SYSC 4805
Robot Task Project - Snow Plough

Team Laser Lemon

Progress Report

February 28th, 2022

Group#: 3

Members:

Timothy Knowles
101097700

Denise Mayo
101044064

Emma Boulay
101073617

Chhavi Sujeebun
101126487

1 Updated Project Proposal

1.1 Project Charter

1.1.1 Overall Objective

The overall objective is to design an autonomous snow plough robot using *CoppeliaSim* that will clear the snow off an area enclosed by a closed path while avoiding fixed and moving obstacles.

1.1.2 Overall Deliverables

The core deliverable of the project is the *CoppeliaSim* model file which implements the final design for the snowplough robot. The model file contains the physical design of the plough as well as the scripts which control the robot's behaviour. The scripts will implement the logic and algorithms necessary to react to sensory input and direct the robot's path to clear as much of the snow as possible within the 5-minute simulation window. Accompanying this model file, other deliverables will focus on documentation and demonstration. For documentation a progress report will be completed midway into project development. There will also be a final-report-documentation deliverable which will summarise all work completed, each member's contributions and describe the end product. Concerning the demonstration, there will be a demo deliverable showing the robot's clearing efficacy on the test maps, and a final project presentation to summarise all the work done while detailing the ultimate result.

1.2 Scope

1.2.1 Requirements

The project must adhere to a number of requirements and constraints. The following list serves as a written agreement between the group members on the specifications that the final deliverable will meet. The requirements are as follows:

- The robot shall be no larger than 0.5 x 0.8 x 1 metres at the start of the simulation, including the custom plough component.
- The robot shall be no larger than 1 x 0.8 x 1 metres at any time during the simulation, including the custom plough component.
- The robot shall have a plough component to push snow spheres outside of the highlighted perimeter.
- The robot shall be able to differentiate between obstacles and snow.
- The starting position of the robot shall be (x, y, z) = (0m, -6.25m, 0m).
- The robot shall not exceed a maximum speed of 2 m/s.
- The robot shall detect and avoid obstacles

- The behaviour of the robot shall be written in Python, with support functions written in Lua as needed to integrate unsupported functionality.
- For each sensor the robot possesses, a real-world equivalent sensor shall be documented.
- The robot simulation time shall not exceed 5 minutes.
- The robot shall avoid falling off the edges of the test map.

1.2.2 Work Breakdown Structure (WBS)

1.2.2.1 WBS Diagram

A work breakdown structure (WBS) is pictured below in Figure 1. The diagram displays a hierarchical decomposition of the entirety of the work to be completed for the project. The work has been divided into four parts; Project Management, Modelling and Design, Testing and Evaluation, and Deliverables.

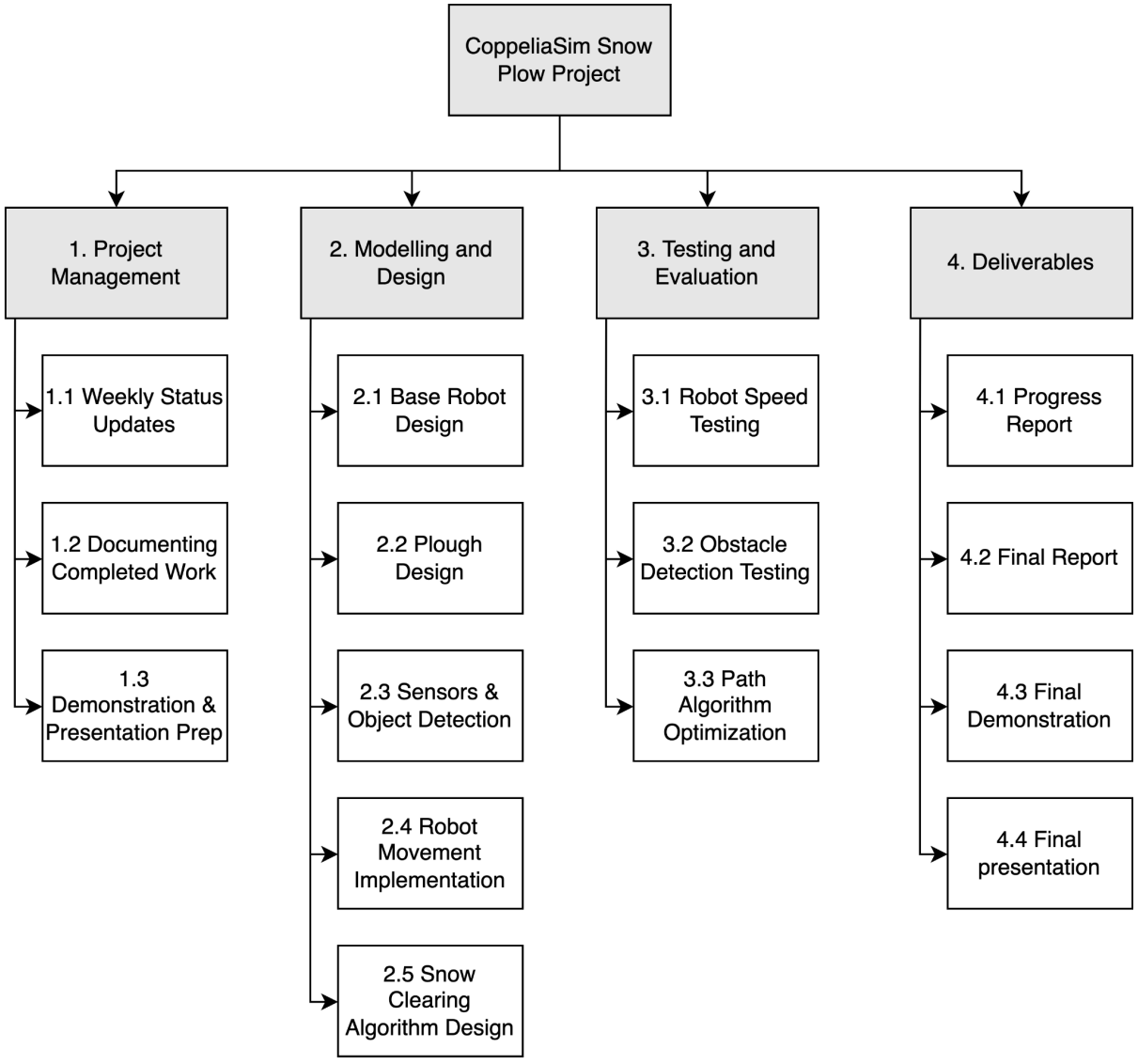

Figure 1: Work Breakdown Structure for the Snow Plough Robot Project

1.2.2.2 WBS Dictionary

Accompanying the Figure 1 WBS, pictured below is the WBS dictionary. The dictionary (shown in Table 1) provides specific details for each terminal element, including descriptions, constraints and quality requirements.

Terminal Element #	Work Description	Constraints	Quality Requirements
1.1 - Weekly Status Updates	- Consists of updating communication through Discord & follow-ups with the T.A. in weekly lab meetings	- Each member must contribute	- Team members are expected to maintain consistent communication
1.2 - Documenting Completed Work	-Document any tasks, research, algorithm completed	-Team members must follow the gantt chart and schedule network diagram to complete tasks by required deadlines	- Ideally work is documented by individual that completed it
1.3 - Demonstration & Presentation Prep	- Meetings to go through and prepare presentations and the demonstration	- Must be completed reasonably before the presentation due dates	- Clear and effectively describe work accomplished
2.1 - Base Robot Design	-Identify and implement the components required for the base robot design	- Robot cannot be larger than 1x0.8x1 metres (includes plough).	-Robot must be able to move -Robot components must remain attached to the robot body when the robot moves
2.2 - Plough Design	-Identify and Implement the components required for the plough design	- Robot cannot be larger than 1x0.8x1 metres (includes plough).	- Plough must be designed for optimal snow moving
2.3 - Sensors & Object Detection	-Identify and Implement appropriate sensors so that the snow plough robot can follow a closed path and avoid obstacles -Proximity and vision sensors are to be used	- Must be able to differentiate between obstacles and snow - Must have real world equivalent sensor - Scripts will be written in python	-Sensors must be able to detect paths, moving and fixed obstacles
2.4 - Robot Movement Implementation	- Involves having the robot properly move through the simulation environment	- Robot cannot fall off the edge of the test map - Robot cannot exceed 2 m/s	-Robot must follow a closed path
2.5 - Snow Clearing Algorithm Design	-How the robot uses the plough to move the snow out of the arena	- Robot cannot collide with obstacles	
3.1 - Robot Speed Testing		- Simulation time cannot exceed 5 minutes	
3.2 - Obstacle Detection Testing	-Test to ensure that the robot does not hit both fixed and moving obstacles	-Obstacle avoidance algorithm must be correctly implemented	-Obstacle avoidance algorithm must be designed to avoid both fixed and moving obstacles
3.3 - Path Algorithm Optimization	-Test to ensure that the robot follows a closed path	-Path finding algorithm must be correctly implemented	-Robot must be able to follow any closed path
4.1 - Progress Report	-Document the progress made in the project	- Due end of lab 6	-Team members follow the Gantt chart and complete the required tasks
4.2 - Final Report	-Document the progress made, the tests performed, the training maps tested and the challenges faced.	- Due end of lab 12	Team members follow the gantt chart and complete the required tasks, tests and training
4.3 - Final Demonstration	-Simulate and demonstrate our snow plough robot on training maps provided	- Due during last two labs	-Robot must be able to remove most amount of snow in the map while avoiding obstacles
4.4 - Final Presentation	Present our project	- Due during last 4 lectures	-Robot must be working and satisfying all requirements

Table 1: WBS Dictionary for the Snow Plough Robot Project

1.2.3 Testing

The system shall be tested rigorously to ensure that all agreed upon requirements are met as documented in Section 2.1. We plan on conducting the following tests:

1.2.3.1 Unit Testing

Test ID	Test Description	Success Criteria
T.U.1	Test that the robot can detect non-moving obstacles	The robot can detect non moving obstacles ranging from 10 to 300 cm.
T.U2	Test that the robot can detect moving obstacles	The robot can detect moving obstacles ranging from 10 to 300 cm and at a speed up to 20 m/s.
T.U.3	Test the robot's obstacle avoidance algorithm	The robot can avoid collisions with obstacles.
T.U.4	Test that the robot can detect the path	The robot can detect a black line.
T.U.5	Test that the robot can move snow outside of the detected path	No snow is left inside of the designated area.

1.2.3.2 Performance Testing

Test ID	Test Description	Success Criteria
T.P.1	Record how the robot performs on training map 1	All snow is cleared from the designated area, without colliding with obstacles, in a maximum of 5 minutes.
T.P.2	Record how the robot performs on training map 2	All snow is cleared from the designated area, without colliding with obstacles, in a maximum of 5 minutes.
T.P.3	Record how the robot performs on training map 3	All snow is cleared from the designated area, without colliding with obstacles, in a maximum of 5 minutes.

1.2.3.3 Stress Testing

Test ID	Test Description	Success Criteria
T.S.1	Test at what speed the robot can no longer reliably detect a moving obstacle	The robot can detect an object moving at a speed of 20 m/s

1.2.3.4 Integration Testing

Test ID	Test Description	Success Criteria
T.I.1	Test that the robot movement control and plough control modules work together	The plough does not impede the robot's vision or proximity sensors at any stage.

The unit tests will be performed by measuring the robot module's performance on *CoppeliaSim* scenes generated by a team member.

If the robot can successfully pass the tests outlined above, then it behaves as expected and should successfully remove the snow from the testing maps in the time limit.

1.3 Schedule

1.3.1 Schedule Network Diagram

The schedule network diagram for the *CoppeliaSim* Snow Plough project is depicted in Figure 2 below. The diagram is a graphical representation of the logical relationships and dependencies among the project schedule activities that are proposed in Section 4.1.

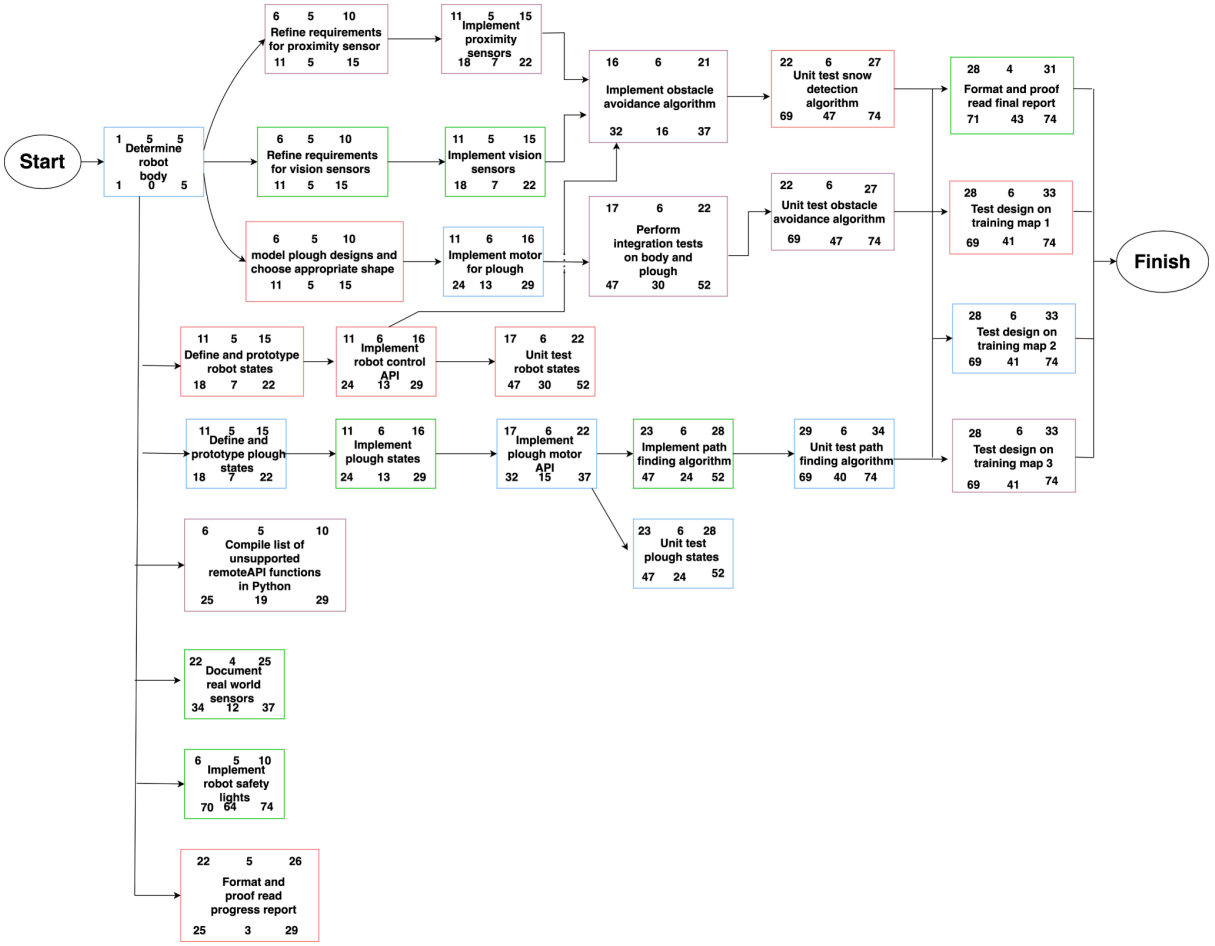

Figure 2: Schedule Network Diagram for the Snow Plough Robot Project

1.3.2 Gantt Chart

The Gantt chart for the project is provided in Figure 3 below. Each team member has one task assigned each week. A colour coded legend is provided to show which team member is assigned which task. The unit tests for the robot and plough controls require their implementation to be finished before testing can start. A contingency plan is made to allow for an extra week of slack time for implementing the robot and plough control in case any unforeseen problems arise.

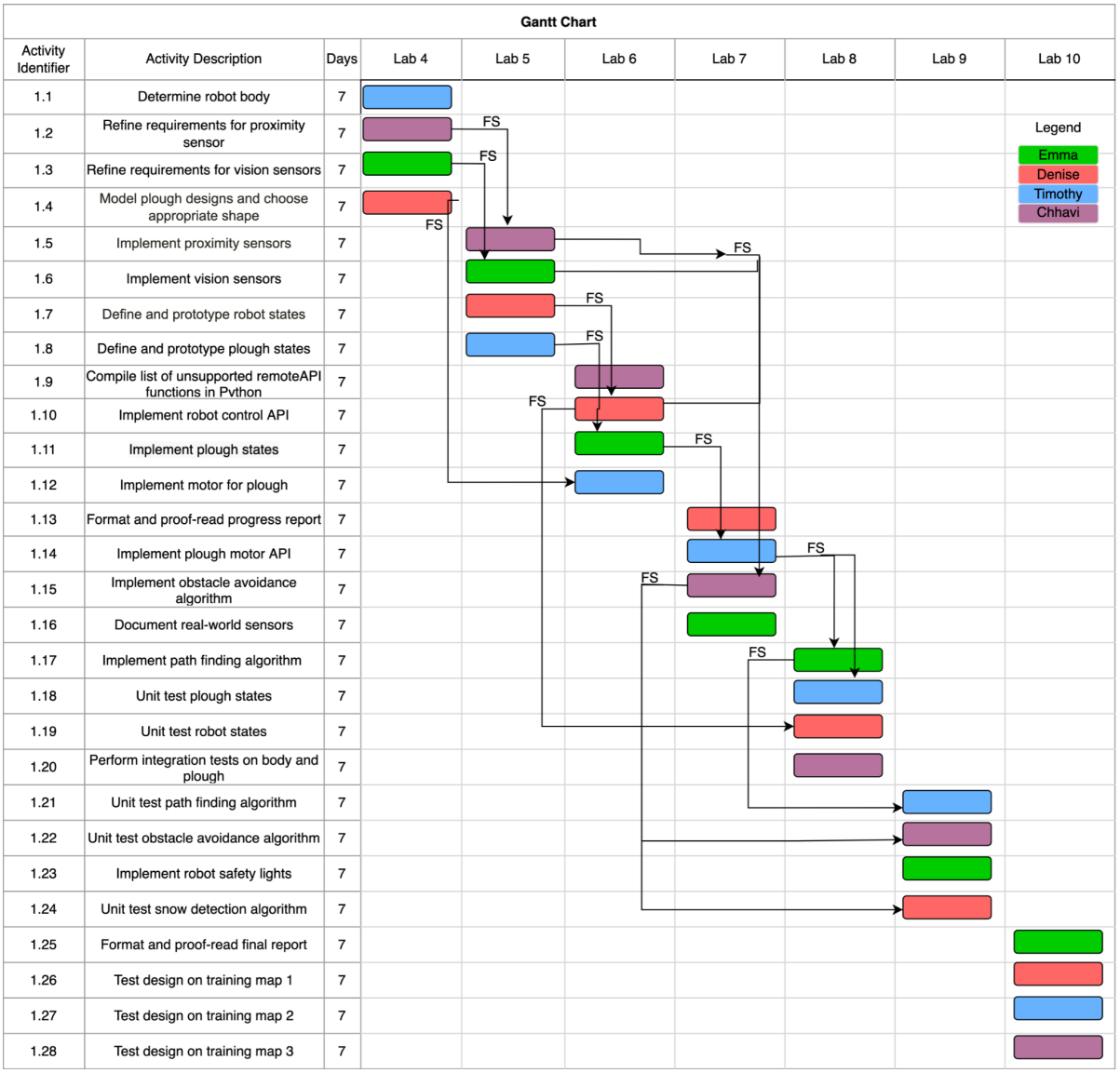

Figure 3: Gantt Chart for the Snow Plough Robot Project

1.4 Human Resources

1.4.1 Responsibility Assignment Matrix

Each member in the team was assigned 7 unique tasks that they are responsible for completing and 7 unique tasks they are responsible for reviewing. Figure 4 below shows the responsibility breakdown for the project.

Activity	Denise	Timothy	Emma	Chhavi
Determine robot body style and shape		R	QA	
Refine and document requirements for proximity sensors		QA		R
Refine and document requirements for vision sensors	QA		R	
Define and prototype robot states	R			QA
Document real-world sensor equivalents	QA		R	
Define and prototype plough states		R	QA	
Compile list of unsupported functions in Python API		QA		R
Implement proximity sensor(s)		QA		R
Implement vision sensor(s)	QA		R	
Model plough designs and choose appropriate shape	R			QA
Implement plough control motors		R	QA	
Implement safety lights	QA		R	
Implement plough states	QA		R	
Implement robot states and control API	R			QA
Implement plough control API		R	QA	
Implement general steering and movement control motor functions	QA		R	
Implement obstacle avoidance movement control functions		QA		R
Unit test robot states and control API	R			QA
Unit test plough states		R	QA	
Unit test obstacle avoidance movement control functions		QA		R
Unit test plough control API	R			QA
Unit test movement control (general steering) functions		R	QA	
Perform integration tests on plough and body		QA		R
Test design on training map 1	R			QA
Test design on training map 2		R	QA	
Test design on training map 3		QA		R
Format and proof-read progress report	R			QA
Format and proof-read final report	QA		R	

Legend	
Responsible for completing work	R
Reviewer (quality assurance check)	QA

Figure 4: Responsibility Assignment Matrix for the Snow Plough Robot Project

2 Overall Architecture

2.1 Architecture Diagram & Description

The figure below describes the overall system architecture of the design solution for the Coppeliasim Snow Plough Robot. The robot is primarily controlled by the path-finding program. The path finding program uses the Movement Control Module to control the robot wheels to move the robot around the map. The Object Detection module will inform the robot when an object is detected and the path finding algorithm can plan the most efficient route to avoid the obstacle. The Line Detection module will inform the robot when it has reached the perimeter and the path finding algorithm knows to change direction.

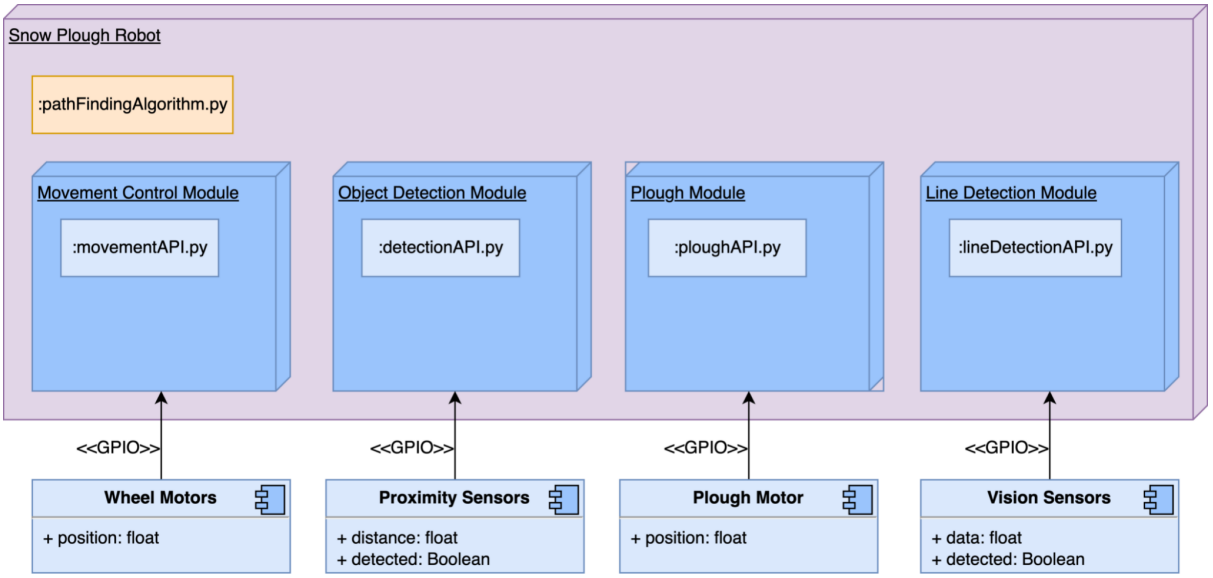

Figure 5: Overall System Architecture of the Snow Plough Robot

2.1.1 Model Design

The primary design for the snow plough robot is documented in the figures below. The robot will have vision sensors at the front base of the robot to detect the outer perimeter and will have proximity sensors at the top of the robot to detect obstacles.

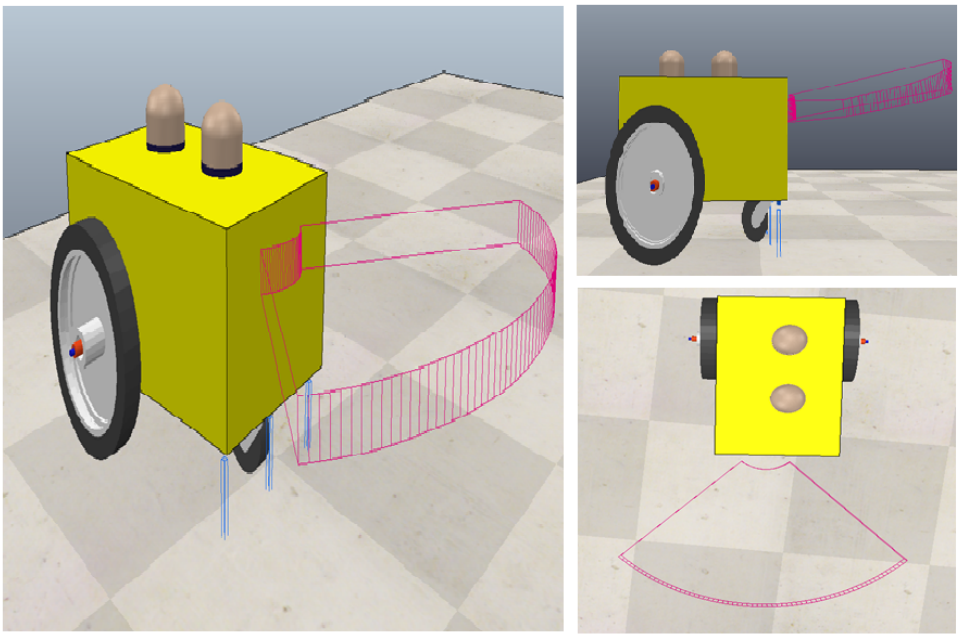

Figure 6: Robot Body Design with Front Proximity Sensor

2.1.2 Plough Designs

The team's initial plough designs are documented in the figures below. The team is currently evaluating having a plough with either a single or dual hinge door system. The team is currently planning on implementing the design shown in Figure 8 but if time allows will implement the plough shown in Figure 9.

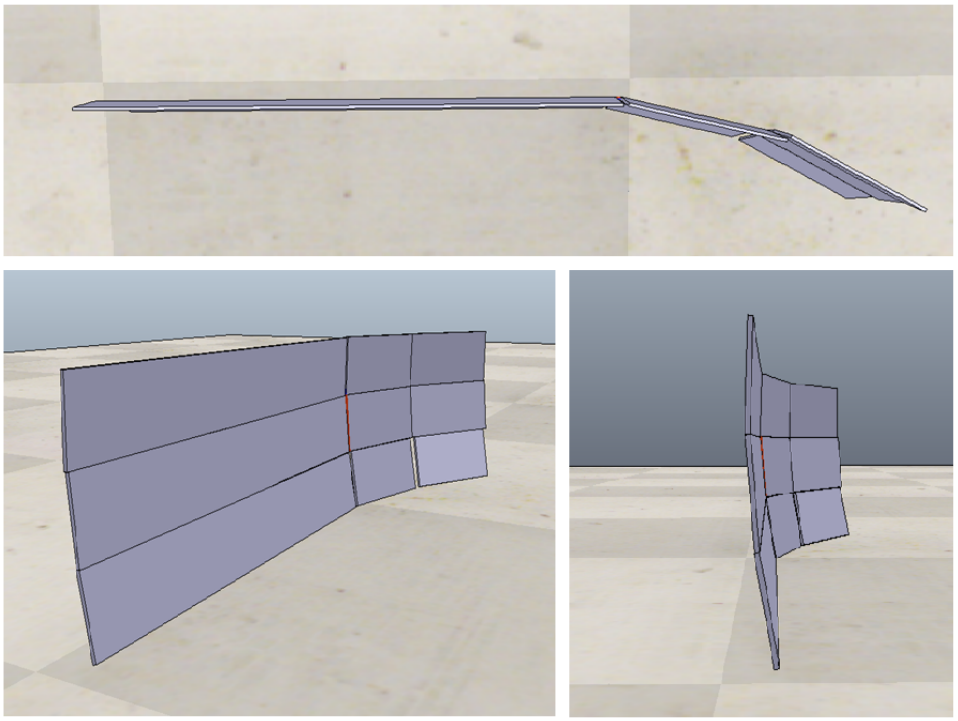

Figure 7: Robot Plough Design Number 1

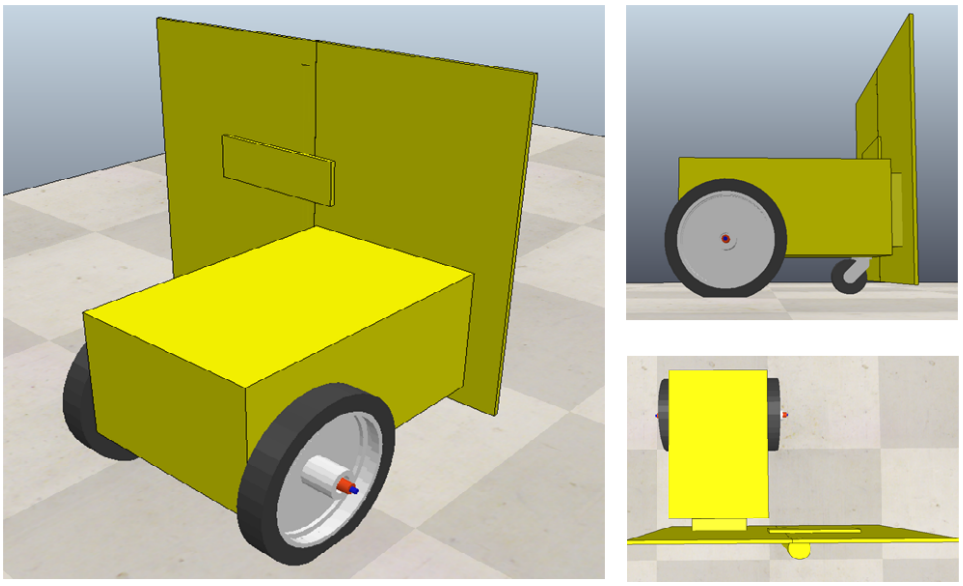

Figure 8: Robot Plough Design Number 2 (for testing purposes)

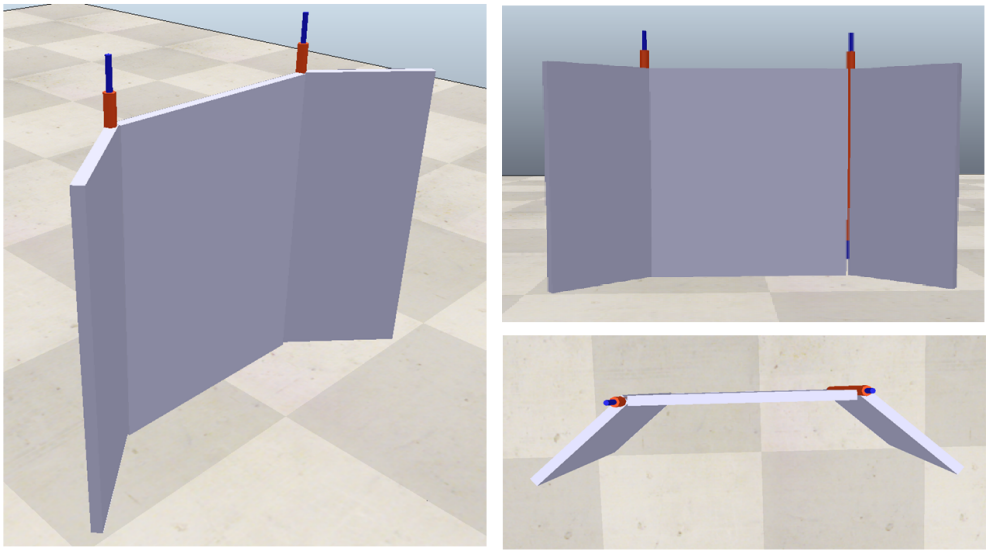

Figure 9: Robot Plough Design Number 3 (for testing purposes)

2.2 Time Triggered or Event Triggered

The design of the robot is time triggered. This means that the robot's vision and proximity sensor will be polled in a loop and the robot will perform the necessary manoeuvre action. A time triggered approach to communication is chosen over an event triggered approach because it is more predictable and is easier to implement. However, an event triggered design consumes less power as the microcontroller unit is in sleep mode until an event triggers an interrupt. Since predictability is more important to the design than efficiency, a time triggered approach is selected.

3 System States

The state machine governing the robot's movement and plough position is detailed below. The two state machines will interact using an additional state in the Robot Body State Chart (Figure 5), the details of which are to be determined.

3.1 Robot Body and Movement States Chart

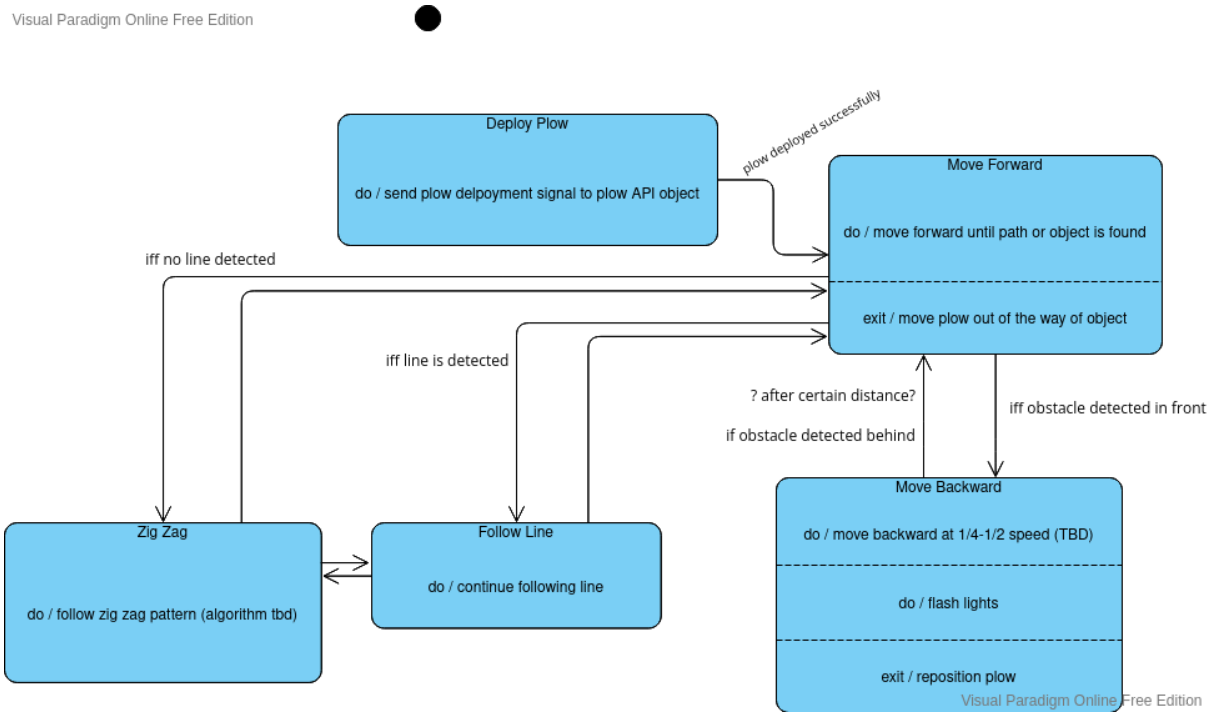

Figure 10: Robot Body State Chart as of February 26, 2022.

3.2 Plough States Chart

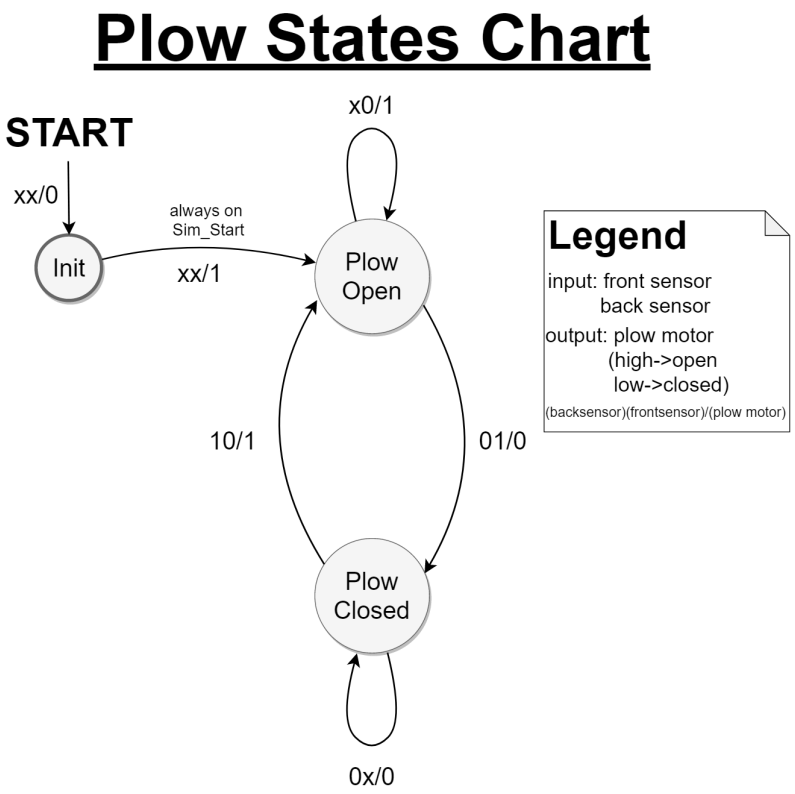

Figure 11: Plough States Chart

```
Current state is now: PlowOpenState
Would you like to exit? (Y/N)n
What is new front sensor value? (0 or 1)1
What is new back sensor value? (0 or 1)0
New Sensor Input:  0 1
ACTIVATING PLOW MOTOR TO CLOSE PLOW
Current state is now: PlowClosedState
Would you like to exit? (Y/N)n
What is new front sensor value? (0 or 1)0
What is new back sensor value? (0 or 1)1
New Sensor Input:  1 0
ACTIVATING PLOW MOTOR TO OPEN PLOW
Current state is now: PlowOpenState
Would you like to exit? (Y/N)
```

Figure 12: Plough States Prototype Script Output (including sample inputs)

4 Sequence Diagrams

4.1 Robot Body Pathfinding Sequence Diagram

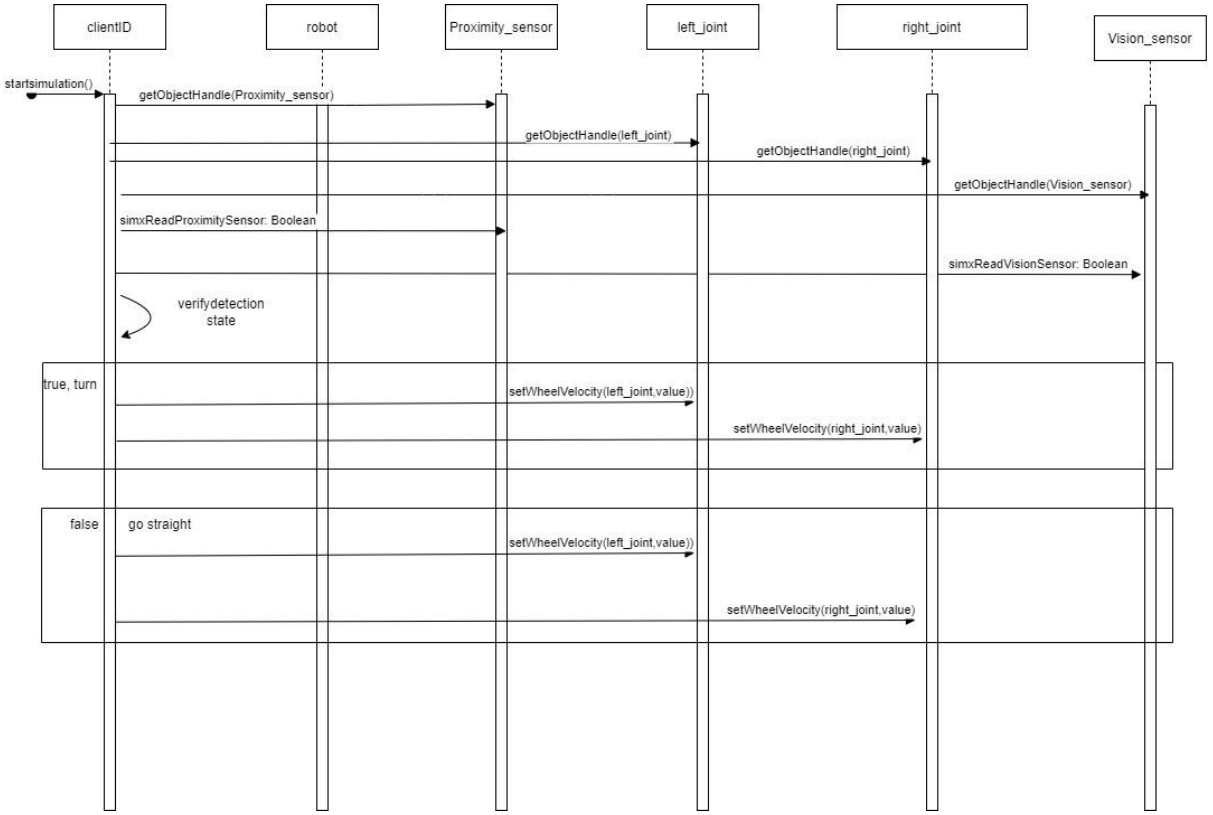

Figure 13:

4.2 Robot Plough Sequence Diagram

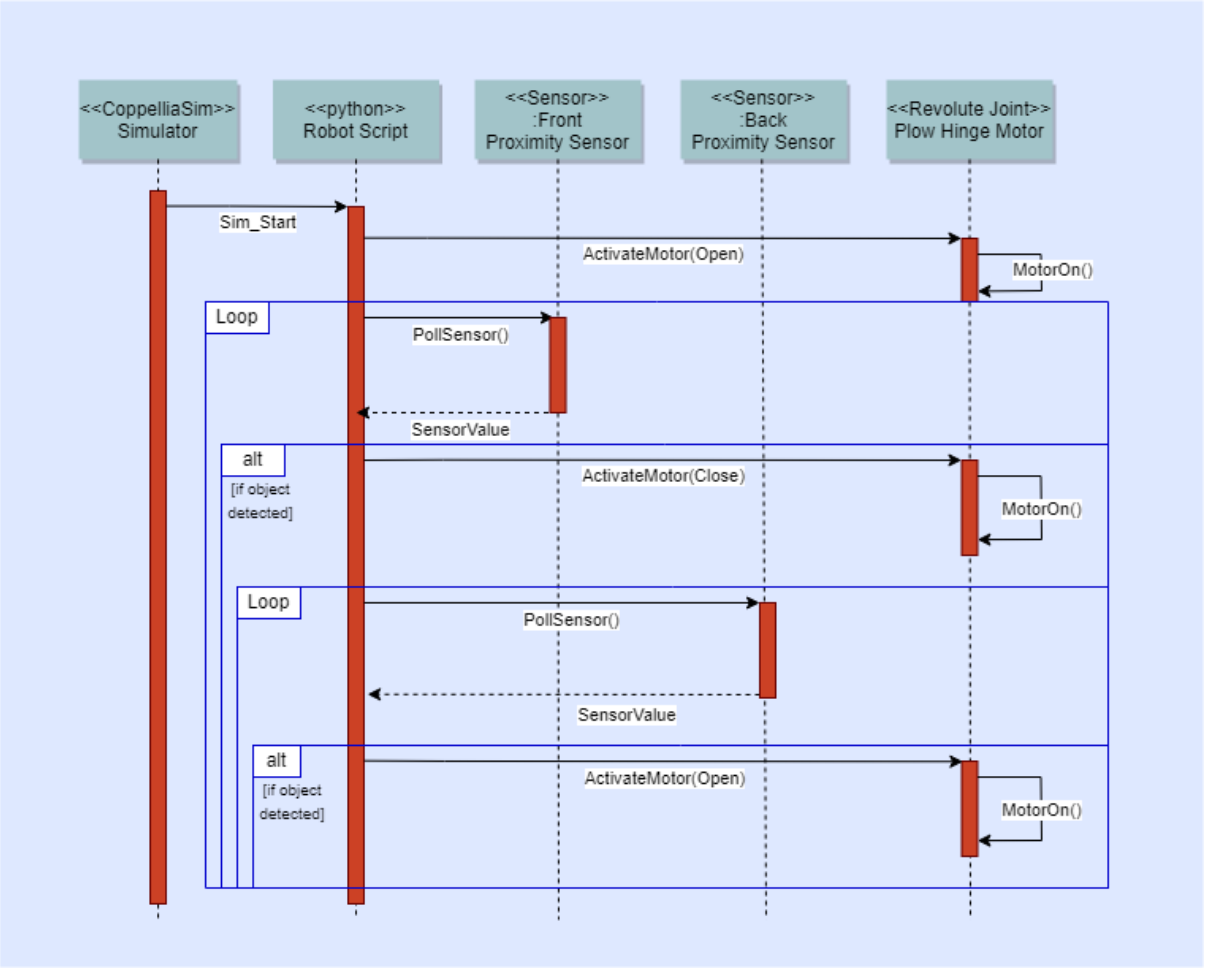

Figure 14:

5 Project Budget

5.1 Planned Value and Budget at Completion

The planned value (PV) is the authorised budget assigned to the scheduled work. Each team member is assigned an activity corresponding to a work breakdown structure component each week to complete. It is estimated that each activity takes approximately 2.5 hours to complete. Since the team has 4 members, the planned value of work for each week is 10 hours. The budget at completion (BAC) is the total planned value for the project. The BAC is estimated to be 70 hours, this accounts for 10 hours each week performed by the team over the span of 7 lab periods. The designated lab period is 4 hours each week, this allows each team member to have an extra 1.5 hours a week to complete their assigned activity without interfering with

other school and personal commitments. This extra time is shown as the Management Reserve in the graph below.

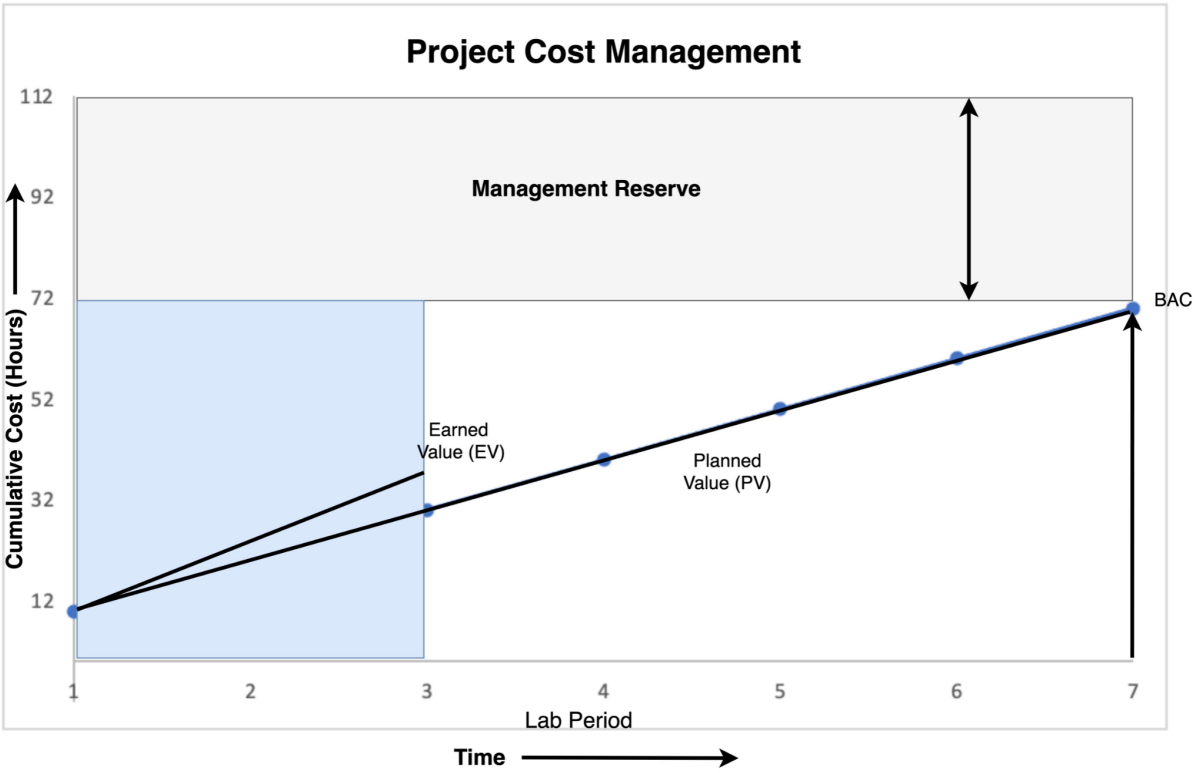

Figure 15: Project Cost Management

5.2 Hardware Cost

An IR sensor will be used to detect the outer perimeter that is marked by a black line. The IR sensor has an IR transmitter module that emits infrared radiation, and the reflected radiation is detected by the IR receiver module. The black line will absorb the radiation and will allow the robot to detect the outer perimeter. As the robot only needs to detect the black line, as opposed to differentiating between different surface colours, a digital sensor is chosen. The sensor will also be mounted to the base of the robot body and must detect the line with a range of 0.5 mm. The Infrared Sensor Module (TCRT5000) [1] is selected as it satisfies all criteria. This sensor can be purchased from Canada Robotix for $3.19 CAD.

An ultrasonic sensor will be used to detect moving and stationary obstacles. The ultrasonic sensor has a transmitter module that emits sound waves, and the reflected wave is detected by the receiver module. Using Sound navigation and Ranging (sonar) the robot can detect and avoid objects in its path allowing it to safely achieve its goal of clearing snow from designated areas. The sensor must be able to detect objects from 10 cm to 300 cm. The sensor must also have a measuring angle of 30

degrees. The HC-SR04 Ultrasonic Ranging Sensor [2] is selected as it satisfies all criteria. The sensor can be purchased from Canada Robotix for $5.09 CAD.

The cost for all sensors required to complete this project is summarised below

- 3 IR sensors X $3.19 CAD = $9.57 CAD
- 5 ultrasonic sensors x $5.09 CAD = $25.45 CAD

The total cost for all sensors is determined to be $35.02 CAD.

6 References

Data Sheets
[1] Vishay Semiconductors, “Reflective Optical Sensor with Transistor Output”, TCRT5000 datasheet, Aug 2009.
[2] Elec Freaks, “Ultrasonic Ranging Module HC - SR04”, HC-SR04 datasheet, Nov. 2011.

7 Appendix

7.1 GitHub Repository Link

https://github.com/SYSC4805-Winter2022/sysc4805-project-group-3-l2-laser-lemon

7.2 Completed Activities Per Student (As Located on GitHub)

Activity Name	File Name	Completed By:
Define Robot Body	project_robotbody_v1.ttt	Timothy Knowles
Define and Prototype Plough States	plow_states_chart.pdf	Timothy Knowles
Define and Prototype Plough States	plow_states_prototype.py	Timothy Knowles
Implement Motor for Plough	plow_v2_implementationTesting.ttt	Timothy Knowles
Refine Requirements for Vision Sensor	project_robotbody_with_proximityandvisionsensor_v3.ttt	Emma Boulay
Implement Vision Sensor	project_robotbody_with_proximityandvisionsensor_v3.ttt	Emma Boulay
Document Real-World Sensors	Hardware Cost.pdf	Emma Boulay
Implement Robot Safety Lights	safetyLight.ttt	Emma Boulay
Refine requirements for proximity sensor	project_robotbody_with_proximitysensor_v2.ttt	Chhavi Sujeebun
Implement proximity sensor	project_robotbody_with_proximitysensor_v2.ttt	Chhavi Sujeebun
Implement obstacle avoidance algorithm in Training Map1	Training_Map_1_proximitysensor_implemented.ttt	Chhavi Sujeebun
Obstacle avoidance algorithm in python	obstacle_avoidance_proximitysensor.py	Chhavi Sujeebun