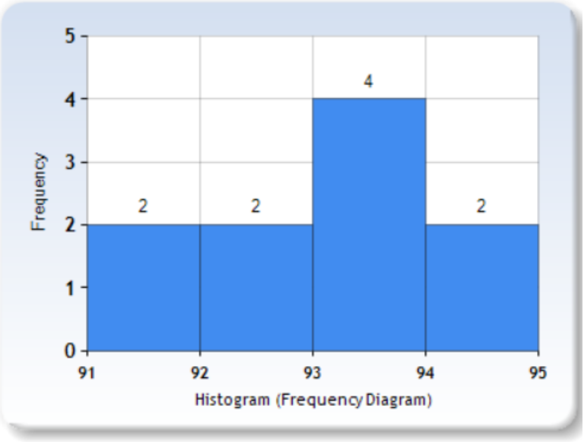
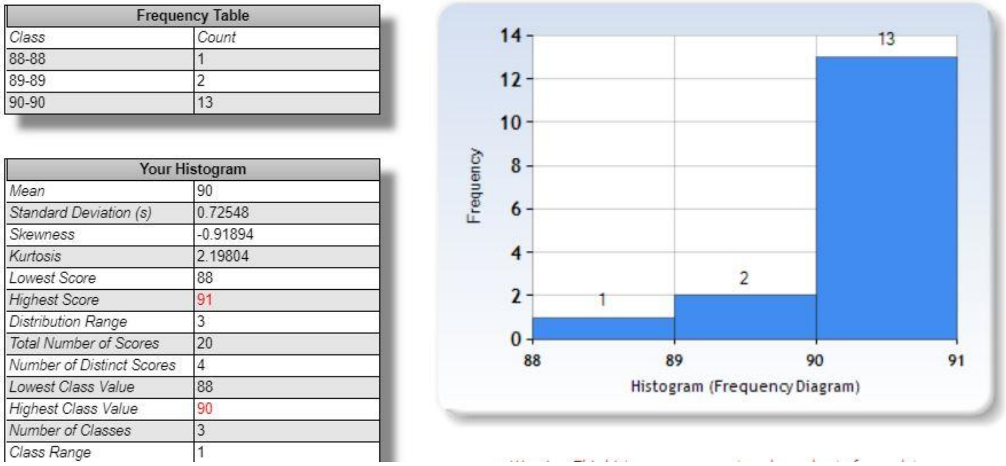


Figure 16: Hypothesis Test for Secondary Turning Data in Both Directions



Frequency Table	
Class	*Count*
88-88	1
89-89	2
90-90	13

Your Histogram	
Mean	90
Standard Deviation (s)	0.72548
Skewness	-0.91894
Kurtosis	2.19804
Lowest Score	88
Highest Score	91
Distribution Range	3
Total Number of Scores	20
Number of Distinct Scores	4
Lowest Class Value	88
Highest Class Value	90
Number of Classes	3
Class Range	1

Cultural and Ethical Considerations

There were many variables that had to be taken into account when designing the cargo labels, namely what symbols would best represent the items and what angle should be taken to understand the strength of the symbols.

The first symbol that needed designing was the one for food and water. It was believed that this would be the most easily recognizable symbol on a universal level. For the initial design, a water droplet and an apple were placed on a black background (Figure 17). When surveyed, 38 of 54 respondents replied that the symbol represented food and water or something similar, 6 of 54 responded with something related to agriculture, and 10 of 54 responded with some other type of response. Some feedback provided included the idea to add a food that was slightly more universal; the apple is a common fruit to North America, but in other areas may not be as common. Another slight adjustment was the recommendation to reorder the the symbols on the design because there were some respondents who were seeing the apple and then the water and labeling the design as apple juice. This led to the final design that featured the apple and water droplet with the added image of grain and a new order (Figure 18).

Figure 17: Initial Graphic for Food and Water

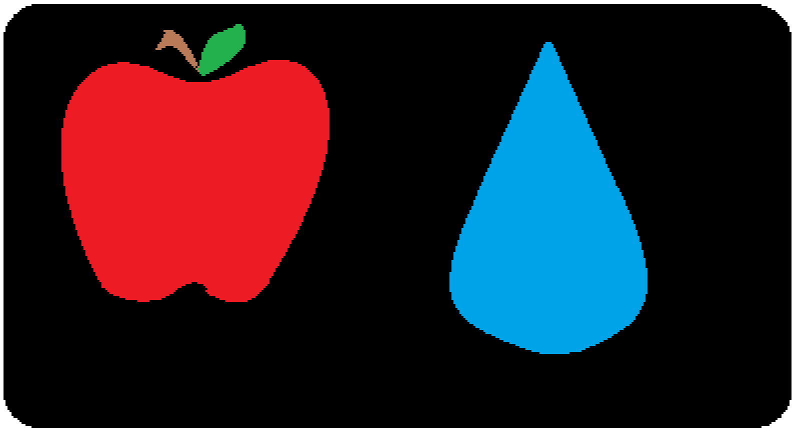
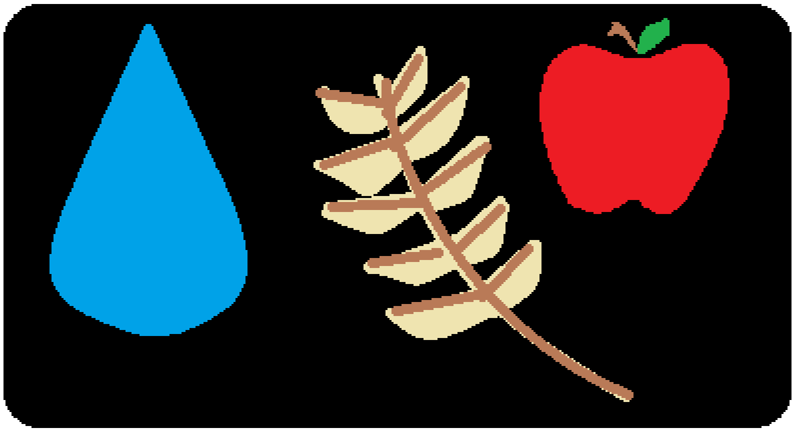


Figure 18: Final Graphic for Food and Water

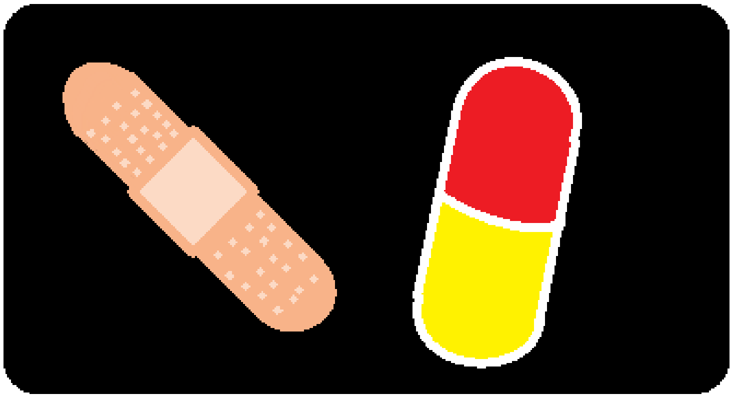


The second symbol that needed to be designed was the one for fuel. It was known that this would be a much more difficult symbol to represent. While gasoline and oil are common ideas in the industrialized world, they may not be common in other developing countries in which the GEARS may respond to a disaster. For the initial design, a gasoline can featuring a symbol of fire, and a log of wood placed on a black background were used (Figure 19). This symbol returned results showing that 17 of 54 respondents believed that the image represented fuel, 23 of 54 responded with fire and wood, 5 of 54 responded with flammable materials, and 9 of 54 responded with some other response. While this data is not the most admirable, it was decided that because fuel, fire and wood, and flammable materials had fairly the same connotation, this data was overall acceptable. No major changes were made to the final design of the fuel symbol.

Figure 19: Final Graphic for Fuel

The next symbol was the design for first aid. There was much discussion about what would represent this well; included in this discussion were a syringe, a drop of blood, and the Caduceus. These images were ultimately rejected for the slightly threatening connotation associated with them. This led to the choice of an adhesive bandage and a time release medicine capsule on a black background (Figure 20). When surveyed, 13 of 54 respondents replied that the images represented first aid, 23 responded with medicine/pills/bandages, 7 responded with injury, and 11 responded with another response. These results were fairly acceptable being that 43 had some resembling of something medical which would ultimately lead to the conclusion that the contents were medical supplies of some sort. The final design featured no alterations following the results of the survey.

Figure 20: Final Graphic for First Aid



The final design to be decided was the emergency shelter. This symbol was thought to be a very easy one to create a representation for, but that was later discovered to be a disadvantage. The initial conception featured the image of a standard home on a black background (Figure 21). When surveyed, 7 of 54 respondents replied that the image represented a shelter, 42 of 54 responded with house or home, and 5 responded with some other reply. While these results sound acceptable, it is troubling that 42 of 54 responded with house and home because that is what is directly represented. To fit a much larger cultural base, a much more general image needed to be added to the symbol. This led to the addition of a tent symbol to the graphic to aid in the overall representation of the design (Figure 22).

Figure 21: Initial Graphic for Emergency Shelter

Figure 22: Final Graphic for Emergency Shelter

For the overall strength of these symbols, two major events were used to help put the images into perspective: Japan following WWII and Puerto Rico following Hurricane Maria. The first way in which these were both looked at is the plausibility of the GEARS being deployed to these scenarios if it were around at the time. In both cases, it would be very plausible. Both areas were in massive distress following the man-made or natural disaster that occured and left the citizens vying for aid from foreign allies. The deployment of the GEARS would have allowed for a safe way to deliver supplies through the hazardous zones that took incredible lengths of time for human groups to be able to get through. This helped lead to using these as a strong analysis source for the symbols. The symbols were very much based on developed industrialized civilizations, and in both cases would have been something distinctly recognizable to the civilians in distress. This research also helped provide some perspective on how ethical the interactions may be. When the United States occupied Japan, it claimed to be focusing on maintaining the Japanese culture. Whether that occured remains to be seen, but this posed the question about whether or not the symbols would be pushing Western ideas and culture onto any civilization they may be helping. The answer is simply that, while they are more easily recognizable to those civilizations, they will not be of large enough focus to necessarily infringe on the systematic and historical culture a civilization has.

All of this led to the conclusion that the symbols will do an acceptable job. But what about the extreme case? What about the isolated Cargo Cult that has no knowledge of the outside world and may worship these symbols? While these symbols, possibly excluding the food and water design, may not be tailored to the needs of such a civilization, it would be rather hard to account for a scenario where this may be the case; not only from the perspective of when the GEARS would be deployed to this civilization, but also how their ideas might be taken into account. There is no clear answer as a way to successfully integrate their needs into the overall design. That being said, the design of the symbols would most likely be able to serve the greatest common denominator, meaning that they a success.

The design of the GEARS itself is one that poses more questions on the overall ethical strength. The design of the GEARS was focused more on ability rather than approachability. There are no major aspects of the design that promote a friendly and welcoming stance, but there are also no major aspects that counter this idea. The design in turn is rather neutral in how it may be viewed by the civilization.

Once the cargo was released, the deposition mechanism would ‘wag’ back and forth to signal to the recipients that the cargo had been released. This mechanism is rather small in size and is well off the ground. It was the overall goal that this would indicate a peaceful delivery of aid to the civilization. This was never able to be demonstrated in the trials as the GEARS never successfully reached that point.

This all leads to the final question: Is it ethical to use the GEARS for aid? The answer to this is quite unclear. When the aid is in an industrialized country or a developing country that has contact with the developed world, the answer is most likely yes. Aid is being provided to those who need it as is often done by countries or large entities like the United Nations. In the cases of Puerto Rico and Japan, the answer would quite clearly be yes as well. Both major disasters that required aid from others that occured in industrialized countries. The only time that it may not be ethical is when politics would be in play. When the governments do not request aid, it would not be ethical to deploy the GEARS because it would most likely be violating the wishes of the civilization in question. While the civilians may be in danger because of such, the wishes of the government must be at the forefront of ethical consideration because of the consequences that may occur. If the government does request aid, however, there is no question about the ethics of deploying the GEARS. Simply, if it is the best option, the GEARS must be deployed to help the civilization.

The issue of ethics gets rather unclear once a different scenario is posed: is it ethical to use the GEARS when a disaster occurs in an isolated civilization such as the isolated Nambikwara tribe of the Amazon? If this were the case, it would most likely not be ethical. This is a civilization that does not have the luxuries that the industrialized world does. The supplies may be more harmful to the civilization than helpful, and the GEARS may be taken as a threat and damaged by the tribe. The culture they live in would be infringed upon, and this could be detrimental to their future. Similar to what happened with the Nambikwara, most of their population could be wiped out by using these resources and establishing contact with a world they do not know. If something like this were to occur, it would be the fault of those who deployed the GEARS. All of this leads to the conclusion that using the GEARS for disaster aid only when requested by an industrialized or developing country.

Results and Discussion

During testing, GEARS was able to perform some of the desired tasks individually. More specifically, it was able to make 90 degree turns, output and update a map of the terrain, detect hazards, and use the cargo container to hold the material being transported. However, the GEARS was unable to perform some tasks successfully due to hardware and software limitations. Specifically, it was unable to take the hazard input and successfully avoid hazards, organize the map data into an organized and coherent csv file for more accessibility, and carry the cargo through the maze due to the instability of the cargo-holder and mass of the cargo. During testing, the GEARS was able to make successful turns and compile the map on limited trials. However, its ability to make multiple turns, detect hazards, and map altogether was uncertain due to time constraints when designing the robot.

With regards to the final demonstration, the robot was able to successfully complete two consecutive 90 degree turns while mapping the total distance that traveled in the correct directions as intended. Specifically, GEARS was able to make one right turn and one left turn, correcting itself as it went. It also mapped both the distance and direction on a 2-D array, taking the form of a map for the robot. It also detected the magnetic sensor right before checkpoint two, which was both good and bad. It was able to detect the magnet, which was desirable, but it turned at the wrong time, which was a negative. The ability of GEARS to make successful 90 degree turns was most likely due to the implementation of PID (primarily P) control into the turning mechanism, allowing for the robot to slow down during turns and reach the most accurate position when turning. The success of the mapping functions was due to the calibration of the motor encoder data, which gave off angular data that was converted to linear distance data. The cargo container was also able to successfully hold the material without spilling in every direction, even though GEARS could not support the cargo and the material inside.

One of the major issues that GEARS experienced during the demonstration was detecting the hazards and reacting appropriately. Before the demonstration, the IR and IMU magnetic sensors were calibrated to determine reading increments for different distances away from the sources. During the demonstration, the robot never encountered the IR hazard, so it never had the opportunity to avoid/map the IR hazard. Conversely, the IMU magnetometer was able to detect the hazard and acted as it was designed. However, the problem arose from the fact that the detection radius, which was approximately 7.5 inches instead of the intended 5 inches, was too large and caused a turn at the wrong moment. This deviation from the test values could have been due to the battery, as during testing, the battery was at a lower charge. Therefore, a method in the code to filter the magnetic and IR sensor readings based on orientation and distance could allowed for more accurate mapping and would fix the issues associated with the large radius.

Another major issue arose from the organization of the mapping data. During the demonstration, after each turn, the robot would output a two-dimensional array with values of 1 substituted in for the distance values. However, this data was not outputted to a csv file, causing confusion and disorganization with regards to the map and its relative accuracy. This also caused confusion when checking to see if the mapping function was able to map the hazards correctly. This was primarily due to time constraints and complexity with compiling the data during testing, as all members of the team had limited experience with manipulating data and therefore took time to learn the usage of the sensors. An easy solution to this problem would be implementing a for loop nested in the mapping function to organize the data into a proper map in a csv file, giving a more visual and easily comprehensible map that could be tested and refined.

In conclusion, GEARS demonstrated exceptional turning capabilities, incremental map updating, and the detection of the magnetic hazard. Some areas GEARS could improve on include organizing the mapping data, refining the hazard avoidance system, and the cargo-holding system. In all, with minor adjustments, GEARS demonstrates the potential to be used for true disaster areas.

Conclusions and Recommendations

In the end, the final design of the GEARS robot was successfully able to make 90 degree turns up until checkpoint one, continually update the map during the course, contain the cargo in such a way as it would not spill, and detect the magnetic hazard upon contact. However, since the GEARS robot did not successfully complete the maze, deposit the cargo, or generate an organized map of the maze, there are some improvements that can be made for future iterations of its design.

With regards to the mechanics, it would be an important adjustment to add a sturdier cargo-holding mechanism so that GEARS could successfully carry the cargo through the maze while experiencing little resistance from the cargo while turning. To do this, placing the cargo near the center of rotation would be optimal. Additionally, creating a holder for the battery, so that there would be little likelihood of the battery falling out during future demonstrations, would be desirable.

With regards to the software, adding an algorithm to organize the two-dimensional array into a csv file would clean up the look of the data, making it more comprehensible, testable, and easily changeable. Adding an extra sensor, such as the Hall effect sensor, would help improve the hazard sensing capabilities, as it would provide more data to compare to the IMU to give more accurate conditionals for avoiding the obstacles and managing the data. Additionally, when avoiding hazards, taking into account the direction and orientation would improve the accuracy of the avoidance system, using more data to improve the overall performance. Taking advantage of the IMU's accelerometer and gyroscope in different capacities could also improve the distance tracking method, relying on angles while using the encoders to give different distances not reliant on exact 90 degree turns.

In all, the aforementioned areas for improvement would allow the GEARS to better carry and deposit the cargo, hold the batteries, organize the mapping data, improve hazard avoidance, and improve tracking capabilities. With these improvements, the GEARS would be even more suited to assisting is disaster situations.

References and Appendices

References

(n.d.). Retrieved from https://history.state.gov/milestones/1945-1952/japan-reconstruction

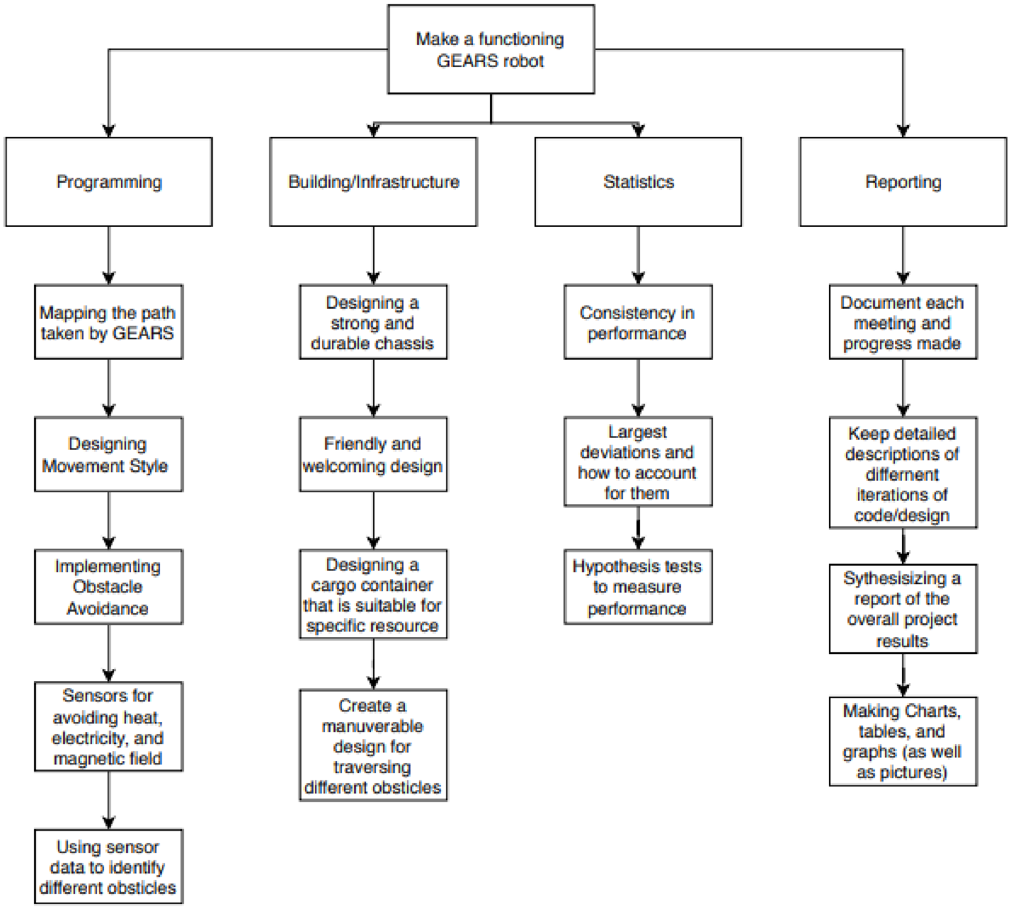
(n.d.). Retrieved from http://afe.easia.columbia.edu/special/japan_1900_occupation.htm

Hurricane Maria Relief. (2019, March 14). Retrieved from https://www.directrelief.org/emergency/hurricane-maria/

Kluger, J. (2015, June 04). Uncontacted Tribes: Should We Leave Them Alone? Retrieved from http://time.com/3909470/tribes-uncontacted-ethics/

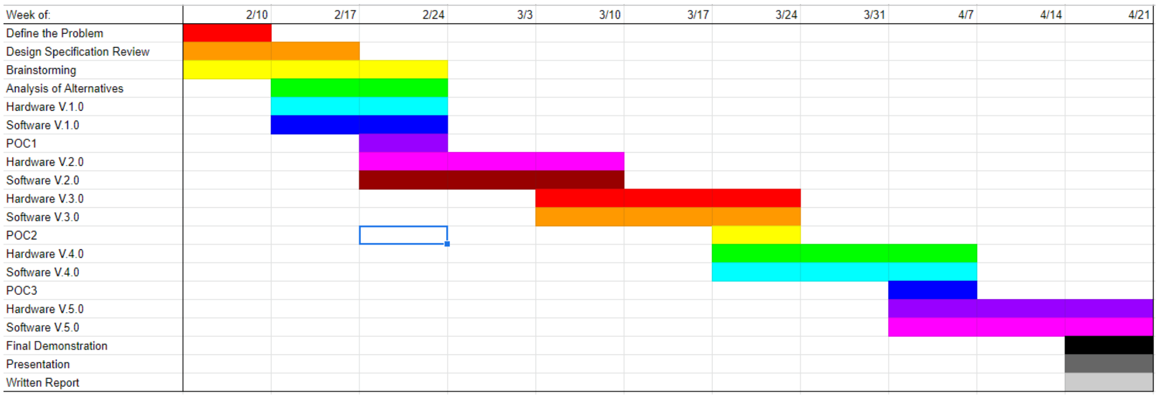
Appendix A: Project Management

Figure 23: Work Breakdown Structure



This work breakdown structure was created to represent the main sections of the project and what needed to be completed. This was used as a way to figure out and assign duties to individual members to create the best possible route to successful completion of the project.

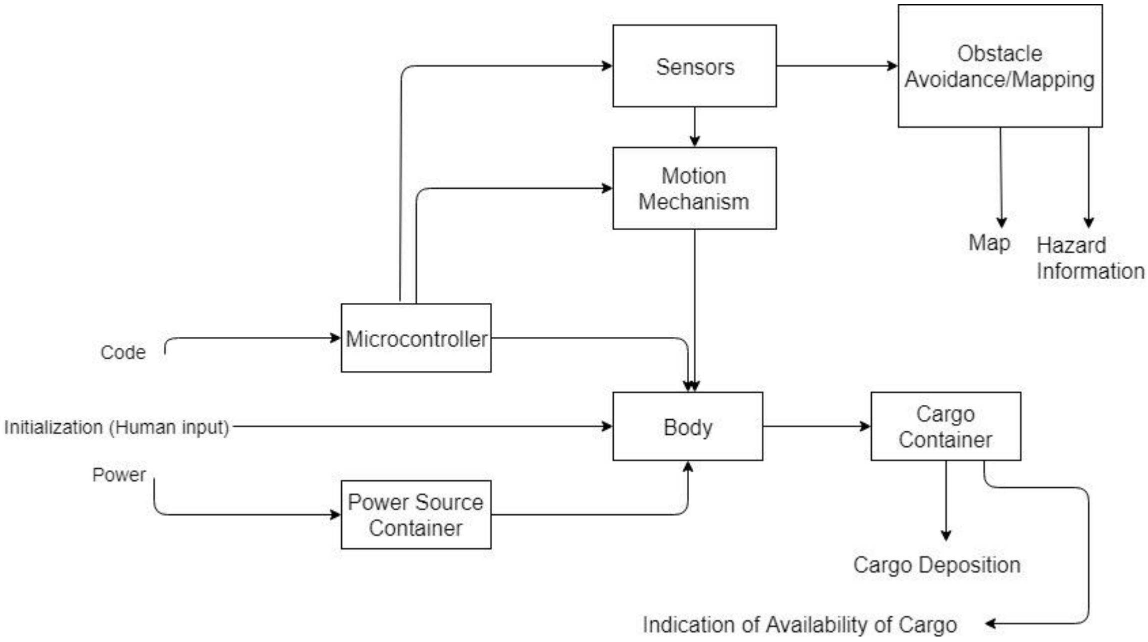
Figure 24: Gantt Chart



Week of:	2/10	2/17	2/24	3/3	3/10	3/17	3/24	3/31	4/7	4/14	4/21
Define the Problem	■										
Design Specification Review	■	■									
Brainstorming	■	■	■								
Analysis of Alternatives		■	■								
Hardware V.1.0		■	■								
Software V.1.0		■	■								
POC1			■								
Hardware V.2.0			■	■	■						
Software V.2.0			■	■	■						
Hardware V.3.0					■	■	■				
Software V.3.0					■	■	■				
POC2							■				
Hardware V.4.0							■	■	■		
Software V.4.0							■	■	■		
POC3									■		
Hardware V.5.0									■	■	■
Software V.5.0									■	■	■
Final Demonstration											■
Presentation											■
Written Report											■

This Gantt chart was created as a rough timeline for the completion of the project. It featured time for up to five iterations for both the hardware side and the software side of the GEARS, as well as times for each POC and the final demo.

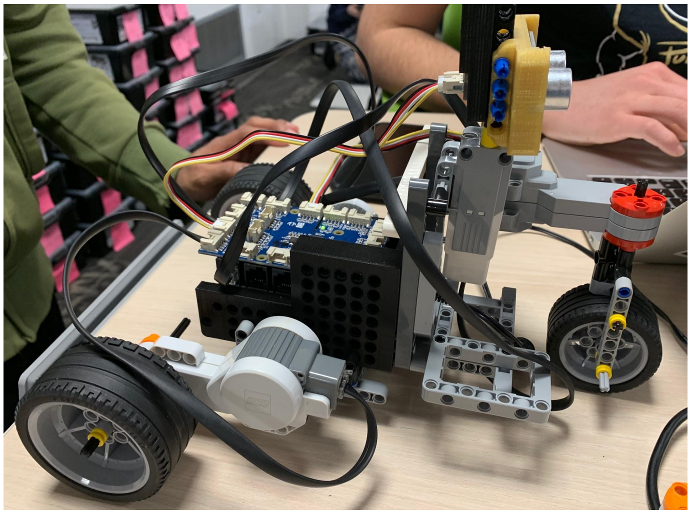
Figure 25: Functional Block Diagram



This functional block diagram was created to identify the main components of the GEARS and identify the relevant inputs and outputs of the system. This made it easier to construct specific subsystems that took care of the individual functions, such as the chassis for the body, the cargo deposition mechanism for the container, and the maneuvering and mapping algorithms for obstacle avoidance/mapping.

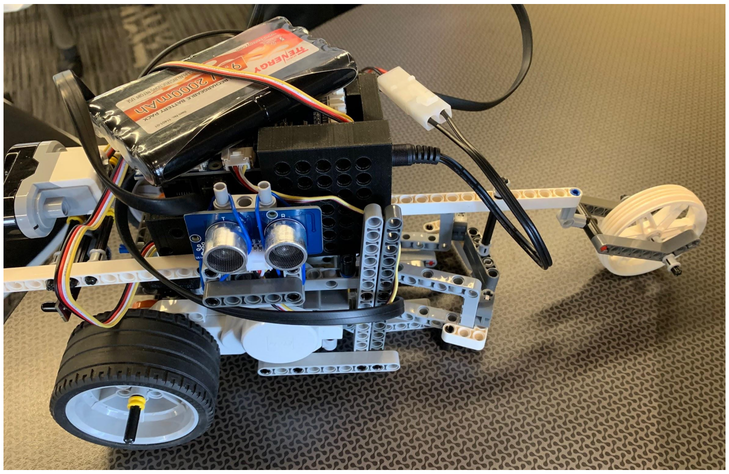
Appendix B: Design Iterations

Figure 26: Iteration 1



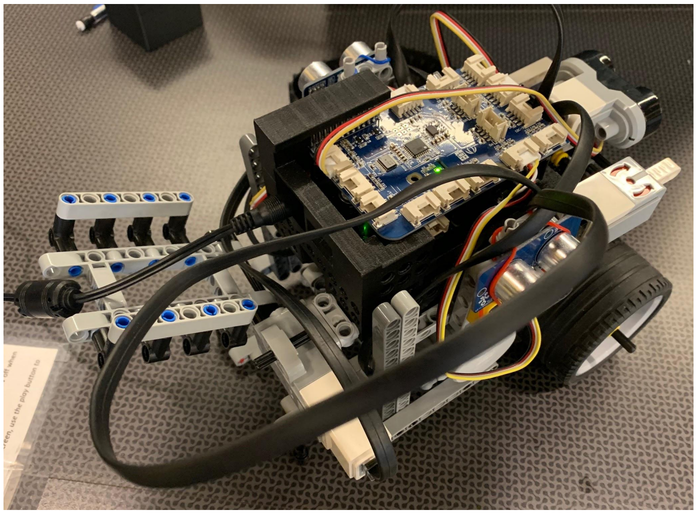
This was the initial design, featuring a tricycle-like turning mechanism and a rotating, centrally-located ultrasonic sensor and a minimalistic chassis. On the right side is the turning wheel and on the left side is the rear-wheel drive setup.

Figure 27: Iteration 2



This was the second iteration, featuring a two-wheel differential drive setup on the left and a swivel wheel on the right. The ultrasonic sensor setup has been altered to have one on each of the front, left, and right sides, and a small cargo carrier is located behind the swivel wheel.

Figure 28: Iteration 3



This was the third and final iteration, keeping the two-wheel differential drive from before and replacing the swivel wheel with a skid plate. The cargo deposition mechanism can be seen on the left, controlled by a single motor.