Location and Resource Identification System (LARIS)

Team 21

Garrett D. Armstrong, Dave L. Guo, Joshua D. Herzfeld, and Aidan C. Roy

Executive Summary

The objective of this project was to create a Location and Resource Identification System (LARIS) that was capable of navigating itself through a mock abandoned hospital. The LARIS must be capable of precise navigation through an unknown space, and autonomously explore the entire facility to determine the layout of the area. It should be capable of locating and identifying various resources within the facility, create a map of the layout of the facility, and maintain communication with a base station and return data. Once all of the tasks are fulfilled, the LARIS should exit the building. The LARIS was constructed with all these factors in mind.

The LARIS is unique in its compact design and drive train design. The small footprint of the robot combined with the tread system allow for point-to-point navigation of the robot to be the strong point of the LARIS. The robot code implements a system to maintain distance from the wall when useful. The placement of the battery pack near the rear of the robot and the mounting of the sensors on the front allow for even weight distribution across the robot. Finally, the prototype implements a swiveling ultrasonic sensor mounted on a motor to be able to actively look in multiple directions as required, while minimizing movement of the robot as a whole.

The robot did not perform as expected due to unaccounted complications with the course. Due to tape being present on the course's surface, the LARIS failed to properly recognize the distance traveled. Due to this issue, the robot was unable to properly navigate through the first three-way junction. While the LARIS did properly navigate into a room with a resource, the light sensor did not activate successfully due to a failure in the touch sensor used to signal the presence of a waste container. Therefore, the LARIS did not retrieve proper readings and was incapable of determining the resource. Since the robot did not travel to the point in which the IR or magnetic beacons were present, it is uncertain if the LARIS would have properly detected the beacons. While there were many issues, the robot was able to successfully output a map of the route taken. The LARIS was generally a success, but more work is needed to perfect the product.

Design Considerations

At the beginning of the design process, completing every task perfectly was daunting. In order to complete tasks deemed more important, engineering design tradeoffs were made to increase performance in certain areas at the cost of limiting performance in other areas. With the project goals, the most important task would be determining a safe path through the hospital, so making sure the LARIS was capable of navigating and relaying a safe path was placed as a higher priority than identifying the resources. A set of general specifications were created to monitor performance of the LARIS throughout the design process.

Customer Need	Technical Need	Technical Requirement	Target Value
Demonstrate LARIS ability to navigate using walls as reference	Distance from edge of Laris to walls	LARIS should stay >= 8 cm away from any wall	Laris should stay 10 cm away from any wall
Demonstrate LARIS ability to make accurate point turns	Angle that the LARIS can turn	LARIS must be able to turn 90 degrees ± 4 degrees	LARIS should turn 90 degrees ± 2 degrees
Demonstrate LARIS ability to navigate using magnetic and infrared sources	Distance from resources	The center of the LARIS must stay at least 10 cm from the magnetic / IR beacon	The center of the LARIS should stay more than 12 cm away from the magnetic beacon
Demonstrate LARIS ability to perform point-to-point navigation	Distance between center of LARIS and navigation point	The center of the LARIS must be less than 5 cm from the desired navigation point	The center of the LARIS should be less than 3 cm from the desired navigation point
Demonstrate LARIS ability to build map of unknown area	Accuracy of map compared to actual route taken	Map should match the route taken by the LARIS	Map should match the route taken by the LARIS
Demonstrate LARIS ability to	Number of resources correctly	LARIS should properly identify 2	LARIS should properly identify 2

identify and differentiate between unknown resources	identified	different resources	different resources

Table 1: Engineering Specifications

The initial design of the LARIS was created through traditional group brainstorming. These ideas included a short overall length to point turn effectively, a system in which both the front and back set of wheels were capable of turning, using swords as low-friction skis, and mounting a third motor containing an ultrasonic sensor to the top of the robot to be able to detect walls.

Decision matrices were created for determining the overall size of the LARIS and ultrasonic sensor placement (Refer to Appendix, Table 2, Table 3). While a larger design would allow for more flexibility in regards to sensor placement and stability, a more compact design would allow for better point turning ability and less probability of colliding with walls, which better help meet the engineering specs shown in Figure 1. When determining ultrasonic sensor placement, the debate was primarily between turning the entire robot or placing the sensor on a swivel. While having multiple sensors would be the best solution, only one sensor was provided. It was ultimately decided that having a sensor on a swivel would allow for less overall movement, minimizing the risks for error.

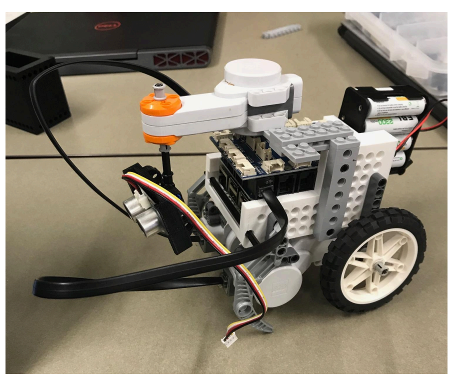

Figure 1: Initial Design of LARIS

The initial design shown in Figure 2 used a drive motor on each side, with LEGO swords to support the front half of the LARIS. The ultrasonic sensor was attached to a motor in order to swivel and look in all directions.

The LARIS successfully completed both Tasks 1 and 2 at the first Presentation of Competency (PoC), but the first specification was not met, so testing was performed to find an alternative wheel option. This testing showed that the wheel based design usually had more error than the tread based design, because the treads allowed the force to be applied along the whole length of the frame instead of just the back.

Trial	Big Wheels Error	Treads Error
1	3	1.5
2	3	1
3	2.5	2
4	1	1.5
5	2	1
6	1.5	~0
7	2	1

Table 2: Degree Error of 90 Degree Turns

A decision matrix was created to determine the optimal wheel design to meet the engineering specifications (Refer to Appendix, Table 4). While the wheels did allow for better consistency (no tread skipping) and higher speeds relative to treads, the ability to point turn carried higher precedence. The battery pack was also moved to the back of the frame in order to distribute weight.

Figure 2: PoC 2 Iteration

Following PoC 2, the light sensor was placed underneath the IR sensor to detect the waste bins to meet the last engineering specification. An Inertial Measurement Unit (IMU) was also placed above the light sensor to detect magnetic fields to avoid the magnetic beacons. After determining that the treads would not be able to consistently stay intact after extended use, the best solution was to reinforce the exterior of the treads with braces. This stabilized the drive train and the frame in general, and served to increased tightness of the treads. To attach the IR sensor, a mount was designed and 3D printed. This was done to prevent damage to the cables that could occur from interference from other pieces mounting the sensor to the robot.

In order to detect a resource, a touch sensor was placed near the light sensor so the button would activate when pressed up against the bin. This would activate the light sensor and run the function to weigh the container. In order to weigh the resource, an arm attached to a motor was placed upon the rear of the robot. The weighing process involved rotating the LARIS 180 degrees, and using the rear mounted arm to lift the container. The change in motor power was detected and converted to mass using testing.

Software was developed in a six-stage process: create a rudimentary flowchart, write test code for data collection, gather data, create an algorithm, refine the algorithm, and finally assimilate it into the final python code. This process was used to create working programming logic for the main functionalities of the LARIS; the main functionalities of the LARIS were determined to be hallway navigation, mapping, resource detection, and weighing.

Throughout the development, focus was extensively placed on the navigation subsystem of the LARIS software. The navigation subsystem consists of scanning the hallways, moving, and turning. Scanning the hallways was accomplished with a swiveling ultrasonic sensor, as previously described. The logic dictates that the sensor would check the left, then the front of the LARIS, and lastly the right. Movement was developed based on ease of use of the given “BrickPi3()” functions provided by the teaching staff. Initially, once the flowchart was created, motor power functions were used to drive the motors (Figure 3). However, after experimenting with degrees per second and finally motor position, it was determined that motor position was the best option. This is because using motor position allows for greater accuracy while moving a set distance. The only downside to this method is the constant need to offset the motors and use power limits, but this is easy to overcome. Based on the radius of the drivetrain wheels, one distance can be achieved with a certain number of degrees of rotation. The final code uses motor position to go forty centimeters, or one grid space.

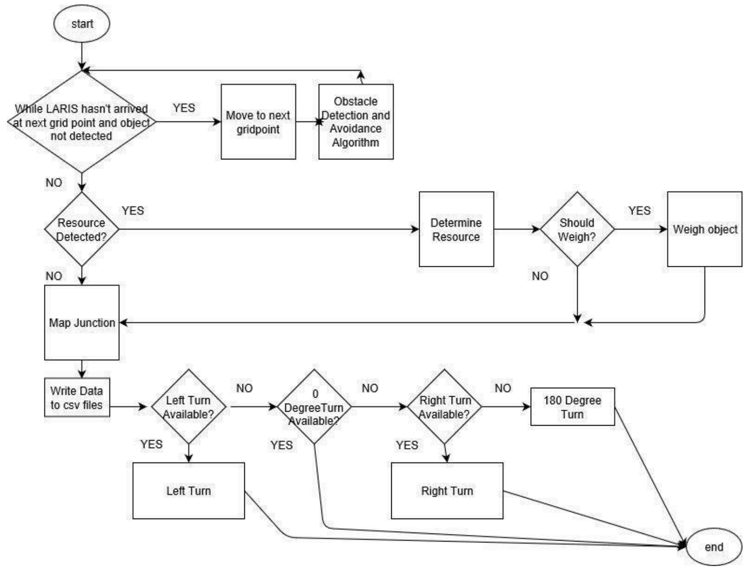

Figure 3: Navigation Algorithm Flowchart

Turning was a simple, yet crucial aspect of the LARIS software. Success using motor encoder position for LARIS movement spurred the push to use position for turning. After flowcharting and creating a demo code, both turning and basic lateral movement were tested on big wheels and treads with small wheels. The python scripts were adapted to both for testing purposes. In order to make a ninety-degree turn, the big wheels had to rotate roughly 138 degrees in opposite directions and the treads had to rotate 560 degrees. These numbers were discovered through tuning the code. Point turning code was used with both types of drive trains to help determine which one is better. Turning also has a precedence order: left has priority, followed by straight, then right, and lastly backwards.

After PoC 2, the mapping algorithm was written, based off Design Challenge 3 from ENGR 162. The LARIS utilizes a two-dimensional array to store values from two counters that correspond to the direction of movement (each counter represents an axis of motion). An ‘x’ denotes that a directional counter was altered and that the robot has explored that grid space of the hospital. This algorithm was already proven to work, so it was implemented in addition to the navigation scripts so that movement and turning changes the directional counters, and thus LARIS orientation, to produce an accurate map. Because of the turning precedence previously described (left, straight, right, backwards), the mapping algorithm has a left side bias that may result in an incomplete map.

To detect the four different types of resources, various sensors were implemented. An NXT-Light sensor and NXT-Touch sensor were added to accommodate for detecting the hazardous and non-hazardous waste, an IMU was added to detect the MRI, and an IR sensor pair was added to detect the Cesium-137 Source. Test code was created for the IMU and IR sensors to determine what values exist at the minimum radii (five inches from the MRI and four inches from the Cesium-137). The IMU read an average absolute value of 120 for the x-axis, 256 for the y-axis, and 638 for the z-axis. The IMU was oriented so that the z-axis faces forwards because it was more sensitive (value ranging from zero to 640) than the other axes. Using one axis that is sensitive comes at the cost of only sensing the MRI on that axis. The IR sensor sensed that the Cesium-137 gave off a raw value over 200 when four inches away (the sum of both halves of the sensor). See Figure 4 in the Appendix for specific data.

The sensors were used in such a way that, if any of them sensed the values featured above, then the code would work to determine the resource type based off the triggered sensor, weigh it if possible, and map it. Touch sensors were used to sense non-hazardous waste and hazardous materials (i.e. blue and gold resources). No further iteration of the object detection was created because it worked as expected. This logic is ideal because it deals with all resources.

The weighing system used the NXT-Touch sensor and NXT-Light sensor specifically. The logic behind this subroutine is outlined in the flowchart below (Figure 5).

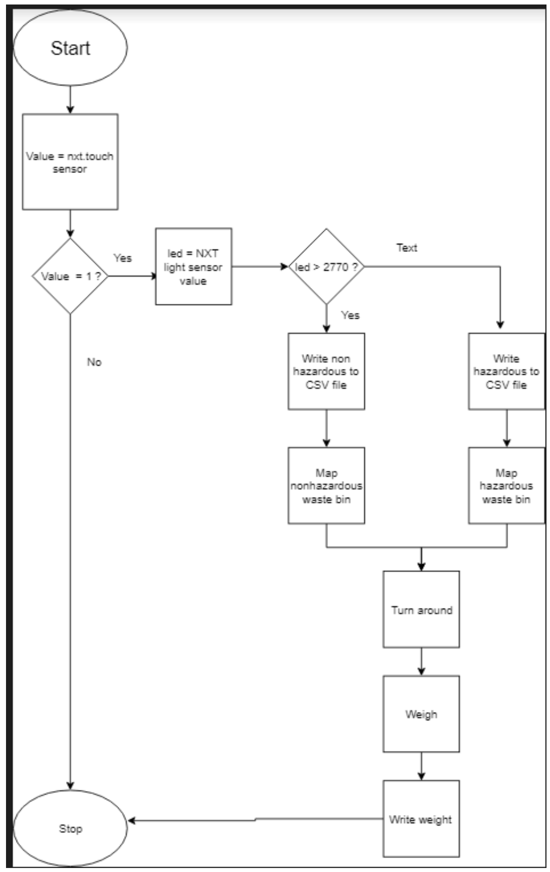

Figure 5: Weighing Algorithm Flowchart

Once the front touch sensor is triggered, the weighing motor lifts the resource until the arm triggers the second touch sensor which ends the weighting algorithm. This code is efficient because it both turns itself on and off, and it accurately reads the correct lifting motor power and associated mass (see LARIS Data Analysis). The downside is that it is very dependent on the movement logic and whether or not the LARIS successfully completes a 180 degree point turn.

The final code works so that the movement algorithm runs until a sensor is triggered or the desired displacement of forty centimeters is accomplished by each motor, followed by object detection logic, and lastly by mapping and point turning (Figure 6). Once the demo ends (i.e. ctrl C), the two .csv files are outputted to the base station.

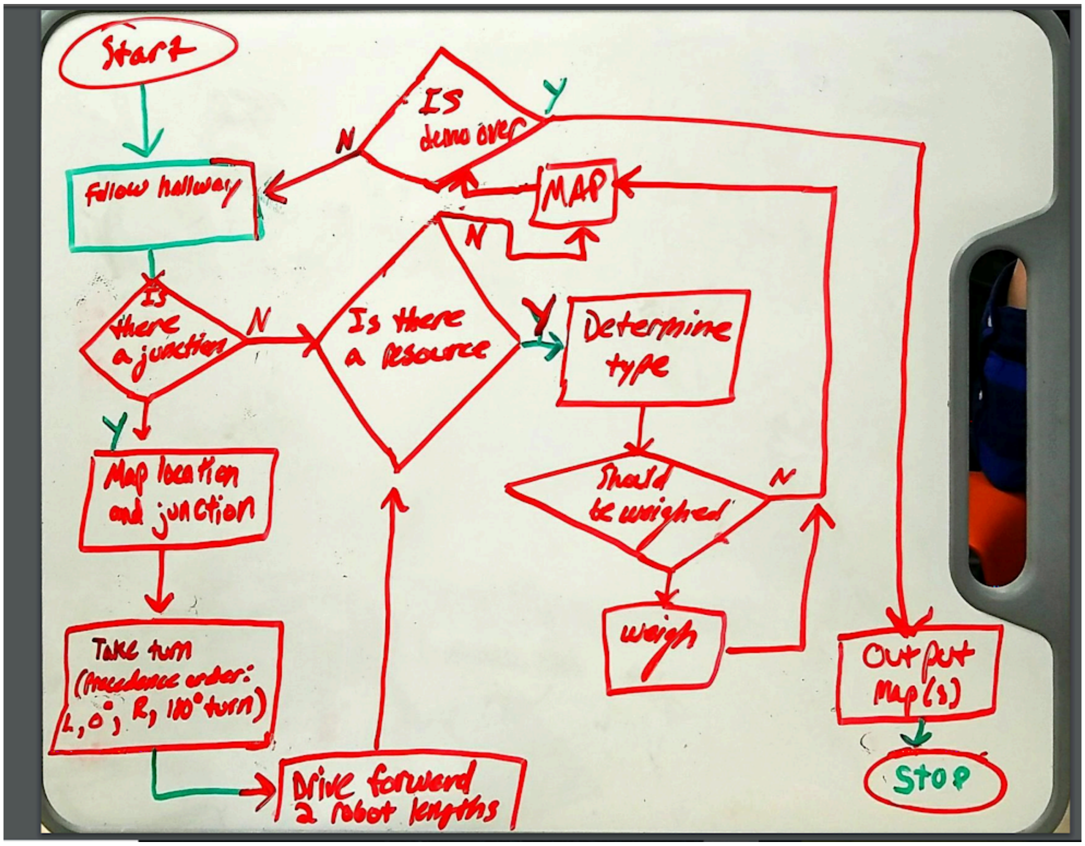

Figure 6: Final Code Flowchart

There are various unique hardware aspects to this LARIS prototype. The drive train was designed with treads, which supplies a drive force to the entire length of the robot, to allow the LARIS to turn effectively. The amount of contact with the ground also will keep the LARIS moving in a straight path while the motors are supplying the same level of power. However, the treads require extensive frame support in order not to disassemble after extended usage. The compactness of the robot allows for better ability to point turn and less potential for the LARIS to collide with the walls. However, this lack of space sometimes made it difficult to place sensors in preferred locations, and made it difficult to manage the wires properly. Placing the sensor on a swivel allowed for a design that minimized the amount of movement while still staying within the collection of sensors available. The design was constructed with relatively few materials, and in conjunction with its compact design, allows it to be easily transportable. The final hardware design is shown in Figure 7.

The presented software is unique for several reasons. The code implemented on the LARIS exclusively uses the set motor position and motor limits: no degree per second or power functions are used. This ensures that all motors go exactly as far as they are supposed to and at the correct power/DPS. Its uniqueness also stems from the code infrastructure. No functions are used in the program; counters are used instead to dictate what is mapped and how the robot moves. Lastly, the weighing system code is special. This subsystem is programmed so that the motor turns itself off when it hits a touch sensor, and then it relays the power required to lift the object and returns to its original state.

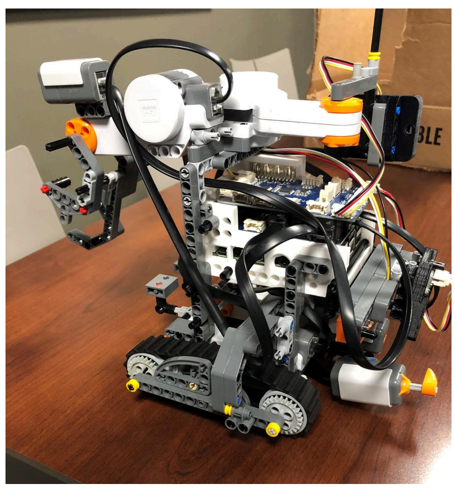

Figure 7: Final Hardware/Software Iteration

LARIS Data Analysis

Several sensors of all types were required for accurate operation of the LARIS. An ultrasonic sensor is the main sensor to be used for navigation. This sensor measures how far away it is from something in front of it, in centimeters. Testing was done to ensure the reliability of the measurements, as shown in Table 5 in Appendix B.

This sensor did not need calibrated, as the data gathered was easy to read and use. However, the distances that would cause the LARIS to take action were determined experimentally. The main use for the sensor on the prototype was to ensure an adequate distance was maintained from the walls in the mock hospital. This distance was tested qualitatively, and the preferred distance was found to be 10 cm from the left wall. The control algorithm uses this 10 cm limit to ensure the path along the hallway is straight. The ultrasonic sensor is also used to determine which direction to turn and move when a junction is reached. The same 10 cm limit is used for that purpose, scanning for and recording the location of the walls around the LARIS.

Another sensor that was used, and needed calibration was the NXT light sensor. The light sensor was used to determine the color of the resource bins, and output the color. That sensor was the easiest and most important to calibrate, due to the large range of light values that can be recorded by the sensor. The color values of the bins are very specific, so the ranges of color values to be read should be equally specific. Each color of resource bin was tested with the light sensor several time, at different angles to create a table of potential values that would be expected depending on the distance away from the bin. This data is shown in Table 6, a sample of which is shown below.

Trial	HAZARDOUS (GOLD)	NON-HAZARDOUS (BLUE)
1	2564	2851
2	2498	2829
3	2573	2836

Table 6.1: Light sensor data

This data was used to create logic allowing the LARIS to identify the color of the object in front of it whenever the light sensor is activated. The light sensor calibration entirely affected the output of the sensor. The data showed that every hazardous container returned a light value less than 2600, and the non-hazardous containers gave a values greater than 2800. Setting these ranges correctly controlled whether the LARIS outputted the correct color of resource when displaying the information.

A related sensor to the color detection method was the touch sensor attached to the front of the LARIS. This sensor simply signaled the presence of an object when pressed. Because the touch sensor is a binary sensor, either pressed or not, there was no calibration needed. The object will press the sensor, which then causes the prototype reversing the motors and rotating to lift the resource bin and weight it. The input is the state of the sensor, and if that state is pressed, the output is a break in the movement of the LARIS to initiate the bin measuring protocol.

Other resource measurement sensors included the Inertial Measurement Unit (IMU), and the Infrared sensor. The IMU was used to measure the strength of the magnetic field around the LARIS, which provided the information about the proximity of the MRI. In testing and calibrating, the data was collected to show the impact of distance on the readings taken by the IMU, as shown by Figure 8. Once this data was found, the distance limit from the MRI could be directly correlated to a magnetic field magnitude read by the sensor. As the distance between the sensor and the magnet source and magnetic sensor decreases, the magnetic field strength increases, until it reaches the limit value, at which point the MRI is located and mapped. The infrared sensor was tested and calibrated in a similar sense.

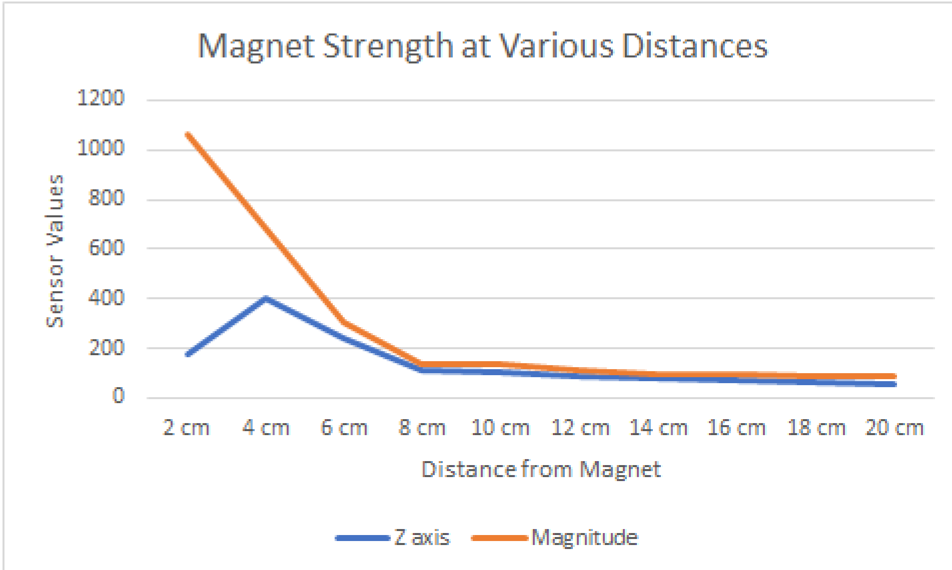

Figure 8: Graph of Magnet Strength vs Magnitude

The infrared sensor was the only method to identify the radioactive material. The method to calibrate the sensor was similar to calibrating the magnetic sensor. A Infrared light beacon was placed a predetermined distance from the prototype, and the readings were analyzed to relate distance from the beacon to strength of the light. The IR sensor has two sensor bulbs, and so reads two individual measurements. Because of the directionality of the sensor, both measurements were utilized to determine the distance from the LARIS to the IR beacon. After collecting the data, the distance was found to be too small if the sum of the two readings was greater than or equal to 200. The units are not clear, which is why the testing was necessary. The data from the test is shown in Figure 4.

To ensure that the sensor values would lead to the correct actions, a hypothesis test was performed. The null hypothesis was set at 200, which was the limit set in the python script. The mean value of the data in Figure 4 is 232.07, and the standard deviation was 32.394. The number of data points is higher than 100, so a normal distribution test can be used. Using a z-score statistic test, the p-value was found to be less than $1x10^{-5}$, which is lower than a significance level of 1%.

A component other then the sensors that needed calibration was the drive system. The motor encoders were used to set the wheels and treads to a certain position, and it was required to calculate that position. The desired distance was determined to be 40 cm using the information given by the project oversight team. Using the radius from the center to the edge of the tread, and the equation to relate the angle and radius to arc length was used to calculate the correct angle to turn to move the tread the correct distance. This equation is as follows: $\theta = 7200/(\pi * r)$. Leaving the variable "r" in the equation allows for the wheels or treads to be substituted and still find the correct distance.

The other motor that was used as a sensor is the lifting motor for finding the mass of the bins. This motor was calibrated by lifting several masses and determining the minimum motor power that would lift the mass. The mass was increased for the full range of masses offered by the oversight team. The maximum mass of a bin was 150 grams. Using small increments of masses, a graph was created and a line of best fit was found to make a relationship between motor power and mass. When the LARIS is lifting an object, the motor power is read from the encoder, and using the relationship found by the graph, the mass is calculated. The relationship is shown in the following graph.

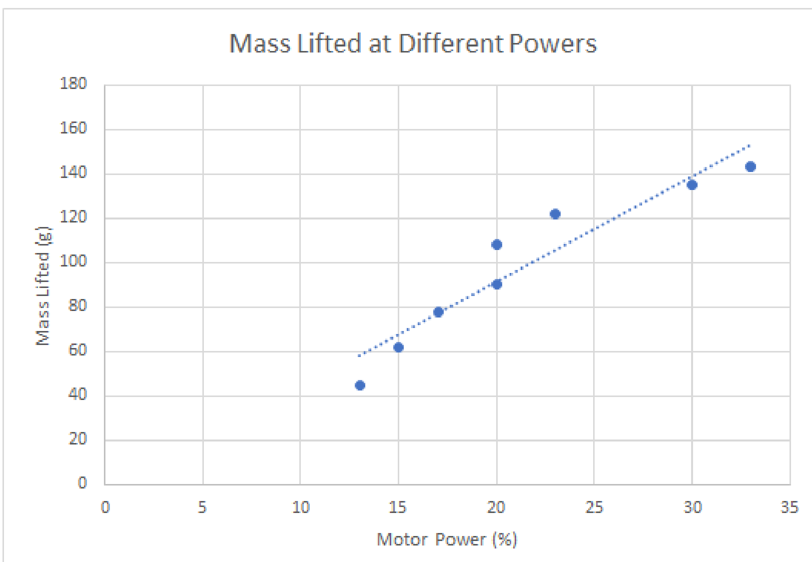

Figure 9: Relationship Between Mass and Motor Power

Results and Discussion

Going into the demonstration, confidence was highest in the LARIS's ability to navigate the maze and successfully exit the hospital. Extensive testing was done on the logic for decision making at the junction, corrective PID control, and point turning ability. The mapping algorithm was created by building off past projects, so few issues were anticipated when outputting a map to the CSV file. However, the LARIS's capability to properly detect the IR beacons and magnetic beacons was not prioritized, so expectations were low. Because of the small design of the LARIS and placing detection of bio-hazardous materials as a secondary objective, performance at detecting and weighing containers was uncertain. The creation of an arm that would be capable of lifting the resource was also implemented quite late in the design process.

During the demonstration, several complications arose that were not accounted for beforehand. Prior to the demonstration, the incorrect assumption was made that the entrance was located at the center of the hospital, which was rectified by by offsetting the beginning point of the map.. The robot's travel distance and turn radius performed all as intended. The first issue arose when the robot passed through the first three-way junction. The navigation array created beforehand was smaller than the actual course, so the robot stopped entirely once it traveled to the point where the index was out of range of the premade grid. The grid dimensions were remade to fit the actual track. After this issue was reworked and the robot was reset, another issue was encountered. In multiple consecutive instances, the LARIS failed to progress past the first junction due to the motors not traveling far enough to exit the programmed while loop. Tape was placed upon the track at the junction, so the surface was different from the anticipated paper surface the testing was done on. The tape was slicker than the paper originally tested upon, so even with the tread design, the robot did not travel the necessary distance to enter its next sensing phase. This issue was alleviated by pushing the robot approximately a centimeter. Afterwards, the robot progressed towards the location of the resource with no issues, but was unsuccessful in activating the light sensor. The touch sensor responsible for activating the light sensor sloped upwards when pressure was placed against it by the resource bin, so the button was not actually pressed. Because of this, the light sensor did not activate to detect the resource and the weight function was not called.

The PID control also did not work as intended. While the PID did allow for the LARIS to stay the intended distance from the left wall, it would fail to counter-correct properly to keep the robot parallel with the hallway. Because of this, when the robot scanned the junction, the LARIS would be at a slight angle, and so the sensors would pick up “walls” that were actually corners of the junction. Finally, in attempting to have the robot successfully exit the first segment of the track, the code was adjusted to abandon the turn precedence by the scanners and turn right, but the robot would turn right twice without time to make appropriate adjustments.

The robot did not successfully navigate past the first junction and did not detect or weigh the radioactive resources. Because the LARIS did not progress far enough to actually be in range of the IR and magnetic beacons, the LARIS did not perform on those tasks. However, the LARIS was successful in outputting the CSV file of the map and showed clear usage of PID control.

Following the demonstration, the group was able to determine which specifications were met properly when compared to the engineering specifications. The LARIS was able to properly navigate using the walls as a reference, staying consistently 8 - 12 cm away from the wall it was following, therefore completing point to point navigation as well. It was successful in properly turning 90 degrees during every turn. When not including the turns that were already unparallel to the hallways, turns in which the LARIS was properly aligned always resulted in the LARIS still being properly aligned after a point turn. The robot was unable to demonstrate its IR and magnetic beacon sensing abilities as it did not travel far enough within the course. The map outputted matched the route taken, and was capable of properly outputting the CSV file. The robot was unsuccessful in properly identifying the blue resource, as the button did not activate properly during the demonstration. The robot was overall mostly successful in achieving the design goals.

Conclusions and Recommendations

After the demonstration of the performance of the LARIS, improvement opportunities were found for the software and hardware design. Almost all the individual components of the design of the prototype were functional, but the integration and total system performance could be improved. The hardware was functional, but still requires some work to be reliable and structurally sound. The majority of the software functioned as desired, with one exception in the function maintaining the correct distance from the wall.

In particular, correcting to a parallel direction with the wall after PID correction away from the wall did not function properly. The LARIS occasionally failed to correct far enough, and as a result, would make contact with a wall during the "Leap of Faith" maneuver that immediately followed. A software revision would be changing the central logical statement in the software controlling the PID correction from an "and" statement to an "or" statement, as well as testing different motors, as the mindstorm motors all have slightly different encoders, which cause them to behave differently. Additionally, it is recommended that the code be modularized into functions to allow for easier editing and understanding of the LARIS software.

Physically, it is recommended that the brace-like device that holds both the motor for the ultrasonic sensor and the motor for the lifting arm be redesigned. A 3D-printed part would be best for this. A customized piece would be best suited to support the motors in the proper position. Currently, the motors are held in the correct position, but not in a stable manner. In addition, the trigger mechanism for detection of waste bins needs a redesign It was not activated by the waste bins as they were too light.

The LARIS, after the few changes suggested, would be recommended as a viable design for an autonomous mapping robot designed for abandoned hospitals. The LARIS is versatile, easy to use, and performs at a high level, making it ideal for its intended use.

Appendix

Decision Factor	Point Turning Ability	Ability to avoid walls	Part Placement Flexibility	Stability	Total
Weight	4	4	2	1	
Compact	5 * 4 = 20	4 * 4 = 16	2 * 2 = 4	1 * 2 = 2	42
Large	2 * 4 = 8	2 * 4 = 8	4 * 2 = 8	1 * 5 = 5	29

Table 2: Design Matrix for Size of LARIS

Decision Factor	Amount of movement	Consistency	Quantity of sensors used/available	Total
Weight	2	3	5	
Sensor on Swivel	4 * 2 = 8	3 * 3 = 9	4 * 5 = 20	37
Multiple Sensors	5 * 2 = 10	5 * 3 = 15	1 * 5 = 5	30
Turning entire robot	1 * 2 = 1	2 * 3 = 6	4 * 5 = 20	27

Table 3: Decision Matrix for Ultrasonic Sensor Placement

Decision Factor	Point turning ability	Consistency	Speed	Total

Weight	4	3	2	
Wheel	2 * 4 = 8	4 * 3 = 12	3 * 2 = 6	26
Tread	4 * 4 = 16	3 * 3 = 9	2 * 2 = 4	29

Table 4: Decision Matrix for Wheel Construction

Actual distance	Measured distance
10	10
10	10
10	10
10	10
10	10
10	10
10	9
10	10
10	10
10	10
10	10
10	10
10	10
10	10
10	10
10	10

10	10
10	10
10	10

Table 5: Testing Data for Ultrasonic Sensor

Trial	HAZARDOUS (GOLD)	NON-HAZARDOUS (BLUE)
1	2564	2851
2	2498	2829
3	2573	2836
4	2588	2832
5	2541	2803
6	2476	2898
7	2483	2847
8	2574	2844
9	2592	2846
10	2597	2860
11	2522	2817
12	2590	2822
13	2484	2831
14	2570	2845
15	2566	2852

Table 6: Light sensor data

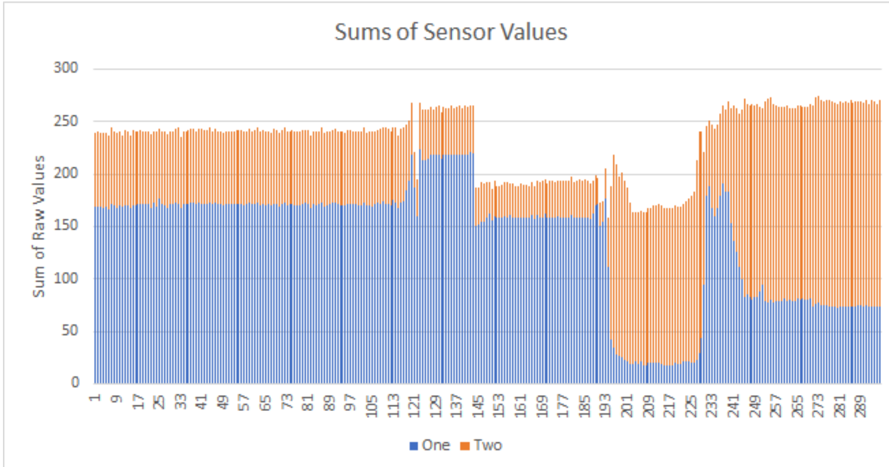

Figure 4: Sum of Raw IR sensor values