SYSC 4805 A - FINAL REPORT

Prepared for: Mostafa Taha

Prepared by: Group 3 - Team Blizzard Blue

Taiye Davies – 101116265

Rami Haddad – 101076255

Table of Contents 2

1.0 Project Charter

Objective

The principal objective of this project is to construct a working model of an autonomous snowplow to travel and clear snow off an area enclosed by a black path using obstacle avoidance capabilities. Team Blizzard Blue aims to manufacture a robot that should complete this task without hitting any obstacles within a specified timeframe. The robot will consist of a designed snowplow to fend away snowballs while remaining within the $6m^2$ black enclosed path. Furthermore, using sensors that induce obstacle detection, the robot shall avoid any obstacles.

Overall Deliverables

Our team has proposed a list of deliverables alongside the overall objective of the project which we believe are high priority requirements to be presented.

Our first deliverable will consist of a model of a snow plow robot that can detect its surrounding path and move obstacles out of the robot's way. This will be performed by writing scripts and constructing an algorithm that will aid the robot in planning routes to optimize the distance, detect an object and reroute when needed, create a snowplow, and ensure the wheels all follow a path in the same direction. Afterwards, we will be testing all sensors and start performing mapping algorithms before the progress report due on November 11. Finally, prior to our final report submission on December 9th, we will be having our project presentations in class and in our lab, section showing the working representation of the robot. The following milestone and final deliverables are listed below:

Milestone Deliverables

- Progress Report due on November 11, 2022
  - The progress report will be a re modification of the project proposal with a few added features. We will be redefining our project goals, schedule diagrams and estimated cost for the project as well as its requirements.

- In Lecture Presentation on December 5, 2022
  - During the presentation we will be explaining our gradual progress over the course of the month towards building the robot. We will be talking about the team goals, challenges faced, design choices, testing and chart showing our process to reach our overall objective.

Final Deliverables

- In Lab Presentation on December 2, 2022
    - o During the presentation in the lab, we will be presenting the final product of our robot and perform a demonstration in the 6m^2 enclosed area with obstacles.
- Final Report on December 9, 2022

We will be updating our project proposal to incorporate raised concerns and add the required changes. We will also be adding a control chart detailing the contribution of each team member, results of testing the system and all code added to GitHub repository.

2.0 Scope

Requirements

We have outlined a list of requirements which we will be delivering by the end of this project:

List of Requirements:

Software Used:

- The main language to develop and accumulate data would be Python in conjunction with the Arduino Due board.

Functional Requirements:

- Clear all the golf balls(snow) from an area labeled by a black path with a given perimeter.
- The robot avoids and detects all the moving and standing obstacles that are laid out on the testing area.

Physical Restrictions:

- The robot must complete the task within the assigned 5 minutes.
- The robot speed must not exceed 30 cm/s.
- The maximum robot size: Width x Length x High = 216 x 252 x 150 mm. Given that the actual robot is 176 x 232 x 79.5, this means that the plow can extend in width by a maximum of 40 mm (20 mm in each direction), and extend in length by 20 mm.
- The testing area is a square of almost 2.5 x 2.5 meters. Testing area is subject to being changed with similar dimensions.

Requirement Activities

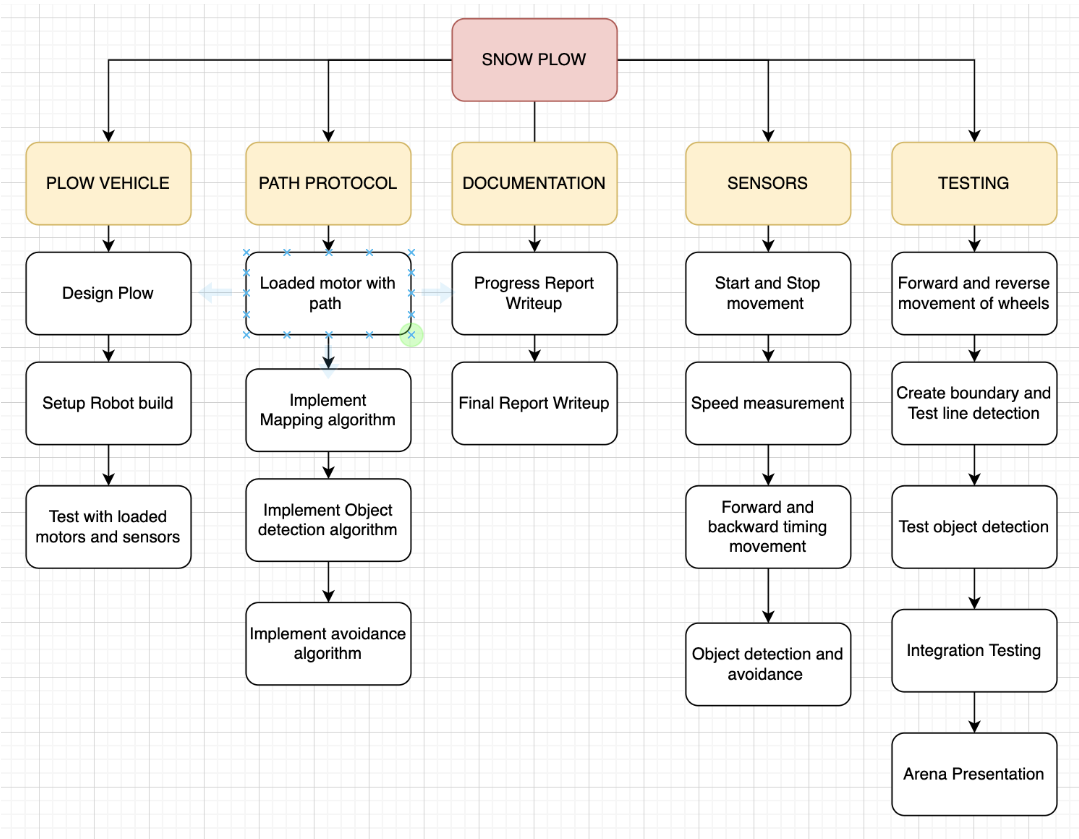

Figure 1: Work Breakdown Structure Diagram

Testing Plan

In unison with the physical restrictions the project is bound by, the project will proceed with the following tests to ensure that the robot fulfills its task and meets the desirable functional requirements. The first test is to ensure that the robot does not exceed the maximum speed limit of 30 cm/s. In addition, the robot is tested to ensure that the robot is compatible with any testing arenas, and the robot avoids any obstacle while staying within the boundaries. Once those tests are conducted, testing on the sensors are executed to establish a secure and stable result to begin integrating the sensors with the robot. Tests regarding integrating the system with the robot are in effect to ensure that the system can work in unison. Finally, final tests are in place to ensure that the robot can match all the functional requirements flawlessly no matter what simulation is compiled.

Individual Testing:

Individual testing is composed of testing on each specific sensor and motor to ensure that it matches the desired functionality and compatibility with the project.

Test Description	Test Pass Condition
Tests on each sensor at different ranges and surfaces	Sensor detects changes in range accurately
Tests on each motor with different speed and frequencies	Motor able to change speeds without exceeding the top speed limit.
Tests on each motor to ensure speed and movements	Motor able to move at the required speed and in unison with other motors

Integration Testing:

Integration Tests are tests that ensure all components of the robot match together and work together to fulfill a task. Testing also ensures that units can work in unison and all the code works together to build a result.

Test Description	Test Pass Condition
Testing on the sensors to effectively detect obstacles and differentiate between them.	The sensors should be able to allocate and distinguish the distance between the sensor and obstacle.
Testing on the sensors to illustrate a sketch of the testing arena and boundaries	The scopes on the robot should be able to map out the position of the robot and map its pathway.
Testing any scope's positions and its axis to establish direction for the robot.	Using the scopes, the robot should be able to establish which direction is best suited to drive towards.
Testing that the motors move within arena dimensions	In agreement with the sensors, the robot should drive within the arena boundaries.
Testing that the robot can hard stop if an obstacle appears unexpectedly.	The robot should be able to continuously navigate the sensors and abruptly stop if an obstacle is presented at any time.

Final Tests:

Final tests are executed to ensure that all the integration and unit tests are secured and well- formatted. It is also implemented to observe if the robot completes the final task and achieves all the requirements.

Test Description	Test Pass Condition
Ensuring that the snowplow is attached securely	The snowplow should be securely attached in event of hitting an obstacle.
Testing that the robot can stop when an obstacle is detected	When the robot is moving, the sensors should be able to detect obstacles approaching.
Testing that the robot directs according to the direction of the obstacle approaching.	When the robot approaches an obstacle, the robot should be able to steer in a desired direction to avoid the obstacle.
Testing that the robot stops when approaching the arena boundaries	The robot should be able to stop when approaching the black enclosed path of the arena boundaries.
Determining if the obstacle is moving or stationary.	If the obstacle is moving, the robot should be able to determine whether to wait or move out of the way.

3.0 Schedule

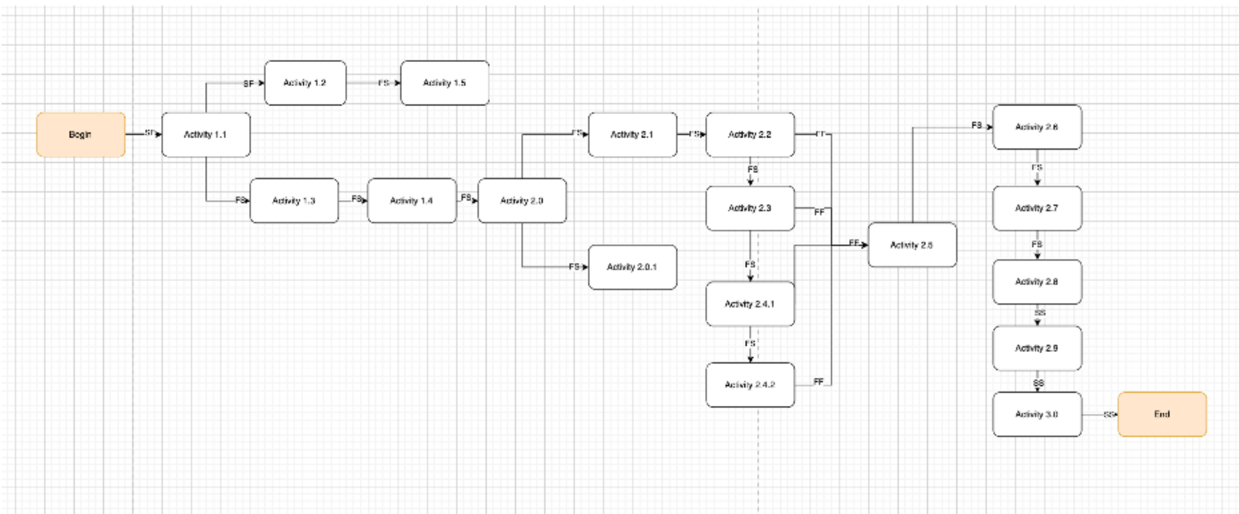

Figure 2: Schedule Network Diagram

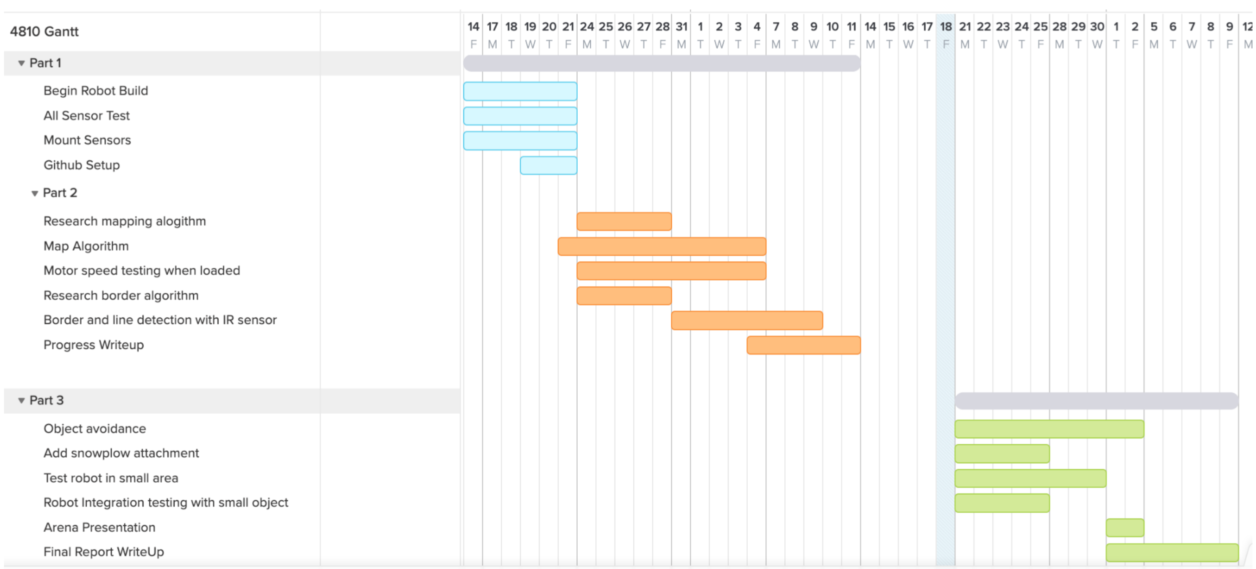

Figure 3: Gantt Diagram

Note: The blue colored bars indicate the building phase of the robot.
The orange colored bars indicate the mapping phase of the robot.
The green colored bars indicate the testing and implementation phase of the robot.

4.0 Cost

For each of the Labs that are conducted throughout the term of this project, 4 hours are assigned for each developer. There are a total of 6 Labs to present a final project. In addition, each member will partake in an extra 2 hours of work per week to provide a report and any additional documentation. However, during the final 2 weeks, each developer will add an additional 2 hours per week to ensure final testing and preparation for the final demonstration and report. Each developer/member will receive a pay rate of $50/hour.

Week No.	Hours Consumed by Group	Total Cost Summation
7	12 (4Hrs for Lab, 2Hrs for Report)	$600
8 (Fall Break)	4 (2Hrs to investigate materials)	$800
9	12 (4Hrs for Lab, 2Hrs for Report)	$1400
10	14 (4Hrs for Lab, 4Hrs for Report)	$2200
11	14 (4Hrs for Lab, 4Hrs for Report)	$3000

Table 1: Cost Breakdown per week

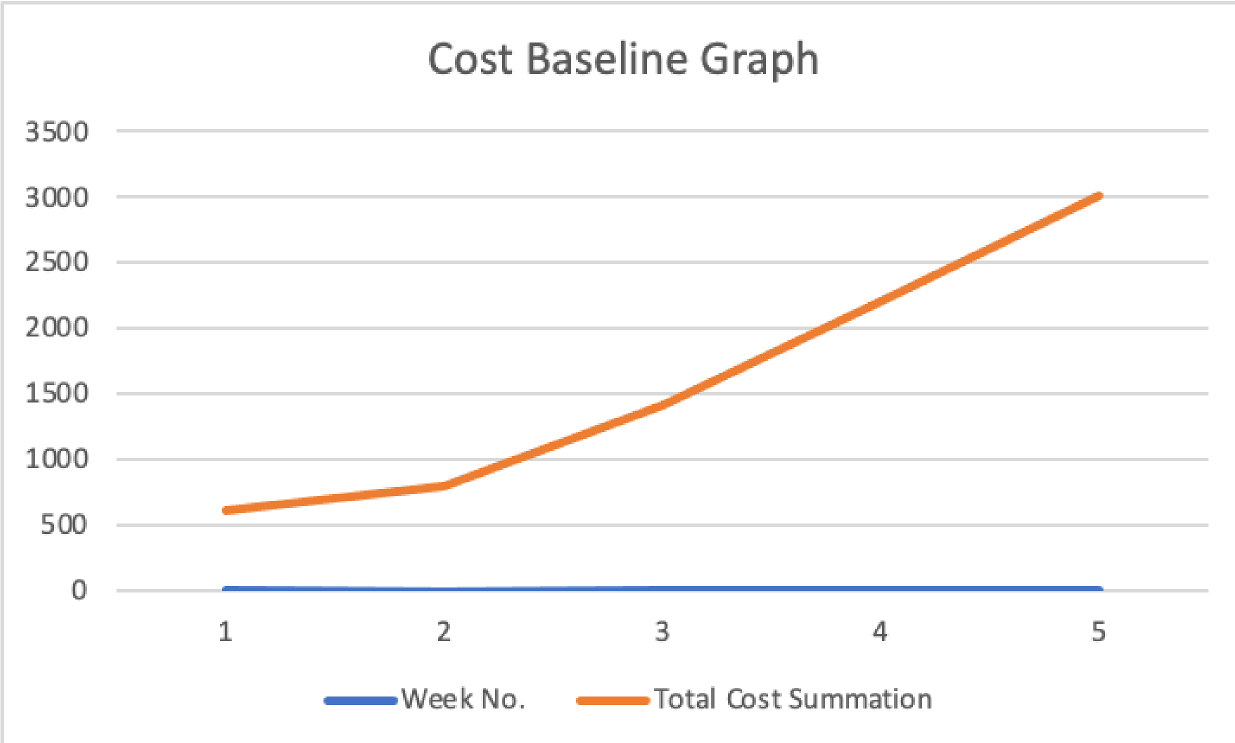

Figure 4: Cost Baseline graph of number of weeks spent vs the cost as the week progresses

Package Cost Estimate: Sum of the Cost for the Work Package (Lab Kit)

- Lab Kit is estimated to be $500

Contingency Reserves: Emergency funds for causes that are known

- Estimated costs for any known causes are estimated to be around $150. Management Reserves: Emergency funds for causes that are known

- Estimated cost for any issue that may arise unknowingly are estimated to be around $50.

Total Baseline Cost: Cost Summation + Contingency Reserves + Package Cost Estimate

- $3000 + $500 + $150 = $3650

Project Budget Worksheet: Total Baseline + Management Reserves:

- $3650 + $50 = $3700

5.0 Planned Value Analysis

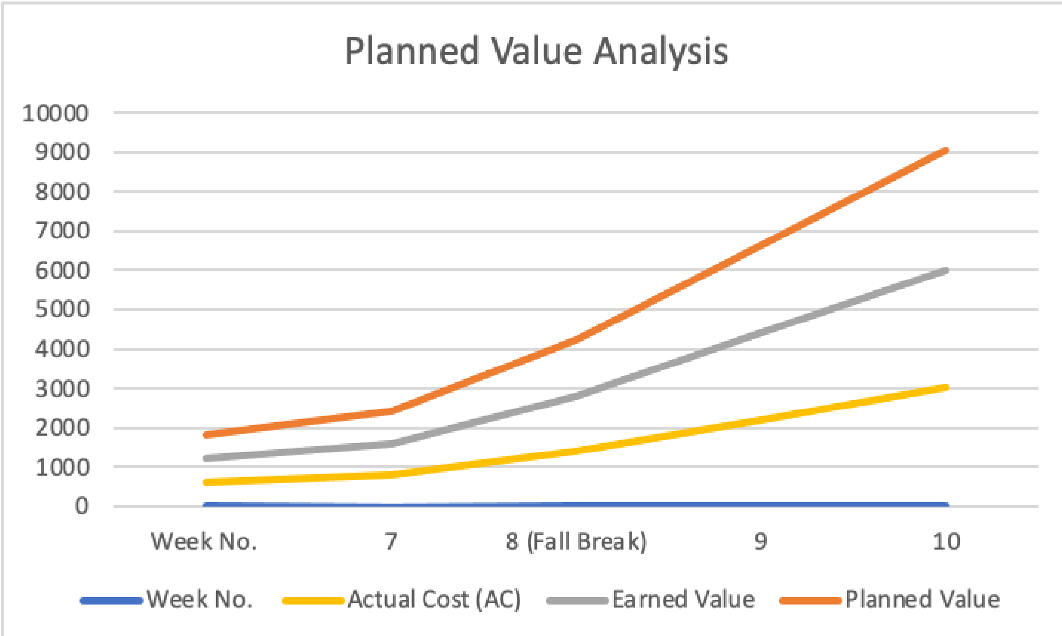

Figure 5: Planned value analysis of number of weeks spent vs the cost as the week progresses

Week No.	Planned Value	Earned Value	Actual Cost
7	$606	$600	$600
8 (Fall Break)	$812	$800	$800
9	$1418	$1400	$1400
10	$2224	$2200	$2200
11	$3030	$3000	$3000

Table 2: Earned Value Analysis (EVA) breakdown

In Figure 5, we see that the planned value is greater than the actual cost and earned value of the project. The Planned Value (PV) details the authorized budget for all activities to date
The PV from week 7 to the week 11 is : $9000

The Earned Value is the authorized budget for actually completed activities to date. This entails the portion of the project which has actually been completed. From our graph we can see that our earned value is estimated to be $6000.

The actual cost represents the actual budget for completed activities to date. This entails the true cost of each task for each week including replacement of tools. From our graph we can see that our earned value is estimated to be $3000.

6.0 Human Resources

R = Responsible, A = Approver

Activity No	Activity	Taiye	Rami
1.1	Setup GitHub repository	RA	
1.2	Finalize robot setup and begin running test drive on motor	R	A
1.3	Start testing on all sensors	A	R
1.4	First implementation of testing line sensor with loaded motor	R	A
1.5	Progress Report Writeup	R	A
2.0.0	Research on border detection algorithm	A	R
2.0.1	Implement border detection algorithm	R	A
2.1.0	Test forward and backward movement of robot	A	R

2.2	Research on mapping algorithms	A	R
2.3	Implement object detection algorithm using small object	RA	AA
2.4.1	Research on object avoidance algorithm	A	R
2.4.2	Implement object avoidance algorithm	RA	RA
2.5	Add snow plow attachment	A	R
2.6	Integration testing of robot with small object	R	A
2.7	Create boundary and test robot navigation	A	R
2.8	Test on sample testing area	RA	RA
2.9	Arena presentation	RA	RA
3.0	Final Report Write up	RA	RA

Table 3: Responsibility Matrix

This is the testing arena where the robot moves the snow and avoid the obstacles shown in the image below:

Figure 6: Testing arena

7.0 Overall System Architecture

Our System design consists of two sections : how the sensors are applied and how the robot moves according to sensor outputs.

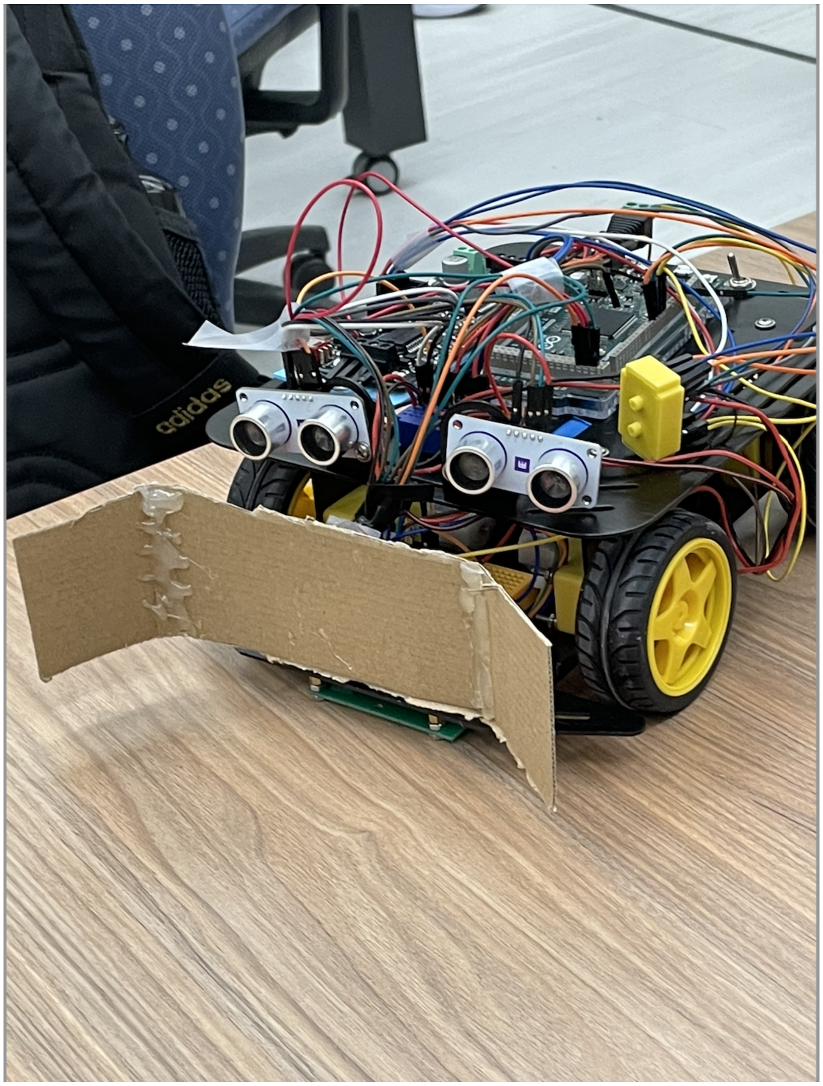

Figure 7: System Architecture Design

Figure 7 resembles the robot as a whole with the placement of the sensors and motors perceived. The design features an Arduino Due mounted at the top of the chassis to aid with connections to the other devices. In addition, a Cytron Motor Board was mounted at the top of the chassis to aid with assembling the motors and deriving each motor's connections. The robot also features two ultrasonic sensors which transmit and receive waves in a cone form. The placement of the ultrasonics sustains complete vision of the robot's incoming path in order to inform the robot of stopping or avoiding an obstacle. In addition, a line follower sensor was placed at the bottom of the chassis with a clearance of 3mm. This was done in accordance with a given data sheet of the line follower sensor which optimizes the performance of the sensor. Lastly, a snow plow was mounted at the front of the chassis to aid in collecting balls and diverting them compactly with a design that interrupts any balls from flowing through the side of the robot.

7.1 Sensors

The sensors we used in our robot design was a line sensor and two ultrasonic sensors. The line sensor was used to detect the black line of the enclosed path and the white floorboard to guide its pathway process. In Figure 7, the line sensor was mounter underneath the chassis leaving a space of about 3mm from the ground. This design was to produce better detection of the border when detected. The ultrasonic sensors were mounted at about a 15 degree angle unto the front of the robot. The initial placement was at an angle of 30 degrees but the robot was not able to detect obstacles directly in front of its spectrum.

7.2 Robot Movement

We used two main boards for robot movement. The motor driver board will be in charge of powering the motor drivers and wheels. The arduino was used to power our one line sensor while the buck regulator will be used to power the ultrasonic sensors.

8.0 Overall State Chart

This section displays our overall state chart of our system. Our main algorithm states as follows:

- The robot will initially drive forward and use the distance sensors to check if there is any obstacle.
- If an obstacle is detected then, we reverse a little bit and turn in the opposite direction of the obstacle by about 30 - 60 degrees and continue driving forward.
- If a line is detected, we turn around and proceed forward until we detect the border or an obstacle. This is measured by a particular threshold to aid in detecting if we are perceiving a black surface underneath or a white surface. If the threshold is greater than 750 it is seen as a black line, if it ranges less than 750 then it is seen as white surface.
- In the case where we drive forward and the distance sensors were not able to detect the obstacle quickly, the vehicle stops and turns around. If the sensors detect an object within a 15 cm range of their view point then it proceeds to adjust its path.

-

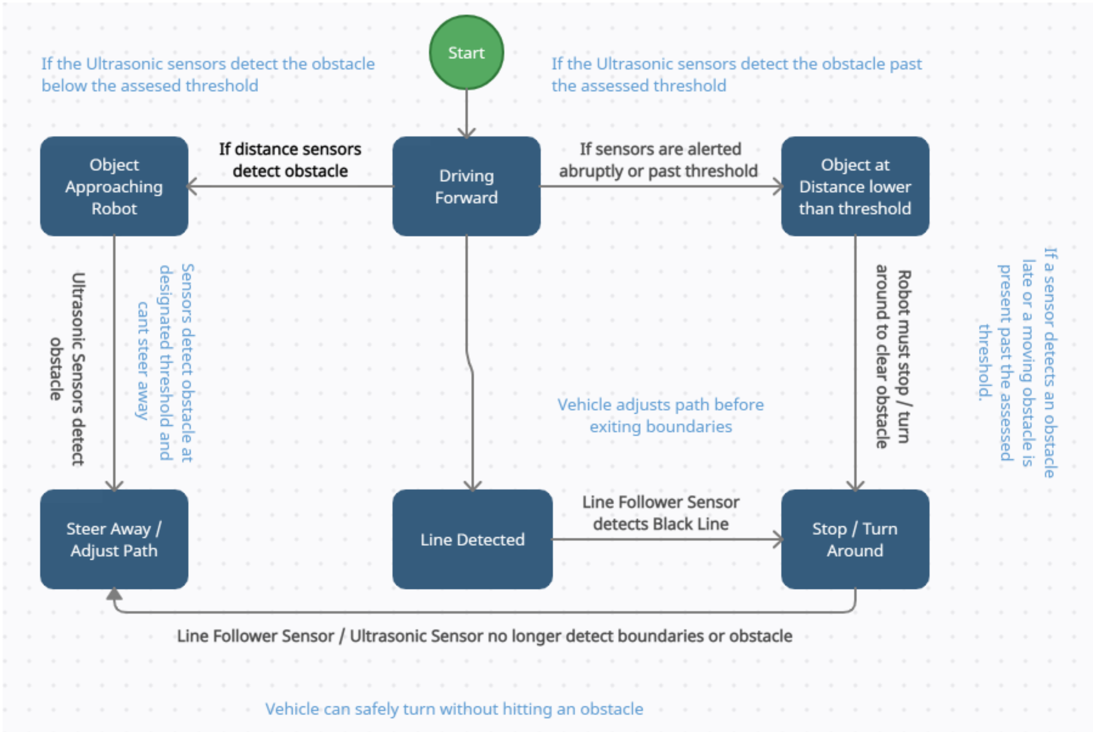

Figure 11: State Chart of System Driving Algorithm

9.0 Sequence Diagrams

The following section details the sequence diagram for our Autonomous snowplow system below:

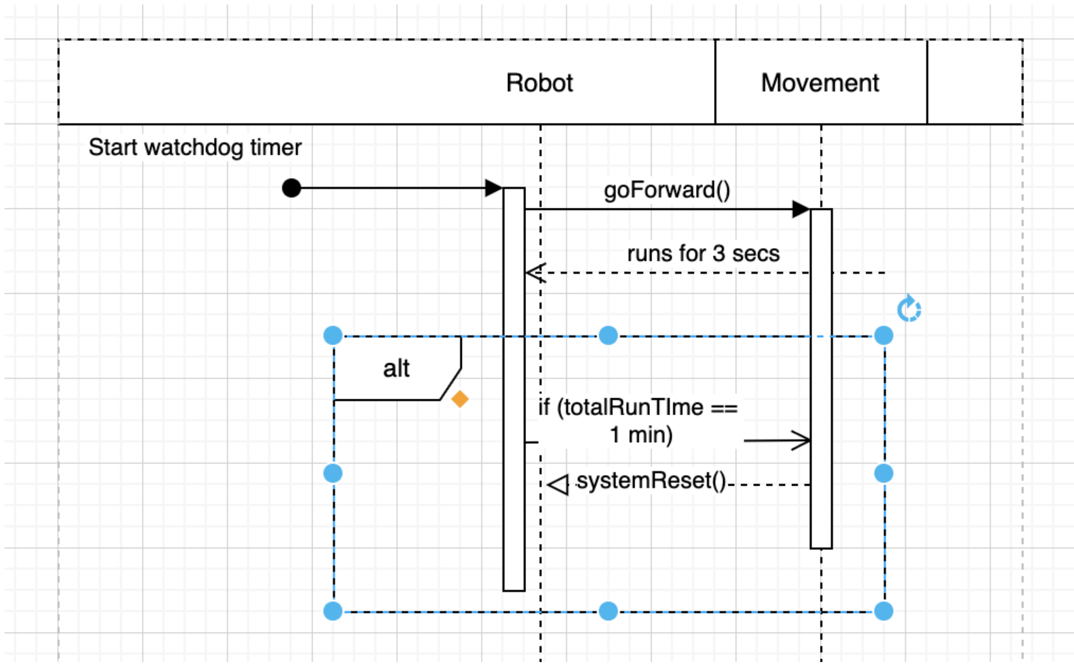

Figure 11: Sequence Diagram for watchdog timer set and reset

The figure above explains our ideal logic for applying the watchdog timer. The vehicle is expected to move forward continuously once the watchdog timer is started. If all sensors are triggered abruptly and the vehicle ends up in a consistent loop in the case of "totalRunTIme == 1min" meaning there has been a function occurring for a maximum 1 minute , then there will be a system reset.

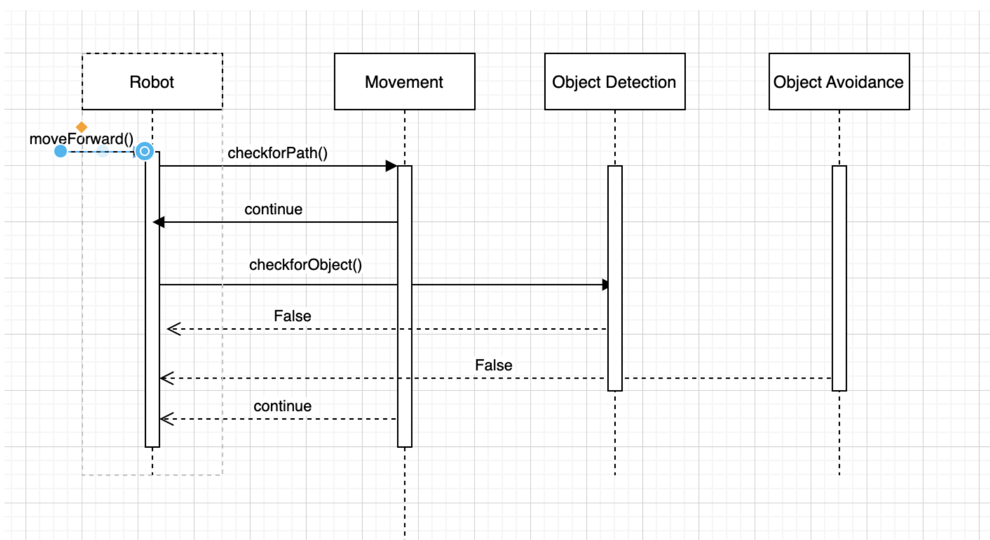

Figure 12: Sequence diagram for no border and no object detected

The figure above explains our main algorithm for no obstacles or lines detected. The vehicle is expected to move continuously forward while the ultrasonic sensors and line sensors are simultaneously detecting an obstacle and boundaries in its way. The threshold for the ultrasonic sensor is set to 15 for obstacle detection. If it senses no object within a 15 cm range of its line of sight, then it continues forward. With the line sensor, if the threshold is less than 750, then it does not detect a black border and continues moving forward.

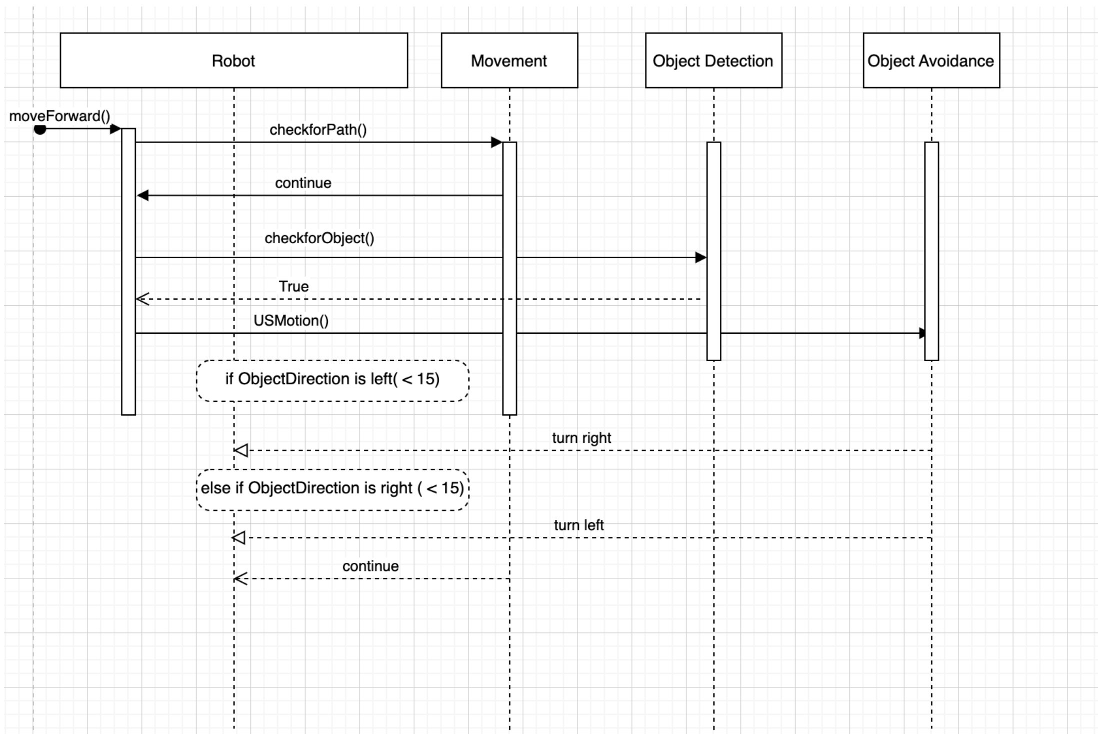

Figure 13: Sequence diagram when object is detected

The figure above explains our main algorithm for obstacles or lines detected. The vehicle is expected to move continuously forward while the ultrasonic sensors and line sensors are simultaneously detecting an obstacle and boundaries in its way. The threshold for the ultrasonic sensor is set to 15 for obstacle detection. If it senses an object within a 15 cm range of its line of sight, then it adjusts its path and steers away from the object. With the line sensor, if the threshold is greater than 750, then it detects a black border, else, it continues moving forward.

10.0 Use of Watchdog Timer

We have shown an implementation of the watchdog timer on our github page in the file “wheel movement for both sides.ino” file. In this file, we have enabled the watchdog timer for 4 seconds and set it to reset in the loop each time there is a delay longer than the threshold of the watchdog timer or there is an infinite loop running within the system.

11.0 Control Charts

The control chart helps the stakeholders and developers the ability to determine points which they can implement corrective actions to prevent future mishaps. The following control charts can be seen below:

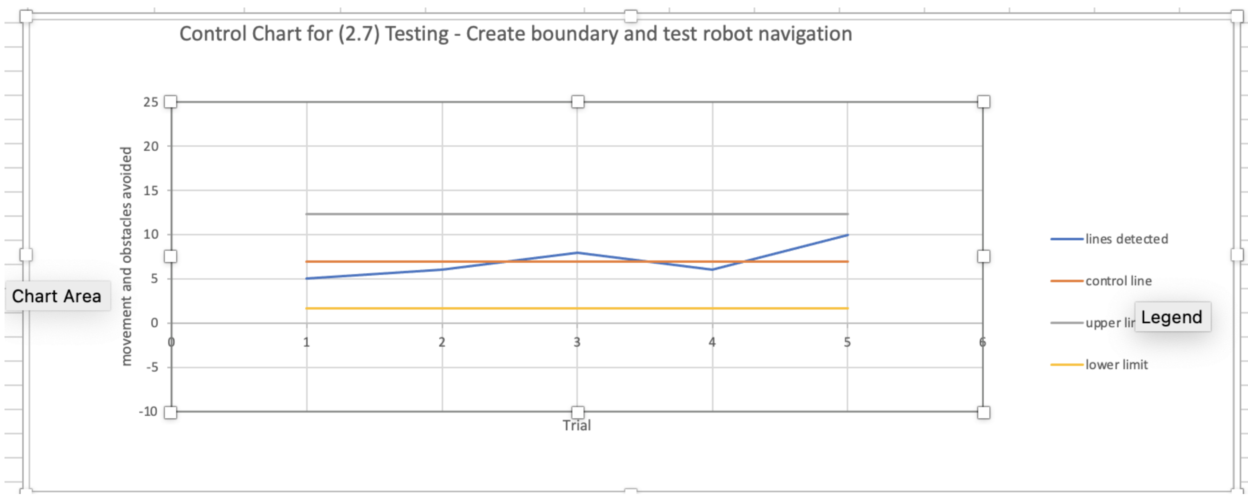

Figure 14: Line test in arena

The chart above tests line detection of the robot. While placed in a sample border, the robot was able to detect the line a maximum of 10 times accurately and 5 times minimum. Due to threshold detection problems, the threshold was decreased and increased as the robot was tested concurrently with the line sensor.

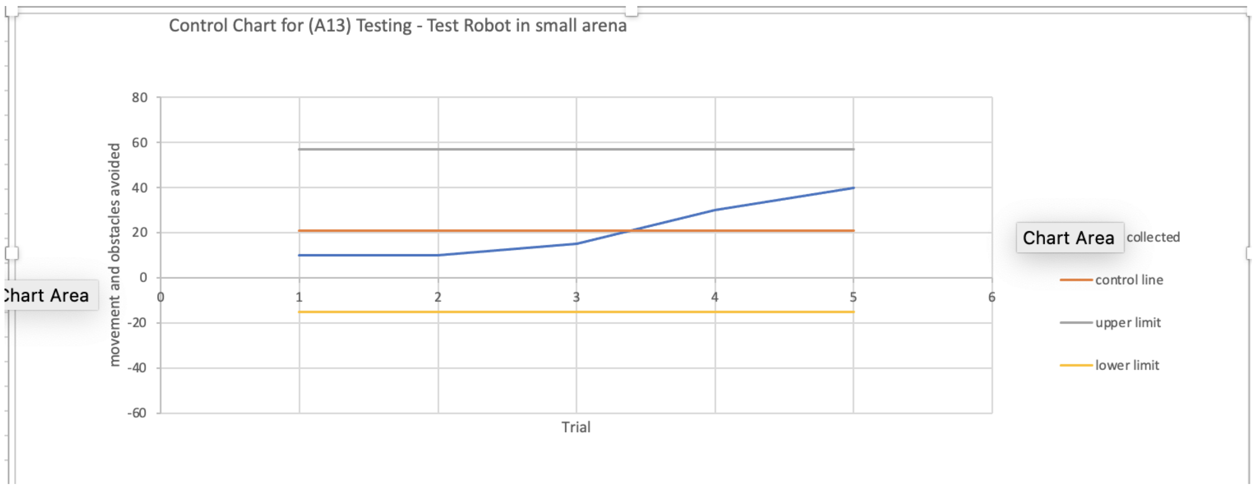

Figure 15: Control chart for Robot in arena

In the above chart, the wheels of the robot was tested in a bounded arena to ensure it can stay within the boundary. Small balls were placed in the arena and the rovbot was able to get a range of 10-40 balls in 5 minutes.

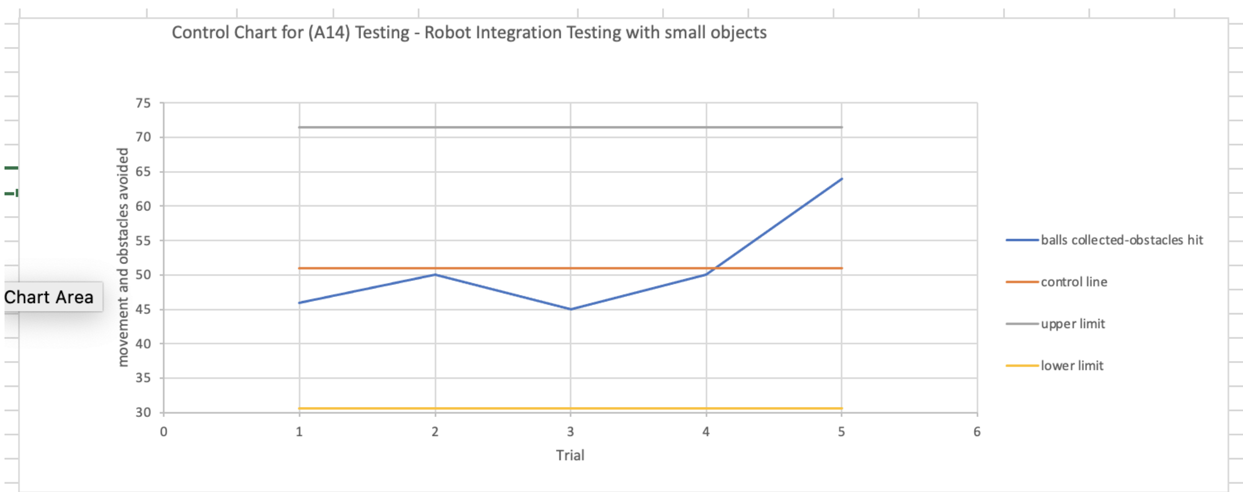

Figure 16: Control chart for Integration Test

The above chart shows the various obstacles hits made by the robot chassis. Due to various challenges with software and hardware components , our overall hit time for obstacles was greater than the control line which was about 52 . The maximum number of balls obtained were 64 and the hit number was about 10 times under normal operation. The upper limit was set to 72 and the lower limit was about 31. The averaged length of balls collected and obstacles hit was 51. Overall the ball was able to perform the required objective.

12.0 Results of System Testing

The testing for the system was performed during Lab 11 and 12. During Lab 11 the robot was not fully applicable for testing since there were concerning issues regarding the wiring of the robot. Moreover, the robot was built and the sensors were mounted but a few of the wires regarding the Cytron Motor Board were malfunctioning thus creating an anomaly when testing. To begin with, when deploying the sensors alongside the motors, a number of the motors would fail to move accordingly. The robot was able to move some of the wheels but not the whole system respectively. After isolating each motor and disassembling the motor driver to represent each connection separately, thus performing motor tests individually.

Furthermore, in between Lab 11 and 12, further testing was done to elaborate on the system and present further mishaps. Once integrating the line follower sensor to the system, the robot was able to move but the line follower sensor was not able to detect the black lines. To achieve better results and allow the robot to detect the line follower sensor to detect the black lines, an analog function was implemented with a threshold that was grasped over several attempts and with different scenarios. Once a final threshold was met, the robot was able to mark and discern the black line in order to keep the robot within the given boundaries. However, implementing a design with two line follower sensors was given to be inaccurate at times, so dropping one of the line followers to create a single line follower system resulted in testing that was deemed more accurate and praised to be more sufficient for the system.

In addition, prior to Lab 12, Integration tests were done to assess the system's response to the added ultrasonic sensors. The system responded well but had some blindspots when an obstacle was heading on straight ahead. When some adjustments were made to the placement of the sensors, the robot was able to detect obstacles and avoid them.

During Lab 12, the customer performed two sets of tests. The first test was a five minute run where the robot was placed in the arena and two obstacles were placed adjacent to each other at the center of this enclosed space. The robot was able to detect the black lines and the obstacles thus avoiding them and staying within the specified boundaries. However, due to some noise and echo from the system, the ultrasonic would receive data which interfered with the system and the robot maintained one corner for the whole majority of the given time. The robot did not hit any obstacle and did not leave the arena. Moreover, the robot was stuck in a loop during its stay in the corner. Nonetheless, the robot was not able to provide its main purpose which was to clear snowballs.

After some adjustments to the system in between the two tests, some problems regarding the line follower sensor arose, and the system was not able to detect any black lines. In order to fix this issue, some changes to the code were made and some thresholds were adjusted. However, the system was not completely fixed which altered the results of the second test. During the second tests, the line follower sensor appeared to be dysfunctional and the robot left the boundaries four times during the run. In addition, the ultrasonics were implemented well but the system hit an obstacle a number of times. Nevertheless, during this second run, the robot was able to displace and clear 64 snowballs during the given timeframe.

13.0 Github Repository

All sample, testing and final code used for the robot can be seen in the group's GitHub repository:

https://github.com/SYSC4805-Fall2022/TeamBlizzardBlueProj

The final demo code can be found in the TaiyeDavcode branch.