MANCHESTER
1824
The University of Manchester

Preliminary Design Review

Submitted to The University of Manchester for the degree of

Master of Science in Robotics

in the Faculty of Science and Engineering

Year of submission

2025

Student ID

Team 9 (14341814, 11103012, 14255884, 14215307, 14316539, 10826766)

School of Engineering

Contents

Word count:

1 Introduction

The objective of this project is to design and develop a fully autonomous robot capable of navigating a rectangular search area with randomly placed obstacles to collect 3 coloured cylinders and return them to matching-coloured bins at the rover's starting point within a 20-minute mission time.

2 System Design

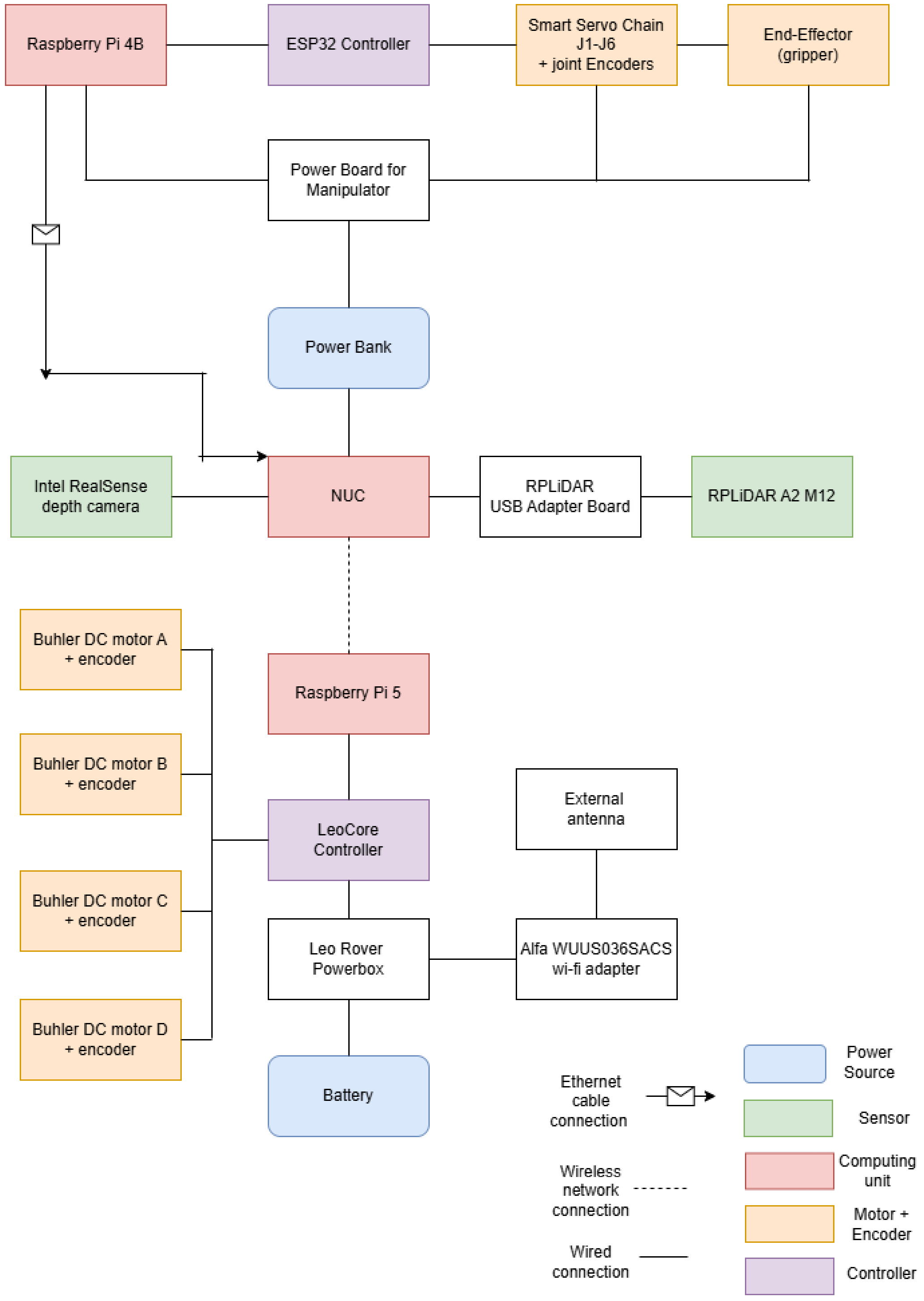

Raspberry Pi 4B
ESP32 Controller
Smart Servo Chain J1-J6 + joint Encoders
End-Effector (gripper)
Power Board for Manipulator
Power Bank
Intel RealSense depth camera
NUC
RPLiDAR USB Adapter Board
RPLiDAR A2 M12
Buhler DC motor A + encoder
Buhler DC motor B + encoder
Buhler DC motor C + encoder
Buhler DC motor D + encoder
Raspberry Pi 5
LeoCore Controller
External antenna
Leo Rover Powerbox
Alfa WUUS036SACS wi-fi adapter
Battery
Ethernet cable connection
Wireless network connection
Wired connection
Power Source
Sensor
Computing unit
Motor + Encoder
Controller

Fig. 1. Complete system architecture of the autonomous rover and manipulator platform.

2.1 System Components and Justification

The final hardware architecture combines sensing, computation, actuation, communication and power delivery in a structure suitable for reliable real-time operation. Each component in the system fulfills a specific role that is required for correct functionality, integration and performance during the 20-minute autonomous mission. Below, all components are described together with the justification for their inclusion.

Computing Units:

Intel NUC (Primary Compute Unit): The NUC is the main processing platform for the system. It runs all high-level ROS 2 nodes, including perception, LiDAR processing, object detection, mapping, and manipulation control. These tasks exceed the computational capability of embedded boards, so the NUC is required to guarantee real-time performance and unify the software stack on a single central machine. By including the NUC in the design, the rover can handle all computation locally, thereby satisfying FR12.

Raspberry Pi 5 (Leo Rover Host): The Pi5 interfaces with the LeoCore motor controller and acts as the rover's internal control computer. It runs the rover firmware, publishes wheel odometry, and provides the API used by the rest of the system. The Pi5 is required because the Leo Rover electronics are designed around it.

Raspberry Pi 4B (Manipulator Host): The MyCobot 280 manipulator requires a compact Linux board to interface with the ESP32, handle USB communication, and run the manufacturer's ROS packages. The Pi 4B serves as this dedicated control host and ensures the manipulator subsystem operates independently while still communicating with the main NUC.

Controllers:

ESP32 Controller: The ESP32 communicates directly with the smart servos, handles low-level joint commands, and streams encoder feedback. It provides deterministic timing and simple communication pathways that are required by the manipulator's control protocol.

LeoCore Controller: This module drives the four DC motors, reads wheel encoders, and provides odometry to the Pi 5. It is responsible for all low-level rover motion control and is necessary for safe and reliable operation of the Leo Rover drivetrain.

Sensors:

Intel RealSense Depth Camera: The depth camera provides RGB-D data needed for object detection, distance estimation, and visual guidance of the manipulator. It is also used for environment perception tasks where depth information is essential. This sensor directly supports FR6 by enabling accurate detection and localisation of target objects and FR8 by allowing the robot to identify the correct sorting bins.

RPLiDAR A2M12 and USB Adapter Board: The 2D LiDAR provides planar distance data used for localisation, obstacle detection, and mapping. It is a requirement for the navigation portion of the project which directly addresses FR1 (autonomous navigation), FR2 (colission avoidance), FR3 (obstacles detection and mapping) and FR4 (self-localisation). The USB adapter board ensures stable power and communication, which is necessary for reliable LiDAR performance.

Actuation:

Smart Servo Chain J1–J6 with Encoders: These servos provide all six joints of the manipulator with position feedback and closed-loop control. Encoders are essential to achieve accurate joint angles and repeatable motion.

End Effector (Gripper): The gripper is required for interaction with objects, enabling the system to perform manipulation tasks such as picking and placing

Bühler DC Motors A–D with Encoders: These motors power the rover's wheels. The encoders provide odometry for navigation and closed-loop control. They are essential for movement and localisation.

Communication Components:

Ethernet Cable Connection: A direct wired connection between the NUC and the Pi 4B (on the manipulator) ensures stable, low-latency communication with the manipulator subsystem. Wired communication avoids packet loss and timing uncertainty, which is important for real-time control.

Alfa AWUS036SACS Wi-Fi Adapter: Provides reliable wireless connectivity, allowing the robot to transmit and receive data, commands, and software updates throughout development and testing.

Power system components:

Leo Rover Battery: The main rover battery powers the drivetrain (DC motors), the LeoCore controller, the Pi 5, and onboard networking components. It is essential for all base rover operations. Powering the rover with this onboard battery satisfies PR9.2.

Power Bank (Additional Power Source): Power calculations showed that the rover battery alone cannot supply both the drivetrain and the high-load components (NUC and manipulator) for the full 20-minute mission duration. A separate power bank is therefore required to guarantee enough capacity and to avoid voltage drops. It cleanly separates the high-demand compute and manipulator power from the rover's mobility power. This ensures that the robot is capable of meeting PR9.1 by providing power to its sub-systems for the entire duration of the mission with some extra capacity to also account for energy losses. Having this extra power bank onboard also satisfies PR9.2.

Power Board for Manipulator: This board distributes power to the, Raspberry Pi 4B, smart servos and provides stable voltage to the manipulator subsystem. It is required to handle the current draw of multi-joint motion and protects the upstream power source.

2.2 Summary

Each component in the system plays a defined role in sensing, computation, actuation, or communication. The NUC serves as the central compute unit due to the computational requirements of perception and manipulation tasks. Additional power was added to ensure sufficient runtime and electrical stability. Together, the components form a robust architecture that supports the rover's autonomous and manipulation capabilities, equipping it with the necessary functionalities to meet the overall objective of completing the search, retrieval and sorting mission within 20 minutes (FR11).

3 Mechanical Design

3.1 Design Overview

Figure 2 shows a labelled diagram of the robot's CAD model containing the Leo Rover and a payload sled which holds the LiDAR, Next Unit of Computing (NUC), camera, manipulator and onboard storage bin.

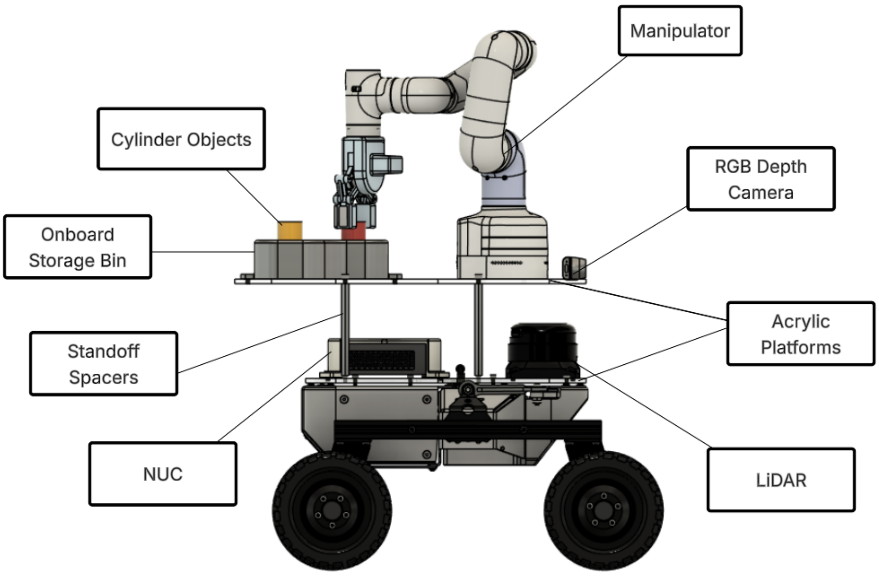
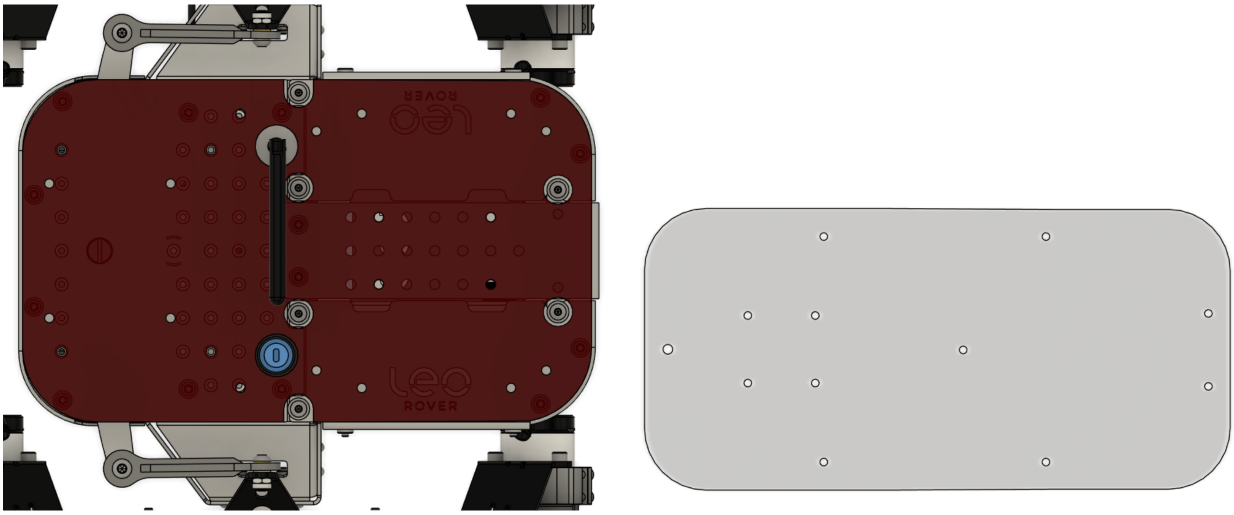


Fig. 2. Labelled diagram of the robot.

The payload sled consists of two 5mm thick acrylic platforms. The base platform mounted on top of the rover, supports the LiDAR at the front and NUC at the rear. The upper platform holds the camera at the front, the manipulator just behind the camera and the onboard storage bin at the back.

3.2 Design Considerations

When designing the acrylic base platform, extra care was taken to maintain access to critical components on top of the rover such as the antenna, micro-USB port and the 6 bolts which secures the battery and power boxes. To minimise flexing due to reaction forces from the movements of the manipulator, the base platform is secured to the top of the rover using four pairs of bolts along its length. The design of the upper platform is comparatively simpler as it only required attachment points for the camera, manipulator, storage bin and the four standoff spacers. Figure 3 shows the design of the base and upper platforms.



(a) Design of the base platform

(b) Design of the upper platform.

Fig. 3. Design of base and upper platforms.

It was decided that no support structures will be placed 180 degrees in front of the LiDAR to provide an unobstructed field of view ahead of the robot. This design choice aligns with FR3 as it prevents occlusion, enabling the robot to accurately detect and map obstacles in its field of view. The NUC is held in place using a 3D printed clamp that only covers its corners, maintaining accessibility to all its peripheral connection points. Figure 4 shows the NUC clamp and the LiDAR holder.

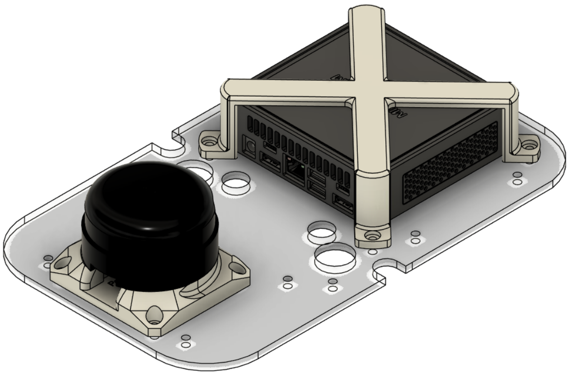


Fig. 4. Design of LiDAR holder and NUC clamp.

The onboard storage bin shown in Figure 5 was designed to hold 3 cylinders of 26mm diameter and 60mm height. It was decided that the storage space for these cylinders would have dimensions of 35 x 35 x 40 mm. This leaves enough of the cylinder exposed for the gripper to comfortably grasp it during retrieval from the storage bin. This also leaves enough tolerance for the manipulator to place the cylinder in the storage bin while still being able to hold the cylinder upright, in a predictable position. This is advantageous for picking the cylinder from the storage bin during the sorting process as the pose of the cylinder is fixed. The edges of the storage space are also angled at 30 degrees so that the cylinder can slide into the bin if the manipulator is slightly off-centre. As 66% of the cylinder is within the storage space, it is very unlikely that it would topple out of the onboard storage bin during the mission, especially when considering that the robot will be operating on a flat and even terrain. These design choices directly address FR7 and PR7.1.

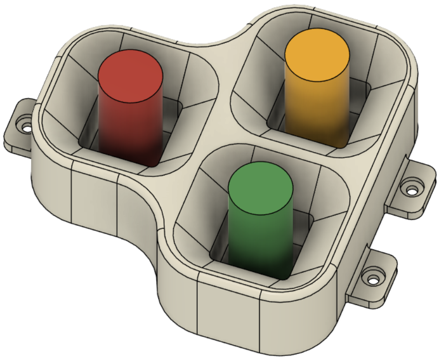


Fig. 5. Design of the storage bin.

3.3 Component Placement Considerations

The LiDAR is placed on the base platform at the front of the robot to increase its maximise its field of view and its chances of detecting obstacles. The 3D printed LiDAR holder was slightly modified from the version provided by fictionlab, its height was increased by a 5mm to raise the LiDAR such that it could see over the NUC towards the back of the rover. This increased the LiDAR's field of view, allowing the rover to map more of its environment in a single scan, maximising the robot's ability to satisfy FRs 1-4.

The camera is placed at the front of the upper platform to provide a clear view of the environment, maintaining a large field of view and increasing its chances of detecting a target object, addressing FR5.

The manipulator is placed just behind the camera. This simplifies the coordinate transform that needs to be computed from the camera's frame to the manipulator's base frame whenever an object is detected. Additionally, keeping the manipulator towards the front improves its ability to access the target objects during pickup without colliding with the upper platform or the camera. This ensures that the robot can successfully pick up the target object consistently, enabling it to meet PR6.1. Figure 6 shows the pose of the manipulator when picking up a target object.

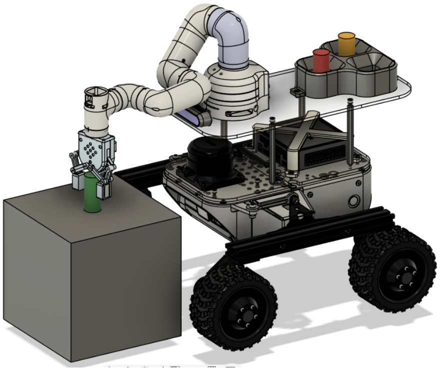


Fig. 6. The manipulator pose when picking up cylindrical object from the box that it is placed on.

The storage bin is placed far enough from the manipulator to allow it to comfortably access the closest storage space as illustrated in Figure 7 showing the arm picking the nearest cylinder. This improves the manipulator's likelihood of successfully placing cylinders inside the onboard storage bin's spaces, supporting compliance with PR6.2.

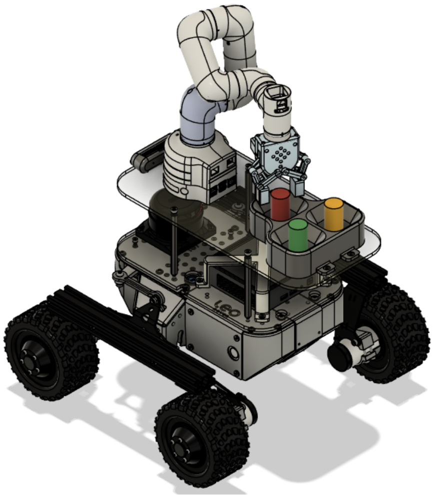


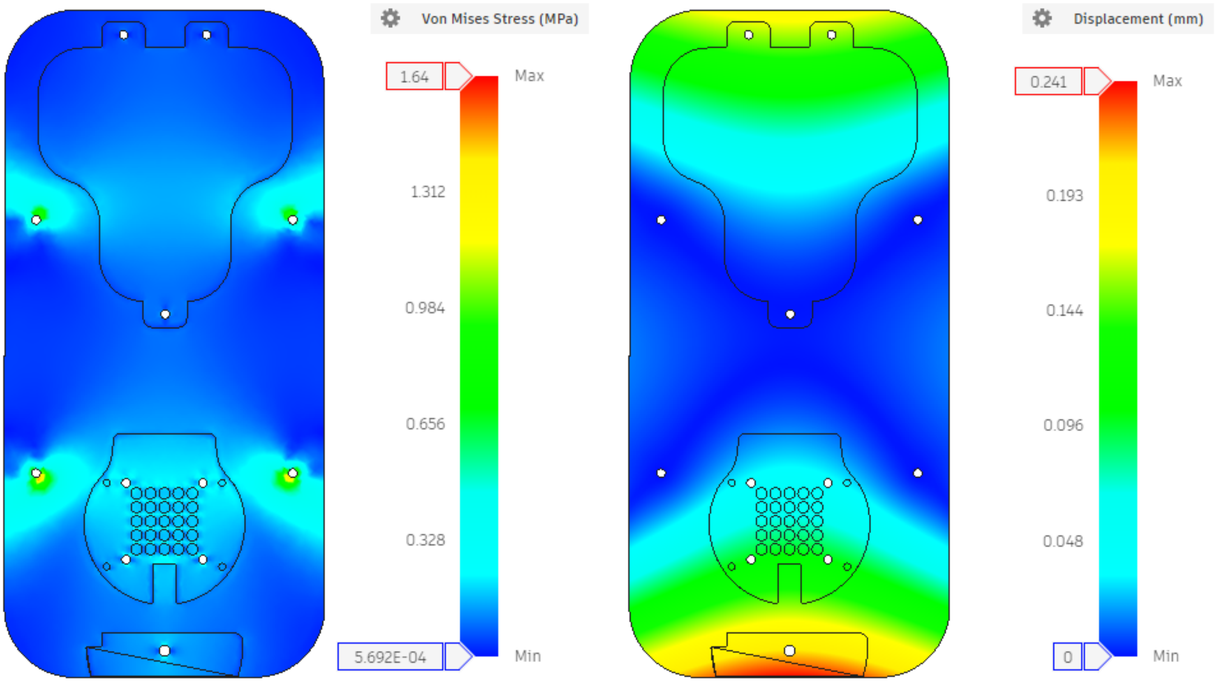
Fig. 7. Pose of the manipulator when picking the nearest cylinder from the onboard storage bin.

3.4 Material Choice and Fabrications

The base and upper platforms were made from 5mm thick acrylic as it is compatible with laser cutting and it is strong enough to withstand the weight of the components mounted on the rover. As the LiDAR holder, NUC clamp and onboard storage bin have complex geometries, they were fabricated using 3D printed PLA.

3.5 Strength and Stability

A finite element analysis was conducted to study the stress and deformation of the upper platform under the weight of the camera, manipulator and storage bin. The weight of these components was simulated as a distributed pressure acting on an area equivalent to their footprint on the platform and the locations were the standoff spacers connect to the platform was set to have fixed constraints.



(a) Stress distribution

(b) Deformation

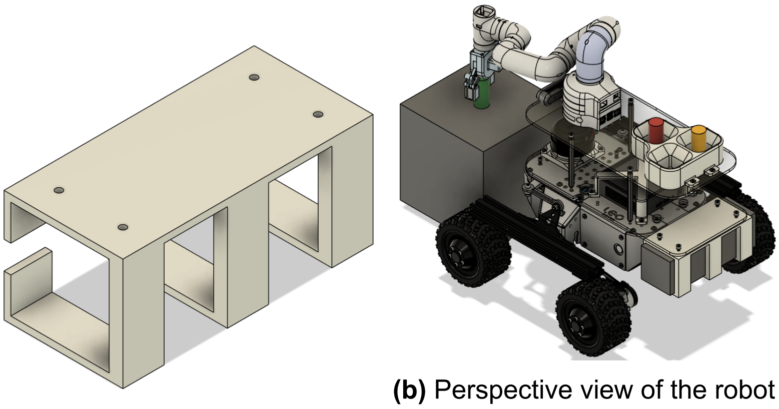
Fig. 8. Finite Element Analysis of the upper platform.

The stress distribution results show that the maximum stress observed is 1.64 MPa near the points where the standoff spacers meet the platform. This is well under the 70 MPa failure stress of clear acrylic material. A maximum deformation of 0.241mm is observed at the front of the upper platform which is also negligible compared to the size of the platform. As no significant sagging was observed in the FEA simulations of the upper platform, it can be concluded that the platform has sufficient strength to hold the camera, manipulator and storage bin. These results directly support the requirement that the robot shall maintain mechanical stability (FR10).

3.6 Future Considerations

The current design uses standoff spacers that are consistently detected by the LiDAR as nearby obstacles. This can interfere with Nav2 navigation, causing the planner to perceive false obstructions and limiting performance. A potential solution is to implement a scan filtering node to ignore LiDAR returns within a defined radius around the sensor. This would prevent self-detection of structural components while preserving mapping and navigation functionality.

Additionally, power budget analysis showed that an additional power source will be required to support all the components used in this system. and a 120 W laptop power bank was chosen. This power bank is a simple cuboid with dimensions of 71 x 71 x 210 mm. To accommodate this on the payload sled, the base platform was extended further from the back of the rover. The power bank will be housed in a 3D-printed holder, which will be bolted to the extended part of the base platform, as shown in Figure 9. Accommodating this extra power source in the mechanical design of the robot enables it to be fully self-powered, meeting PR9.2 by eliminating the need for trailing power cables.



(a) The power bank holder.

(b) Perspective view of the robot with the additional power bank.

Fig. 9. Additional power bank mounted at the back of the robot.

Extending the base platform has increased the footprint of the robot. This will be updated in the nav2 configuration parameters to avoid the risk of collision when the robot is reversing.

4 Electrical Design

4.1 Power Connection Diagram

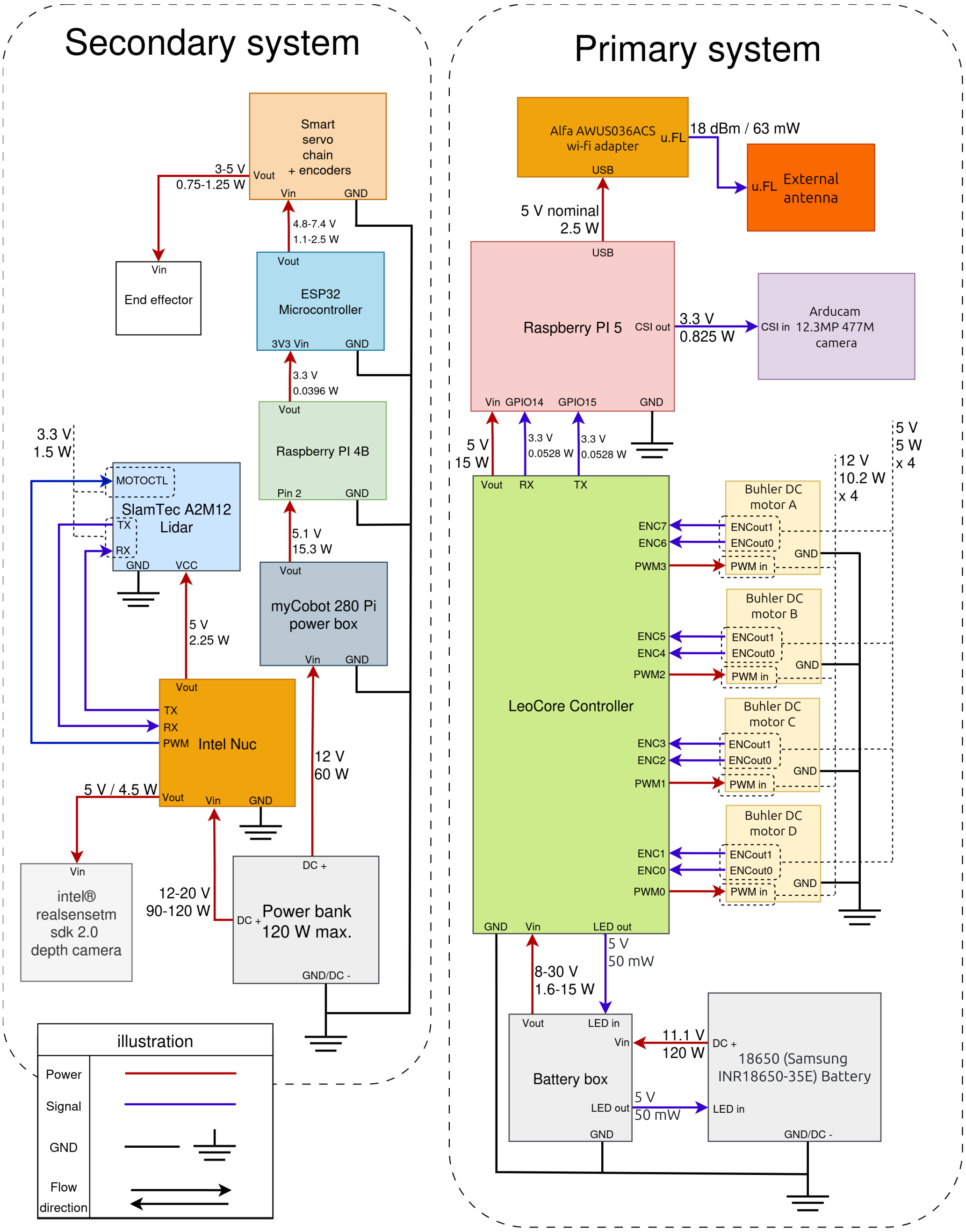


Fig. 10. Power connection diagram.

The power attributes of the manipulator's servo motors were estimated using the rated torque, stall torque and the maximum joint angular velocity.

The electrical diagram includes the typical voltage and power value of each sub-component and ports.

The gripper's power attributes were estimated using two similar products for the myCobot 320 gripper models (1).

The power attributes of the manipulator's servo motors were estimated using the rated torque, stall torque and the maximum joint angular velocity (2). The power can be calculated by using the torque and the angular velocity of the motor joints assuming 100% efficiency:

$$P = \tau \times \omega \quad \text{assuming } P_{\text{mech}} \approx P_{\text{elec}} \tag{1}$$

The rated current can be calculated as $I = \frac{P_{elec}}{V}$

From that we obtained: Rated power: 2.5W, Stall power: 1.14-1.23W.

The voltage and power values of all other electrical components were obtained directly from their datasheet or the corresponding online sources.

4.2 Power Budget Calculations

The power is calculated based on typical current usage. This was done assuming all devices are powered on simultaneously at constant rate. In real-life, the rover's total power drawn is highly dynamic and varies with time. Many electrical characteristics such as the transient effect, the frequency/phase response or the PWM duty cycle can change how much power the system uses over time. Hence, to calculate the power demand of the base rover (without the on-board manipulator and intel NUC), we used the power supplied by the battery and assume the base rover would be continuously operating for 4 hours. We then obtained a time average power usage of the base rover to estimate the actual power consumption. Apart from the base rover system, the on-board system consists of the manipulator, the intel NUC, the lidar and the depth camera can be estimated more easily as they have an operation power rating and requires constant power supply, making their power to be approximately time-constant. The power usage of the manipulator is relatively more time-varying so we used the minimum and maximum power for a more reliable estimation.

Table 1. Electrical characteristics and energy demands of rover subsystems and battery/power bank.

Item	Voltage (V)	Current (A)	Power (W)
Base rover (Primary system)			20
Base rover (Primary system) total electrical energy demand (assume 30 min operation)			10 Wh
Manipulator (2)	12	5	24–60
Lidar (3)	5 + 3.3	0.45 + 2.2	2.25 + 1.5 = 3.75
Depth camera (4)	5	0.5	2.5
NUC (5)	20	6	120
Secondary system total power			150.25–186.25
Secondary system electrical energy demand (assuming 30 min operation)			75.13–93.13 Wh
Whole system total power			170.25–206.25
Whole system electrical energy demand (assume 30 min operation)			85.13–103.13 Wh
Battery power			120
Battery total electrical energy			77.7 Wh
Power bank power (6)			Port1: 120 Port2: 65 USB Port1: 18 USB Port2: 12
Power bank total electrical energy (assume 30 min operation)			Total: max. 31.2 Ah / 116 Wh Port1: min. 60 Wh Port2: min. 32.5 Wh USB Port1: min. 9 Wh USB Port2: min. 6 Wh

The primary system consists of the LeoCore controller (7), the LeoRover RaspberryPI 5 board (8), 4

Buhler motors (9), the Arducam camera (10), the Wi-Fi adaptor (11) and an antenna. This system is responsible for navigation and movement of the rover. The primary system is supplied by the Samsung INR185650 battery pack which has a maximum output power of 120W and a total electrical energy of 77.7 Wh (7). This would supply enough power to the base rover which demands only 10 Wh of electrical energy for 30 minutes operation duration. A battery box is connected to the battery and re-distribute the power from the battery to all the components in the primary system. The controller accepts the power supply from the battery box and then distribute it to all the other attached components. The Raspberry PI board also act as a power supply to all the components that are attached to it.

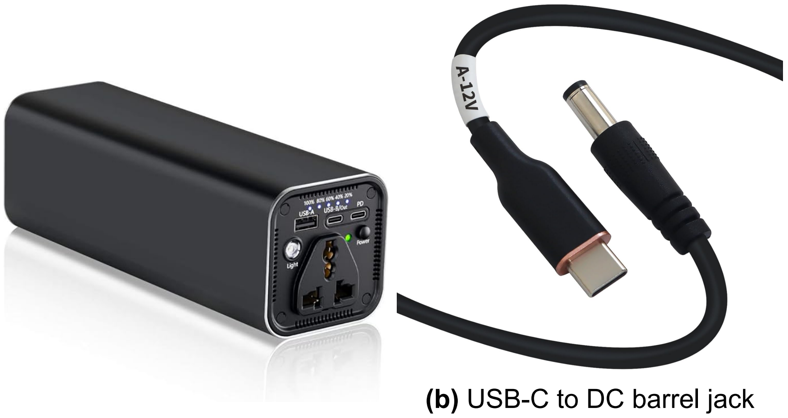
The secondary system is responsible for computer vision (mainly object recognition) and manipulation. It consists of the intel NUC, the myCobot 280 PI manipulator (with gripper attached) the lidar and the depth camera. The secondary system requires an additional power supply of 150.25-186.25 W or a total energy supply of 75.13-93.13 Wh. The manipulator system consists of a Raspberry PI 4B board (8), an ESP32 microcontroller (12), 6 metal servo motors (2) and the myCobot gripper (1). The manipulator has its own power box which will take the power supply from the additional power bank and distribute the power to all the connected components.

4.3 Proposed Solution

The power budget calculations address FR9. From the power calculations above, the battery provided is not able to supply the full electrical power or energy required by the entire system to perform the task for 30 minutes. However, it is in fact capable of supplying enough energy (assuming the operation time does not exceed 30 minutes) to the rover alone without add-on systems. The proposed solution uses an additional power source to power the manipulator as well as the NUC along with its sensors. The 120 W charging port of the power bank would be enough to power the intel NUC with required 120 W power supply. The 65 W outlet port would be able to supply enough power to the manipulator and the two remaining USB ports can supply enough power for the lidar and the depth camera. The power pack has total energy of 116 Wh (6) which is sufficient in supplying the secondary system which demands a maximum energy of (60 + 30 + 6.25) / 0.5 = 93.13 Wh (as the total task duration is 30 minutes the multiply factor is 0.5). Therefore, this proposed solution could supply more energy than the system requires and would enable the robot to meet PR9.1. As the robot is using an onboard lithium-ion battery and the remaining components are powered by a power bank, which is also mounted on the robot, it is not going to require any wired connections for power, meeting PR9.2.

Hence, we split the whole system into a primary system consists of the mobile base of the rover and all the onboard sensors, and a secondary system which consists of the manipulator, the Intel NUC and all its sensors.

We propose using the 31200mAh 120 W portable power bank shown in Figure 11 (6). This power bank offers a 120W AC port, a 65W PD Type-C port, a 12W USB port, and an 18W USB port, making it suitable for devices with different energy ratings. An additional USB-C to DC barrel jack cable is also required to connect the manipulator to the battery pack. The 12 V power cable shown in Figure 11 (13) is a suitable choice for this application as it is compatible with the manipulator's 12V operating voltage. The Intel NUC can be connected directly to the powerbank using a USB-C to USB-C cable.



(a) 120 W laptop power bank

(b) USB-C to DC barrel jack cable

Fig. 11. Components of the proposed solution.

5 Software Design

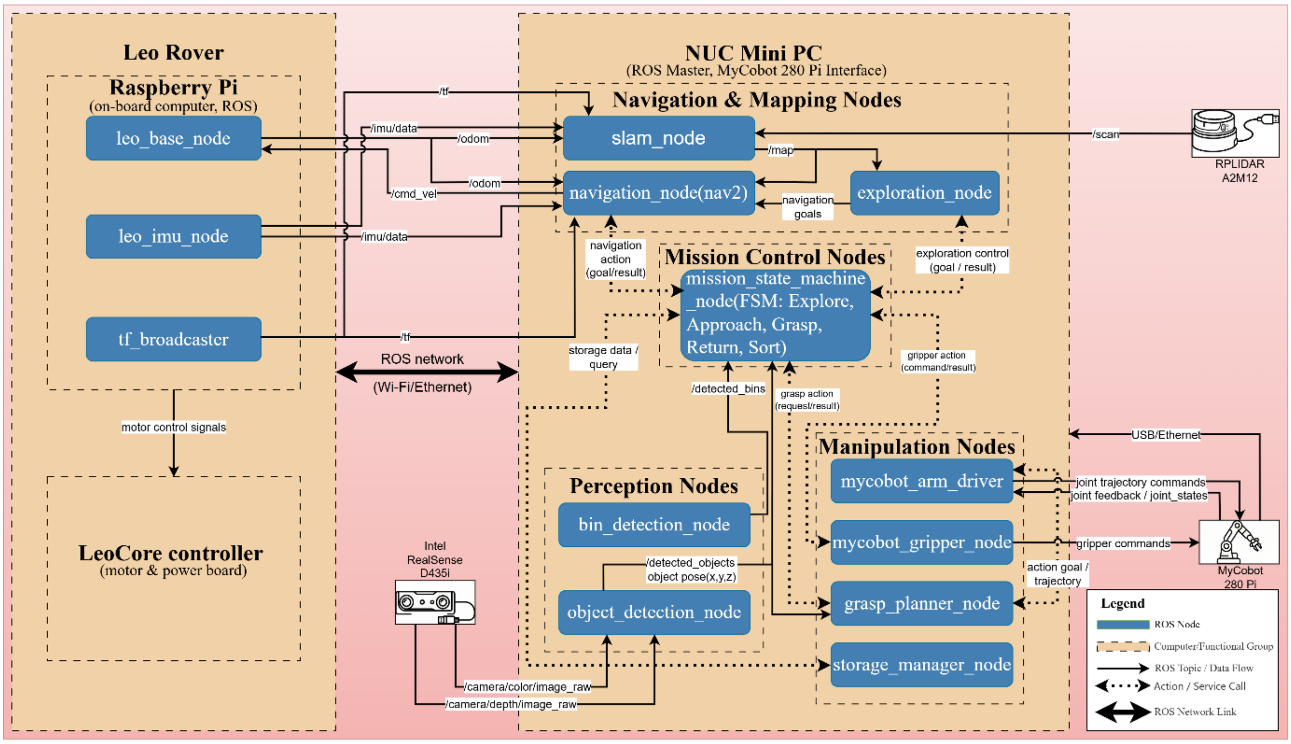


Fig. 12. Software block diagram.

5.1 Exploration Phase

After the state machine enters the Explore state, it starts exploration_node via exploration control. Based on /map and /tf, exploration_node performs frontier search on the global map, selects unexplored areas, and sends the corresponding exploration target points to navigation_node in the form of navigation goals. navigation_node conducts global path planning and local obstacle avoidance by integrating /map, /odom, /tf, and /scan, and controls the chassis motion via /cmd_vel. At the same time, the visual perception module operates continuously, and object_detection_node updates /detected_objects constantly. When mission_state_machine_node finds a graspable target object in /detected_objects, it stops exploration_node via exploration control and prepares to switch from the Explore phase to the Approach phase; if no target is detected for a long time and exploration_node returns results such as exploration_done or no_more_frontiers, the state machine can choose to end the task or return to the starting area according to the strategy.

5.2 Approach Target

In the Approach phase, mission_state_machine_node selects a target object (e.g., the nearest one) from /detected_objects, generates a navigation target pose (navigation goal) for approaching the object based on the object's position in the map coordinate system, and sends it to navigation_node via the navigation Action interface. navigation_node plans a path based on /map and outputs /cmd_vel to control the chassis motion, moving the robot to a suitable position near the object. After the navigation is completed, navigation_node feeds back results such as successful arrival/failure to mission_state_machine_node via Action result. The state machine switches to the Grasp phase when the arrival is successful; if it fails, it can choose to retry or reselect the target according to the strategy.

5.3 Grasping phase

After entering the Grasp phase, mission_state_machine_node first calls mycobot_gripper_node to execute the OPEN command to ensure the gripper is open. Then, the state machine calls grasp_planner_node via the grasping Action interface (grasp request/result) and specifies the target object to be grasped; grasp_planner_node subscribes to /detected_objects to obtain the latest 3D pose of the object, plans the grasping trajectory by combining with the current posture of the robotic arm, and sends the planned joint trajectory to mycobot_arm_driver in the form of action goal/trajectory for it to send to the robotic arm for execution. After the grasping trajectory is executed in place, the state machine calls mycobot_gripper_node again to execute the CLOSE command to grip the object. The trajectory execution result of grasp_planner_node and the execution feedback of mycobot_gripper_node are returned to mission_state_machine_node together. The state machine judges whether the grasping is successful based on this information: if successful, it updates the state of the corresponding storage slot in storage_manager_node (recording which type of object is placed in the slot) via the storage data interface; if failed, it can retry, switch to another object, or return to other states such as Explore/Approach according to the task strategy.

5.4 Return to Starting Point

When the number of grasped objects reaches the preset threshold or other conditions (e.g., time limit) are met, mission_state_machine_node generates a navigation goal for returning to the starting point from the starting pose recorded at the beginning of the task, and sends this navigation goal to navigation_node via the navigation Action interface. navigation_node uses /map for path planning and outputs /cmd_vel to control the chassis to return to the starting area along the planned path. After the navigation is completed, navigation_node feeds back the execution result to the state machine to prepare for the subsequent sorting and placing phase.

5.5 Sorting

In the sorting phase, the state machine obtains the poses of collection bins of different colors in the environment using /detected_bins provided by bin_detection_node, and learns the color/type and location of the objects in each slot on the vehicle using the storage slot information provided by storage_manager_node. Then the state machine executes the following steps in a loop:

- Query storage_manager_node to select the next object to be placed and the slot where it is located.
- Determine the pose of the corresponding target collection bin from /detected_bins according to the color/type of the object.
- Generate a navigation goal for moving to the front of the collection bin, send it to navigation_node via navigation goals, and navigation_node completes path planning and chassis control to move the robot to the front of the corresponding collection bin. - Control the robotic arm and gripper (via grasp_planner_node and mycobot_gripper_node) to take the object out of the corresponding storage slot and place it into the target collection bin.
- Update the state of the storage slot in storage_manager_node (empty it or mark it as placed).

  Repeat the above process until all objects are sorted or the task termination conditions are met. Then mission_state_machine_node switches the state to the termination state, and the system completes the current task.

5.6 Team GitHub Repository

The team's design documents and code can be found at: `https://github.com/krishnanair2025/Team9_repository.git`. This repository integrates all sub-system code and enables better collaboration between sub-teams during continued development of the rover.

6 Analysis: Sub-Team Progress and Implementation Strategy

6.1 Navigation and Mapping

Thus far, most of the work related to navigation and mapping has been carried out in simulations of the rover in Gazebo. The custom gazebo world shown in Figure 13 with a 6m x 4m rectangular boundary marking the search area and randomly located shapes marking the obstacles has been used to develop the autonomous navigation and mapping capabilities of the rover.

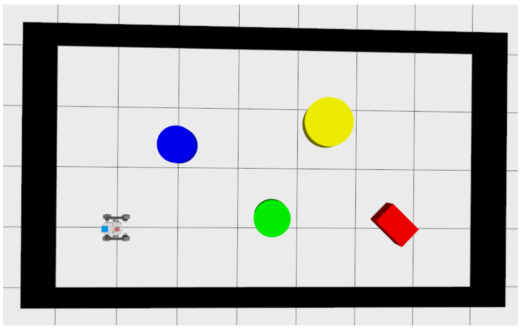


Fig. 13. Custom gazebo world to model the search environment of the robot.

Using the SLAM toolbox, the robot can map its environment using scanning data from its LiDAR and localise itself within the search area using odometry data from the rover's wheel encoders. Figure 14 shows the rover in the process of mapping its environment. This allows the robot to satisfy FR3 and the corresponding PR3.1 in simulation but further tests will be conducted with the physical rover and the specific LiDAR model which will be used on the robot system. Further fine tuning might be required in the SLAM configurations to ensure that the mapping accuracy meets PR3.1. The SLAM toolbox is also responsible for accurate localisation (FR4/PR4.1), in the Gazebo simulation, it has been proven that the robot can update its position as the TF frames move in sync with the simulated rover in Gazebo but the accuracy and magnitude of localisation drift will be measured when conducting physical tests with the LiDAR and rover.

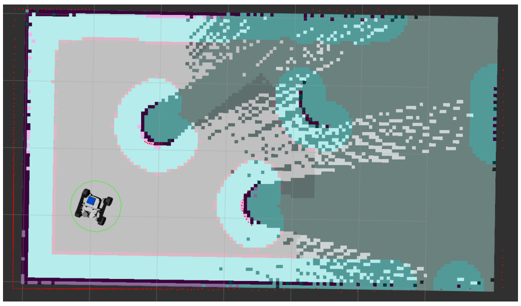


Fig. 14. Simulated robot in the process of mapping its environment in Gazebo.

The Nav2 stack is responsible for navigation, collision avoidance and speed control during the mission. Implementing this in addition to the SLAM toolbox resulted in the rover being able to plan paths to target coordinates within the search area and controlling the robot to travel to that location while avoiding obstacles and actively mapping new areas of the map. In addition to this, a simple frontier exploration node was created to send goals to the Nav2 stack, allowing the rover to explore the map without manual goal input. This resulted in the rover being able to meet PR1.1 and PR1.3 in simulation as it could navigate and map the entire 6m x 4m area without requiring any teleoperation commands. The frontier exploration node will be further developed by improving its awareness of inflation zones on the costmap when sending goals.

PR1.2 was met by setting the Nav2 controller server‘s `vx_max` parameter to be 0.25 m/s which aligns with the risk assessment’s maximum speed limit of 250 mm/s. Nav2 can avoid collision using inflation layers in cost maps and the robot’s radius. To ensure that the robot does not get closer than 50mm to any obstacle or wall, the robot’s footprint was enlarged to include this buffer zone and the inflation radius was also set strategically. The rover has dimensions of 425mm x 448mm, a 50mm clearance was added to each side resulting in the footprint of the robot being defined as a circle with a radius of 0.37m. In addition to this, the inflation zone in both local and global costmaps was set to 0.37m. This resulted in the rover maintaining a 50mm clearance from all obstacles throughout its exploration of the 6m x 4m area. This means that the robot satisfies PR2.1 in simulation.

Moving forward, the navigation and mapping sub-team plans to conduct practical tests to further fine tune SLAM and Nav2 parameters so that the performance of the physical rover fully satisfies the requirements defined under FR1- FR4.

6.2 Manipulator

The myCobot 280 PI manipulator, shown in Figure 15, comes with a complete ROS2 package of command nodes that can be used to execute common robotic manipulation operations. Each node has built-in solver algorithms making it capable of performing kinematic or dynamic tasks without having to re-program from scratch. This ensures the manipulator can be quickly deployed in any real-life applications to perform tasks such as object grasping [FR6, PR6.1, PR6.2], placing [PR6.3, FR7] and moving through trajectories [FR8]. The ROS2 package are fully integratable. Meaning it's easy to add any custom functions or nodes with additional functionalities for perform more complex or varied tasks.

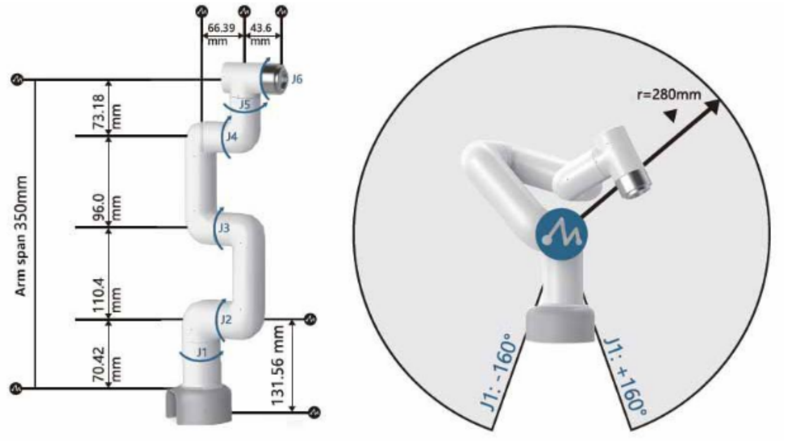


Fig. 15. The myCobot 280 Raspberry Pi manipulator.

The manipulator is equipped with 6 joint servo motors with programmable dynamic behaviours. This means we can easily control the velocity and acceleration to ensure that the manipulator carries out tasks with good stability and controllability [FR6, PR6.1, FR7]. The 6 motors create 6 DoF for the manipulator, making it capable of performing tasks that requires reaching to positions with a large variety of poses [FR8]. The servo joints are linked with a total span of 41.3 cm [11]. Each joint motor has a rotational degree of freedom of 330-358 degrees [11]. These makes the control of picking up objects highly reachable in different environments [FR8].

The on-board myCobot gripper is equipped with rubber contact with a clamp force of 150 g and a maximum clamp size of 4.5 cm [14]. This allows for successfully handling of our target selection of objects with a stable grip [PR6.1].

We will test if the API connections are working between our ros2 nodes and the servo motors by reaching our purposed 3D poses in space. We will also test and confirm if the API works in controlling the gripper from our ros2 nodes. This would ensure that [FR6, PR6.1, PR6.2, PR6.3, FR8] are satisfied.

6.3 Object Perception

The camera object detection pipeline has been prototyped and verified on real images from the lab environment in preparation for the perception requirements [FR5]. A dedicated dataset of the target objects was collected under different viewpoints, distances and lighting conditions, and manually annotated for training. Using this dataset, a YOLOv8-nano model was fine-tuned and deployed on the target hardware, achieving real-time inference on live camera streams. Initial tests show that the detector can reliably distinguish between objects of different colours and return stable 2D bounding boxes, demonstrating that the fundamental detection functionality is in place.

In the next stage, the dataset and model will be refined specifically for the three coloured cylinders defined in the requirements (red, yellow and green), so that the system can explicitly satisfy PR5.1. The 2D detections will then be combined with depth measurements and camera calibration to obtain 3D positions of each cylinder in the robot base frame, enabling the robot to provide target locations with an accuracy on the order of ±10 mm as required by PR5.2, and thereby supplying sufficiently precise pose information for the manipulator to perform reliable grasping and placement.

7 Project Plan

The Gantt chart in Figure 16 outlines the Leo Rover project's Semester 2 work plan, with relevant context from Semester 1, covering five core technical modules and key milestones as well as deliverables:

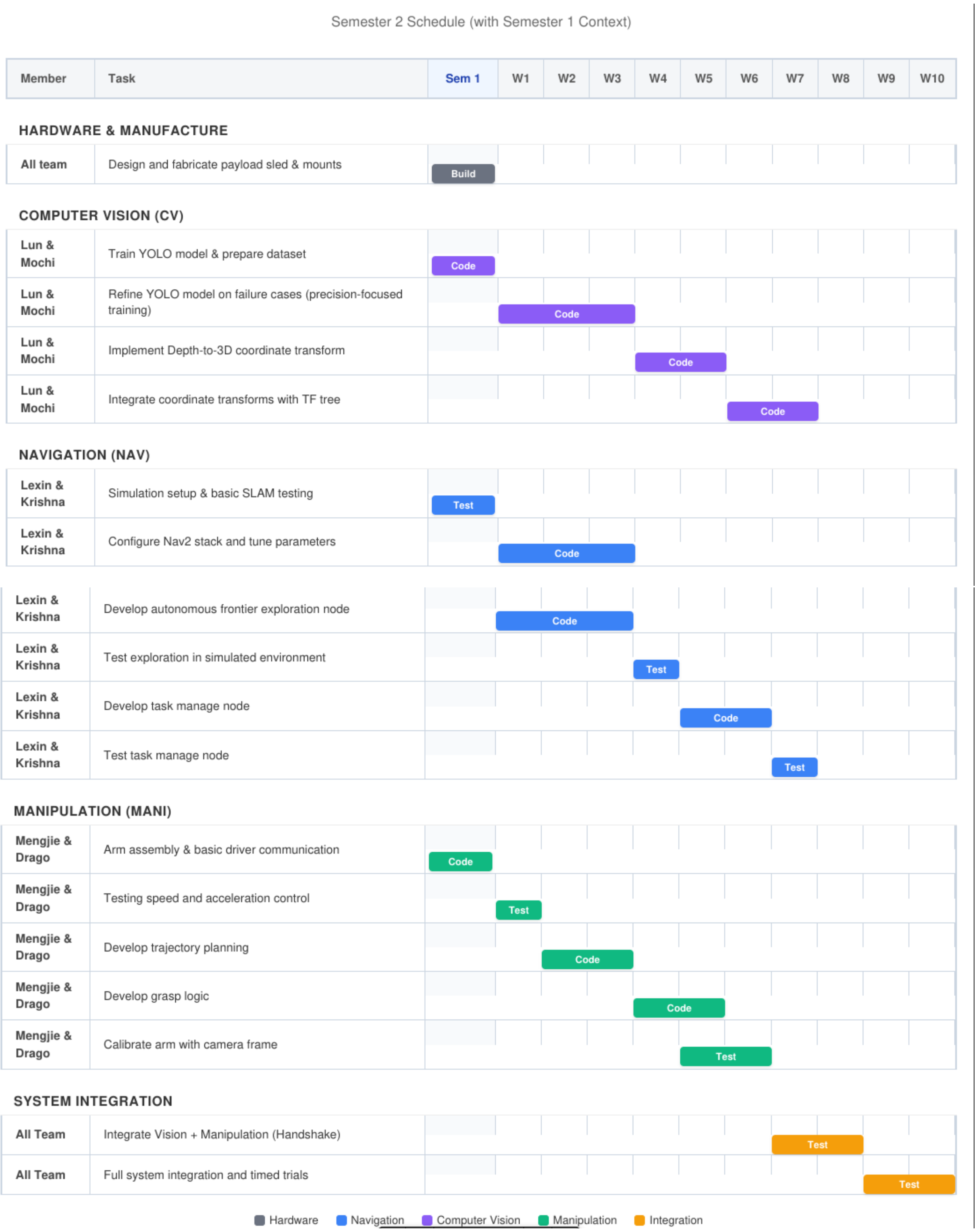
- Hardware & Manufacture: The team undertook the design and fabrication of the payload sled and mounts, a foundational hardware component that was completed in Semester 1.

- Computer Vision (CV): The team carried out YOLO model training, dataset preparation, precision refinement for failure cases, depth-to-3D coordinate transformation, and integration with the TF tree—all essential for object detection and spatial perception.

- Navigation (NAV): The team completed simulation setup, basic SLAM testing, Nav2 stack configuration and parameter tuning, developed an autonomous frontier exploration node, and designed and tested a task management node to enable autonomous navigation capabilities.

- Manipulation (MANI): The team performed mechanical arm assembly, basic driver communication, speed and acceleration control testing, trajectory planning, grasp logic development, and calibrated the arm with the camera frame, laying the groundwork for physical interaction with objects.

- System Integration: The team collaborated to develop a central Task Manager state machine, integrated the Computer Vision and Manipulation modules, and conducted full system integration and timed trials to ensure end-to-end functionality.

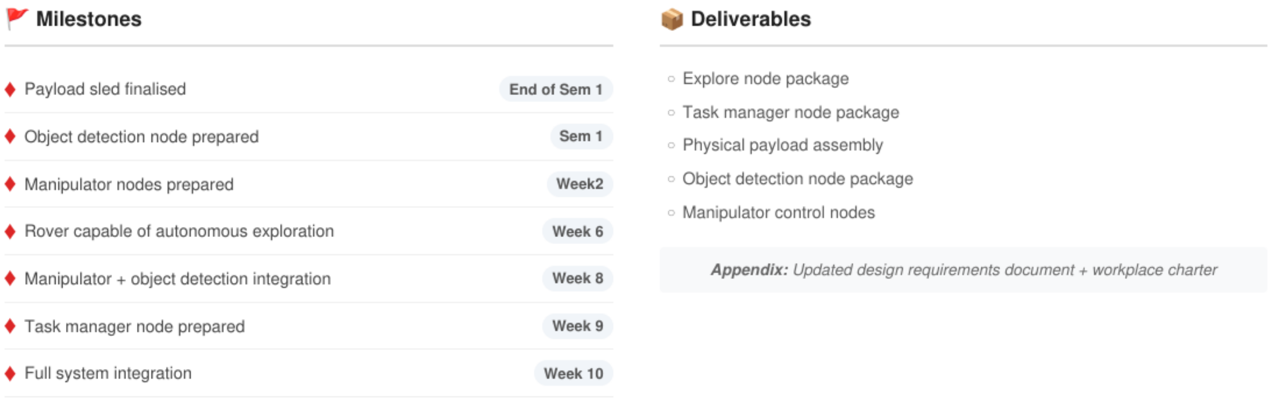


Semester 2 Schedule (with Semester 1 Context)

Member	Task	Sem 1	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
HARDWARE & MANUFACTURE												
All team	Design and fabricate payload sled & mounts	Build										
COMPUTER VISION (CV)												
Lun & Mochi	Train YOLO model & prepare dataset	Code										
Lun & Mochi	Refine YOLO model on failure cases (precision-focused training)		Code	Code	Code							
Lun & Mochi	Implement Depth-to-3D coordinate transform					Code	Code					
Lun & Mochi	Integrate coordinate transforms with TF tree							Code	Code			
NAVIGATION (NAV)												
Lexin & Krishna	Simulation setup & basic SLAM testing	Test										
Lexin & Krishna	Configure Nav2 stack and tune parameters		Code	Code	Code							
Lexin & Krishna	Develop autonomous frontier exploration node		Code	Code	Code							
Lexin & Krishna	Test exploration in simulated environment					Test						
Lexin & Krishna	Develop task manage node						Code	Code				
Lexin & Krishna	Test task manage node								Test			
MANIPULATION (MANI)												
Mengjie & Drago	Arm assembly & basic driver communication	Code										
Mengjie & Drago	Testing speed and acceleration control		Test									
Mengjie & Drago	Develop trajectory planning			Code	Code							
Mengjie & Drago	Develop grasp logic					Code	Code					
Mengjie & Drago	Calibrate arm with camera frame						Test	Test				
SYSTEM INTEGRATION												
All Team	Integrate Vision + Manipulation (Handshake)								Test	Test		
All Team	Full system integration and timed trials										Test	Test

■ Hardware ■ Navigation ■ Computer Vision ■ Manipulation ■ Integration

Fig. 16. Gantt chart with work packages.



🚩 **Milestones**

♦ Payload sled finalised	End of Sem 1
♦ Object detection node prepared	Sem 1
♦ Manipulator nodes prepared	Week2
♦ Rover capable of autonomous exploration	Week 6
♦ Manipulator + object detection integration	Week 8
♦ Task manager node prepared	Week 9
♦ Full system integration	Week 10

📦 **Deliverables**

○ Explore node package
○ Task manager node package
○ Physical payload assembly
○ Object detection node package
○ Manipulator control nodes

Appendix: *Updated design requirements document + workplace charter*

Fig. 17. Milestones and Deliverables

8 References

[1] ElephantRobotics. Adaptive Gripper. 2024. Available from: https://docs.elephantrobotics.com/docs/gitbook-en/2-serialproduct/2.7-accessories/2.7.3%20grip/2.7.3.1-ag.html

[2] ElephantRobotics. myCobot 280 Pi. 2024. Available from: https://www.elephantrobotics.com/en/mycobot-280-pi-specifications-en/

[3] SLAMTEC. RPLIDAR A2. 2017. Available from: https://cdn.sparkfun.com/assets/e/a/f/9/8/LD208_SLAMTEC_rplidar_datasheet_A2M8_v1.0_en.pdf

[4] Intel. Intel® RealSenseTM D400 Series Product Family. 2019. Available from: https://cdrdv2-public.intel.com/841984/Intel-RealSense-D400-Series-Datasheet.pdf

[5] Intel. intel nuc. 2023. Available from: https://www.intel.com/content/dam/support/us/en/documents/intel-nuc/NUC12WS-TechProdSpec.pdf

[6] Amazon. Laptop Power Bank with AC Outlet, 31200mAh 120W. 2025. Available from: https://www.amazon.co.uk/31200mAh-Portabale-External-Dual-Way-Emergency/dp/B0DBZFNC8Y

[7] fictionlab. Leo Rover Specification. 2024. Available from: https://docs.fictionlab.pl/leo-rover/documentation/specification

[8] raspberrypi.com. Raspberry Pi Computer Hardware. 2024. Available from: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

[9] mobot.pl. Gear Motor 31 x 100 / 106 / 88. 2024. Available from: https://www.mobot.pl/download/1_61_077_4xx_en.pdf

[10] ArduCam. 12.3MP 477M MINI Wide Angle Camera Module for Raspberry Pi. 2024. Available from: https://www.arducam.com/12mp-imx477m-mini-wide-angle-camera-module-for-raspberry-pi.html

[11] AlfaNetwork. AWUS036ACS. 2023. Available from: https://alfa-network.eu/media/sparsh/product_attachment/Alfa_Network_AWUS036ACS_datasheet.pdf?srsltid=AfmBOopcdPVykytJk16U6S5i5Dt1BQBBKKkfXhyWRX63XhjmhwDR8-9f

[12] Espressif Systems. ESP32 datasheet. 2016. Available from: https://cdn.sparkfun.com/datasheets/IoT/esp32_datasheet_en.pdf

[13] Amazon. DSD TECH MagicConn SH-CP12A USB Type C PD to DC Power Cable-12V. 2025. Available from: https://www.amazon.co.uk/DSD-TECH-MagicConn-SH-CP12A-Cable-12V-Black/dp/B0B9FDZX7P?th=1

9 Appendix

9.1 Appendix A: Design Requirements

9.1.1 Problem Statement

The objective of this project is to design and develop a fully autonomous rover capable of navigating a rectangular search area with randomly placed obstacles to collect 3 coloured objects and return them to matching coloured bins at the rover's starting point within a 20-minute operating time.

9.1.2 Concept of Operations (ConOps)

Figure 18 shows a diagram of the environment that the rover is required to operate in, based on the problem statement and stakeholder responses.

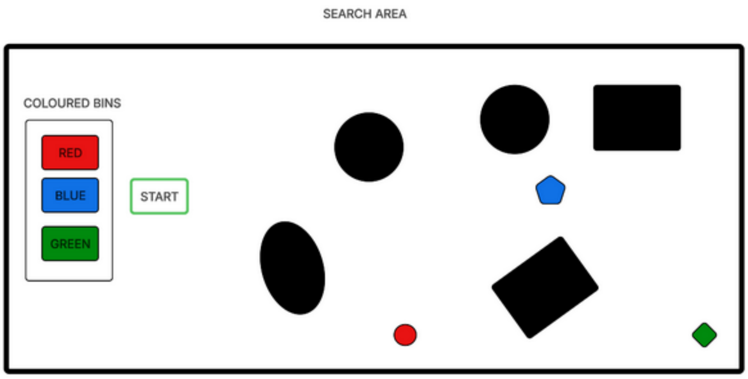


Fig. 18. Diagram of the expected environment.

There is a rectangular search area of approximately 6m x 4m marked by short walls which the robot must operate within. The black shapes represent obstacles and the coloured shapes represent the target objects, both of which are randomly placed inside the search area. On one side of the search area, there are colour-coded bins and a starting point for the robot.

After preparing the robot for the mission, it will be placed at the starting point. During the 20 minute mission time, the robot will autonomously navigate the search area while its vision system monitors the environment for coloured objects. If an object is detected, the robot will approach the detected object and the manipulator will pick and place the object into the onboard storage bin. The robot continues doing this until all three objects are collected at which point it returns to the starting point and the manipulator picks each object from the onboard storage bin and places it in a matching coloured bin. Once all three objects are sorted the mission is complete and the robot can be packed up. The flowchart in Figure 19 shows the ConOps in the form of a flowchart.

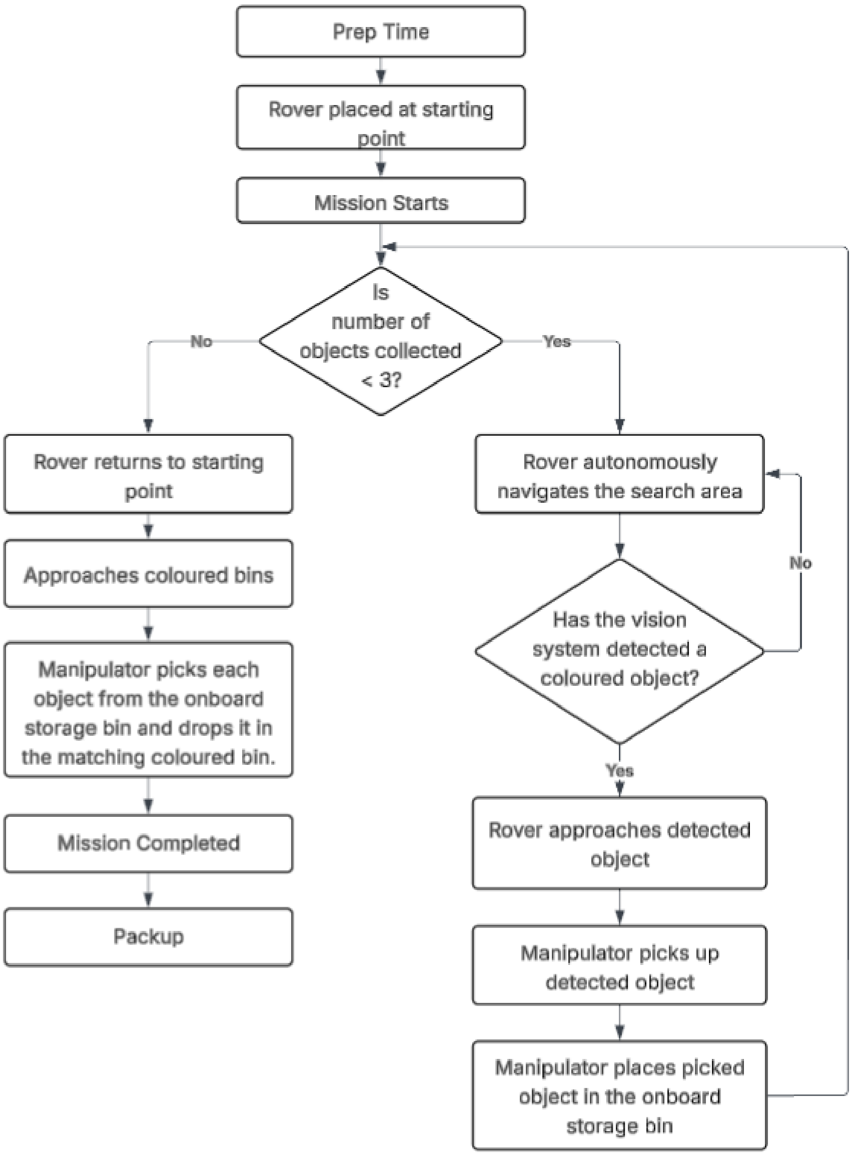

Prep Time
Rover placed at starting point
Mission Starts
Is number of objects collected < 3?
No
Yes
Rover returns to starting point
Approaches coloured bins
Manipulator picks each object from the onboard storage bin and drops it in the matching coloured bin.
Mission Completed
Packup
Rover autonomously navigates the search area
Has the vision system detected a coloured object?
No
Yes
Rover approaches detected object
Manipulator picks up detected object
Manipulator places picked object in the onboard storage bin

Fig. 19. Flowchart of the ConOps.

9.1.3 Functional and Performance Requirements

Table 2. Functional and Performance Requirements with Justifications

Req. ID	**Description**	**Justification**
FR1	The robot shall fully autonomously navigate the search area containing obstacles.	The robot must navigate the search area to find target objects.
PR1.1	The robot shall be able to navigate a 6m x 4m area.	The search area has dimensions of 6m x 4m.
PR1.2	The robot shall not exceed a maximum speed of 250 mm/s.	The risk assessment limits maximum speed to 250 mm/s.
PR1.3	The robot shall navigate the search area without requiring teleoperation or physical intervention.	No physical intervention allowed unless the robot malfunctions.
FR2	The robot shall avoid collisions with walls or obstacles.	The robot must not hit obstacles or walls.
PR2.1	The robot shall maintain a clearance of at least 50mm from walls and obstacles during the mission.	Leaving a minimum clearance of approximately 50% of the robot's width allows for safe navigation within the search area.
FR3	The robot shall detect and map obstacles within the search area.	The robot does not have a map of the environment; it is required to generate its own during the mission.
PR3.1	The map generated shall match the physical layout with a tolerance of ±5cm.	The robot must be capable of making an accurate map of its environment. 5cm is the common mapping resolution used by SLAM.
FR4	The robot shall estimate and update its position within the map during navigation.	The robot must be able to localise itself in its environment for effective navigation and mapping.
PR4.1	The estimated position of the robot shall be within ±0.05m of its actual location in the search area.	The robot must localise itself accurately to improve navigation performance. The mapping resolution is 0.05m for SLAM, so this is a reasonable tolerance.
FR5	The robot shall detect the location of target objects.	The robot should be able to detect and locate target objects in the search area.
PR5.1	The robot shall detect red, yellow and green cylinders.	The shapes placed in the search area will be red, yellow and green cylinders.

PR5.2	The robot shall provide the location of the target object with an accuracy of ±10mm.	The location of the object must be detected accurately to allow the manipulator to pick it without errors. Since the cylinder has a diameter of 26mm and the fully open position of the gripper is 37mm wide, a tolerance of 10mm still puts the cylinder inside the open gripper.
FR6	The robot shall pick up target objects using its manipulator.	The robot must be able to collect the target object from the search area.
PR6.1	The robot shall pick up cylinders of 26mm diameter and 60mm height, weighing 20g.	The manipulator must be capable of picking up three identical wooden cylinders placed in the search area.
PR6.2	The robot shall pick up 3 target objects during the mission.	The mission is only completed if all three target objects are collected.
PR6.3	The robot shall put all three cylindrical objects that it picks into its onboard storage facility.	The robot must take all three target objects to the starting point.
FR7	The robot shall transport collected objects to the starting point.	The objects must be collected and sorted into bins which are located at the starting point.
PR7.1	The robot shall securely hold 3 objects onboard.	The robot must take all three target objects to the starting point.
FR8	The robot shall sort the target objects into bins with matching colour.	The coloured target objects must be sorted correctly into matching-coloured bins.
PR8.1	The robot shall detect red, green and yellow bins.	The robot must detect red, green and yellow bins to sort the red, green and yellow target objects it collected.
FR9	The robot shall operate, untethered for at least 20 minutes.	The mission duration is 20 minutes and no wired connections to the robot are allowed.
PR9.1	The robot's onboard energy storage system shall support all sub-systems for 30 minutes of operation.	The mission duration is 20 minutes. A safety margin of 10 minutes will improve reliability.
PR9.2	The robot shall not use trailing power or data cables during the entirety of the mission.	The robot must operate untethered.

FR10	The robot shall maintain mechanical stability during navigation and manipulation tasks.	Robot must be mechanically stable to enable safe operation during the mission.
PR10.1	All the components on the robot shall securely remain in place over the duration of the mission.	Peripheral components moving and shifting out of place could negatively affect mapping, vision and manipulation functionalities. Collected objects could also be lost.
FR11	The robot shall complete the entire mission within the allocated 20-minute time limit.	The mission must be completed within 20 minutes.
FR12	The robot shall handle all computational work onboard.	The robot will be less reliant on network connections, avoiding issues with latency.

9.1.4 Requirements Verification Matrix

Table 3. Requirements Verification Matrix

Req. ID	Description	Verification Method	Verification Level	Verification Description	Success Criteria	Rationale
FR1	The robot shall fully autonomously navigate the search area containing obstacles.	Demo*	Subsystem Rover base**, Navigation	Run the robot in the search area and observe whether the rover is capable of fully autonomous navigation.	The robot navigates the entire search area independently.	R1 Navigation is one of the fundamental functions of mobile robots.
PR1.1	The robot shall be able to navigate a 6m x 4m area.	Demo	Subsystem Rover base, Navigation	Observe if the rover can completely cover the 6m x 4m area during autonomous navigation.	The robot travels and maps the entire 6m x 4m area autonomously.	R1.1 In real-life scenarios, mobile robots operate in constrained areas.
PR1.2	The robot shall not exceed a maximum speed of 250 mm/s.	Test + Inspection	Subsystem Rover base	Run the robot in the search area 10 times and observe that the velocity does not exceed maximum limit.	Recorded encoder data shows speed remains below 250 mm/s in all repetitions.	R1.2 Ensures safety and power limit compliance.
PR1.3	The robot shall navigate the search area without requiring teleoperation or physical intervention.	Demo	Subsystem Rover base, Navigation	Run the robot and observe autonomous navigation without physical intervention.	Robot navigates independently without teleoperation.	R1.3 Ensures autonomy as required in industrial applications.

FR2	The robot shall avoid collisions with walls or obstacles.	Demo	Subsystem Rover, Navigation	Operate the robot and verify no collisions occur.	Robot never collides with walls or obstacles.	R2 Obstacle avoidance is required to achieve [FR1, R1].
PR2.1	The robot shall maintain a clearance of at least 50mm from walls and obstacles during the mission.	Demo	Subsystem Navigation, Camera	Mark 50mm clearance and run the robot 10 times to check it never crosses lines.	Rover never crosses tape marking 50mm clearance.	R2.1 Ensures [FR2, R2] are satisfied.
FR3	The robot shall detect and map obstacles within the search area.	Inspection	Subsystem Rover base, Navigation	Run robot and visually confirm live generated map matches surroundings.	Robot correctly maps environment and detects obstacles.	R3 Environment mapping is essential in [FR1, R1].
PR3.1	The map generated shall match the physical layout with a tolerance of ±5cm.	Analysis	Subsystem Rover base, Navigation	Measure dimensions and verify map in Rviz matches within tolerance.	Map accuracy ±5cm in 10/10 trials.	R3.1 Ensures [FR3, R3] are satisfied.

FR4	The robot shall estimate and update its position within the map during navigation.	Inspection	Subsystem Rover base, Navigation	Observe that robot location matches Rviz visualization.	TF in Rviz matches real robot movement.	R4 Self-localization is a key SLAM task.
PR4.1	The estimated position of the robot shall be within ±0.05m of its actual location.	Test	Subsystem Rover base, Navigation	Pause exploration every 30s, record position, and compare to actual coordinates.	Estimated position does not drift beyond ±0.05m in all 3 repetitions.	R4.1 Accurate tracking ensures correct navigation.
FR5	The robot shall detect the location of target objects.	Test	Subsystem Rover base, Computer vision, Depth camera	Place cylinders at known positions and verify camera detection 10 times per color.	Camera detects all cylinders correctly each trial.	R5 Ensures objects can be detected for manipulation.
PR5.1	The robot shall detect red, yellow and green cylinders.	Demo	Subsystem Rover base, Computer vision, Depth camera	Place cylinders in 10 positions each and verify detection.	Camera detects all cylinders 10/10 times.	R5.1 Ensures [FR5, R5] are satisfied.
PR5.2	The robot shall provide the location of the target object with ±10mm accuracy.	Test + Inspection	Subsystem Rover base, Computer vision, Depth camera	Place cylinder at known (x,y,z) and verify coordinates 10 times.	Camera outputs coordinates within ±10mm in all 10 repetitions.	R5.2 Provides accurate spatial coordinates for manipulator.

FR6	The robot shall pick up target objects using its manipulator.	Demo + Inspection	Subsystem Rover base, Manipulator	Run pick-up process and verify successful grasp.	Manipulator picks up object without dropping or knocking it.	R5.3 Ensures [FR5, R5] are satisfied.
PR6.1	The robot shall pick up cylinders of 26mm diameter and 60mm height, weighing 20g.	Demo + Inspection	Subsystem Rover base, Manipulator	Place object in front of manipulator and verify pick-up 10 times.	Object remains secure in gripper without being knocked over.	R6.1 Ensures successful grasping.
PR6.2	The robot shall pick up 3 target objects during the mission.	Demo	Subsystem Rover base, Task manager, Manipulator	Run mission with 3 randomly placed objects and verify collection.	All 3 objects collected correctly in all trials.	R6.2 Ensures [FR5, R5] are satisfied.
PR6.3	The robot shall put all three cylindrical objects into onboard storage.	Demo	Subsystem Rover base, Manipulator	Place cylinders and verify correct placement 10 times.	All cylinders stored correctly in each trial.	R6.3 Ensures [FR5, R5] are satisfied.
FR7	The robot shall transport collected objects to the starting point.	Demo	Subsystem Rover base, Manipulator, Navigation, Depth camera	Run mission and verify objects reach starting point without dropping.	Rover collects and returns all objects successfully.	R7 Ensures [FR5, R5] are satisfied.
PR7.1	The robot shall securely hold 3 objects onboard.	Demo + Inspection	Subsystem Rover base, Mechanical chassis	Observe objects stay secured in storage during mission.	Objects never fall from storage.	R7.1 Ensures successful grasping.

FR8	The robot shall sort target objects into bins with matching color.	Demo	Subsystem Rover base, Manipulator, Depth camera	Run sorting operation and verify correct bin placement.	All objects placed in correct bins.	R8 Ensures [FR5, R5] are satisfied.
PR8.1	The robot shall detect red, green and yellow bins.	Inspection	Subsystem Rover base, Depth camera	Place bins and verify detection in robot interface.	All text output matches object class.	R8.1 Ensures [FR5, R5] are satisfied.
FR9	The robot shall operate untethered for at least 20 minutes.	Demo	Subsystem Rover base, Power system	Run full mission and verify >20 min operation.	All components remain functional.	R9 Ensures [PR1.3, R1.3] and power limits are satisfied.
PR9.1	The robot's onboard energy storage shall support all subsystems for 30 min.	Demo	Whole system	Fully charge and run mission; verify subsystems powered for 30 min.	All subsystems remain operational.	R9.1 Confirms subsystems supported with safety margin.
PR9.2	The robot shall not use trailing cables.	Demo	Whole system	Run mission untethered.	Robot completes mission without cables.	R9.2 Confirms FR9 satisfied.
FR10	The robot shall maintain mechanical stability.	Demo + Inspection	Whole system	Run full mission and verify no structural issues.	No structural issues detected.	R10 Ensures stability for safe operation.
PR10.1	All components shall securely remain in place.	Demo	Whole system	Run full mission and verify no shifting.	No shifting of components observed.	R10.1 Ensures task completion.

FR11	The robot shall complete mission within 20 minutes.	Test	Whole system	Run mission 10 times and record completion time.	Robot completes mission within 20 minutes in all trials.	R11 Confirms time requirement.
FR12	The robot shall handle all computational work onboard.	Demo	Whole system	Run mission and observe onboard computation.	All computation handled onboard without external support.	R12 Confirms onboard computation.

*Demonstration

**Rover base – the subsystem that consists of only the main body, the motors, the battery, the intel NUC and the lidar. Other components such as the manipulator are not required for verification.

9.2 Appendix B: Workplace Charter

9.2.1 Purpose

This document is used to define the general style of our teamwork environment and act as a common standard for each team member's behaviours as well as any issues that may arise during or outside the working schedule.

Each team member should read and agree to the rules and guidelines outlines in the document. Regular checks shall be carried out to ensure the solid execution of this charter. Each team member should also refer to this document first should any concerns occurs.

The document will be updated constantly as the project progresses. This is to ensure all newly occurred constraints or events will be considered to prevent mis-communication.

9.2.2 Principles and Commitments

The workspace should be a motivating, resilient, inclusive and professional environment. This will maximize working efficiency and more importantly the trust and team working spirits between the team members.

Each team member shall commit to:

- Being respectful to everyone in the team.
- Fostering an inclusive workspace.
- Taking ownership of their tasks and meeting deadlines.
- Communicating proactively and asking for help as early as possible.
- Sharing knowledge and ideas to help the team develop as a whole.
- Listening to everyone’s ideas .
- Upholding ethical standards and honesty.
- Being open to learning and growth.
- Addressing conflicts constructively.
- Escalating issues through appropriate channels.
- Behaving in a professional manner at all times.

9.2.3 Team Rules

Working hours and Availability:

- Each team member is expected to be present at the labs on Mondays and Wednesdays.
- All deadlines must be respected and delays must be communicated in advance.
- Members shall coordinate meetings based on the availability of the group.

Communication:

- All team communications will primarily be in English.
- Team members shall support each other and share knowldege wherever possible.
- Team members shall avoid unconstructive criticism.
- Team members shall be willing to compromise and problem-solve during discussions.

Safety:

- Team members shall stick to the guidelines stated in the lab risk assessment documents.
- Team members shall ensure their safety and each other's safety at all times.
- Team members shall report any incidents, near misses or unsafe equipment immediately.

Collaboration:

- Members shall take ownership of assigned tasks and deliver on time.
- Team members shall support each other and share knowldege wherever possible.
- Team members shall respond to work-related messages within 24-48 hours.
- All team members shall be encouraged to participate in meetings.

9.2.4 Daily Activity Conflict Avoidance

To avoid escalating situations and getting to the grievance procedure, the team members shall:

- Communicate clearly, honestly and professionally at all times.
- Raise concerns as soon as possible.
- Address any signs of tension within the team as soon as possible.
- Listen to everyone's perspective during group discussions.
- Provide constructive criticism.
- Respectfully escalate issues to module leader when issues cannot be resolved within the team.

9.2.5 Conflict Resolution Guidelines

Definitions:
Minor: Disagreements and tensions between team members.

Examples: accidental verbal disrespect, slightly overdue of individual workload (1-3 days), arguments over best idea

Moderate: Persisting tensions/disagreements, repeated minor conflicts, hindering of team performance.

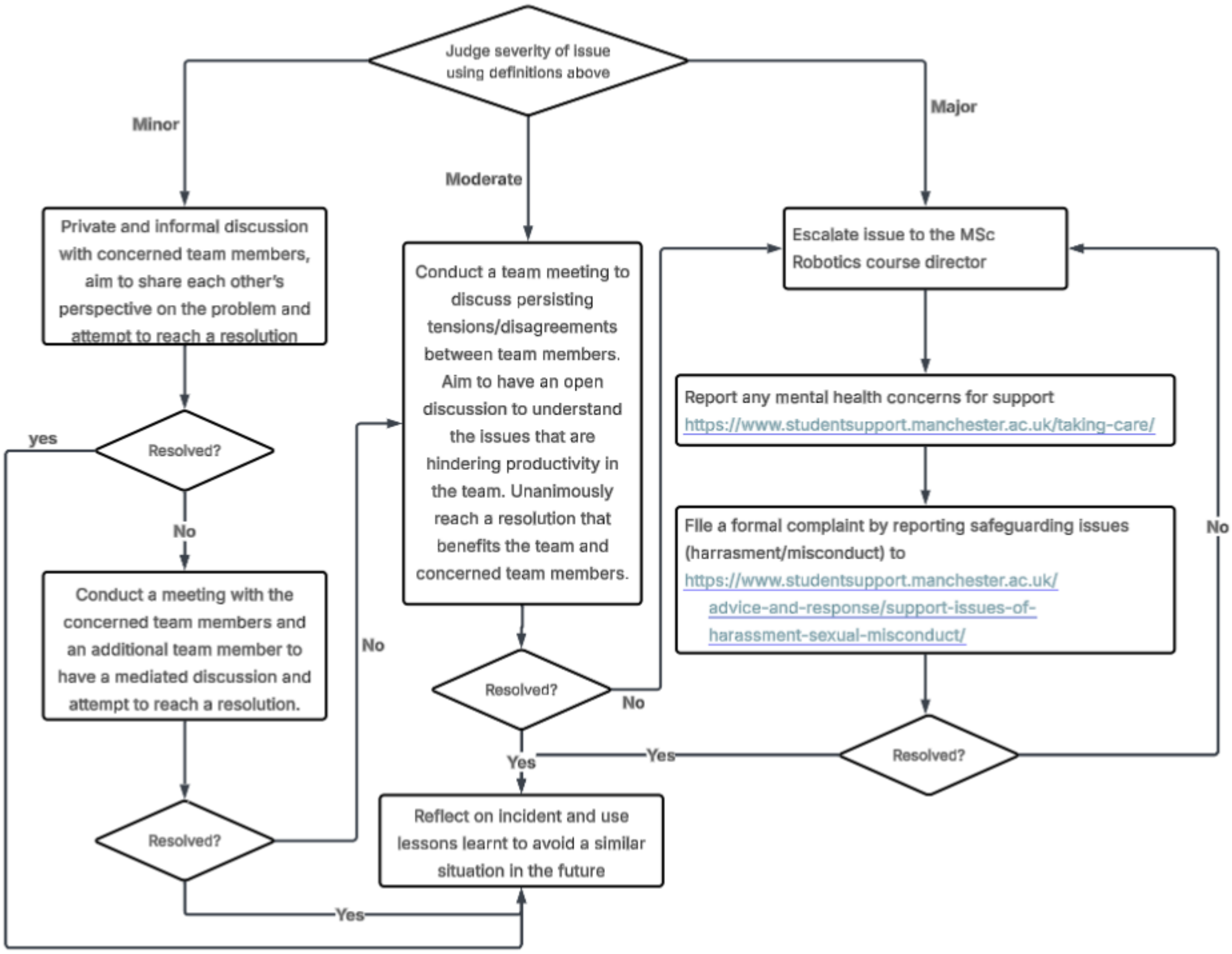
Examples: deliberate/repeated verbal disrespect, longer overdue of individual workload (4-7 days), slight physical injury due to mistake

Major: Behaviour that violates the University's policies or puts wellbeing of team members at risk. Repeated moderate issues that are hindering team performance.

Examples: severe verbal insult/slander, long overdue of individual workload (> 7 days), deliberate physical injury

9.2.6 Conflict Resolution Flowchart

Note: The procedure shown in Figure 20 shall be followed whenever any of the above conflicts arises.



Judge severity of issue using definitions above
Minor
Moderate
Major
Private and informal discussion with concerned team members, aim to share each other's perspective on the problem and attempt to reach a resolution
Resolved?
yes
No
Conduct a meeting with the concerned team members and an additional team member to have a mediated discussion and attempt to reach a resolution.
Resolved?
Yes
No
Conduct a team meeting to discuss persisting tensions/disagreements between team members. Aim to have an open discussion to understand the issues that are hindering productivity in the team. Unanimously reach a resolution that benefits the team and concerned team members.
Resolved?
No
Yes
Reflect on incident and use lessons learnt to avoid a similar situation in the future
Escalate issue to the MSc Robotics course director
Report any mental health concerns for support https://www.studentsupport.manchester.ac.uk/taking-care/
File a formal complaint by reporting safeguarding issues (harrasment/misconduct) to https://www.studentsupport.manchester.ac.uk/advice-and-response/support-issues-of-harassment-sexual-misconduct/
Resolved?
Yes
No

Fig. 20. Conflict Resolution Flowchart