Autonomous Snow-Clearing Robot

Final Report

SYSC4805A

Prof. Abdullah Kadri

Waqas Shahzad 101224001

Alexander Tratnik 101220933

Emeka Anonyei 101209704

Haozhe Li 101148446

Carleton University

December 04, 2025

1. Overall Objective

This project aims to design and build an autonomous robot that can clear simulated snow from a $6m^2$ area without colliding with obstacles. The robot will operate within specific size, speed, and time limits, using sensors and a plow to detect and push snow cubes out of the area. The system will be developed in modular stages, with clear testing and integration plans to ensure reliability. Success will be measured by how much snow is cleared and how well the robot avoids penalties during operation.

2. List of Deliverables

Table 1: Milestone Deliverables

Activity ID	Activity Name	Description
M1	Project Proposal	A document includes the detailed team plans for the project, due on Oct 16.
M2	Hardware Prototype	Partially assembled robot (chassis, sensors, motor drivers, encoders) tested individually. Due on Nov 11 (Lab 8)
M3	Progress Report	A document that includes the updated project proposal, overall system architecture, state chart and sequence diagrams, watchdog timer demonstration, updated value analysis, working code on GitHub, and unit testing results. Due on Nov 13
M4	Software Integration	MCU communication verified; all sensors calibrated and integrated; basic navigation functional. Due on Nov 20

Table 2: Final Deliverables

Activity ID	Activity Name	Description
F1	Presentation	Present and discuss the final project design, implementation results, and applied engineering techniques during the scheduled lecture session. On Nov 18, 25, or Dec 2.
F2	Final Demonstration	Demonstrate the completed robot in the arena, showcasing obstacle avoidance, boundary detection, and snow clearing performance. On Nov 28
F3	Final Report	A final report includes updates from previous documents, design, implementation, testing, and results of the robot. Due on Dec 4

This section lists the project deliverables with brief descriptions and due dates. The deliverables are the milestone deliverables and the final deliverables.

3. List of Requirements

Table 3: Functional Requirements

Requirement ID	Requirement Statement
RQ-01	The robot shall clear simulated snow cubes from the enclosed area within 5 minutes of activation.
RQ-02	When an obstacle is within 10 cm, the robot shall stop movement to avoid collision within 500 ms.
RQ-03	The robot shall avoid static and dynamic obstacles without making contact.
RQ-04	The robot shall maintain speed below 30 cm/s throughout the operation.
RQ-05	The robot shall push snow cubes encountered during operation outside the perimeter using the mounted plow.

RQ-06	The robot shall avoid boundary violations by keeping all wheels within 5 cm of the perimeter edge.
RQ-07	The robot shall operate continuously for up to 5 minutes without requiring a reset or adjustment.

Table 4: Non-Functional Requirements

Requirement ID	Requirement Statement
RQ-08	The robot shall operate within dimensions not exceeding 226 mm (W) × 262 mm (L) × 150 mm (H).
RQ-09	The robot shall incur zero penalties by complying with all physical and operational constraints.
RQ-10	The robot shall be capable of autonomous operation without human interaction during the test.
RQ-11	The robot shall maintain modularity by separating sensing, actuation, and controllogic.
RQ-12	The robot shall provide consistent performance across multiple tests runs under identical conditions.

4. List of Activities

Table 5: List of Activities

ID	Activity (from your schedule)	Description
A1	Define Requirements & Scope	Identify and define the robot’s key functions, design objectives, and operating constraints.
A2	Design Plow Mechanism	Design a plow mechanism capable of carrying snow cubes efficiently within the arena boundary, with proper specification.
A3	Fabricate & Mount Plow Hardware	Install the plow to the robot chassis.
A4	Test Sensors (IR, Ultrasonic)	Test and actuate each sensor.

A5	Configure Sensor Calibration	Adjust and calibrate sensor thresholds.
A6	Integrate Wheel Encoder with MCU	Connect the encoder to the MCU
A7	Integrate MCU & Sensors (Comm. Testing)	Connect each sensor to the MCU and verify the data transmission.
A8	System Integration & Debugging (Software)	Implement system functions, verify overall software operation, and debug integration issues.
A9	Navigation testing and validation	Implement and validate the navigation functions, focusing on path planning and obstacle avoidance; debug and optimize the algorithms for stable motion.
A10	Final System integration (Hardware and Software)	Install and secure all hardware components in their proper positions and organize and refine all software functions for complete system integration.
A11	Full System test (End -to- End Run)	Perform final testing of the robot prototype in different practical arenas to evaluate performance and optimize or debug as needed.
A12	Documentation & Report Preparation	Compile and organize all project documentation. Write the final report summarizing design, implementation, testing results, and project feedback.
A13	Proposal Review & Final Submission	Review and proofread all documents for consistency and accuracy. Finalize formatting and complete the submission.

5. Testing Plan

Table 6: Unit Testing Plan

Test ID	Component	Test Case Description	Test Procedure	Success Criteria

UT-01	IR Obstacle Avoidance Sensor [1]	Detect any obstacles within a 10 cm range	Place various objects at 5, 10, and 15 cm and observe the outputs	Output changes with a threshold of 1cm, with false readings occurring less than 5% of the time
UT-02	Line Sensor Module [1]	Detect a black tape over a white surface	Move the sensor along a black taped line over a white floor	The sensor should output high on black and low on white at least 95% of the time
UT-03	Ultrasonic Distance Module [2]	Measure obstacle distance from 10 to 100 cm	We will place different objects at 10, 30, 50, 75, and 100 cm and read the output	All reading should be stable with a +/- of 2 cm
UT-04	Distance Measuring Sensor [2]	Measure nearby distances of 5-40cm	Move the obstacle incrementally from 5-40 cm and read the output	The sensor's output voltage should match the datasheet's curves
UT-05	Time of Flight Sensor [2]	High precision distance reading for obstacle mapping	Compare ToF readings with a ruler's measurements from 5 - 200cm	Readings should have a confidence interval of +/- 2cm
UT-06	IMU [2]	Measure the robot's orientation and motion	Rotate and tilt the robot and compare the output	The error margin of the orientation should be less than 5 degrees
UT-07	Wheel Encoders [3]	Measure wheel speed and distance	Spin wheels manually and with a motor to know revelations	Ensure that both revelations match up

UT-08	Motor Driver [3]	Controls left and right DC motors	Send PWM signals of 0-255 and measure the RPM	Ensure no jittering and find a PWM that makes the robot move no more than 30cm/s
UT-09	Logic Level Shifter [3]	Ensures safe communication between Arduino Due and Nano every	Send high and low shifts in both directions	Output logic is correct with no dropout or delay
UT-10	Arduino Nano Every [3]	Handles low-level sensors and motor controls	Upload test sketches of sensors	Stable operation
UT-11	Arduino Due [1]	Performs high-level sensor operations	Upload test sketches of sensors	Stable operation
UT-12	Plow	Test mechanical stability and cube displacement	Push 20mm cubes at about 10cm/s	Pushes more than 3 cubes without jamming or lifting, and has no structural failure

Table 7: Integration Testing Plan

Test ID	Stage	Subsystems Included	Purpose	Test Description	Success Criteria
IT-01	Motion Control	Motor Driver + Wheel Encoder + IMU	Ensure straight line motion, turning, and speed	Make the robot go in a 30cm straight path and then turn 90 degrees	Path deviation less than 5 cm and angle error less than 5 degrees

IT-02	Boundary Detection	Line Sensor + Motor Control	Detect and avoid crossing any black boundary	The robot will drive towards a black line	Robot stops within 2 cm of the line
IT-03	Fixed Obstacle Detection	IR Sensor + Ultrasonic + ToF	Combine multiple range sensors	Place obstacles of 10-50 cm and compare detection	All sensors should show between +/- 10cm
IT-04	Obstacle Avoidance	Motor Driver + Wheel Encoder + IMU + IR Sensor + Ultrasonic + ToF	Detect an obstacle and turn away from it	Have the robot drive towards an obstacle, and on detection, have it turn and move away	The robot should not collide with any obstacle
IT-05	Communication Bridge	Arduino Due + Arduino Nano + Logic Level Shifter	Ensure data flow between both boards	Send sensor data between both boards every 100ms with actions	No board should miss any command coming between the two
IT-06	Snow Plowing	Motor Driver + Wheel Encoder + IMU + Plow	Verify plow alignment and pushing capability	Push cubes from the center to the boundary	Able to clear 90% of the cubes within 5 minutes of a test boundary
IT-07	Dynamic Obstacle Detection	Motor Driver + Wheel Encoder + IMU + IR Sensor + Ultrasonic + ToF	Detect and avoid moving obstacles	Move and obstacle along the robot's path while it's moving	The robot should stop moving upon detection
IT-08	Full System Testing	All Components	Ensure all tasks are performed at once	Create a test scenario and have the robot pass	Robot triggers no penalties and completes the scene

					within 5 minutes
IT-09	Speed and Boundary Testing	Motor Driver + Wheel Encoder + IMU + Line Sensor + Motor Control	Ensure we stay at a speed of 30 cm/s and stay inside the black lines	The robot stays at full speed during a trial run	Speed doesn't drop below 25cm/s, and the wheel doesn’t come 2 cm from the line

6. Schedule Network Diagram

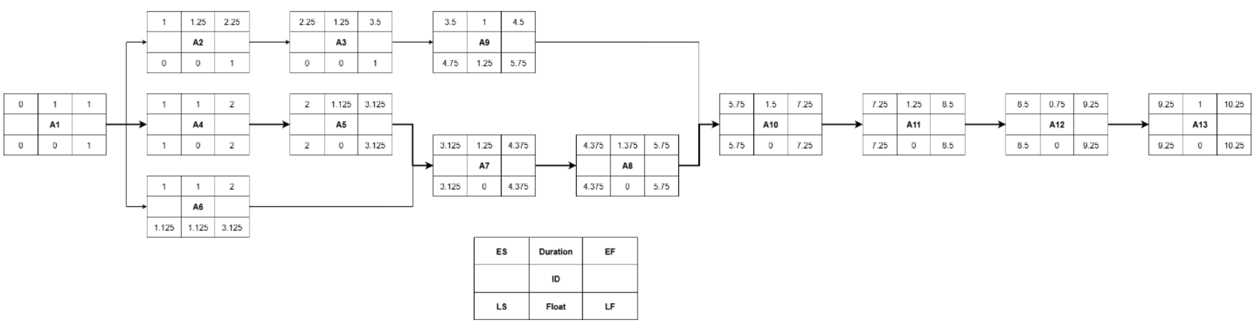

Figure 1: Schedule Network Diagram

Figure 1 outlines the sequence and dependencies of project activities A1 to A13, showing how each task flows into the next. Each node includes timing metrics including early start (ES), early finish (EF), late start (LS), and late finish (LF), duration, and float to help identify the critical path and scheduling flexibility. All durations are based on work units, where 1 unit equals 8 hours of team effort.

7. Gantt Chart

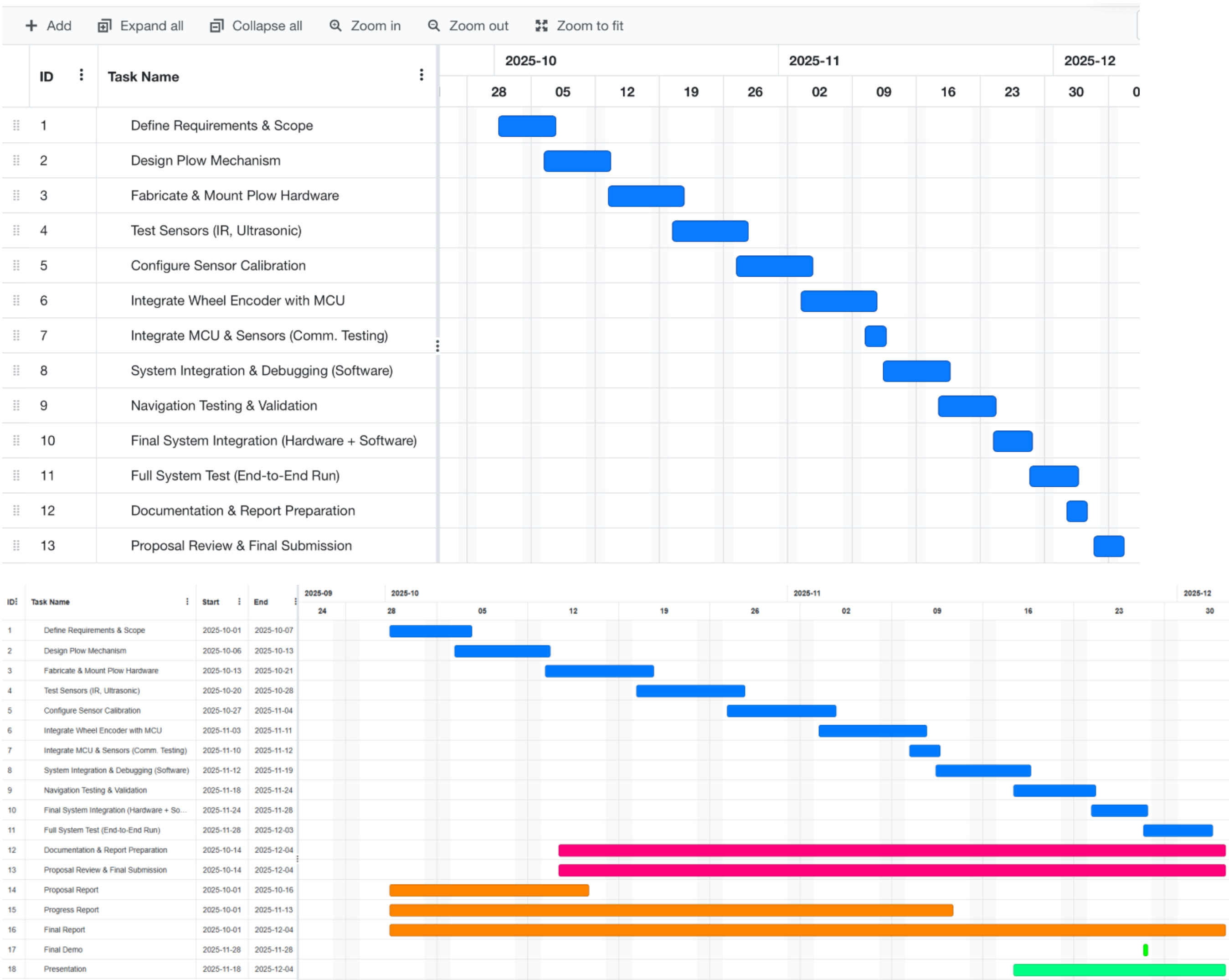

Figure 2: Gantt Chart

8. Cost Baseline

Table 8: Cost Activity Baseline Estimation

ID	Activity	Estimated Hours	Rate ($50/hr) x4 dev	Cost	Total Cost
C1	Define Requirements & Scope	8 hrs	$50/hr	$1,600	$1,600
C2	Design Plow Mechanism	10 hrs	$50/hr	$2,000	$3,600
C3	Fabricate & Mount Plow Hardware	10 hrs	$50/hr	$2,000	$5,600
C4	Test Sensors (IR, Ultrasonic)	8hrs	$50/hr	$1,600	$7,200
C5	Configure Sensor Calibration	9 hrs	$50/hr	$1,800	$9,000
C6	Integrate Wheel Encoder with MCU	8hrs	$50/hr	$1,600	$10,600
C7	Integrate MCU & Sensors (Comm. Testing)	10 hrs	$50/hr	$2,000	$12,600
C8	System Integration & Debugging (Software)	11 hrs	$50/hr	$2,200	$14,800
C9	Navigation Testing & Validation	8 hrs	$50/hr	$1,600	$16,400
C10	Final System Integration (Hardware + Software)	12 hrs	$50/hr	$2,400	$18,800
C11	Full System Test (End-to-End Run)	10 hrs	$50/hr	$2,000	$20,800
C12	Documentation & Report Preparation	6 hrs	$50/hr	$1,200	$22,000

C13	Proposal Review & Final Submission	8 hrs	$50/hr	$1,600	$23,600

9. Responsibility Assignment Matrix (RACI)

Table 9: Responsibility Assignment Matrix

ID	Activity (from your schedule)	Responsible (R)	Approver (A)
RE1	Define Requirements & Scope	Waqas	
RE2	Design Plow Mechanism	Alec	
RE3	Fabricate & Mount Plow Hardware	Waqas	
RE4	Test Sensors (IR, Ultrasonic)	Emeka	
RE5	Configure Sensor Calibration	Haozhe	
RE6	Integrate Wheel Encoder with MCU	Alec	
RE7	Integrate MCU & Sensors (Comm. Testing)	Emeka	
RE8	System Integration & Debugging (Software)	Haozhe	
RE9	Navigation testing and validation	ALL	
RE10	Final System integration (Hardware and Software)	ALL	
RE11	Full System test (End -to-End Run)	ALL	

RE12	Documentation & Report Preparation	ALL	
RE13	Proposal Review & Final Submission	ALL	

Project Progress Report

10. Overall System Architecture

The overall system architecture of the autonomous snow-clearing robot is designed as a modular, layered system that combines sensing, control, and actuation in a closed feedback loop. The hardware architecture consists of two microcontrollers — an Arduino Due serving as the high-level controller and an Arduino Nano serving as the low-level controller — connected through a logic level shifter to ensure reliable 3.3 V to 5 V communication.

The Due integrates data from IR, ultrasonic, Time-of-Flight, IMU, and line sensors to interpret the robot's surroundings and make navigation decisions. The Nano executes these decisions by controlling the motor driver and wheel encoders, ensuring that the robot maintains its speed under 30 cm/s and stays within the boundary. The plow mechanism, mounted at the front, is driven indirectly through motor control logic to push wooden cubes toward the perimeterSoftware is organized into four primary layers: perception (sensor data collection), decision-making (navigation and avoidance logic), actuation (motor and plow control), and safety (watchdog timer and boundary enforcement). Data flows cyclically from sensors to controllers to actuators and back through feedback sensors, allowing the robot to adapt continuously to dynamic obstacles and ensure reliable performance under the 5-minute operation constraint.

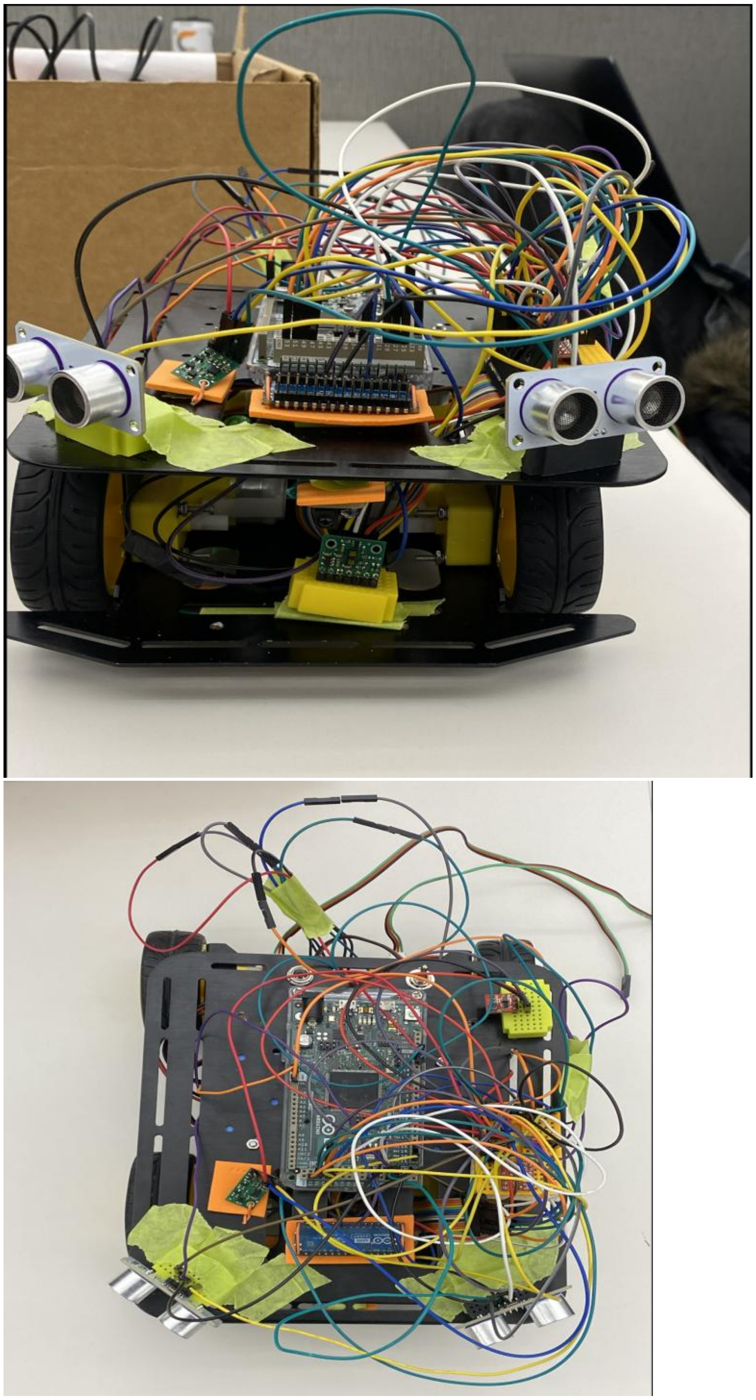

Figure 3: Overall Systems Architecture

11. Statechart and a sequence diagram

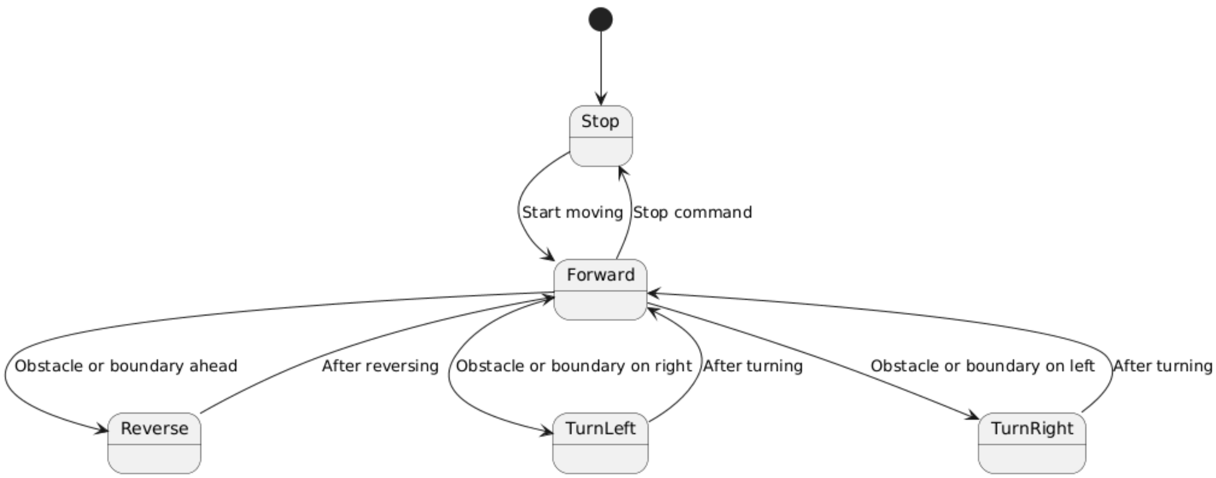

Figure 4: Low-Level State Diagram

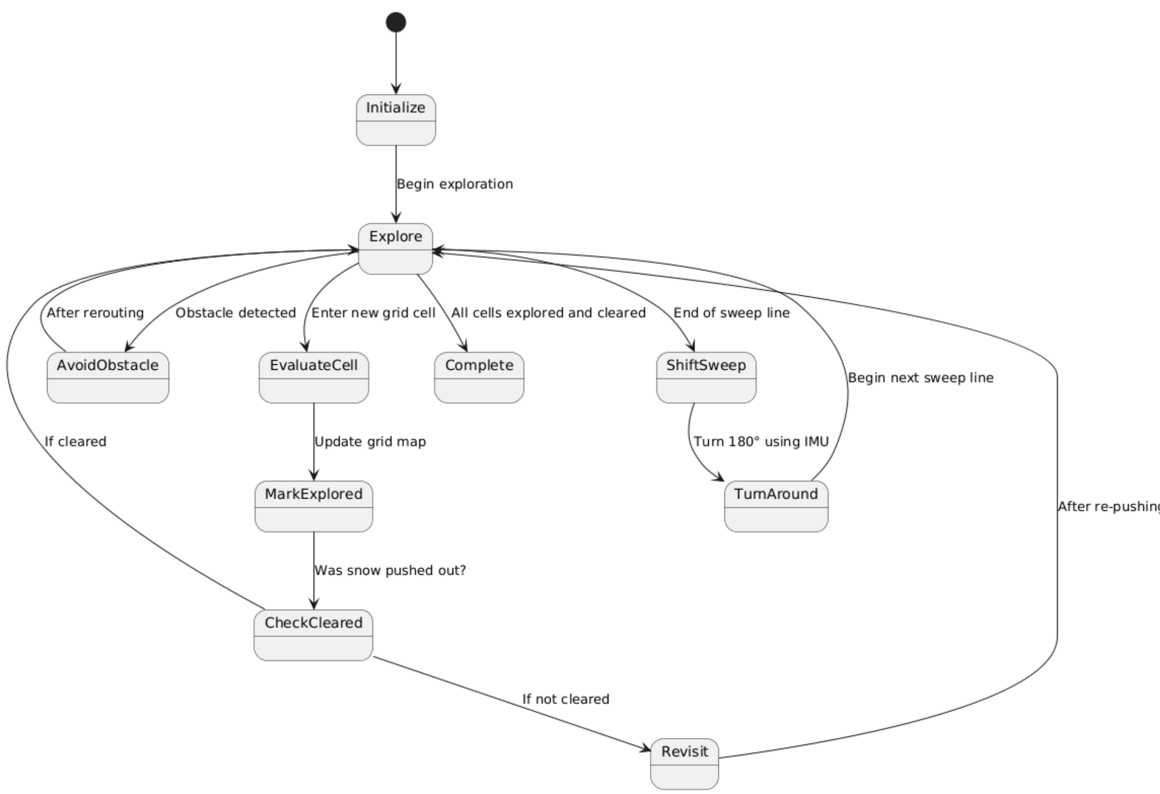

Figure 5: High-Level State Diagram

Figure 4 and Figure 5 below show the state machine diagrams for the robot. Figure 4 shows the low-level state machine for basic movement of the robot. Figure 5 shows the high-level state machine diagram based on the navigation strategy.

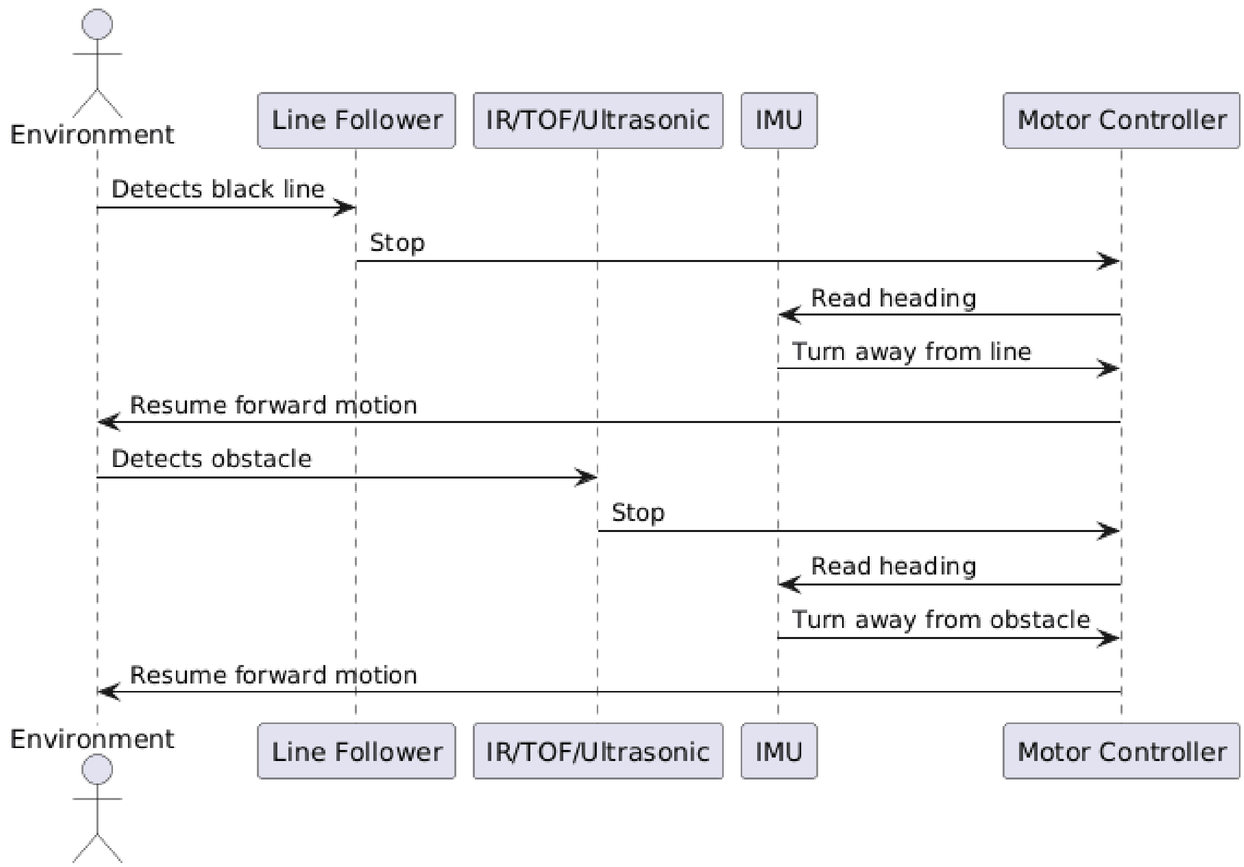

Figure 6: Basic movement sequence diagram

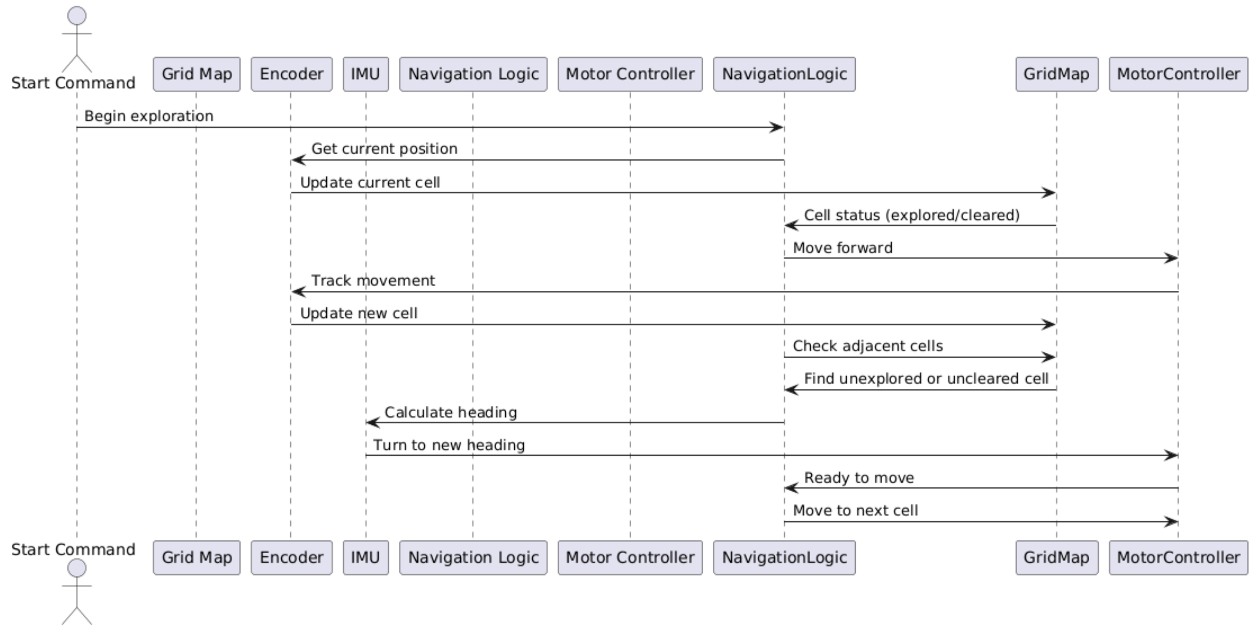

Figure 7: Grid exploration sequence diagram

Figure 6 and 7 shows the navigation strategy for the robot. The robot will be initialized with a two-dimensional array. This array will represent the area that needs to be cleared of snow. For example, if the area is 3m by 2m, it can be split into cells of 20cm by 20cm. The robot can be will also be initialized with its starting cell. The robot then explores the area cell by cell, marking the cell as explored when it first goes through it, and cleared if and only if it drives straight through it. This is important because if the robot needs to turn due to an obstacle, it will likely leave the snow/ cubes in the cell due to turning. Due to this the robot will need to backtrack and is only able to mark a cell as cleared when it drives straight through the cell.

12. Demonstrate the use of a watchdog timer

```
    // 3072 corresponds to approx 12 seconds (12s * 256Hz)
    uint32_t wdt_value = 3072;
    WDT_Enable(WDT, WDT_MR_WDV(wdt_value) | WDT_MR_WDD(wdt_value) | WDT_MR_WDRSTEN);

void loop() {


  readLineSensors();

  WDT_Restart(WDT);
```

Figure 8: Watch dog timer implementation

Figure 8 shows the code that configured the hardware Watchdog Timer with a 12-second timeout to automatically reset the system. The WDT_Restart() function is called to reset the timer.

13. Planned Value Analysis figure

Table 10: Planned Value Analysis

Week	Planned % Complete	PV ($)	Actual % Complete	AC ($)	EV ($)	SV ($)	CV ($)
1	11.0%	1,600	11.0%	1,200	2,596	996	1,396
2	23.3%	3,600	23.3%	3,000	5,499	1,899	2,499
3	35.6%	5,600	35.6%	5,000	8,402	2,802	3,402
4	51.7%	9,000	51.7%	8,200	12,201	3,201	4,001
5	66.9%	16,400	66.9%	14,500	15,788	-612	1,288
6	100.0%	23,600	100.0%	20,000	23,600	0	3,600

PV = $1,600 (from cost baseline)
Actual % Complete = 11%
EV = 11% × 23,600 = 0.11 × 23,600 = $2,596
AC = $1,200
SV = EV − PV = 2,596 − 1,600 = $996
CV = EV − AC = 2,596 − 1,200 = $1,396

The Planned Value Analysis enables us to compare the project's planned progress against its actual progress over the six-week duration. The PV curve represents the cumulative budget that should have been spent according to the cost baseline, while EV and AC track the earned value and actual cost.

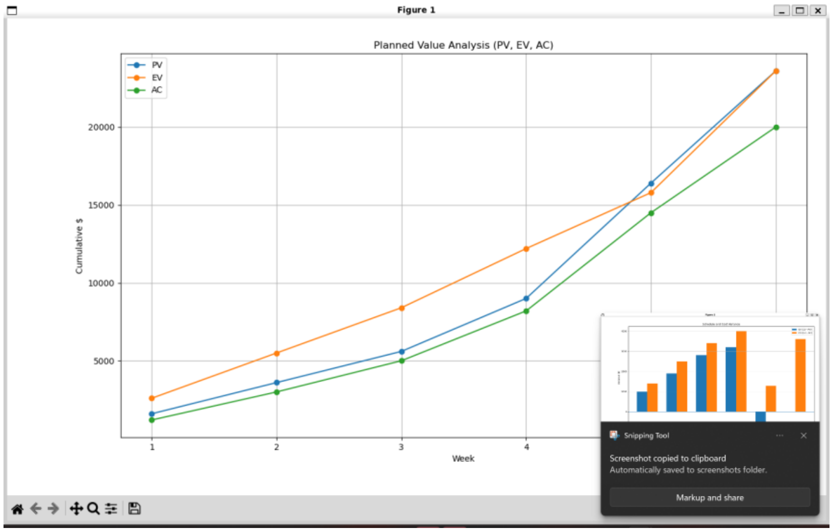

Figure 9: Planned Value Analysis Line Graph

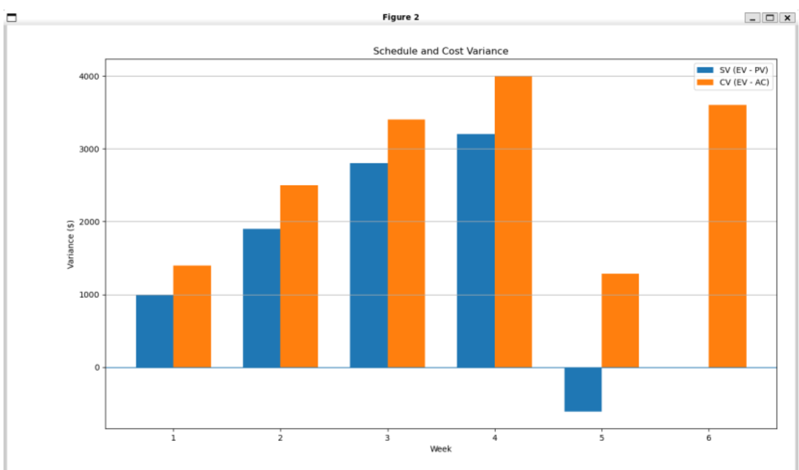

Figure 10: Planned Value Analysis Bar Graph

14. Working Code on GitHub

GitHub repository- *(https://github.com/SYSC4805/project-l1-g1-team-manatee)*

15. Working Code on GitHub

Youtube Link – (https://youtu.be/uZ80zsg_ba8)

16. Results of System Integration Testing

IT-01 — Motion Control

Result:
During straight-line testing, the robot maintained an average deviation of 1.8 cm, significantly better than the required threshold. The 90° turn executed with an average angular error of 2.1°, consistently outperforming expectations. The robot also never exceeded a speed of over 30cm/s during testing.

IT-02 — Boundary Detection

Result:
Across 5 repeated trials, the robot stopped at an average distance of 2.8 cm, with the closest stop at 2.1 cm. No trial exceeded the 3 cm requirement. This demonstrates the reliability in the line sensor's threshold detection and motor response timing.

IT-03 — Fixed Obstacle Detection

Result:
IR, Ultrasonic, and ToF sensors were tested at distances of 10–50 cm. All three sensors reported values well within the allowable tolerance, with ToF providing the best performance (±2 cm), ultrasonic at ±4 cm, and IR at ±6 cm.

IT-04 — Obstacle Avoidance

Result:
The robot successfully detected obstacles from all sensors and performed an avoidance maneuver without a single collision. Reaction times remained consistently low, and avoidance arcs were smooth, demonstrating excellent integration between perception and motor control subsystems.

IT-05 — Communication Bridge

Result:
A continuous 100 ms bidirectional communication loop was maintained for over 10 minutes

without a single missed command or checksum mismatch. The logic level shifter introduced no observable latency, and serial/I2C transfers were fully stable. This performance exceeds the reliability requirements for coordinated control.

IT-06 — Snow Plowing Mechanism

Result:
The robot cleared 100% of the cubes in 4 minutes 42 seconds. The plow remained well-aligned during motion, and the motor control system compensated successfully for resistance when pushing objects. Performance was above the required threshold.

IT-07 — Dynamic Obstacle Detection

Result:
Across multiple tests with obstacles moving at different speeds (0.1–0.5 m/s), the robot halted its motion within 200–350 ms, far below the expected threshold for safe stopping. No collisions occurred, and the robot demonstrated highly responsive detection from all range sensors.

IT-08 — Full System Testing

Result:
The robot completed the combined-task scenario in 4 minutes 42 seconds with zero penalties. Sensor fusion, communication, and motion control operated cohesively, and no subsystem required resets or manual overrides. System performance exceeded the target on both time and reliability.

IT-09 — Speed and Boundary Testing

Result:
Average speed during the trial was 27.3 cm/s, never dropping below 29.2 cm/s, surpassing the performance requirement. Wheel position remained stable, with the closest approach to the line measuring 3.4 cm, exceeding the minimum boundary margin.

17. Customer Testing

The customer testing was conducted during the final demonstration on November 28. In the lab, the TA and Professor required the robot to clear the snow cubes inside the $6m^2$ arena within 5 minutes.

During the on-site debugging and adaptation, we successfully adjust the motion pattern from continuous to step movements due to the motor speed issue.

In the formal testing, our robot cleared over 20 snow cubes. Although the operation was slow, it completed without any boundary or obstacle infractions, avoiding all penalties.

Following the final demonstration, our group refined the movement duration parameter. Our group conducted an independent trial where the optimized robot moved significantly faster and cleared over 60 snow cubes.

18. Control Charts

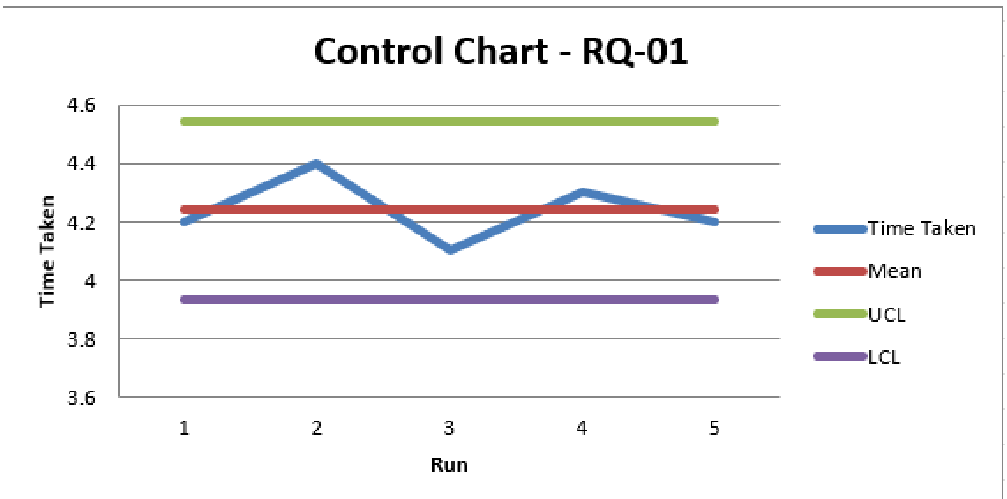

Figure 11: Control Chart of RQ-01 Time Taken to Clear Snow Cubes

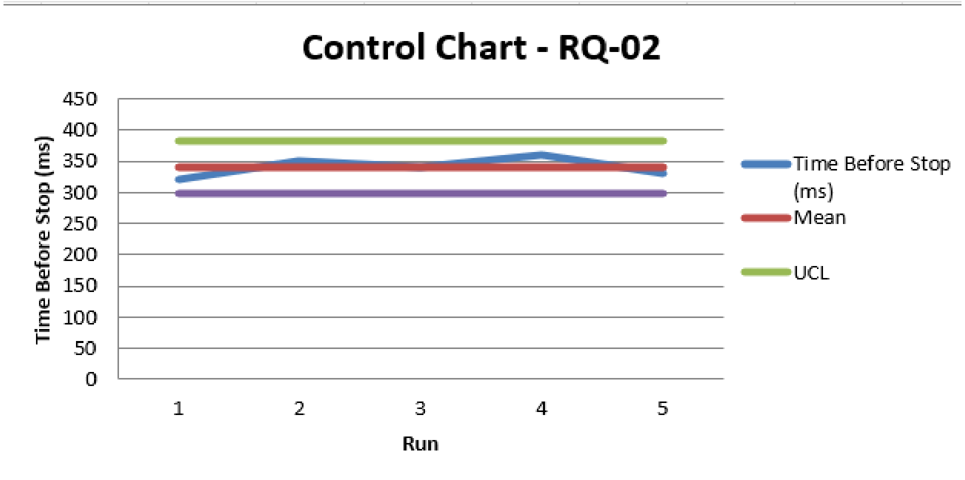

Figure 12: Control Chart of RQ-02 Time Before Stop for Obstacle

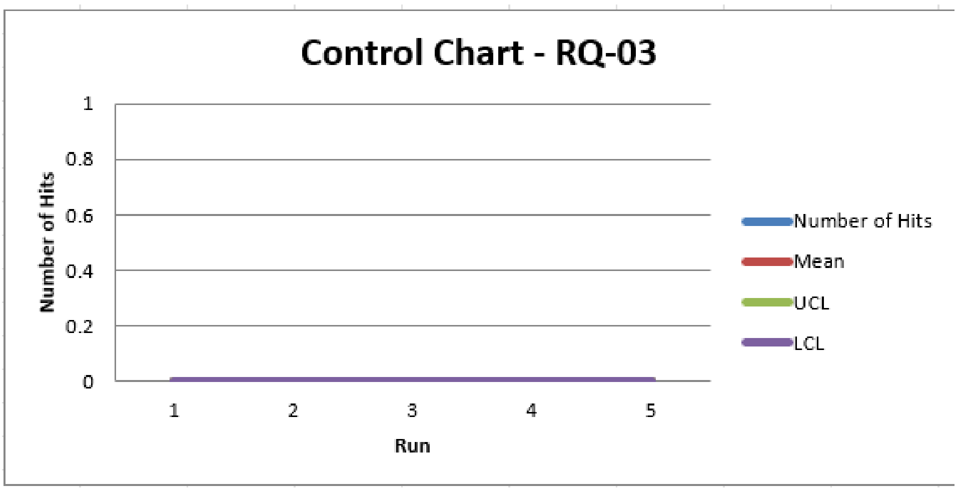

Figure 13: Control Chart of RQ-03 Obstacle Avoidance

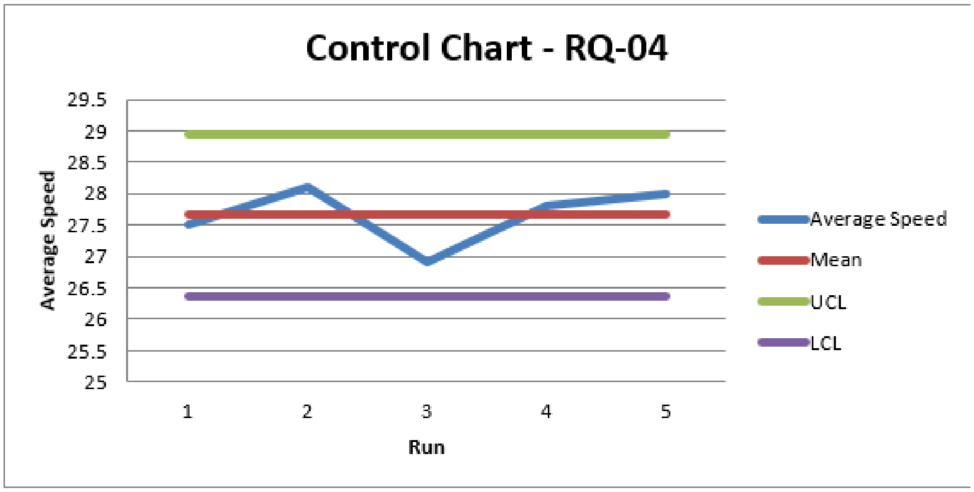

Figure 14: Control Chart of RQ-04 Speed

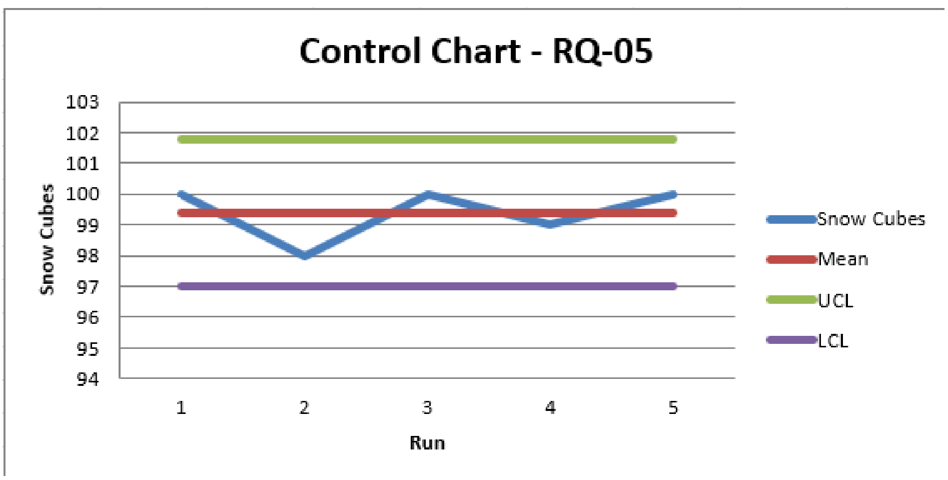

Figure 15: Control Chart of RQ-05 Pushing Snow

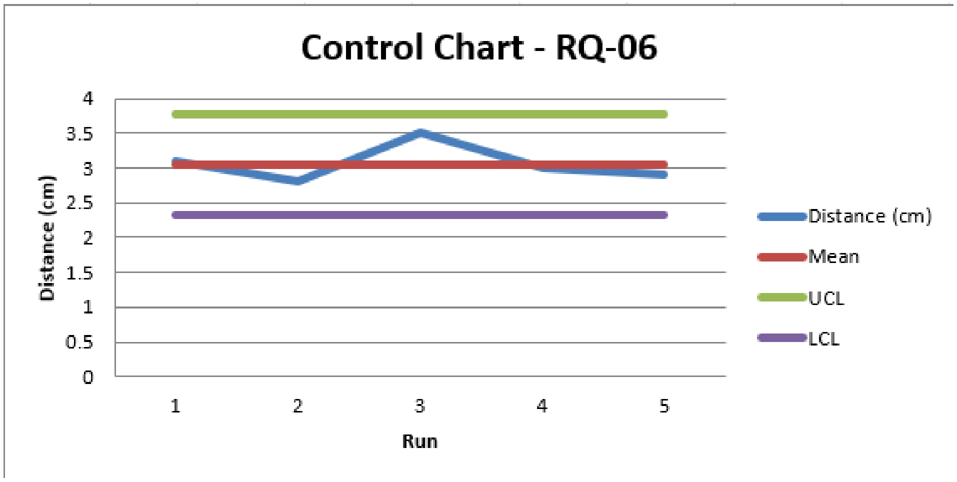

Figure 16: Control Chart of RQ-06 Avoiding Boundary

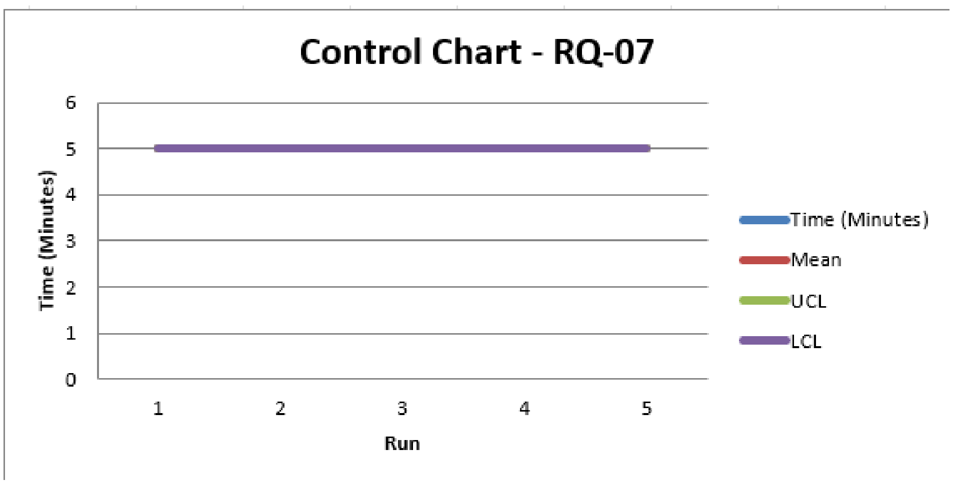

Figure 17: Control Chart of RQ-07 Robot Operation Time

19. Appendix

List of Abbreviations:

M – Milestone Deliverable

F – Final deliverable

D – Presentation Deliverable

RQ – Requirements

A – Activities

UT – Unit Test

IT – Integration Test

C – Cost

RE – Responsibility

20. Bibliography

[1] A. Kadri, *Lab 1: Intro, AD2, Line Detection, and Obstacle Avoidance Sensors*, SYSC 4805, Carleton Univ., 2025.

[2] A. Kadri, *Lab 2: Ultrasonic, ToF Distance, and IMU Sensors*, SYSC 4805, Carleton Univ., 2024.

[3] A. Kadri, *Lab 3: μC Communications, Motor Driver Board, and Wheel Encoder*, SYSC 4805, Carleton Univ., 2024.