Computer Systems Design Lab

Instructor: Mostafa Taha

SYSC4805 Project Proposal Report

Team Egyptian Blue, Section L3 Group 5

Ahmed Ali, 101181126

Duncan MacLeod, 101160585

Tauheed Alamgir, 101194927

Brendan Kennedy, 101203359

Report Start Date: October 4th, 2024

Report Due Date: October 18th, 2024

Table of Contents

1.0 Project Charter

1.1 Team Name

Egyptian Blue

1.2 Objective

The aim of this project is to create a working prototype of a snow removal autonomous robot. The robot is tasked with removing simulated snow in the shape of small cubes. The robot will navigate around both static and dynamic obstacles in the shape of wooden boxes within an established 6 m^2 circular arena marked by a black tape. The basis of our concept is the integration of three separate sensors on Arduino DUE and Nano Every microcontroller boards, which will give the robot strong navigational capabilities and enable it to maneuver around both static and dynamic obstacles. The robot's design will meet specific criteria, such as a speed limit of 30 cm/s and a 5-minute operational window, which are crucial to avoid penalties. The plow design is equally as important, as we need to maximize our snow clearance to earn most of the points. Our start point is trial and error and make sure we are logging different types of data. This logging will not only let us monitor the speed and accuracy of our robot, but it will also be a crucial tool for analyzing sensor data, improving our design, and pinpointing possible improvement areas. By working on this project, we are hoping to gain knowledge in both embedded systems and robotics fields to apply in our future careers.

1.3 Project Deliverables

1. Project Proposal (October 18th)
   - An informational report about the requirements, scope, schedule, cost, and risk analysis.
2. Progress Report
   - Here we will highlight the challenges that have been faced during the project and the solutions that have been implemented to address them. An updated project timeline and milestones will be presented. Any changes made to the project's scope, or its requirements will be detailed comprehensively.
3. In Lecture presentation
   - The presentation will discuss team roles and individual contributions. An interactive session with the audience will be held, which will include a Q&A segment. The highlight of the presentation will be a live demonstration showcasing the robot's functionalities.
4. Lab Demonstration

- The demonstration will showcase the robot's capability to avoid obstacles and remove the simulated snow from the arena. The robot's final software interface and its control functionalities will be provided on the GitHub codebase.

5. Final Report
   - The Final Report milestone will start with an executive summary, providing an overview of the entire project. This will be followed by an in-depth analysis of the project's results and outcomes. Recommendations for future work and potential improvements will be discussed. The report will conclude with appendices containing detailed data, charts, and code.

Final Deliverables

- Finalized robot design
  - A fully functional robot equipped with integrated sensors, the plow attachment, and any other necessary modifications based on milestones' feedback.
- Source code:
  - Complete code base that shows the logic of all the sensors.
  - Documentation explaining the logic of the sensors.
- Final Report
  - Detailed description of the finalized robot prototype with integrated sensors and plow attachment.
  - Comprehensive documentation of the software, including code logic, execution flow, and operational procedures.
  - Sensor data interpretation, providing insights into environmental interactions, obstacle detection accuracy, and sensor reliability.

2.0 Scope

2.1 Requirements

Functional Requirements

- Upon activation, the robot’s system shall start its operation sequence within the boundaries of a space of 2.5 x 2.5 meters.
- On power up, the system shall perform a full calibration of all sensors
- During movement, the system shall detect small wooden cubes with a side length of 20mm and push them towards the designated outside perimeter.
- When approaching the black path boundary, the robot shall prevent itself from crossing the line by more than 5cm. The wheels going outside the boundary by more than 5cm results in the deduction of 5 cubes.
- Upon detecting an obstacle in its path, the system shall navigate around the obstacle without causing any significant displacement.
- During operation, the robot shall maintain dimensions within 226 x 262 x 150 mm.
- The robot’s speed shall stay below or at the limit of 30cm/s.
- Upon activation, the robot should clear all the wooden cubes within a duration of 5 minutes.
- In any case of a small hit with the obstacle which does not move the obstacle, results in the deduction of 5 cubes.
- In any case of a strong hit with the obstacle, which pushes the obstacle away, results in the deduction of 10 cubes.
- Any human interaction with the robot for adjustment or resets, results in the deduction of 10 cubes.
- If the robot’s speed exceeds the permissible limit at any point during the test, results in the deduction of 10 cubes.

Non-Functional Requirements

- During all operational phases, the robot’s speed should maintain a consistent speed not exceeding 30cm/s.
- Throughout the task, there shall be a timer to maintaining a countdown not surpassing a total of 5 minutes for task completion.
- Throughout the tasks, there should be a smooth and precise movement to effectively move around obstacles without causing displacements.

Software Requirements

- For enhancing modularity, the function handler of at least three sensors should be submitted to the designated GitHub repository.

2.2 Deliverables

2.2.1 Work Breakdown Structure (WBS)

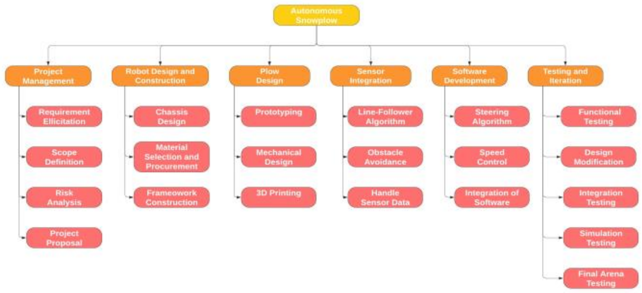

Figure 1: A list of detailed deliverables that cover all the requirements, organized in a WBS.

2.2.2 Movement Module

The Movement Module will be based on a PID control system loop. The system will have two input signals: The voltage signals driving the left and right sides of the robot. To help illustrate, a rough figure of the robot chassis is shown below.

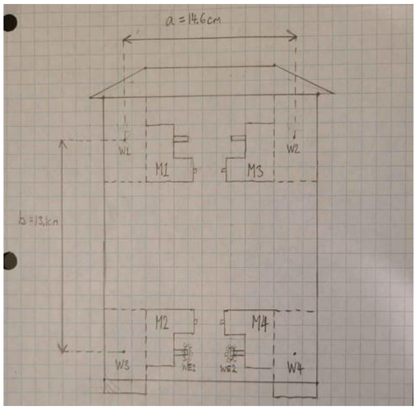

Figure 2: Diagram of rough robot chassis with labelled motors

Each motor is being driven by a voltage waveform and producing an angular velocity as a result. Given that motor pairs (M1, M2) and (M3, M4) are a fixed length apart, we simplify this problem by reducing from four independent elements to just two, one for each side. Let $vL\ (t)$ be the PWM signal driving the left motors M1,M2 and $vR\ (t)$ be the PWM signal driving the right motors M3,M4. $vL\ (t)$ and $vR\ (t)$ are the inputs to the PID system. The outputs to the PID system are the left and right angular velocities $\omega L\ (t)$ and $\omega R\ (t)$. These output angular velocities can be captured by the left and right wheel encoder outputs. Having access to the output velocities is critical because it allows us to create a closed-loop system with input signals $vL\ (t)$ and $vR\ (t)$ controlling outputs $\omega L\ (t)$ and $\omega R\ (t)$.

Figure [] below shows the general block diagram of the movement module

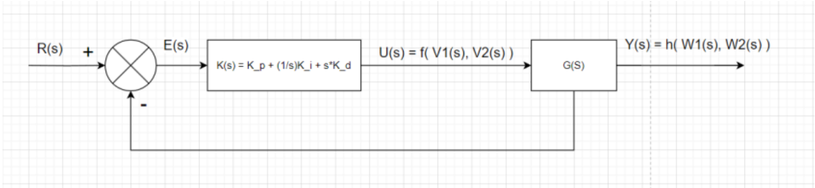

Figure 3: Block diagram of movement module

Where:

$R(s)$ is some desirable output state

$E(s)$ is the difference between the desired output, $R(s)$, and the actual output, $Y(s)$

$K(s)$ is the actual PID controller

$U(s)$ is some function of the two voltage inputs

$Y(s)$ is the output, which is some function of the two outputs

The exact form of $\mathrm{U}(s)$ and $Y(s)$ will depend on the task being carried out. If, for example, the Movement Module was tasked with producing straight line movement, then $Y(s)$ would simply be the difference between the two angular velocities, and the desired output would be zero.

2.2.3 Event Manager

The Event Manager will be the main interface between the sensors and the rest of the system. All relevant sensor information will be fed into the Event Manager, which will monitor incoming data and send commands to the Movement Module as needed.

2.2.4 Coordinate System

The Coordinate System will act as a bare-bones map of the environment and will assist the robot in keeping track of its position in the demo environment. The Coordinate System will feed into the Event Manager, allowing the Event Manager to cross reference between its expected surroundings, according to the Coordinate System, and its observed surroundings, according to various sensor feeds being received. Having an estimated position reading may prove useful for anticipating potential stops/redirections earlier in advance, allowing for more reliable and robust movement

2.3 Testing

Unit Testing

Ultrasonic Distance Sensor

- This sensor allows the robot to be able to detect obstacles and avoid them.

ToF Distance Sensor

- The ToF sensor allows the robot to detect both static and moving objects.

Analog Distance Sensor

- Test the visibility scope, reliability, and precision of each sensor for variable distances and angles. With a large enough dataset

IMU Gyroscope, and Accelerometer, and Magnetometer

- The gyroscope reading can reliably be used to determine how fast the car is turning

VMA330 Obstacle Detector

- These sensors can detect obstacles from a range of 2cm to 40cm.

Line Follower Sensor

- The line follower sensor allows the robot to keep track of the black painter's tape.

Robot Design and Construction

- Chassis Design: We will physically measure the dimensions of the chassis to verify it against the dimension of 226 x 262 x 150 mm. There should be no deviation beyond this tolerance.
- Material Selection and Procurement: Materials will be checked against the procurement list to ensure alignment with the design requirements.
- Framework Construction: Our team will examine the robot's framework for any visible structural defects and conduct a stability test under various load conditions. Framework should be stable under all conditions and free from defects.

Plow Design

- We will review the AutoCAD design and ensure it matches the physical prototype. We will also check the plow's ability to push lightweight wooden cubes simulating snow.

Sensor Integration

- Each sensor will be tested and evaluated (previous labs) to ensure the system effectively processes and responds to senor data. For instance, with the line follower algorithm, we will mark a black path and observe the robot's behavior when approaching. Each sensor should provide accurate readings and respond correctly to its designed parameters.

Integration Testing

Integration testing focuses on evaluating the interfaces and interactions between integrated components or systems. For our robot, the goal of integration testing is to confirm that individual modules, when combined, work as expected in unison.

- Chassis and Plow Integration: This is to confirm that the attachment mechanism between the chassis and the plow is secure. We would integrate the plow's mechanism with the robot's movement controls and test that the robot navigates and pushes lightweight wooden cubes with the plow attached. There should be no hindrance in the robot's movement.
- Sensor and Chassis Integration: With all the sensors mounted on the robot, we shall evaluate if the chassis supports them without causing interference or obstructing their functions. The robot's movement will be tested to determine if the sensors maintain their functionality and provide accurate reading consistently.
- Error and Exception Handling: We will intentionally induce errors, like blocking a sensor or overloading the robot, to see if the integrated system can identify the issue and handle it gracefully. The robot should not crash or behave erratically. Instead, it should follow predefined procedures for error handling, like stopping and signaling an error with a meaningful message.
- Functional Requirement Integration
  - Controlled Environmental Testing: In the designated testing area of 2.5 x 2.5 meters, we must make sure the area is free from external disturbances and distractions. Within this area, we will then set up specific conditions such as different cube placements and boundary markers to test the robot's behavior against these requirements.

- Boundary Checks: We will mark the testing area with a clearly defined black path boundary (black tape) and observe the robot's behaviors when it approaches this boundary. The robot should not cross the line by more than 5cm.

Movement (Wheel Encoders + Motors):

The testing for the robot movement will include 3 stages:

1. Speed: Can the robot maintain a speed within a 1cm/s error

Test: The test for speed will be operating the robot to move at 20cm/s. The test will pass if the robot can maintain a speed of 20cm/s +- 1cm/s. If the robot's speed drops below 19cm/s or above 21cm/s, the test is a failure.

2. Direction: Can the robot maintain a straight line without veering off path

Test: To test straight line operation, a makeshift boundary (line of tape) will be setup, and the robot will be setup right next to it and will move straight forward for 2.5m. The test will pass if the robot maintains within 2cm of the tape, but fails if it crosses over the tape or moves away from the tape 2cm or more.

3. Turning: Can the robot accurately turn left and right (90 degrees)

Test: To test turning, the robot will be tested at attempting a 90 degree turn in both left and right directions. The test will pass if the robot is able to turn in a range of 88 degrees to 92 degrees (98% accuracy). If the robot turns more than 92 degrees or less than 88 degrees in either direction, the test is a failure.

Navigation:

The testing for navigation will include 2 stages:

1. Mapping: Can the robot accurately map a rough grid of the area in which it clears

Test: A makeshift grid of tape will be made, and the robot will converse around the grid for 1 minute, and must accurately map the border of the grid. The test will pass if the robot can accurately map the grid with 98% confidence, meaning a +- 2% margin on dimensions of grid. The test will fail if this confidence margin of 98% is not met, or the robot fails to create the map in the first place.

2. Positioning: Does the robot know it's position and orientation within this grid

Test: To test the positioning, the robot will be placed in a makeshift testing grid, and will navigate to random points within the grid. The test is to have the robot communicate in Arduino its current coordinates within the grid and it/s orientation XYZ. The test will pass if the readings are approximately 95% accurate compared to the expected (observed) data. The test will fail otherwise.

Sensor Communication with Robot:

Each sensor will be calibrated within the unit testing, but will need to communicate with the robot to perform certain tasks:

1. Obstacle Detection -> Speed: Can the robot accurately detect the distance from an object and decelerate accordingly

Test: An obstacle will be set up 1 meter from the robot, and the robot will move at 20cm/s towards the obstacle. The robot must decelerate smoothly as it gets closer to the obstacle and comes to a stop about 5cm from the obstacle. The test pass/failure will be observed as so: If the robot decelerates relatively smoothly proportional to its distance from the obstacle, and stops within a range 4.5 to 5.5 cm, the test will pass. If the deceleration is sudden or the robot drives into the obstacle, the test will fail.

2. Line Detection -> Speed: Can the robot slow down or stop if a line is detected closely in front of it.

Test: A line of tape will be set up to simulate the border. The robot will move towards the tape and the line must be detected by the line detection sensor and stop the robot. If the robot drives over and past the line, the test will fail. If the robot stops at the line, the test is successful.

3. Grid -> Navigation: Can the robot maneuver itself around its constructed grid maintaining knowledge of its position and orientation.

Test: The test for grid navigation will be the final test in our testing plan. Integrating all our previous calibration, the robot will move around an empty grid, following our constructed navigation plan. As the robot navigates it must detect lines to keep itself within bounds, must detect obstacles and avoid hitting them, and will be displaying its position and orientation to the Arduino terminal whilst navigating. The passing criterion for this test is that the robot stays in boundaries (wheels within 5cm of line), avoids obstacles (stops and turns when detected), and can cover the area in our navigation plan. The test will fail if any of these criteria are not met in tandem.

3.0 Schedule

3.1 List of Activities

Table 1: List of activities that covers all the deliverables in the WBS

	Activity	Detailed Activities
1	Project Management (PM)	1.1 Planning and research (PM – 01) 1.2 Defining functional requirements (PM – 02) 1.3 Resource allocation (PM – 03) 1.4 Progress Monitoring (PM – 04)
2	Robot Design and Construction (RDC)	2.1 Metal chassis design and assembly (RDC – 01) 2.2 Material selection (RDC – 02) 2.3 Arduino and motor board placement (RDC – 03) 2.4 Sensor placement (RDC – 04)
3	Plow Design (PD)	3.1 Prototyping with different shapes and materials (PD – 01) 3.2 Develop final design (PD – 02)
4	Plow Final Design (PFD)	4.1 Obtain final structure using 3D printer (PFD – 01)
5	Sensor Integration (SI)	5.1 Sensor selection (SI – 01)
6	Lane Detection (LD)	6.1 Develop line-follower algorithm (LD – 01) 6.2 Sensor function handlers for data acquisition and processing (LD – 02)
7	Obstacle Detection (OD)	7.1 Integrate time-of-flight sensor for obstacle detection (OD – 01) 7.2 Sensor function handlers for data acquisition and processing (OD – 02)
8	Software Development (SD)	8.1 Control software for running all sensor concurrently (SD – 01) 8.2 Path planning and obstacle avoidance algorithm (SD – 02) 8.3 Speed control and steering integration (SD – 03) 8.4 Software-sensor integration (SD – 04) 8.5 Testing and debugging (SD – 05)
9	Testing and Iteration (TI)	9.1 Initial function testing (TI – 01) 9.2 Iterative design modifications (TI – 02) 9.3 Integration testing (TI – 03) 9.4 Simulation testing (TI – 04) 9.5 Arena testing (TI – 05)

10	Document and Reporting (DR)	10.3 Code documentation (DR – 01) 10.2 GitHub update (DR – 02)
11	Final Demonstration (FD)	11.1 Final presentation (FD – 01) 11.2 Demonstrating snow clearing in arena (FD – 02)
12	Project Proposal (PP)	12.1 Develop project proposal (PP – 01)
13	Progress Report (PR)	13.1 Create progress report (PR – 01)
14	Final Report (FR)	14.1 Create final project report (FR – 01)
15	In Lecture Presentation (ILP)	15.1 Perform in lecture presentation (ILP – 01)

Schedule Network Diagram

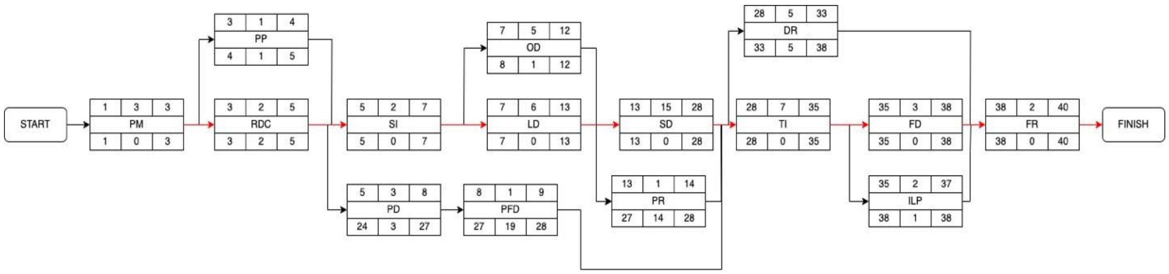

Figure 4: Schedule Network Diagram

Gantt Chart

Figure 5: Gantt Chart

4.0 Project Cost

Carleton University provides a kit valued at approximately $500. This kit includes various sensors and construction elements required for the robot project. These components are essential for the project's successful development and are generously provided by Carleton University to support the endeavor. In addition to the provided kit, there is an outside budget consideration for the snowplow design and implementation. The cost of this component depends on the material used, which is plastic, and the custom figure, which is 3D printed. The estimated cost for the plastic material and 3D printing can range from $20 to $50 per kilogram [2]. It's important to note that the snowplow's weight is a critical factor since a heavier plow would consume more energy for the robot's operation. Ideally, the plow's weight should maintain a ratio of 1:8 in comparison to the robot's weight. Proper cost allocation for these essential components is vital in ensuring the successful development of the project.

4.1 Cost Breakdown per Activity:

Core Movement Patterns – 4 developers, 3 weeks, 7h/week, 85h x $50/h = **$4200**

Navigation - 4 developers, 2 weeks, 9.5 h/week = 4x2x9.5 = 76 h x $50/h = **$3800**

Plow Design - 4 developers, 2 weeks, 0.75h/week = 4x2x0.75 = 6 h x $50/h = **$300**

Sensor Calibration - 4 developers, 3 weeks, 7h/week, 85h x $50/h = **$4200**

Unit Testing – 4 developers, total 7 weeks, 5h/week = 4x7x5 = 155h x $50/h = **$7750**

Integration Testing - 4 developers, total 7 weeks, 5h/week = 4x7x5 = 155h x $50/h = **$7750**

Course Kit = **$500**

Adding Costs Together - Total Cost: **$28,500**

5.0 Human resources

R = Responsible A = Approver

Activity	Ahmed Ali	Brendan Kennedy	Duncan MacLeod	Tauheed Alamgir
Project Management (PM)	R	A	/	/
Robot Design and Construction (RDC)	/	/	A	R
Plow Design (PD)	/	R	/	A
Plow Final Design (PFD)	/	A	/	R
Sensor Integration (SI)	A	/	R	/
Lane Detection (LD)	/	R	/	A
Obstacle Detection (OD)	A	/	R	A
Software Development (SD)	R	/	A	/
Testing and Iteration (TI)	A	/	R	/
Document and Reporting (DR)	/	R	A	/
Final Demonstration (FD)	R	A	/	/
Project Proposal (PP)	R	/	/	A
Progress Report (PR)	/	/	A	R
Final Report (FR)	/	A	R	/
In Lecture Presentation (ILP)	A	R	/	/

References

[1] *Carleton University, Project Description*
https://brightspace.carleton.ca/d2l/le/content/292069/viewContent/3933855/View
https://brightspace.carleton.ca/d2l/le/content/211845/viewContent/3442554/View

[2] U. (2023, March 21). *How much does 3D printing cost?* UltiMaker.
https://ultimaker.com/learn/howhttps://ultimaker.com/learn/how-much-does-3d-printing-cost/much-does-3d-printing-cost/