Automated Snowplow in CoppeliaSim

Final Report

By: Dominique Giguere Samson #100912460

James Runtas #101109175

Ramit Mahajan # 101132991

SYSC4805 – L2 – Group 11

Team Iris

Instructor: Dr. Mostafa Taha

TA: Manas Chamola

April 11th, 2022

TABLE OF CONTENTS

1. Project Outline

1.1 Objective

The objective of this project is to design and build a Snowplow in CoppeliaSim that can clear snow while avoiding obstacles in its path. In addition, the robot must clear the snow out of the parking lot, without colliding into any obstacles and must operate autonomously.

1.2 Deliverables

At project completion, the Robot will be evaluated on 3 hidden maps, based on the amount of snow cleared from the parking lot, and number of collisions. Until then, a progress report and a final report will be submitted.

Table 1 shows all the project deliverables with tentative dates.

Deliverable	Date
Proposal	February 4th, 2022
Progress Report	February 28th, 2022
Robot Model (*.ttm)	March 28th, 2022
Demonstration Lab	April 4th, 2022
Final Report	April 11th, 2022

Table 1: Project deliverables with date of submission

1.3 Requirements

The requirements of the snowplow's functionality include:

- The robot must be designed and programmed in CoppeliaSim (Lua), and/or an external script editor such as MATLAB, python, or any other language supported by CoppeliaSim.
- The robot must navigate autonomously.
- The robot must clear the snow out of the parking lot, without colliding into any obstacles.
- When proximity sensors detect a static obstacle, the robot should pick a path that avoids the obstacle.
- When proximity sensors detect a moving obstacle, the robot should pick a path outside the trajectory of the moving obstacle.
- When vision sensors detect the perimeter, the robot must turn around, without leaving the perimeter.

Robot Specifications (From provided "ProjectDescription.pdf") Include:

- Maximum robot size at parking space = 0.5x0.8x1 (Width x Length x Height) (meters).
- Maximum robot size with fully extended plow = 1x0.8x1 (Width x Length x Height) (meters).
- Robot starting position (x, y) = (0, -6.25)
- Robot starting orientation (Alpha, Beta, Gamma) = (90, 0, 90)
- Maximum Robot speed = 2m/s
- Maximum Simulation time = 5:00 Minutes

1.4 Testing

The model is graded on the remaining amount of snow in the area, 3 training maps are provided to our team to ensure all goals are met. However, the successful operation of the snowplow is not limited to the 'map' itself and should be fully functional in all three *additional* maps only available to the grader of this project. Snow particles are randomly generated for each map to ensure flexibility and resilience for the robot.

Testing will focus on completing all project goals, requirements, and specifications. Important testing milestones can be found in table 2 below.

Date	Test No.	Test(s)
February 12th,2022	1	Plow is attached correctly to the robot and can be extended to/from the robot within dimensions. PASS: The plows correctly extend/retract
February 13th, 2022	2	Motors work properly on the snowplow and snowplow can move within an area. PASS: The vehicle drives and doesn't leave the perimeter
March 5th,2022	3	Snowplow Collison Detection works on obstacles of varying sizes, and types. PASS: The vehicle drives around the parking lot and doesn't collide into obstacles.
March 19th,2022	4	Snowplow detects snow particles and can physically clear the snow off the path using the plow. PASS: When moving snow, the snow particles are pushed by the plow, instead of going under the vehicle or not-colliding with plow.
March 27th,2022	5	Snowplow successfully completes the task of clearing snow on all three training parts as well as avoiding obstacles and having all specifications outlined in section 1.3. PASS: The vehicle autonomously navigates the parking lot without colliding into any obstacles, while doing so, snow is removed from the parking lot.

Table 2: Tests of Snowplow at different stages with dates

2. Schedule & Human Resources

2.1 Work Breakdown Structure

The structure for work done in the project is divided into 5 stages, with accompanying subtasks and activities. The detailed work breakdown structure is seen in figure 1 below.

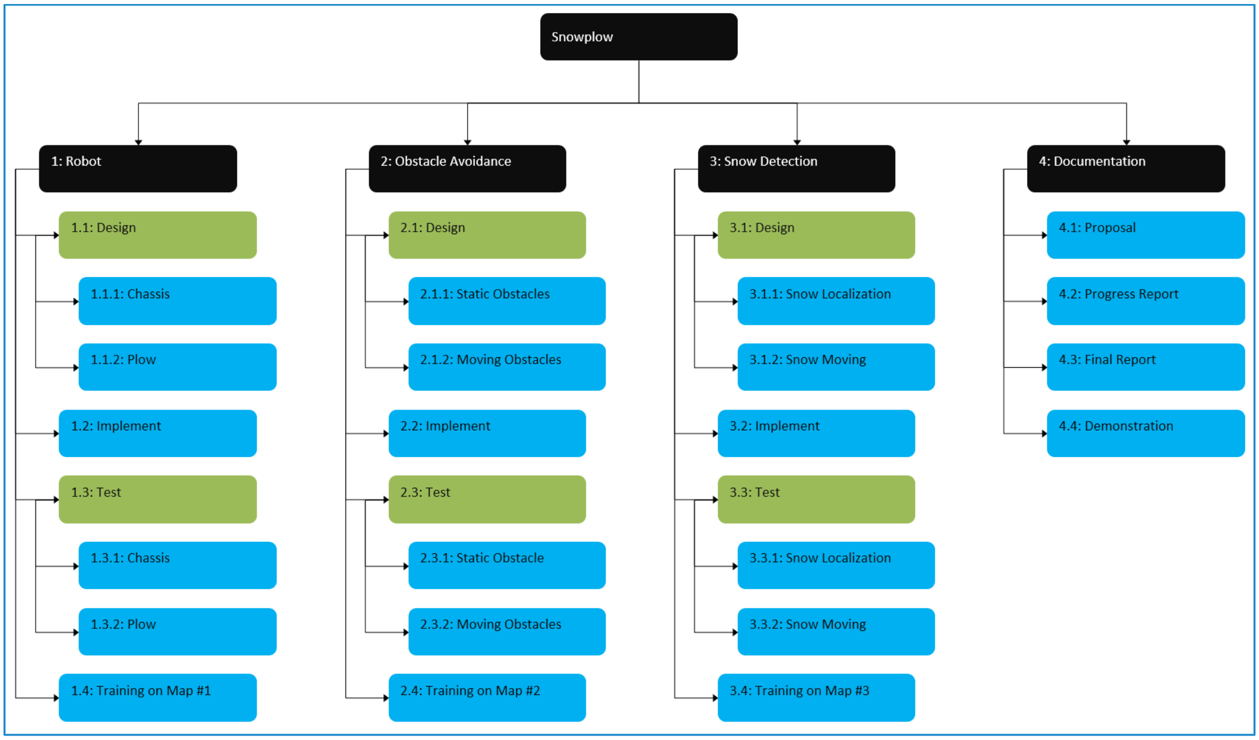

Figure 1: Work breakdown structure of Snowplow project

2.2 Gantt Chart

The Gantt chart in figure 2 displays the project goals with respect to time. The time for completion for each week will need to be accurately followed in order to work efficiently each week.

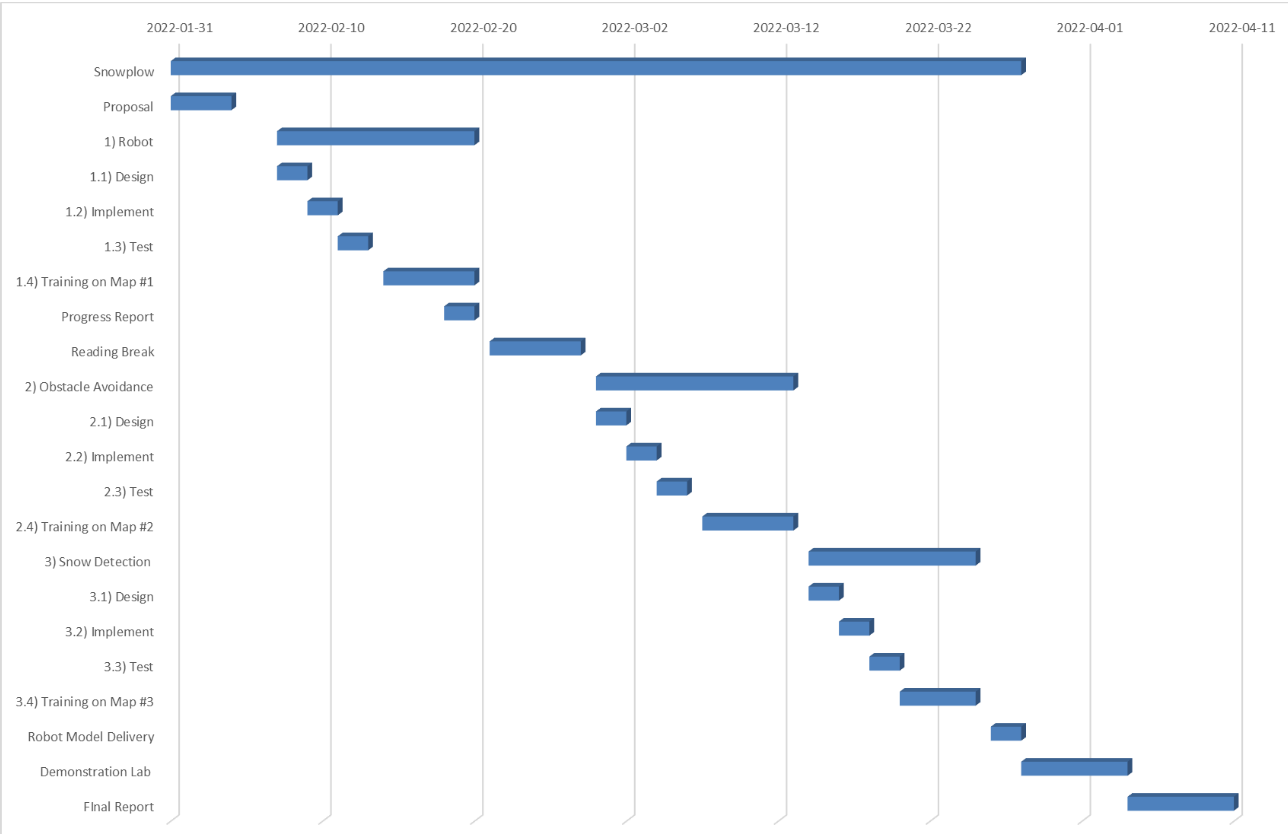

Figure 2: Project timeline with estimated time of completion

2.3 Network Schedule Diagram

This project includes several tasks that relate to previous tasks, as each addition to the snowplow is required to complete the next objective of the project. The dependencies for these tasks have been located and can be seen in figure 3 below.

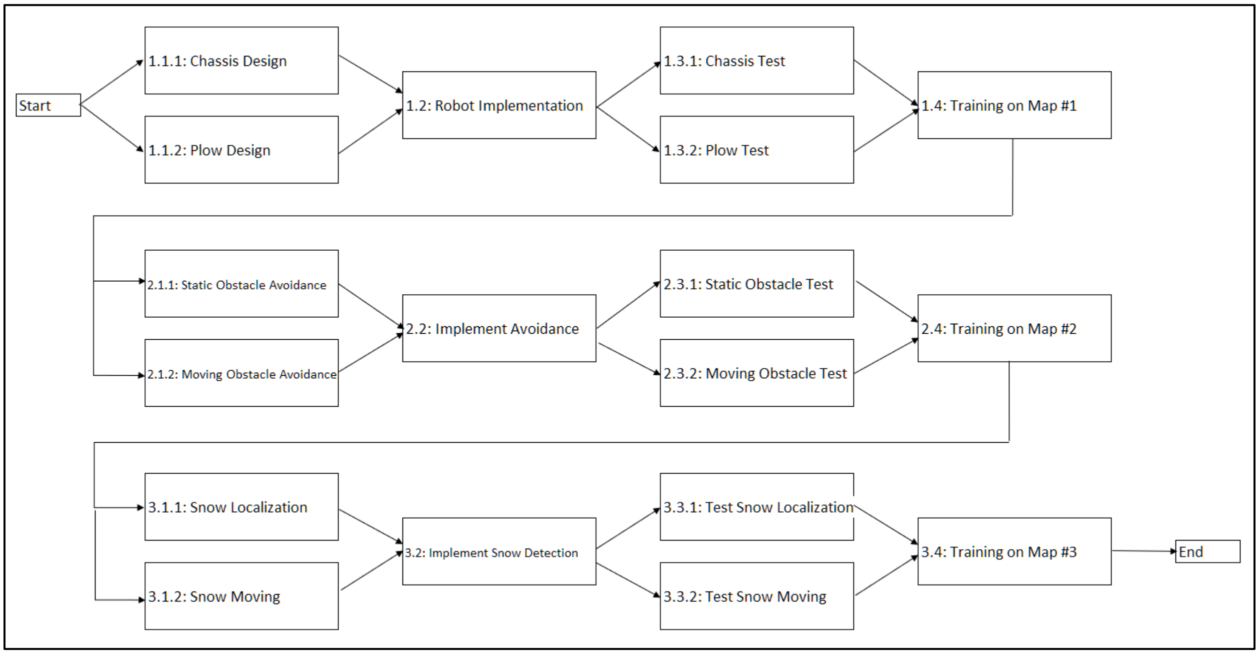

Figure 3: Schedule management diagram with dependencies.

2.4 Responsibility Management

Meetings are conducted every Monday during our lab period. These meetings are to be held through discord where we can upload files, messages, videos, make voice calls, and share our computer screens.

The team has been assigned certain roles and tasks that play to our individual skills. The table below displays the assignments for each team member.

R = Responsible

A = Approver

Responsibility	Team Member		
	James	Ramit	Dominique
Project Proposal	R	R	R
Robot Design		R	A
Plow Design	A		R
Snowplow Implementation	R	A	
Chassis Test		R	A
Plow Test	A		R
Static Obstacle Detection	R	A	
Dynamic Obstacle Detection		R	A
Obstacle Implementation	A		R
Snowplow Collision Test	R	A	
Snow Localization		R	A
Snow Moving	A		R
Snow Implementation	R	A	
Snow Clearing Test		R	A
Map 1 Training		A	R
Map 2 Training	R		A
Map 3 Training	A	R	
Progress Report	R	R	R
Final Report	R	R	R
Demonstration	R	R	R

Table 3: Responsibility Assignment Matrix

2.5 Project budget

This project has a budget at completion of $3 240.00. Table 3 shows a breakdown of the project budget by activity. The cost of each activity is determined by the time required to complete the activity at a rate of $30 per hour.

Activity	Time required (hours)	Cost ($)
Project Proposal	12	360.00
1. Robot	-	-
1.1.1 Chassis Design	2.5	75.00
1.1.2 Plow Design	2	60.00
1.2 Robot Implementation	3	90.00
1.3.1 Chassis Test	1.5	45.00
1.3.2 Plow Test	1	30.00
1.4 Map test #1	12	360.00
Sub Total	22	660.00
Progress Report	12	360.00
2. Obstacle Avoidance*	-	-
2.1.1 Static Obstacle Avoidance	3	90.00
2.1.2 Moving Obstacle Avoidance	3	90.00
2.2 Implemental Avoidance	3	90.00
2.3.1 Static avoidance test	2	60.00
2.3.2 Moving avoidance test	2	60.00
2.4 Map test #2	12	360.00
Sub Total	25	750.00
3. Snow Detection*	-	-
3.1.1 Snow Localization	3	90.00
3.1.2 Snow Moving	3	90.00
3.2 Snow Detection Implementation	3	90.00
3.3.1 Snow Localization Test	2	60.00
3.3.2 Snow Moving Test	2	60.00
3.4 Map test #3	12	360.00
Sub Total	25	750.00
Final Report	12	360.00
Total	108	3 240.00

Table 3: Planned value of project.

3.0 System Architecture

The automated snowplow will feature 3 different extendable snowplows, 2 different obstacle sensors, and a path planning algorithm. Figure 4 shows an overview of the project.

3.1 System Overview

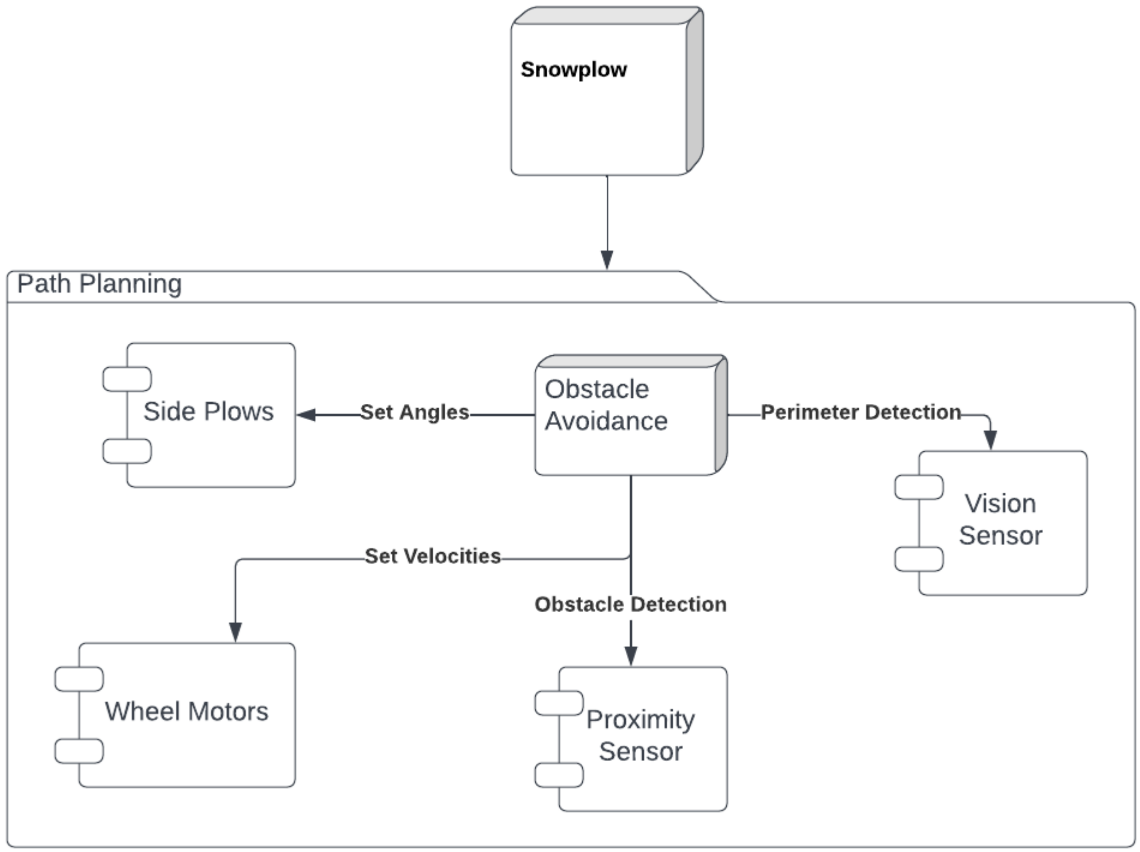

Figure 4: System Architecture Overview

The plow was designed to give the automated snow plow a large surface area to move snow but maintain the robot's maneuverability, while being able to fit within the parking space. In addition, the plows can extend and retract based on the environment around the snowplow. For example, if there is a wall to the left of the robot, the plow on the left of the robot will retract, allowing the robot to get close to the wall. The plows on the front and right side of the snowplow will extended to the right, to guide the snow away from the wall. The plows will also retract to avoid any collisions.

3.2 Side Plow Movement

The sequence diagram, figure 5, illustrates how the movement of the side plows is controlled. The snowplow begins by polling the left and right proximity sensors, for their result and distance. The result is a Boolean, true when the proximity sensor is detecting an obstacle, and distance is the distance to the obstacle in meters. If the result was true, the angle of the side plow is calculated using:

$$Angle = 0.9(Distance * 200)$$

Once both angles are calculated, both side plow angles are updated.

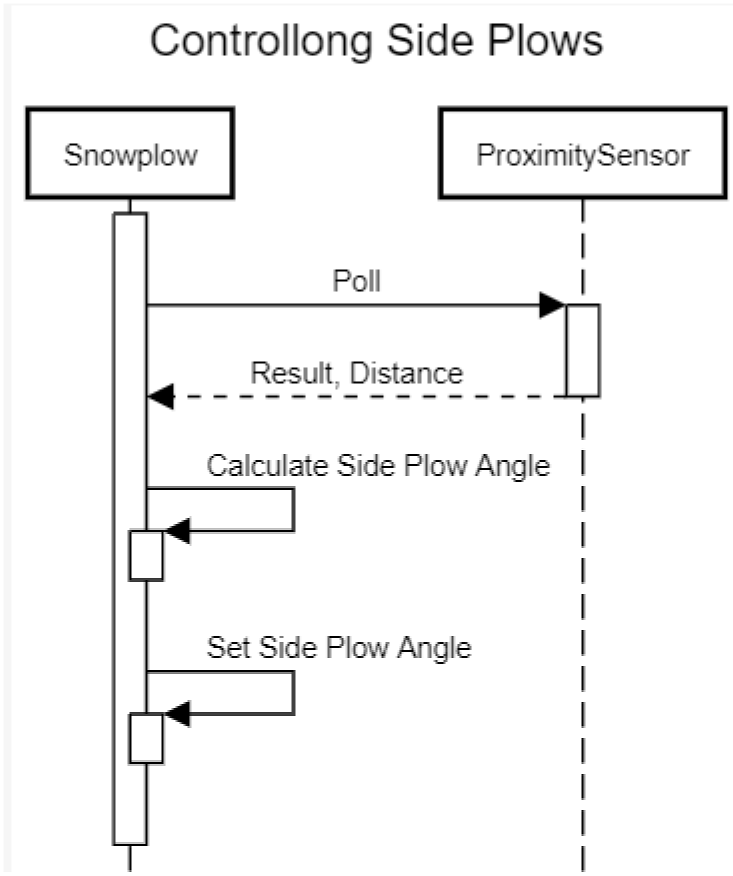

Figure 5: Sequence diagram for the side plow movement

3.3 Path Planning

The snowplow's path planning is based on random walk. A random walk algorithm was used because it is easy and quick to implement and given enough time the snowplow should cover most of the map. Figure 6 shows the state machine of the path planning algorithm. The algorithm goes as follows:

1. Poll vision sensors for perimeter detection
2. Poll proximity sensors for obstacle detection
3. Set control variables
4. Make decision based on control variables
5. Repeat

There are 3 Boolean control variables for perimeter detection (Left, right and center), 4 Boolean control variables for obstacle detection (Left, right, front and back), and 2 floating point variables for obstacle distance measurements (Left and right). The Booleans are set to true when the sensor's detect an object. Based on which variables were true, a decision can be made. The perimeter has precedence over obstacles for the snowplow's movement, this means that if an obstacle and perimeter is detected, the snowplow will make a movement decision based on the perimeter, as it is important the snowplow doesn't leave the perimeter. In the cases where the perimeter, or obstacles are on both sides of the snowplow, the snowplow is biased to turn left.

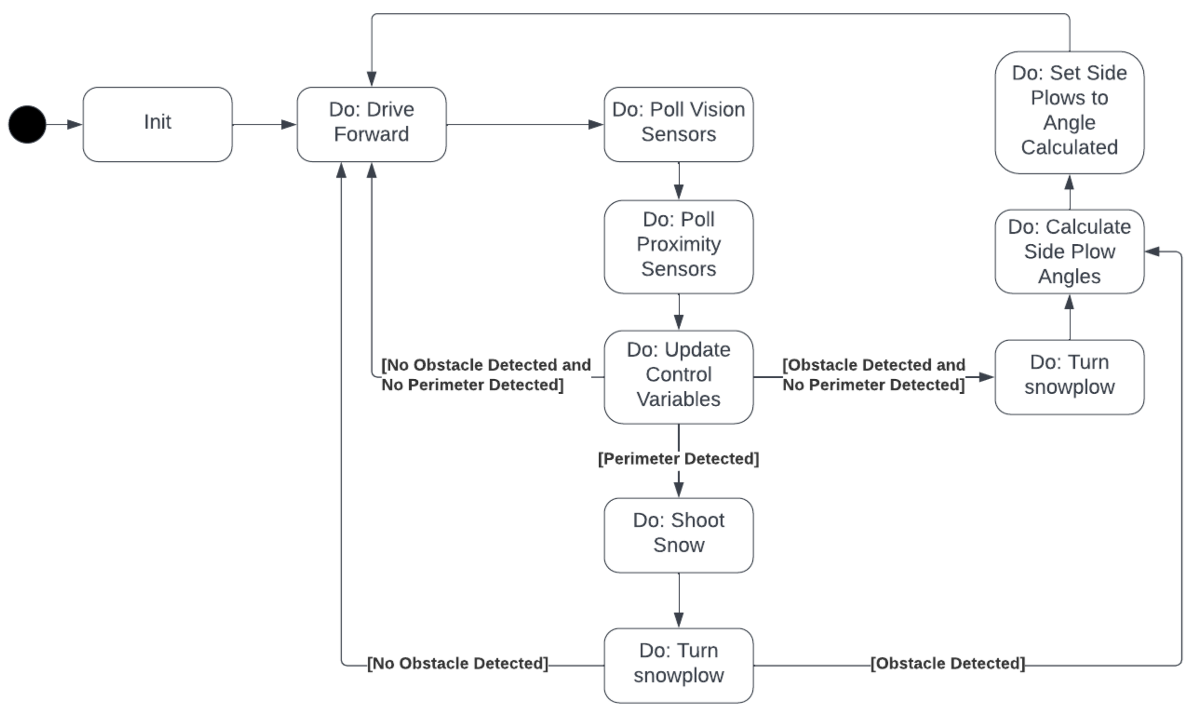

Figure 6: State machine of the snowplow's path planning algorithm.

3.4 Time Triggered Event Polling:

This project will utilize time triggered polling. This will allow this plow to operate with a smaller, limited number of sensors compared to an event triggered design. This lower number of sensors will limit the complexity of polling each sensor after an event. Furthermore, a time triggered design will be far quicker, and faster to program due to two reasons: The plow will not require service routines for each event, and each event can be treated as the same event rather than differentiating between all types of events. It is easier to set up this time-based scheduling compared to event-based because programming the logic to trigger the ETL once the event has occurred is no longer necessary.

4.0 Snowplow Implementation

This section will cover the layout and operation of the snowplow object, including the body design choices, the sensor setup, as well as the snow collection/removal functionality.

4.1 Robot Body

The body of the snowplow includes everything attached to the plow except for the sensors. This includes the wheels, motors, joints, chassis, and plow(s). The chassis is a basic rectangular shape that is a composite of a few additional parts: The cab, the chassis itself, and the ground extension planes.

The ground extension planes are on both sides of the chassis that prevent snow from getting extremely close, or under the snowplow to hinder movement and operation.

Attached to the back sides of the chassis are two revolute joints with cyclical movement enabled. Each wheel is then attached to these joints such that when the joints rotate, the wheels rotate at the same angular velocity. The robot only uses two wheels because of the spherical caster wheel under the chassis near the top, on the other side of the other two wheels. The caster wheel is a ball on a swivel that acts upon the force generated by the two other wheels, making the robot easier to move. The robot only uses two wheels and a caster wheel instead of a more standard 4-wheel approach because of a few conditions. 1. The team understood that all maps and settings did not include any changes in the floor (no uphill or downhill). 2. Budget. And 3. No noticeable improvement was observed with the 4 wheels. In the case of this 3-wheel approach, all sharp corners and turns can be successfully maneuvered when one wheel moves forwards and the other wheel moves backwards.

The last component(s) of the robot body are the 3 plows. The first and foremost is the front plow, a triangular object with its pointed tip at the apex of the robot. This stationary plow acts as an icebreaker, which deflects snow to the left or right side of the robot. The other 2 plows are on the left and right sides of the robot. They are long, thin paddle-like objects that can extend into and out of the body by their corresponding revolute joints. The functionality of the plow arms is detailed in sections 4.2 and 4.4 below.

4.2 Proximity Sensors

The proximity sensors provide the basis for the autonomous motion of the plow when inside the perimeter of the map. There are a total of six proximity sensors attached to the robot body that remain invisible during the operation of the plow. There are 3 located on the front of the robot, and 2 on the sides, culminating for around 200 degrees of total vision. There is also an additional sensor at the back of the robot.

The scope of vision for each sensor is about 1 meter. This is because the robot is only interested in reading Boolean true or false data for each sensor, if the sensors were able to ‘see’ farther, the robot would avoid obstacles too early and miss snow. Obstacle avoidance is the sole purpose of these sensors, constantly reading and sending data from the simulation to the robot in terms of where obstacles are present, and where they are not. As explained in the movement algorithm section of the report, the algorithm will dictate which direction to drive in based on the readings/data of the proximity sensors.

The true or false readings of the sensors also allow for the extension and retraction of the side plow arms to occur. For instance, if the proximity sensors on the right tell the robot there is an obstacle, the right plow arm will quickly retract back into the body of the robot to avoid collision and damage.

Similarly, if the right proximity sensors tell the robot there is nothing on the right side, the arms remain extended. This process occurs for both plow arms.

4.3 Vision Sensors

There are 3 vision sensors used in the project namely left, right and middle sensor. All the vision sensors are pointed towards ground to detect the perimeter/outline of the provided maps/area and perform their respective operations. Two of the vision sensors (left and right) are located on the front left and front right of the vehicle body and one sensor is located at the middle (inside) of the front plow. All of the sensors are hidden.

Left & Right vision sensors are responsible for making the robot stay inside the perimeter of the map. Both sensors are set to the default value of False, and when the vehicle reaches the perimeter/black line the sensor is triggered, and the sensor returns a value of true. After that the velocity of the right wheel is set to zero and the left wheel is set to a value between 1 to 3 so that the vehicle turns towards the right.

The middle sensor is used in the same way to detect the perimeter but is triggered to through snow out of the map once it reaches the black line of the parameter. The process of snow removal is discussed in detail in the next section 4.4.

4.4 Snow Collection & Removal

The snow collection and removal process are built upon the foundation of everything explained in this section, as well as the movement algorithm to successfully collect and remove snow. This includes the robot body, sensors, plows, and code, all of which were developed with the ability to clear snow in mind.

The plow arms are the main source of collecting snow, with snowballs either landing directly in the plow or deflected off the triangular tip. The snow is then released off the map during a pinball-like process in which the plow arms retract extremely fast and hit all the snow off the map and out of the perimeter. This includes reading the vision sensors to determine if the robot is currently located on the perimeter, but snow is only deflected when the perimeter is active on the front of the plow. This means that if the robot is moving along the left side of the perimeter, it does not trigger the pinball motion to remove snow. In the case of moving along one side of the perimeter, it must reach the corner in order to trigger the pinball motion, because the vision sensors will be triggered in the middle sensor.

However, the initial pinball mechanism conflicted with our existing algorithms for the proximity sensors, being that snow collected in the arms will be lost when the robot encounters an obstacle, so the arms retract, losing all the snow. To make up for this, our robot had to be configured to slowly turn to avoid the obstacle if only the top side sensor was activated and slowly retract the arms if both the top side and direct side sensors were activated in order to retain most of the snowballs.

5.0 Testing Results

This section will discuss the testing methodology and results. The snowplow’s obstacle detection, perimeter detection and side plow control has been tested.

The testing maps supplied during the term were used for testing obstacle avoidance. 5 minute simulations were run on each map, and the number of collisions was counted. Initially during testing, the side plows' angle algorithm wasn't perfect, and many of the early collisions in testing were caused by side plows colliding into walls while turning. Thus, we have changed the algorithm so that the side plows close faster, and that the vehicle turns with a sharper radius. The trickiest obstacle is the moving obstacle. In most scenarios, the snowplow will navigate around the moving obstacle well. In the case where the moving obstacle is heading towards the snowplow, sometimes they collide, as the robot move slower than the moving obstacle. As the path planning algorithm improved, the number of collisions decreased.

To test the side plows, a map was created with two long cuboids. The cuboids are angled 8° towards the snowplow and have leave enough space for the snowplow to drive between them. As the snow drives between the two cuboids, the space gets smaller and smaller, and thus the side plows have to be retracted. You can observe the snowplow traverse the map and visually look for any collisions. Using this method, the equation for the side plow angles was refined by scaling the value by 90%. Figure 9 shows the side plow's angles being tested.

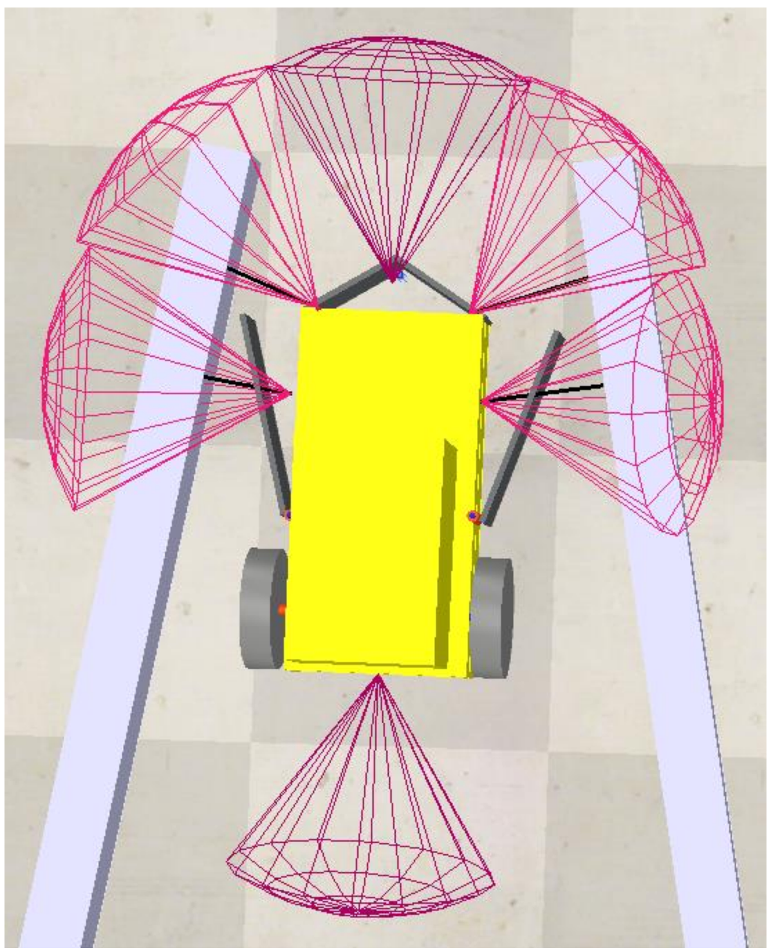

Figure 7: Snowplow side angle testing.

For obstacle avoidance and perimeter detection, the vehicle was tested on all three provided maps. Our algorithm consists of multiple if-else statements which enact different scenarios that the vehicle could face while moving on the map. Initially when tested the vehicle did not perform well due to lack of the decided /thought scenarios. But after testing it frequently on different provided and self made maps, the efficiency of the vehicle movement was brought close to 95% which means the vehicle can move and stay on the map for 95% of the time. The most tricky situation was when there is an obstacle detected on one side and a perimeter detected on the other but it was perfected by rigurous training of the vehicle. Figure 8 shows the testing of the obstacle avoidance and Figure 9 shows the shows the testing of the perimeter detection.

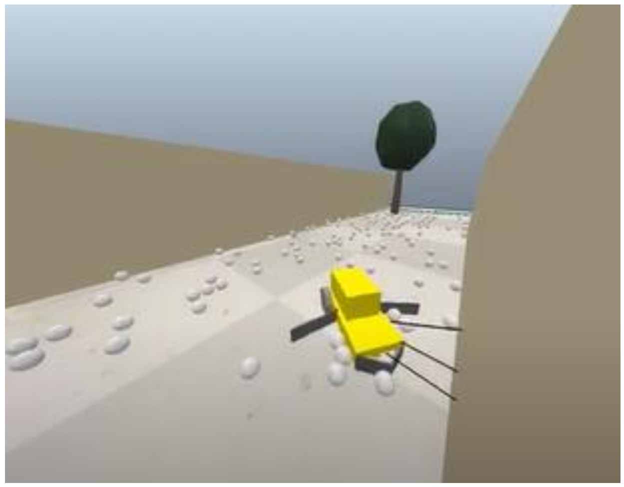

Figure 8: Obstacle avoidance testing.

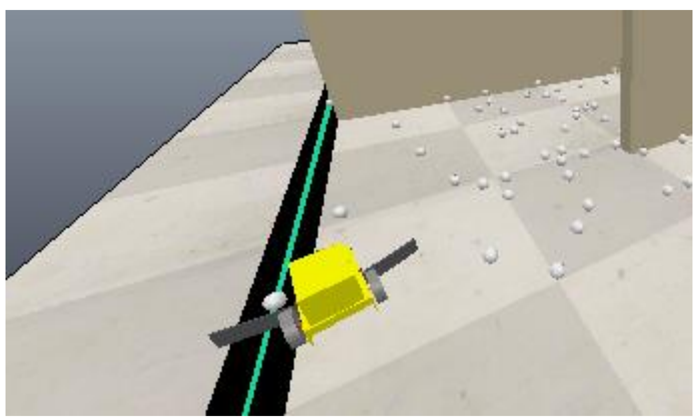

Figure 9: Perimeter detection testing.

To verify that the project is meeting its requirements, the requirements are tested, and the results are logged in control charts.

The control chart in figure 10, displays the variation is the robot's velocity. To verify the snowplow's velocity, we timed the snowplow driving 10 meters in simulation using real-time mode. The upper control limit is 2 m/s as described by the project outline, and the lower control limit is chosen to be 1.5 m/s. We chose this value, because we want the robot to perform it's task quickly, but still leave enough room for variation in velocity.

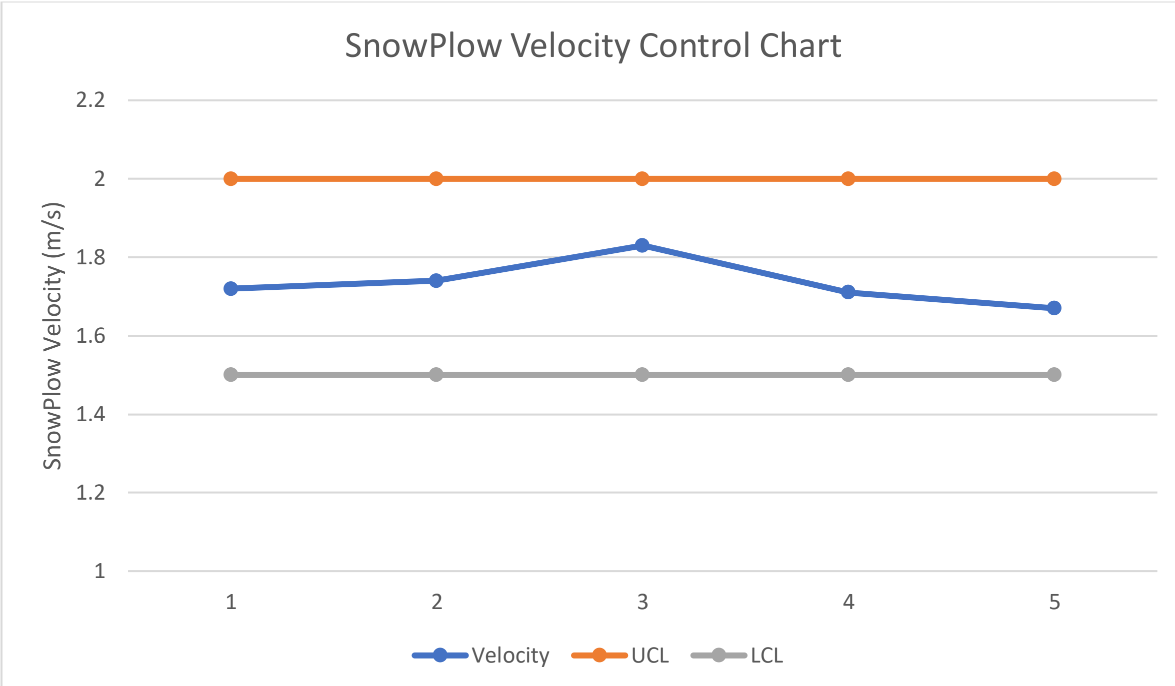

Figure 8: Control chart of the snowplow's velocity.

Link to GitHub repository :

https://github.com/RamitMahajan/SYSC4805-Group11.git

Branch name : Final