SYSC 4805 Final Project Report

L3 Group 3

MAXIMUM YELLOW

December 9, 2022

Corbin Garlough 101101493

Philip Baird 101117932

Shidrath Haider 101102130

Project Charter

Overall objective

The goal of this project is to develop a robot capable of clearing "snow" from an area. A solid black line will mark the boundary of the area, this is so a line follower sensor can detect the edge. Obstacles will be placed in the area to increase difficulty, these include stationary "buildings", and a randomly moving robot. The snow-clearing robot is not allowed to hit any of the obstacles, stationary or moving. Lightweight plastic balls represent the "snow". The robot should be designed with moving components, multiple sensors, and have the ability to make the decisions required to complete the task. Using a variety of sensors, the robot will detect snow and obstacles to inform the path the robot should take.

Overall deliverables

Milestones

- Attach a plow on the front of the robot
- Install IMU for orientation monitoring
- Enable robot to detect buildings using time of flight distance sensor
- Enable robot to detect snow using time of flight distance sensor
- Have the robot navigate the area freely
- Implement an algorithm for path decisions targeting snow
- Enable the robot to detect the enemy robot from all 4 sides
- Implement an algorithm to make movements to avoid enemy robot attacks
- Successfully clear the area of snow without leaving the area, colliding with buildings or being attacked by the enemy robot

Final Deliverables

- (D1) Attach plow (Friday, October 14)
- (D2) Install IMU (Friday, October 14)
- (D3) Project proposal (Due by Friday, October 14 @ 11:59 pm)
- (D4) Robot able to detect buildings and snow (Friday, October 28)
- (D5) Robot navigation (Friday November 4)
- (D6) Path finding (Friday November 11)
- (D7) Progress report (Due by Friday, November 18 @ 11:59 pm)
- (D8) Enemy robot avoidance(Friday November 18)
- (D9) Clear snow (Friday November 18)

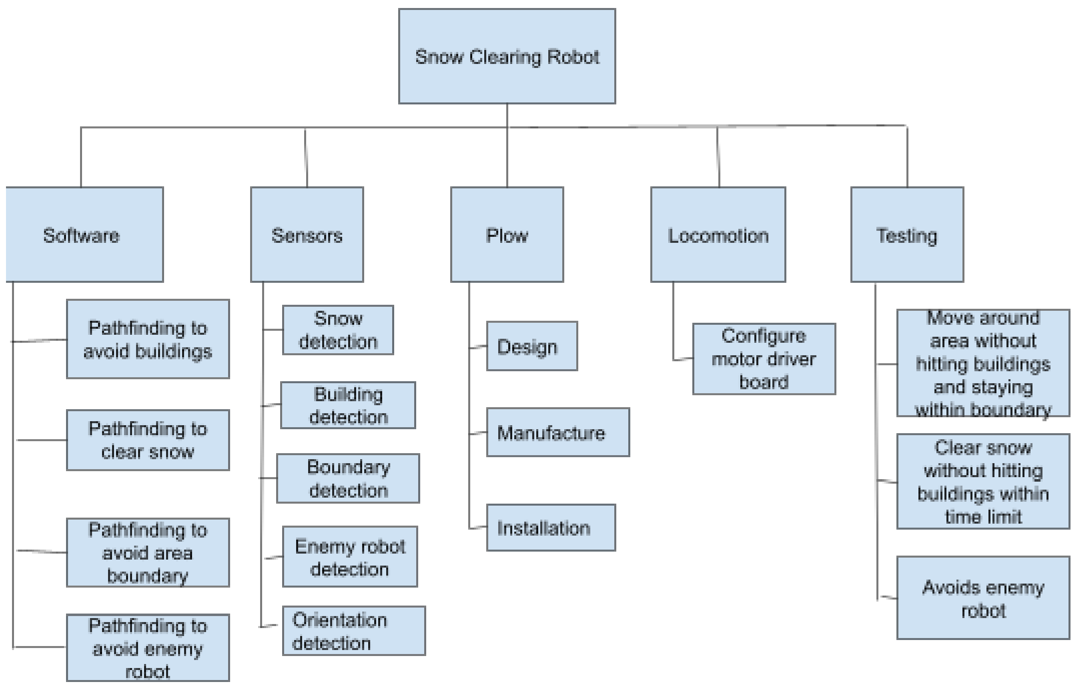
- (D10) Final testing (Friday November 25)
- (D11) Final report (Due by Friday, December 9 @ 11:59 pm)

Scope

Requirements List

- The robot must not travel over 30cm/s
- The robot not not exceed 216 x 252 x 150 mm in size
- The robot shall clear the area of snow in no more than 5 minutes
- The robot shall use line follower sensors to detect the edge of the area
- When a boundary is detected, the robot shall stop and rotate until a path not leaving the area is available
- When clearing snow, the robot shall use distance sensors to detect obstacles and snow.
- The robot shall use ultrasonic sensors to detect the enemy robot moving towards it.
- When the enemy robot is detected coming close to the snow-clearing robot, the snow-clearing robot shall move to avoid a collision.
- An IMU shall be used to detect orientation during snow clearing.

Work Breakdown Structure



Activities List

- Plow design, manufacture, and installation
- Distance sensor installation
- Line follower sensor installation
- IMU installation
- Ultrasonic sensor installation
- Sensor installation verification
- Configure motor connections
- Write driver for distance sensors for differentiating between buildings and snow
- Write driver for line follower for boundary detection and avoidance
- Write driver for IMU for orientation monitoring
- Write driver for movement using motors and encoders
- Write driver for ultrasonics for enemy robot detection
- Coordinate sensor readings for situational awareness
- Create algorithm for path decisions targeting snow, avoiding buildings, and avoiding enemy robot attacks
- Testing

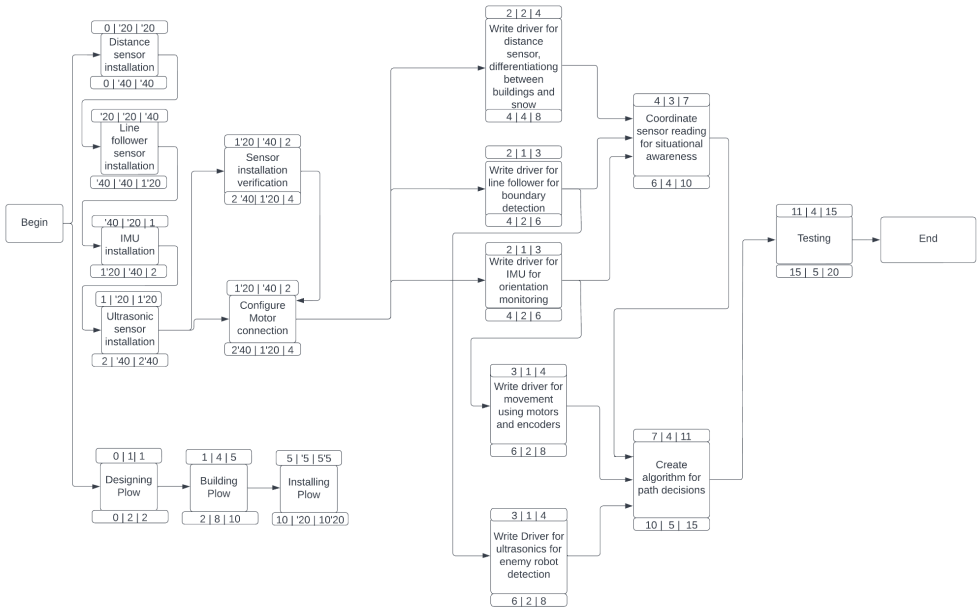
Testing Plan

To ensure success during the final demonstration, rigorous testing is needed. Currently, all components have been individually tested and quantized. It remains that the system needs to be thoroughly tested as assembled. This requires the mounting, wiring, and testing of each component and then the testing as a whole in many situations similar to those that would be encountered in the demo.

- Test the robot's ability to move using the motor driver
  If the robot is able to move forward, backward, turn left, turn right and stop the test is passed, otherwise it fails.
- Test the robot's speed
  If the motors drive forward at the speed used during snow clearing and does not exceed 30 cm/s then the test is passed, otherwise it fails.
- Test the plow's ability to push snowballs
  If the robot is able to move "snow" around the area using the plow the test passes, otherwise it fails.
- Test the orientation detection using the IMU
  If the robot turns a set arc using the IMU for feedback and does not overshoot or stop short of the angle the test passes, otherwise it fails.
- Test the snow detection using the distance sensor drivers
  If the robot is able to detect when a piece of snow is in front of the plow the test passes, otherwise it fails
- Test the boundary detection using the line follower
  If the robot stops and does not cross the area boundary when it is reached then the test passes, otherwise it fails.
- Test the enemy robot detection using the ultrasonic sensors
  If the robot is able to detect the enemy robot approaching and avoid a collision then the test passes, otherwise it fails.
- Test the robot's building avoidance
  If the robot avoids a collision with a building when it is approached then the test passes, otherwise it fails.
- Test the effectiveness of the robot's full operational ability
  If the robot is able to clear the area of snow within 5 minutes then the test passes, otherwise it fails.
- Test the robot's battery life in full operation
  If the robot's battery is able to power it without loss of speed for the whole 5 minutes needed to clear the area of snow then the test passes, otherwise it fails.

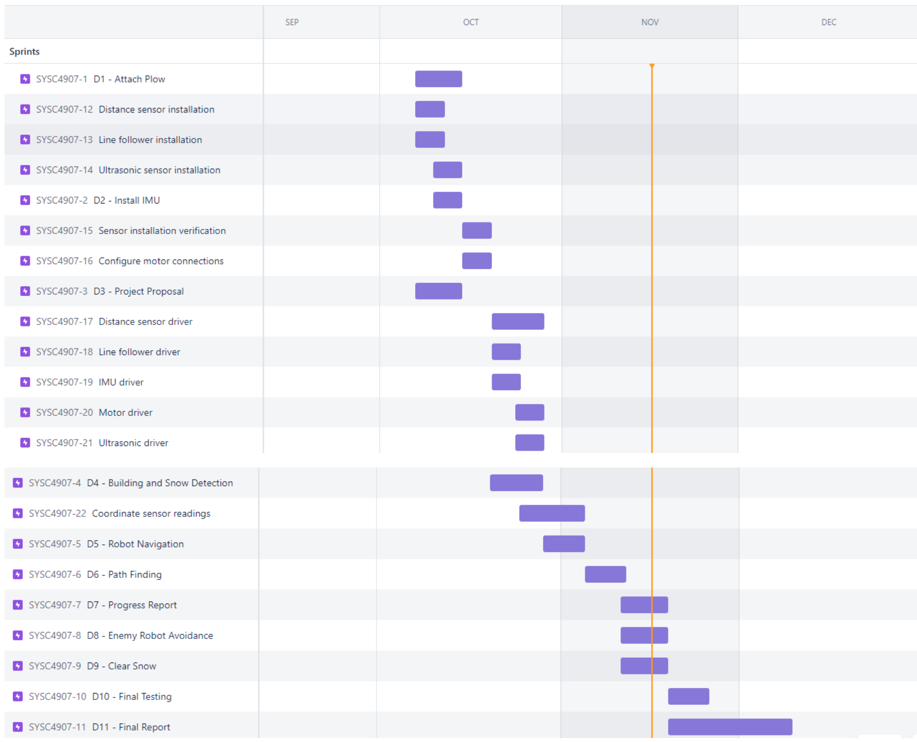
Schedule

Schedule Network Diagram



The diagram shows the relationships between tasks in the development of the robot. It shows that each assembly task was reliant on the previous stage as it would be too crowded to install multiple sensors simultaneously. The plow design was able to be completed independently. The sensor verification and motor configuration were completed at the same time as the drivers were being written. The testing of the drivers occurred after the verification. The coordination of the sensors was successful although the development of the algorithm for path decisions was delayed by the IMU and TOF being disabled by the running motors. This also delayed testing.

Gantt chart



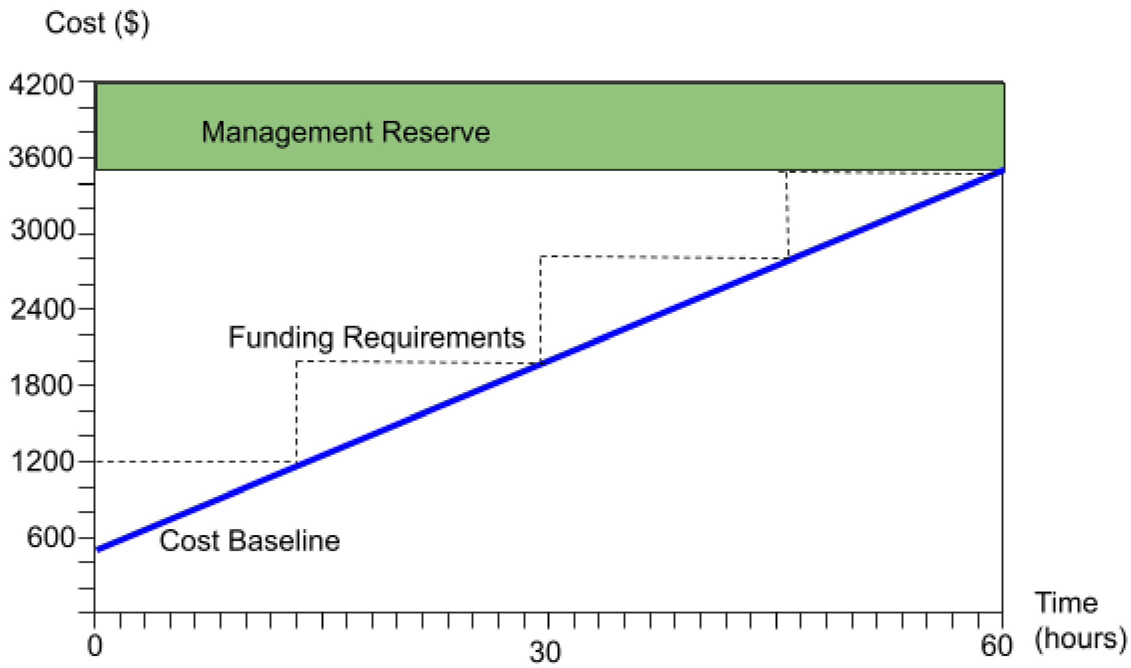
This shows the planned progression for the development of the robot. Each major task has a window of time to be worked on, and the sequence of tasks makes the best use of time by having separate parts of the system developed in parallel.

Cost

Item Description	Amount
Kit of parts	$500
Man hours = (5 labs x 4 hours) x3 members= 60 hours	($50/hour) x 60 hours = $3000
Plow manufacture	$5

Total Estimated Cost: $3505

Cost Baseline figure



We are expecting the project to cost approximately $3500. The initial cost of the robot kit is $500, this includes the main components needed to assemble the frame and also the sensors available to us to use. We are assuming 60 man hours will be dedicated to the project at $50/hour. This means the cost of labor for the project is $3000.

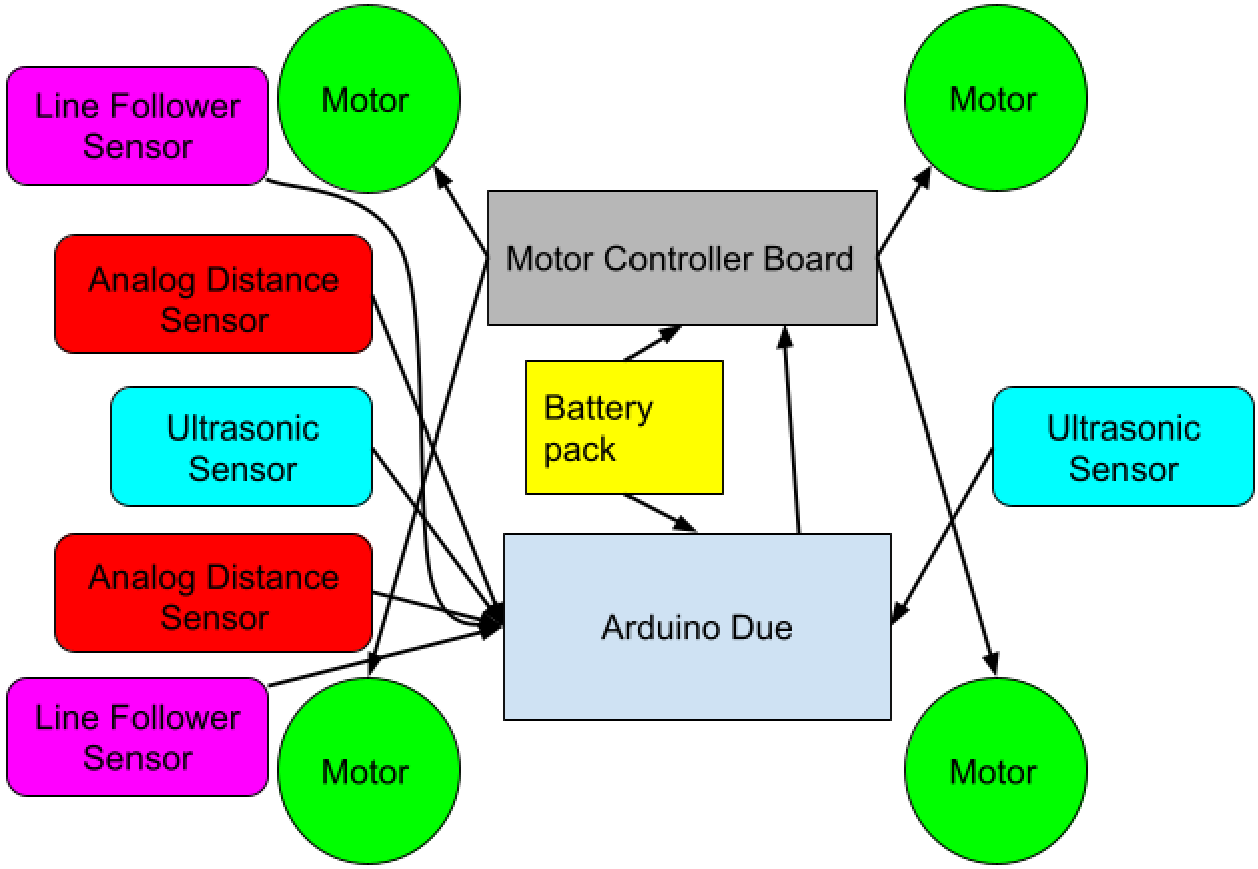
Human Resources

Responsibility Assignment Matrix

Task	Corbin	Philip	Shidrath
Plow design, manufacture, installation	R	A	A
Distance sensor installation		A	R
Line follower sensor installation		A	R
IMU installation		R	A
Ultrasonic sensor installation		R	A
Configure motor connections	R		A

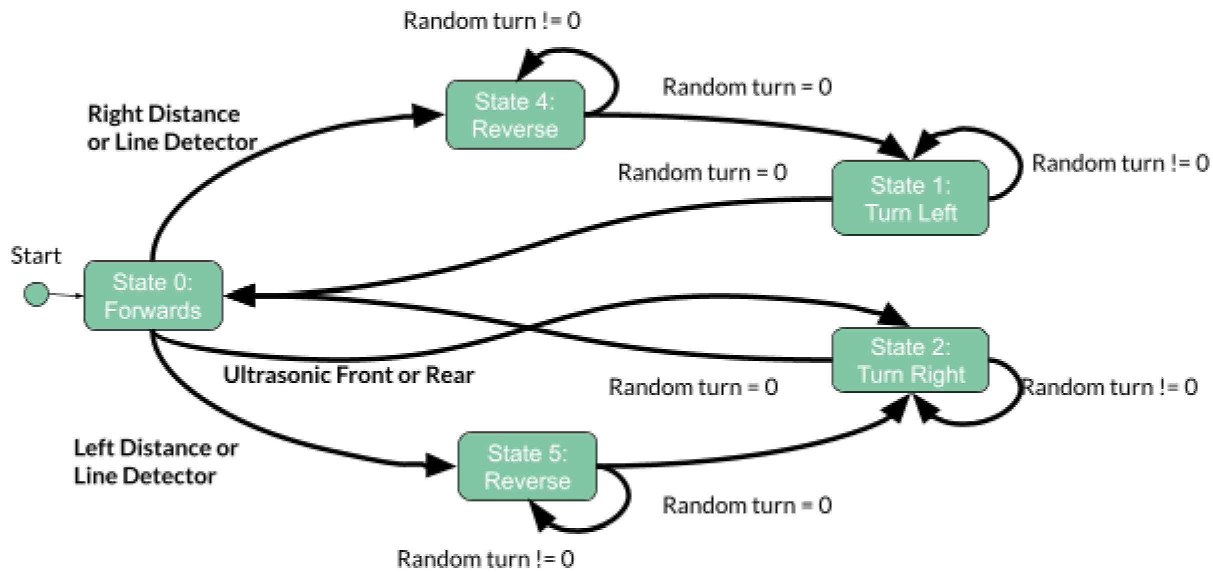
Write driver for distance sensors for differentiating between buildings and snow	A	R	
Write driver for line follower for boundary detection and avoidance		A	R
Write driver for IMU for orientation monitoring	R	A	
Write driver for movement using motors and encoders	R	A	
Write driver for ultrasonics for enemy robot detection	A		R
Coordinate sensor readings for situational awareness	R	A	
Create algorithm for path decisions targeting snow, avoiding buildings, and avoiding enemy robot attacks	A	R	A
Test the robot's ability to move using the motor driver	A		R
Test the robot's speed	A		R
Test the plow's ability to push snowballs	A	R	
Test the orientation and acceleration detection using the IMU	A	R	
Test the snow detection using the distance sensor drivers	R	A	
Test the boundary detection using the line follower		R	A
Test the enemy robot detection using the ultrasonic sensors	R		A
Test the robot's snow targeting		A	R
Test the robot's building avoidance	R		A
Test the robot's coordinated snow clearing ability	A	R	
Test the robot's enemy avoidance		R	A
Test the effectiveness of robot's full operational ability	R	R	R
Test the robot's battery life in full operation	A		R
R = Responsible	9	10	9
A = Approver	9	9	9
Unmarked = Consulted or Informed			

System Architecture



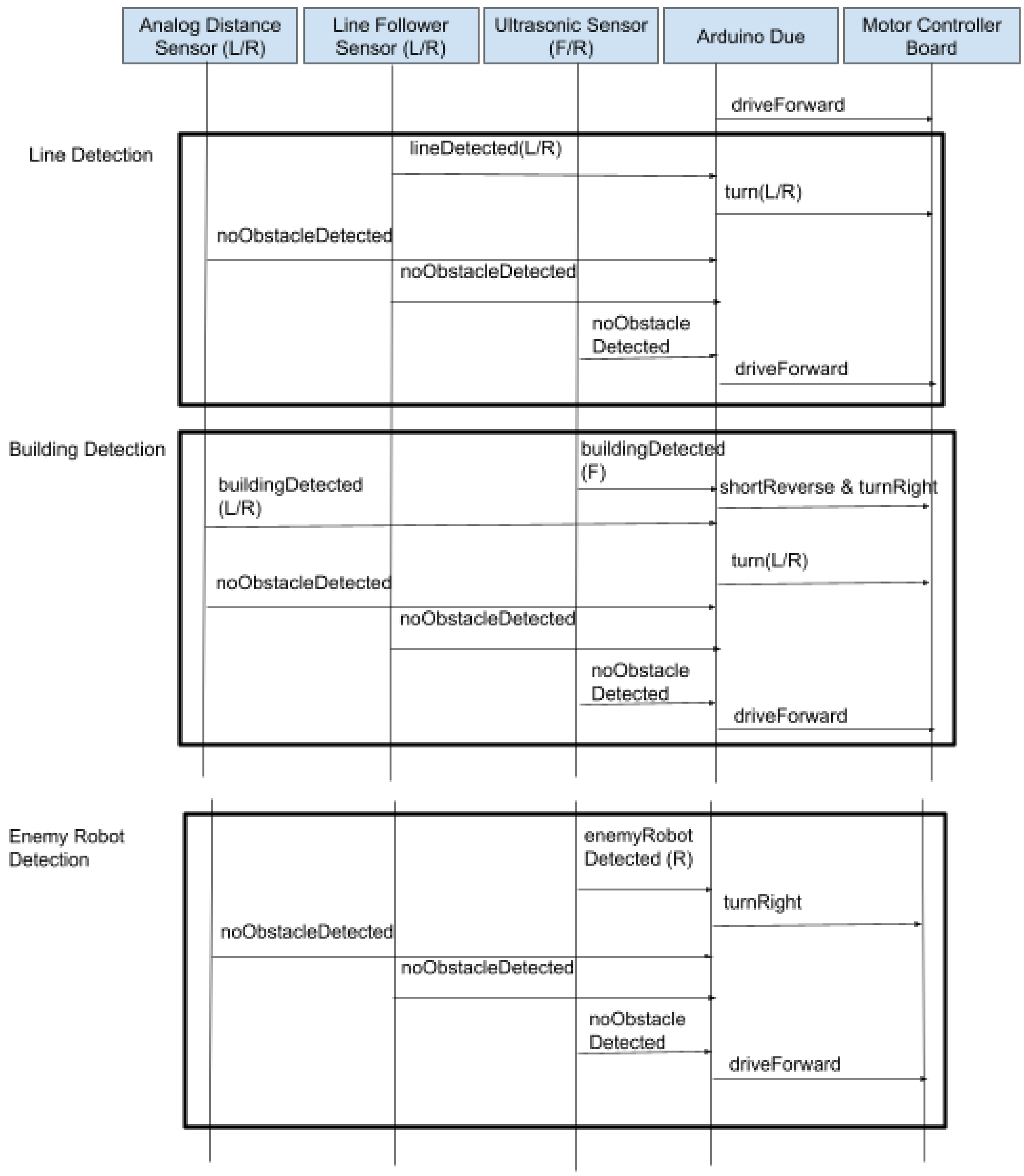
Our robot uses multiple sensors to navigate the area and avoid obstacles. We have 2 line follower sensors, one on each side of the robot, to detect when the robot reaches the perimeter of the area. Having a sensor on each side allows us to determine which side of the robot is on the edge and turn in the proper direction to go back into the area. The analog distance sensors are mounted on either side of the plow and detect buildings at a short distance for fine adjustments when a corner of the plow will collide. With a sensor on each side, we are able to tell which side of the robot is going to hit and adjust our course accordingly so we avoid a collision. The front ultrasonic sensor is used to detect buildings at a distance. The rear ultrasonic sensor is for detecting when the enemy robot is going to hit the robot so that we can change course and avoid a collision. The robot is powered by a battery pack of 5 AA batteries, uses a Cytron Motor Controller Board to drive the 4 motors, and is controlled with an Arduino Due.

Statechart



Our robot starts in the Forward state. If the line follower sensors detect a line, the robot goes to the Reverse state setting the drive pins. If it is the first move instruction it also starts the PWM. A variable randomTurn is set to a random value to randomize the path taken and avoid a situation where the robot executes the same movements over and over when it is stuck. Once randomTurn indicates the movement is finished, the robot moves to the Turn state only setting the drive pins as the PWM should already be running. In the Turn states, the robot will turn until the randomTurn value is 0, then drive Forward. If a building is detected with the ultrasonic sensor, then the robot moves to the Turn state and turns until the randomTurn expires. If the robot sees a building with the analog distance sensors, then it means the building is close and the robot moves to the Reverse state for a short while, and then goes into a Turn state to avoid the building. If the enemy robot is detected with the rear ultrasonic sensor, then the robot moves to the turn right state to avoid the collision. When the randomTurn value reaches 0, it goes back to the Forward state. When entering the state machine into the Forwards state where no sensors detect any obstacles then the robot starts the PWM signal and sets the drive pins to forwards.

Sequence Diagram



The line detection method starts with the line follower sensor detecting the edge of the area on either side of the robot. Depending on which side the line was detected, the robot will turn the opposite way, and once all the sensors indicate that there is no additional obstacle or line being detected, the robot drives forward. The building detection starts with either the front ultrasonic or analog distance sensor detecting the robot approaching a building. First, the robot attempts a short reverse and small turn to the right to avoid the building, but if the analog distance sensors still indicate that a corner of the plow will hit the building, then it turns in the opposite direction to the side on track to collide. Once all the sensors indicate that there is no additional obstacle or line being detected, the robot drives forward. Enemy detection starts with the rear ultrasonic detecting the enemy robot approaching and then turning to the right to avoid a collision. Once all the sensors indicate that there is no additional obstacle or line being detected, the robot drives forward.

Watchdog Timer

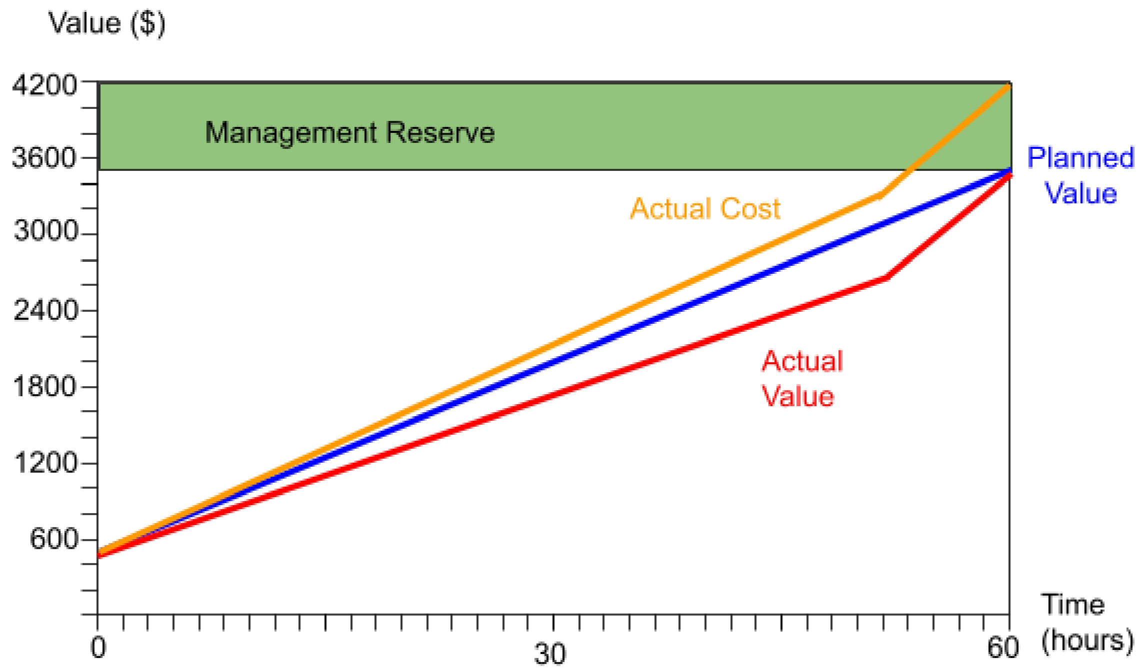
In a computer system a Watchdog timer is used to monitor when the system has encountered a malfunction that causes a task to get held up or stop execution. The timer counts down to 0 with periodic resets, and if a hardware fault or program error causes the timer not to reset, then the system attempts to recover from whatever issue caused the hold-up by going into a safe state. Watchdog timers are used in embedded systems where human intervention is problematic and the system has to be capable of recovering from a fault.

We implemented the Watchdog timer for each sensor reading to avoid situations such as the TOF timeout. A lack of power to the TOF disabling it caused the code to be stuck waiting for the TOF to return a value delaying other functions. The watchdog timer would limit this waiting period as well as block a faulting sensor from blocking again.

```
if(!dl_flag){                        // check if the flag is set
    dl_flag = true;                  // set the flag
    dl = d_sensor1.read();           // read the sensor
    dl_flag = false;                 // unset the flag
    watchdogReset();                 // reset the wd timer
  }
```

In the above snippet is an example of reading the analog distance sensor. If the sensor reading lasts longer than the set timeout, we use 250 ms, then it will reset to the beginning of the loop function. At this point, the flag would not have been reset to false therefore removing the sensor read from future iterations.

Planned Value Analysis



Our Planned Value Analysis shows that the team was over budget and behind schedule until the end of the project. Due to issues with wiring, faulty sensor readings, and issues with power draw, the project took longer than expected. We had to go through multiple versions of solutions before we found one that worked reliably with the sensors available. Towards the end of the project, we had to dedicate additional man-hours to complete the project on time, this resulted in the budget for the project going into the management reserve.

Github Links

Activity	Lead	Link
Distance Sensor Driver	Shidrath	https://github.com/SYSC4805-Fall2022/MaximumYellow_L3G3/commit/c6b09ab47e23fb6bbb4cd9e96e13164e23d8e077
Line Follower Driver	Corbin	https://github.com/SYSC4805-Fall2022/MaximumYellow_L3G3/commit/701f8b7609829be308822b7868b4de71324fa6cc
Motor Driver	Corbin	https://github.com/SYSC4805

		-Fall2022/MaximumYellow_L3G3/commit/edd52ae1394defb783fbb2eeefc89cf7afc5739c
Ultrasonic Driver	Phil	https://github.com/SYSC4805-Fall2022/MaximumYellow_L3G3/commit/d83fbaeb71d9255e84eaf404fe0281e81268bc4f
Coordinate Sensor Readings	Phil	https://github.com/SYSC4805-Fall2022/MaximumYellow_L3G3/commit/3e2d9609166a6be2c0e8e28b2bafb1af73e17fa4
(All other activities do not have portions involving code to put on Github)	Corbin, Phil, Shidrath	N/A

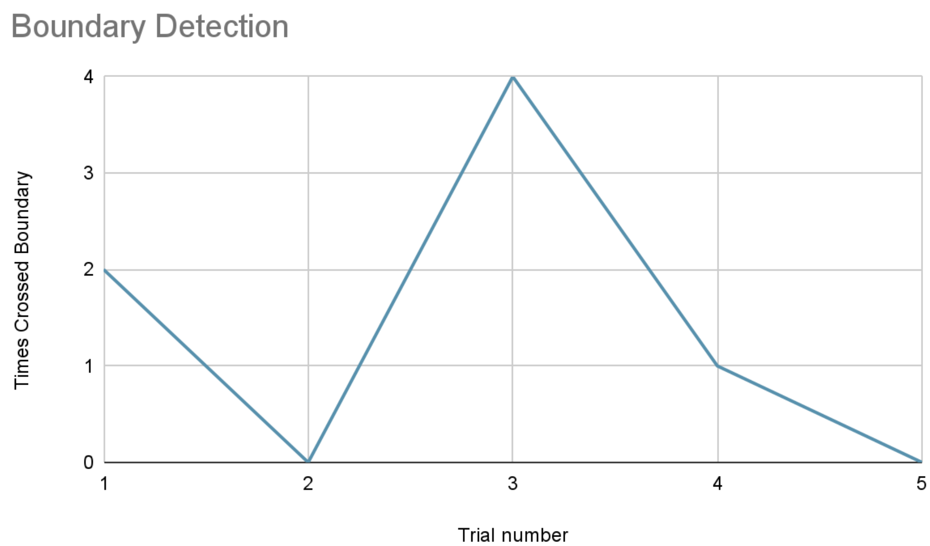
Final Report Additions

Team Member Contributions

Report Item	**Main Contributor(s)**
Overall objective	Corbin
Overall deliverables	Sid
Requirements List	Corbin
Work Breakdown Structure Figure	Corbin
Testing Plan	Corbin
Schedule Network Diagram	Sid and Phil
Gantt Chart	Phil
Cost Figure and Table	Corbin
Responsibility Assignment Matrix	Phil
System Architecture	Corbin
Statechart	Corbin and Phil
Sequence Diagram	Corbin
Watchdog Timer	Phil
Planned Value Analysis	Corbin
Github Links Table	Corbin
Team Member Contributions	Phil
Control Charts	Corbin and Sid
Results of Testing	Corbin

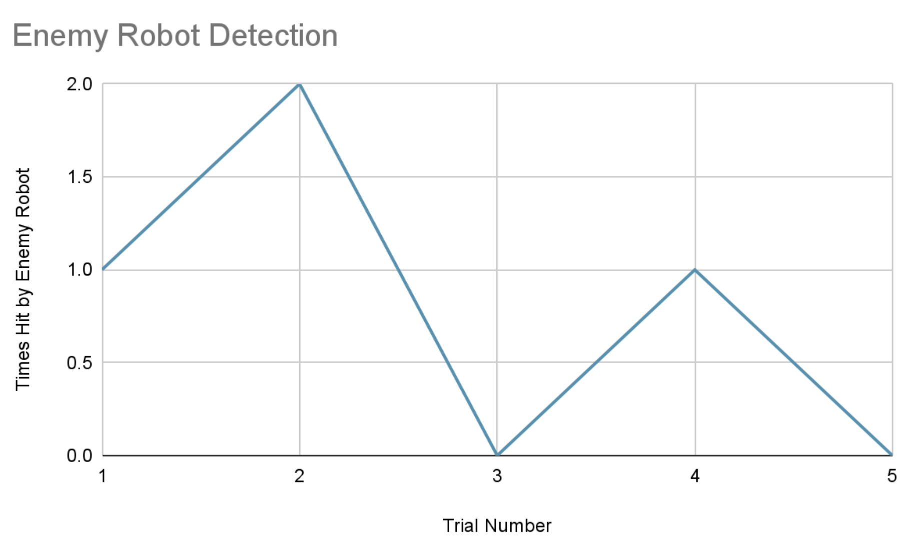
Control Charts

Boundary Detection



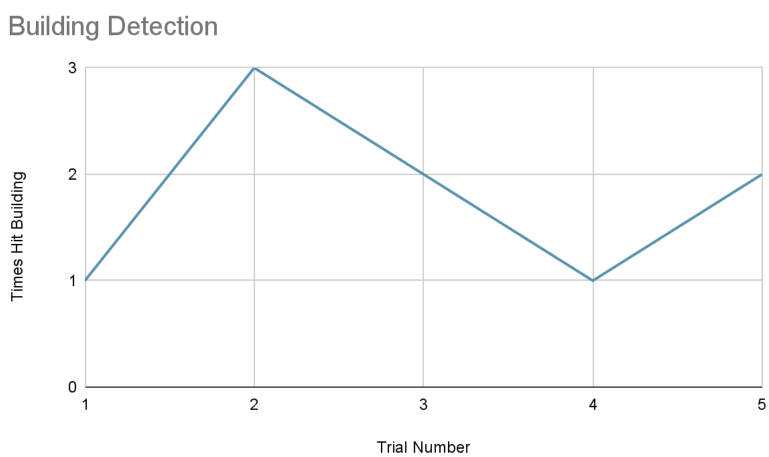
During our boundary detection we ran a number of trials where we tested the number of times our robot crossed the boundary. On average, the robot crossed the boundary twice but we had a successful test in trials 2 and 5. One of the main issues we faced while we ran the robot over the black line is that the robot did not detect the black line when it crossed the line close to parallel, this might have happened due to the middle sensor of the line follower not detecting the line when the robot moves parallel towards the line.

Enemy Robot Detection



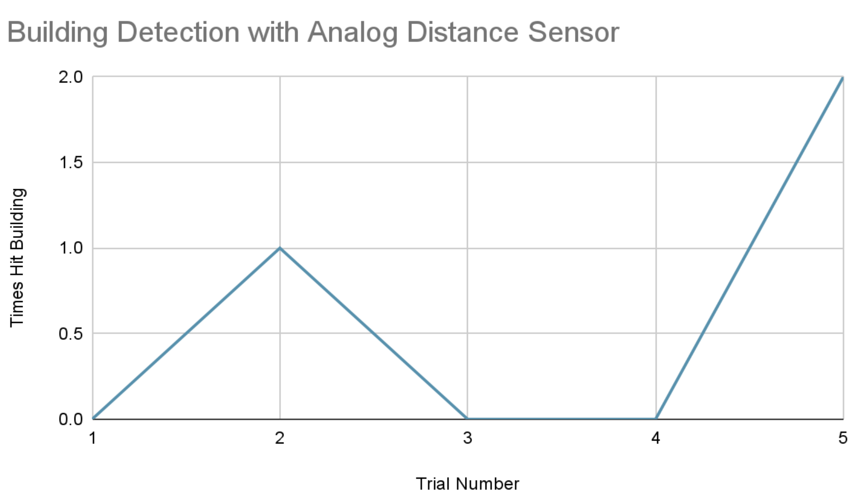
We used a box to represent an enemy robot and moved it towards the robot at various angles and distances to test how the robot would react. When the box approached from the rear, it was reliably detected, but when coming from the sides, the box is not detected. This is a result of the cone of detection projected by the ultrasonic not being at a very wide angle. We repositioned the sensor multiple times in an attempt to increase the detection rate but were not able to get consistent side detection.

Building Detection



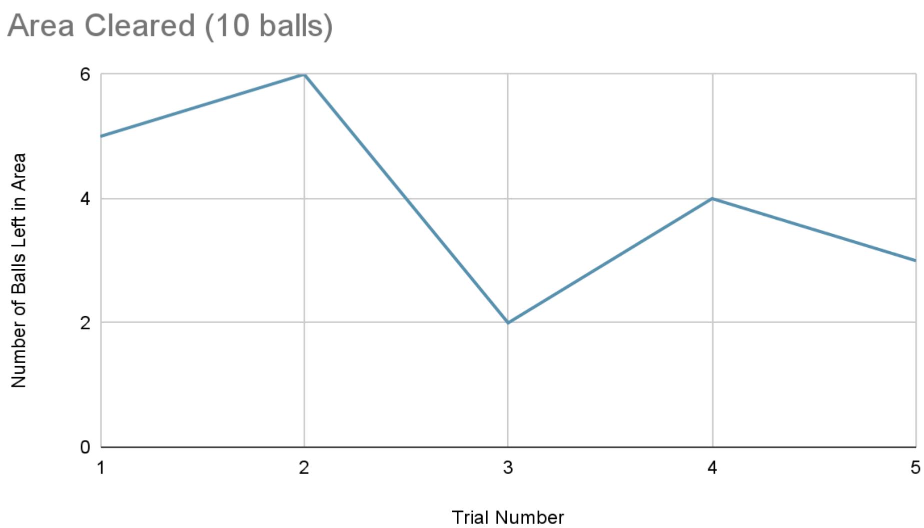
By using only the ultrasonic to detect buildings we had issues with the edge of the plow hitting the building when it is set just off the edge. Since the ultrasonic is placed in the middle of the plow and has a fairly narrow cone of detection, it would miss any box that was not in front of the middle of the plow. This edge clipping issue was significant, so we decided to add the analog distance sensors on the edge of the plow to detect buildings the ultrasonic would miss.

Building Detection with Analog Distance



Once the analog distance sensors were added to the plow, the building detection was much more consistent. It was only very specific angles and placements that resulted in the building not being detected. These situations were difficult to recreate in natural trials, and are confident the robot will perform properly during the demo.

Area cleared



During our testing we managed to clear approximately 50 percent of the snow from the area. In our test the limitation was 10 snowballs, if we had more snowballs then we could have had a more reliable result in our testing on how many more balls we could remove.

Results of Testing

	Trial Number (P/F)						
Test	1	2	3	4	5	In Lab Demo 1	In Lab Demo 2
Movement	P	P	P	P	P	P	P
Speed	F	P	P	P	P	P	P
Plow function	P	P	P	P	P	P	P
Boundary Detection	F	P	F	F	P	P	F
Enemy Robot Detection	F	F	P	F	P	n/a	n/a
Building Detection (with analog distance sensor)	P	F	P	P	F	P	F
Area Cleared	F	F	F	F	F	F	F
Battery Life	P	P	P	P	P	P	P

Testing showed that our robot was constantly able to maneuver forward, backward, turn left and turn right. The speed of our robot was initially faster than the maximum 30 cm/s limit, but this was easily fixed by reducing the duty cycle driving the motors. Our plow worked as planned, being able to push snow out of the area and hold snow in the plow with the box wings on its sides.

During our testing we found that our robot had difficulty handling some specific situations encountering a boundary, having the enemy robot approach it, and approaching buildings. The boundary detection works very well when the robot approaches the line at a steep angle, but when running close to parallel to the boundary, the line was not always detected properly. This is because we used the edge and middle detectors of the line follower to reduce false positives when moving within the boundary, and the tape defining the boundary is ever so slightly thinner than the spacing of the detectors.

Having an enemy robot approach the robot directly from the side resulted in the enemy not being detected. This is because we only have an ultrasonic sensor on the rear of the robot, and its cone of detection is not wide enough to see the sides of the robot. During the in-lab demo, this was not an issue because the enemy robot was not in play.

Our robot is able to detect buildings fairly reliably using analog distance and ultrasonic sensors, but there are cases where the building can be at a certain angle and offset slightly to the side of the robot. This causes the analog distance sensor to see the building as being outside the threshold for detection, and the ultrasonic pulse reflects off the side at an angle so it cannot detect the building either. This was difficult to recreate these issues in natural scenarios so we deemed it not important enough to address.

We were never able to fully clear the area of snow, but we came close to the goal and are confident the robot will clear a significant portion of snow. The battery life of the robot was not an issue in any of our tests.