Preliminary Design Review

MSc Robotics AERO62520 Robotic Systems Design Project

Year of submission
2024

Group Number
Group 5

School of Engineering

Contents

1 Introduction

This document outlines the preliminary design of a robotic system with the objective to□

> **“Develop a robot which can autonomously retrieve coloured objects from the environment and place them in matching storage bins located at the starting point.”**

Specifically, the project focuses on the design and development of a fully autonomous robot capable of navigating an unmapped environment, retrieving specific objects, and accurately placing them in designated storage bins. The robot will be equipped with advanced sensors and a manipulator to perform the task efficiently and safely within a specified time limit. Key objectives include the implementation of autonomous navigation capabilities, the application of computer vision for object detection and classification, and the establishment of reliable collision avoidance mechanisms for interaction with stationary obstacles. Additionally, the system will incorporate safety features, such as an emergency stop function, to ensure safe operation.

This project provides an opportunity to apply robotics knowledge to a practical context, enhancing both technical expertise and teamwork capabilities.

2 System

2.1 Components

- **Battery**: Provides the primary energy source required for the entire robotic system to operate autonomously, ensuring uninterrupted functionality during tasks.
- **Power Distribution System (PDS)**: Regulates and distributes power from the battery to various subsystems with precision, ensuring each component receives the necessary voltage and current for stable operation.
- **LEO Core Controller**: Serves as the central hub for motor control and sensor integration, enabling real-time feedback from the motors and encoders for precise movement and navigation.
- **DC Motors with Encoders**: Propel the robot’s wheels while providing positional feedback, allowing accurate control of speed and direction.
- **Raspberry Pi 4B**: Functions as a key processing unit for sensor data acquisition and communication, bridging data exchange with the Intel NUC and managing wheel velocity inputs from the LEO Core.
- **Integrated IMU**: Provides real-time information on the robot’s orientation, acceleration, and angular velocity, supporting motion control and navigation accuracy.
- **RGB Camera**: Captures high-resolution visual data for object recognition and scene analysis, contributing to navigation and perception capabilities.

- **WiFi Adapter with External Antenna**: Enables wireless connectivity for the robot, facilitating data transmission and remote control during testing and operation.
- **Intel NUC**: Acts as a high-performance computing platform for intensive tasks such as SLAM (Simultaneous Localization and Mapping) and advanced navigation algorithms.
- **RPLIDAR A2M12**: Performs 360-degree environmental scanning to generate precise 2D maps of the surroundings, essential for path planning and obstacle avoidance.
- **RPLIDAR USB Adapter**: Provides a seamless interface between the RPLIDAR and the Intel NUC, ensuring efficient data transfer and compatibility.
- **Intel RealSense Depth Camera**: Captures detailed depth and 3D data of the environment, enhancing the robot's ability to perform object detection and spatial analysis.
- **PincherX 150 Manipulator**: A robotic arm mounted on the LoeRover, capable of executing manipulation tasks such as grabbing and moving objects when detected.

2.2 System block-diagram

The system block-diagram below illustrates the interconnection of all the components in the robotic system. Each component is organized into specific categories: computing, sensors, peripherals and power. The color-coded legend in the top-left corner provides a visual guide to distinguish these categories and identify the functions of each component.

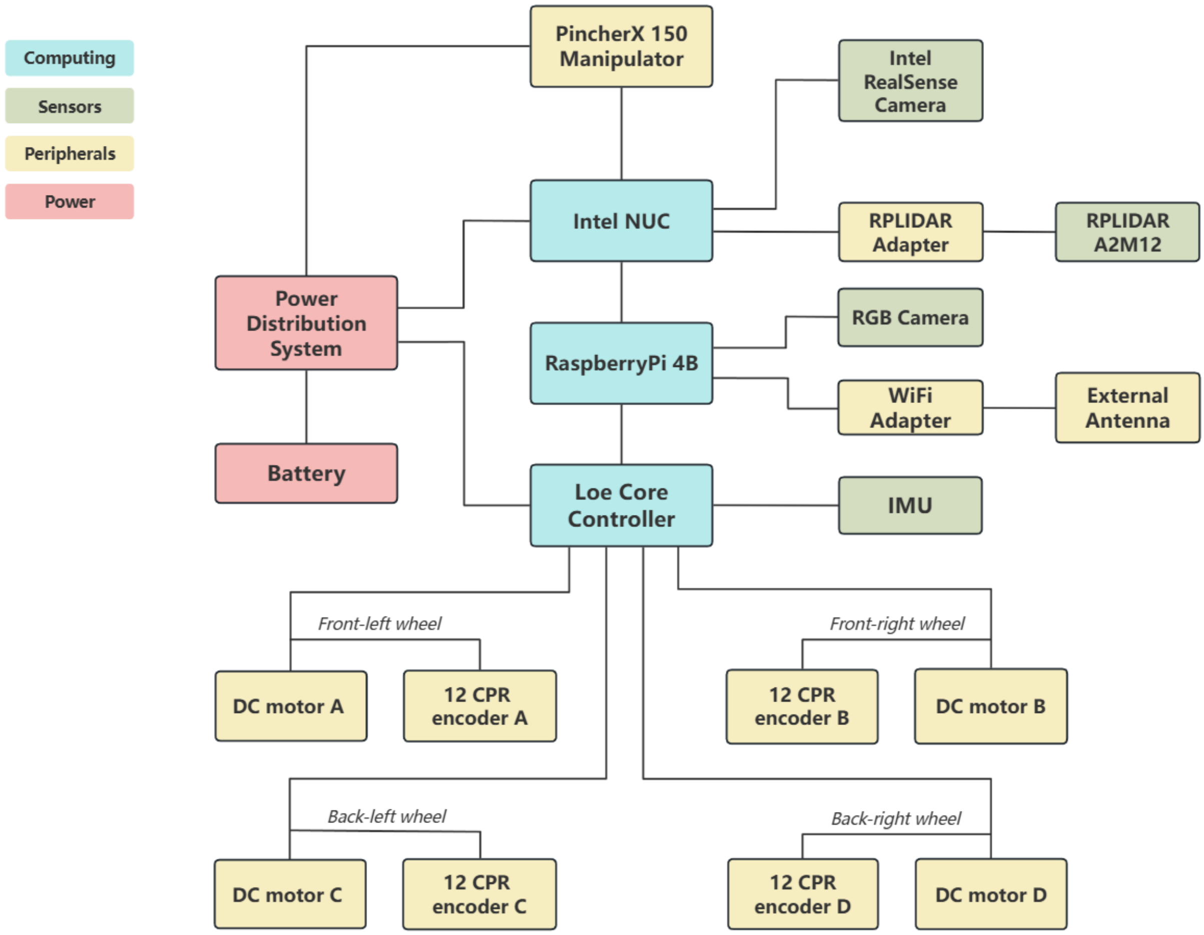


Fig. 1. System block-diagram

3 Mechanical Design

3.1 CAD model

Figure 3.4 illustrate 3D models of the LeoRover robot equipped with a RealSense depth camera, LiDAR, NUC and a manipulator. After carefully analyzing the design requirements and ensuring that all sensors can operate without obstruction, we developed a two-platform design:

1. Base plate: Serves as the mounting area for the LiDAR and NUC.

2. Top Plate: The robotic arm, depth camera, and a storage bin are accommodated to hold objects during operation.

The two plates are connected using six I beams. All components are securely fixed with screws to maintain the alignment and stability of the plates. To optimize mapping and object detection, the depth camera and LiDAR are positioned at the front of the robot. Adequate clearance is provided for all components to facilitate proper wiring and operation.

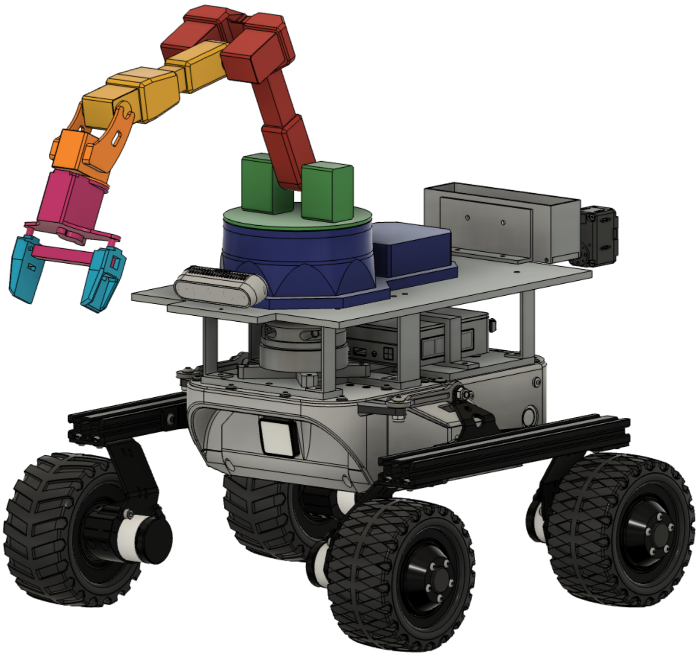


Fig. 2. Isometric View CAD

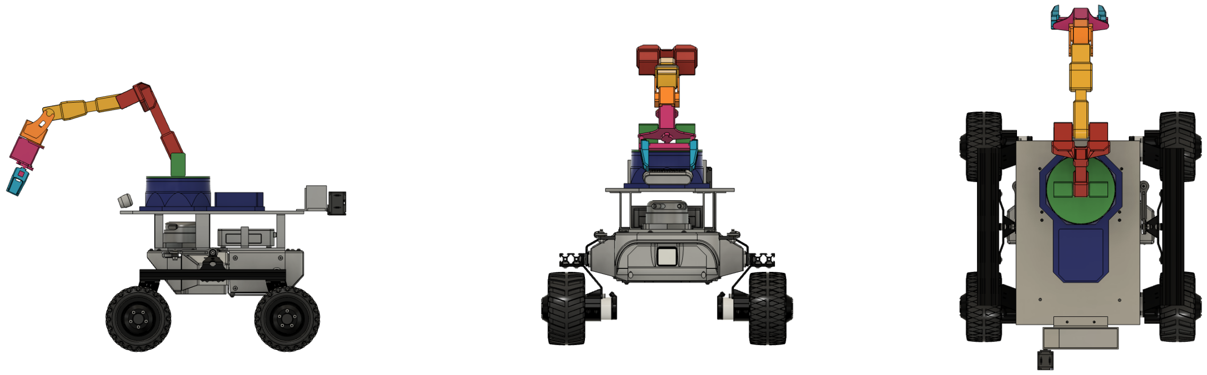


Fig. 3. Three views of CAD model

3.2 Payload Sled Design

The primary challenge was broken down into three key tasks: mapping and navigation, object detection, and manipulation and storage of objects. A critical point of our design approach was to eliminate potential sensor interference, ensuring unobstructed functionality and precise data collection. In order to do this, the lidar and depth camera are located at the very front of the payload where interference from other components can be minimized.

The RPLidar A2M12 is a high-performance 360-degree 2D laser scanner, ideally suited for obstacle detection. Its performance improves when it is positioned closer to the ground, as it operates more effectively at lower levels. Therefore, the LiDAR was strategically mounted on the base plate, bringing it closer to the surface for enhanced detection capabilities. Additionally, positioning it at the front of the robot allows for rapid response during navigation, especially as the robot primarily moves forward, ensuring efficient and timely obstacle avoidance.

The depth camera requires an unobstructed line of sight to accurately identify objects. Mounting it on the top plate not only prevents interference with the LiDAR but also provides a broader field of view. In addition, reducing the distance between the camera and the manipulator minimizes the coordinate transformation, resulting in improved accuracy and reduced error.

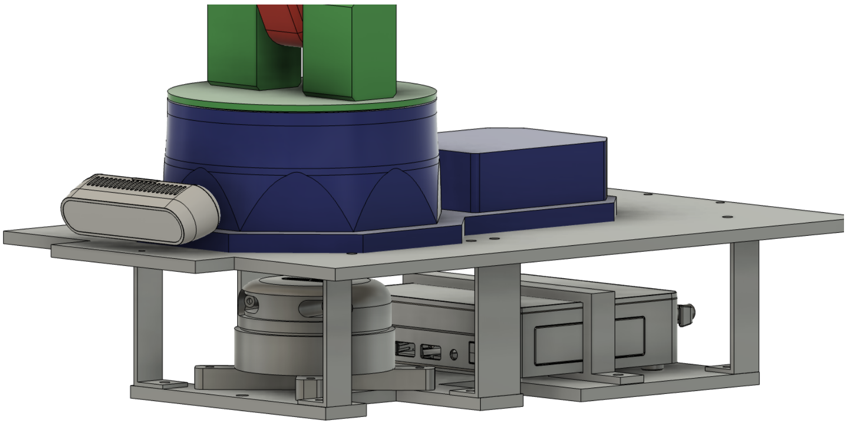


Fig. 4. Payload Sled

3.3 Strength

The components are connected using I-shaped beams, chosen for their structural advantages. The I beam design provides high bending stiffness in the vertical direction, making it ideal for supporting vertical loads. While plastic or acrylic I-beams are less commonly used compared to steel ones, the simulation results shown below demonstrate their suitability for this application.

Since the base plate is fixed to the LeoRover robot, the simulation focuses primarily on the I-beams, which bear the majority of the manipulator's weight. The depth camera, weighing only 72g compared to the manipulator's 770g, exerts negligible force and is therefore excluded from consideration. The simulation results indicate a maximum stress of approximately 0.108 MPa, well below the material's yield strength, ensuring that no strain under these conditions.

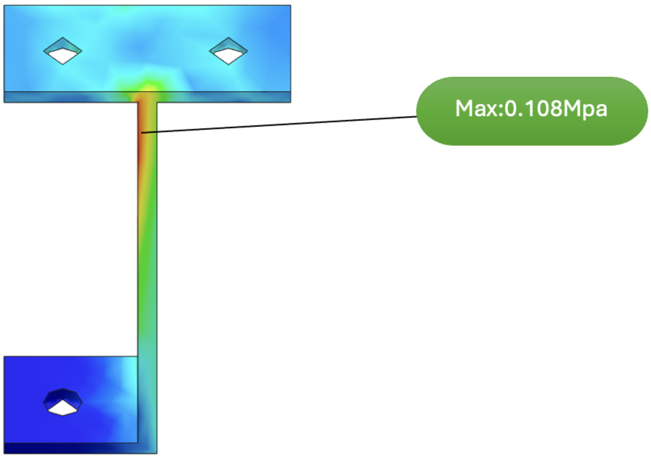


Fig. 5. Payload Sled

3.4 Weight Restriction

The LeoRover has a maximum payload capacity of 5 kg. According to Table 1, the total weight of the components is approximately 3.0 kg. Therefore, the combined weight of the payload sled and the objects temporarily stored in the robot must not exceed 2.0 kg. The weight of the payload is adjustable by modifying the size of the plate and selecting lighter material as needed.

Component	Weight (kg)
Manipulator	0.7
Lidar	0.2
NUC	0.4
Depth camera	0.1
LeoRover	1.6
Total	3.0

Table 1. Component Weight

3.5 Storage Container Design

A simple 140×40×50 mm storage unit is designd to retrieve objects. Upon returning to the destination, a motor tilts the base of the unit around a pivot point, allowing the objects to be dumped into the final bin.

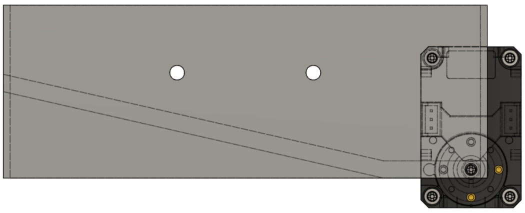
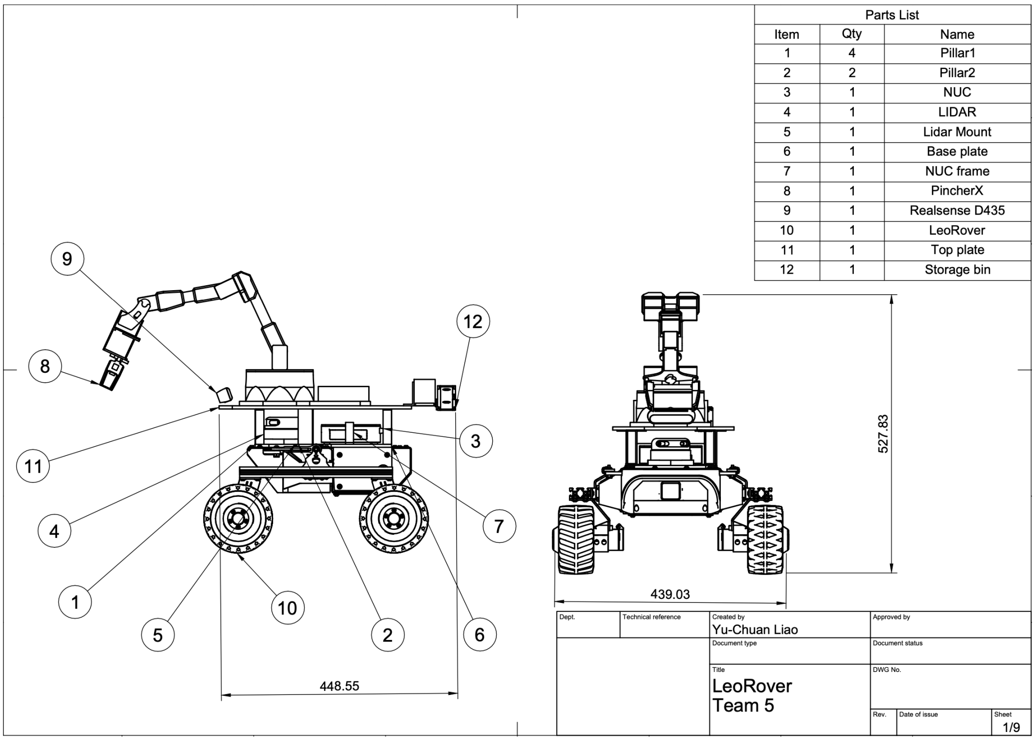


Fig. 6. Storage Container

3.6 Design Files

3.6.1 Overall

The overall design points out all components and the size of the complete robot.



Parts List		
Item	Qty	Name
1	4	Pillar1
2	2	Pillar2
3	1	NUC
4	1	LIDAR
5	1	Lidar Mount
6	1	Base plate
7	1	NUC frame
8	1	PincherX
9	1	Realsense D435
10	1	LeoRover
11	1	Top plate
12	1	Storage bin

Dept.	Technical reference	Created by Yu-Chuan Liao	Approved by
		Document type	Document status
		Title LeoRover Team 5	DWG No.
			Rev. / Date of issue / Sheet 1/9

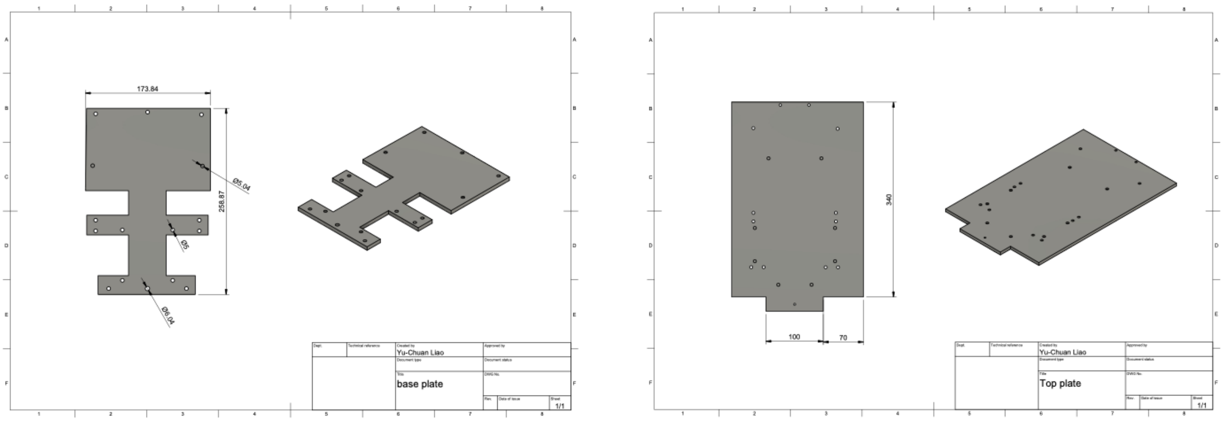
Fig. 7. Overall Drawing

3.6.2 Base plate and Top plate

The base plate and top plate drawing includes the dimensions and area, as well as the sizes of the screws.

3.6.3 Pillar

The drawing indicate the width and length of both type of pillars which are used to connect two plates and the size of screws.



(a) Base Plate Drawing

(b) Top Plate Drawing

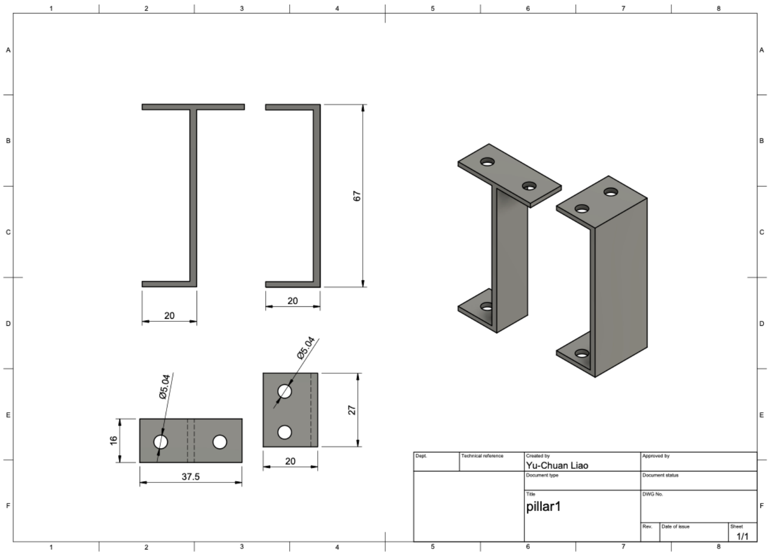


Fig. 9. Pillar Drawing

3.6.4 Lidar Module

The LiDAR is securely mounted on the base plate using an X-shaped bracket with two sets of screws. One set is used to attach the LiDAR to the bracket, while the other set fastens the bracket to the base plate.

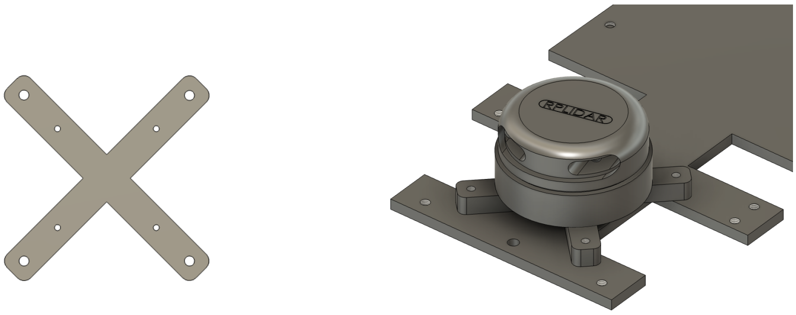


Fig. 10. Lidar Module

3.6.5 NUC Module

The frame is design to securely fix the NUC on the base plate through the force of the screw.

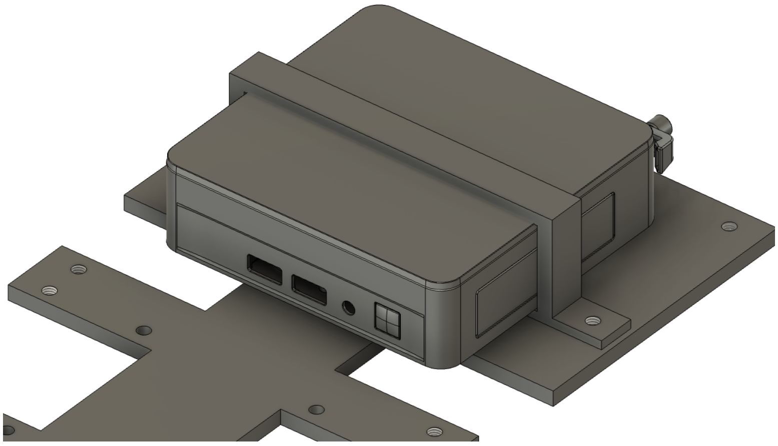


Fig. 11. NUC module

3.6.6 Depth Camera Module

The depth camera is mounted on the front side of the top plate and secured using a component featuring a spherical core structure. This spherical core allows free rotation in all directions, enabling flexible and detailed angle adjustments.

3.7 Manufacturability

Manufacturability is a primary goal of the design. Both laser cutting and 3D printing are excellent options for the payload sled design. Laser cutting offers higher precision, consistency, and faster production. On the other hand, 3D printing takes more time and provides lower precision but is ideal for creating custom prototypes. In conclusion, both methods are suitable for the current requirements, and the final decision may depend on the material chosen after confirmation.

The design is suitable for the goals of mapping and navigation, object detection, and manipulation. As for objective dropping to the final bin, it should control the position it drop down to a 20 centermeters cube with 15 centermeters circular area at the center of the cube.

4 Electrical Design

The power connection diagram illustrates the distribution of electrical power from the battery through the Power Distribution System (PDS) to the various components of the robot. The key details of the power connections are as follows:

- **Battery:** The core energy source of the LeoRover is an 11.1V, 5800mAh Li-ion battery capable of delivering up to 8A of current.
- **Power Distribution System (PDS):** Regulates the 12V input from the battery and distributes power to connected components, including the Intel NUC (12V, 3A), Loe Core Controller (12V, 5A), and PincherX 150 Manipulator (12V, 5A).

- **Loe Core Controller:** Powered by the Power Distribution System (PDS) with 12V, 5A through its dedicated PWR port. It regulates power to the components, converting 12V to 5V for the Raspberry Pi, 3.3V for the IMU, and 12V for the motors.

- **Raspberry Pi 4B:** Powered via the Loe Core Controller with 5V, 3A through a UART connection, it serves as a central hub for peripherals such as the WiFi Adapter and RGB Camera.

- **PincherX 150 Manipulator:** Receives 12V, 5A directly from the PDS for robotic arm operations, including object manipulation tasks.

- **Intel RealSense Camera:** Receives 5V, 0.7A through a USB connection from the Intel NUC, enabling 3D depth perception and object detection.

- **RPLIDAR A2M12:** Powered via the RPLIDAR Adapter, which requires 5V, 0.6A through the NUC.

- **RGB Camera:** Powered by the Raspberry Pi through the CSI interface, it operates on 3.3V, 0.3A, capturing visual data for navigation and object detection.

- **WiFi Adapter with External Antenna:** Powered by the Raspberry Pi with 5V, 1.2A through a USB connection, it facilitates wireless communication for remote monitoring and control.

- **IMU:** It draws 3.3V, 6mA from the Loe Core Controller via UART, delivering real-time orientation and acceleration data.

- **DC Motors with Encoders:** Four motors, each powered by 12V, 0.55A from the Loe Core Controller, drive the robot's movement and ensure precise control through positional feedback.

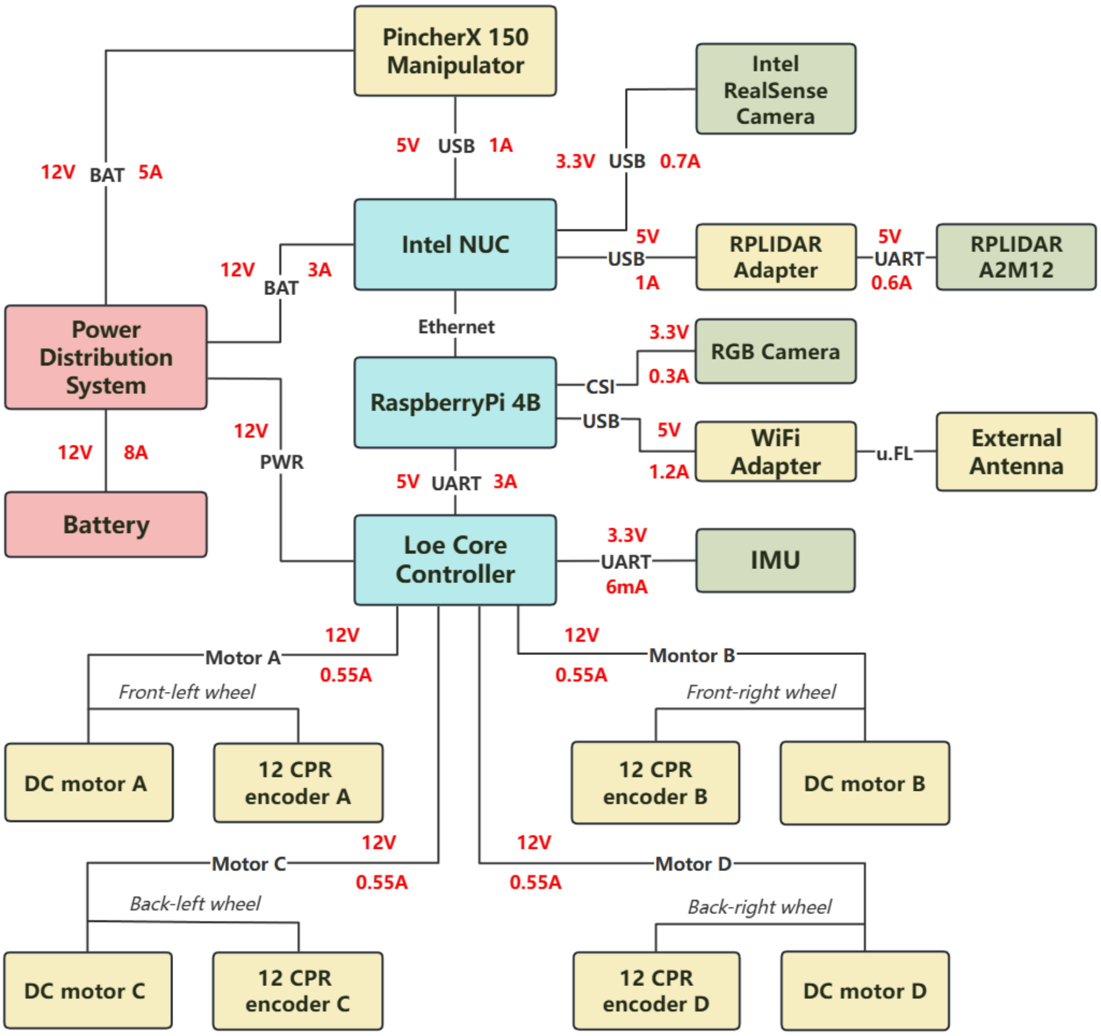


Fig. 12. Power connection diagram

4.1 Power Budget

The power budget analysis confirms that the robotic system has sufficient power to operate all components effectively. The system is powered by a 12V, 8A battery with a total power capacity of 96W, distributed through the Power Distribution System (PDS). The key power consumers include the Intel NUC, Loe Core Controller, and PincherX 150 Manipulator, along with other peripherals such as the Raspberry Pi, motors, and sensors.

By efficiently distributing power, the system meets the demands of all components while staying within the battery's current limit of 8A. Typical operating conditions are optimized to ensure no component exceeds its allocated power, maintaining stable and reliable performance for the entire system. This confirms that the system has sufficient power to run all components effectively.

5 Software Design

5.1 Software Function Overview

Based on the analysis of Design Requirements Analysis, we give the core functional modules of the robot's software as follows. The RQT graph and the modular design sections are structured accordingly, consisting of six modules. Perception Module, Navigation Module, Grasping Module, User Interface Module, Communication Module, and System Management Module, primarily responsible for implementing the robot's environmental perception functionality, path planning and obstacle avoidance functionality, object grasping and placement functionality, user interaction and visualization functionality, data communication functionality, and system state management functionality.

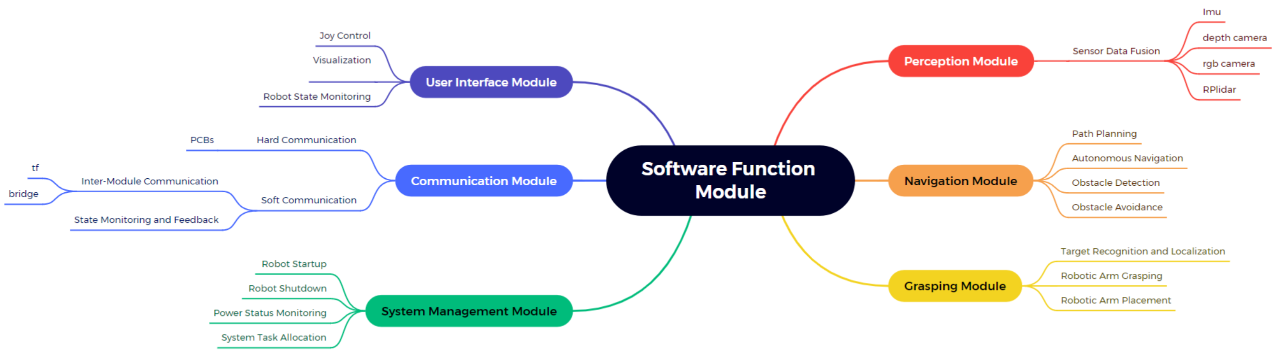


Fig. 13. Overview Diagram of Software Functional Modules

5.2 RQT graph

Consistent with the system diagram presented in Chapter 2, we have initially developed a comprehensive RQT (Robotics Query Tool) node graph (illustrated in Figure 14). The graph employs distinct colors corresponding to the modules outlined in Section 5.1, offering a clear and intuitive depiction of the implementation flow for various nodes within each module, as well as the associated topic communication states. It is important to note that this is a preliminary design, and certain details remain to be refined. Subsequent work will prioritize further elaboration and enhancement of the RQT, with a particular focus on the Navigation Module and Grasping Module.

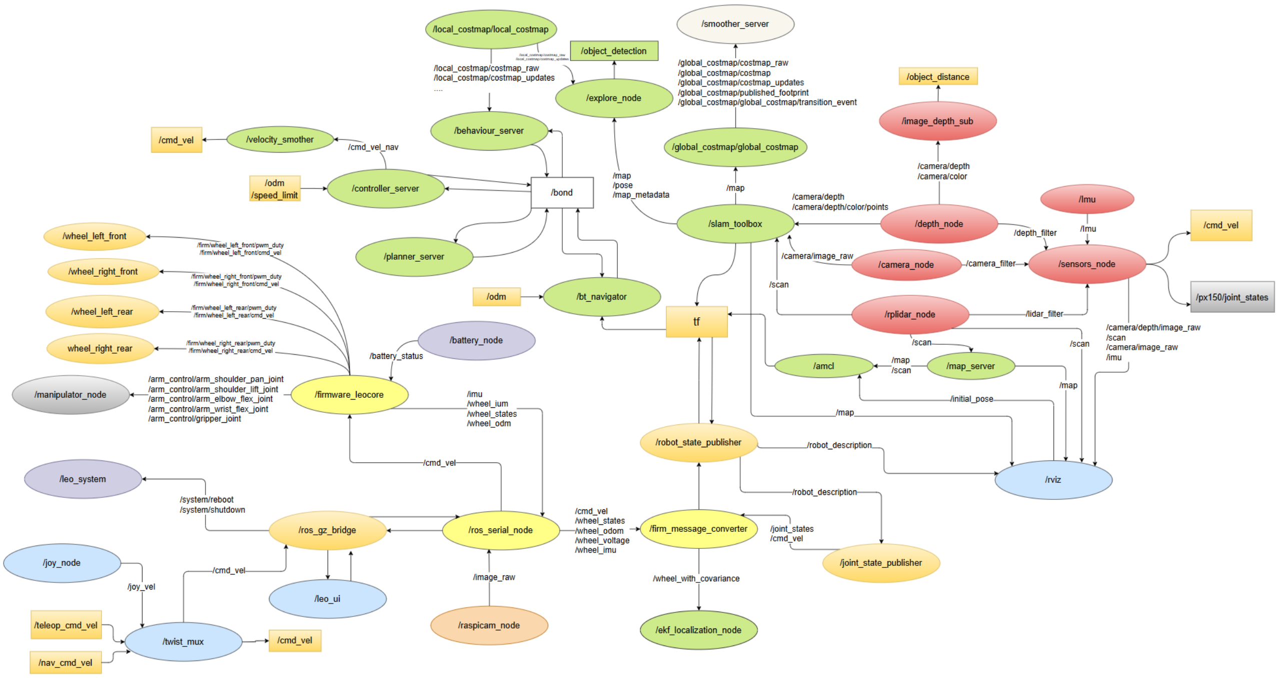


Fig. 14. RQT Graph Corresponding to System Block Diagram

Color	Module	Description
🟡	Communication Module	Direct interaction with hardware via PCBs, converting hardware information into software information through interfaces.
🟠	Communication Module	Data exchange between modules is achieved through the ROS framework.
⚪	Grasping Module	Perform tasks such as target recognition, grasping, and placement.
🔵	User Interface Module	Provide interfaces for user-robot interaction, including remote control and visualization functions.
🔴	Perception Module	Collect sensor data, perform data fusion, and provide raw data for target detection.
🟢	Navigation Module	Implement path planning, localization, navigation, obstacle avoidance, and target detection.
🟣	System Management Module	Manage the system's operational states, including startup, shutdown, restart, and power management.

Fig. 15. RQT Module Division Color-Coded Reference Diagram

5.3 Modular Design

5.3.1 Perception Module Design

This module primarily corresponds to the red section of the RQT graph. In our current configuration, the primary sensors include a LiDAR, an Inertial Measurement Unit (IMU), and a depth camera. Multi-sensor data fusion is utilized to improve the accuracy of robot localization and map construction.

In the RQT graph, the main nodes of the Perception Module include /depth_node, /camera_node, /rplidar_node, and /imu, while /sensors_node integrates these multiple nodes. The relevant topics include /scan, /camera/depth/image_raw, and /imu, providing reliable data support for subsequent mapping (SLAM), obstacle detection, and localization functionalities.

5.3.2 Navigation Module design

This module primarily corresponds to the green section of the RQT graph. We plan to use slam_toolbox to construct real-time maps of dynamic environments, enabling the exploration of unknown areas while performing local localization. Additionally, Extended Kalman Filter (EKF) will be employed to fuse odometry, IMU, and LiDAR data, enhancing the smoothness and accuracy of localization.

The target detection algorithm has not yet been determined. For Global Path Planning, we intend to use the A* algorithm, which is simple and efficient for finding the shortest path. For Local Path Planning, we plan to use TEB.

The core nodes include /planner_server, /controller_server, and /bt_navigator for path planning and navigation. /slam_toolbox is responsible for real-time SLAM and localization. Additionally, the /velocity_smoother node ensures smooth velocity control, and the /explore_node node handles target detection and obstacle avoidance during navigation.
Through communication topics such as /odom, /speed_limit, /map, and /joint_states, real-time updates to /cmd_vel and /pose are achieved, enabling autonomous navigation, obstacle avoidance, dynamic path adjustment, and target detection during navigation.

Implementation Flowchart: We have preliminarily designed the implementation flowchart for the Navigation Module, as shown below (Figure 16). Figure 17 is a detailed implementation flowchart for Frontier Exploration. This flowchart primarily represents the navigation system process for path planning and dynamic obstacle detection, including sensor initialization, localization, frontier exploration, global path planning, local path planning, and target grasping.

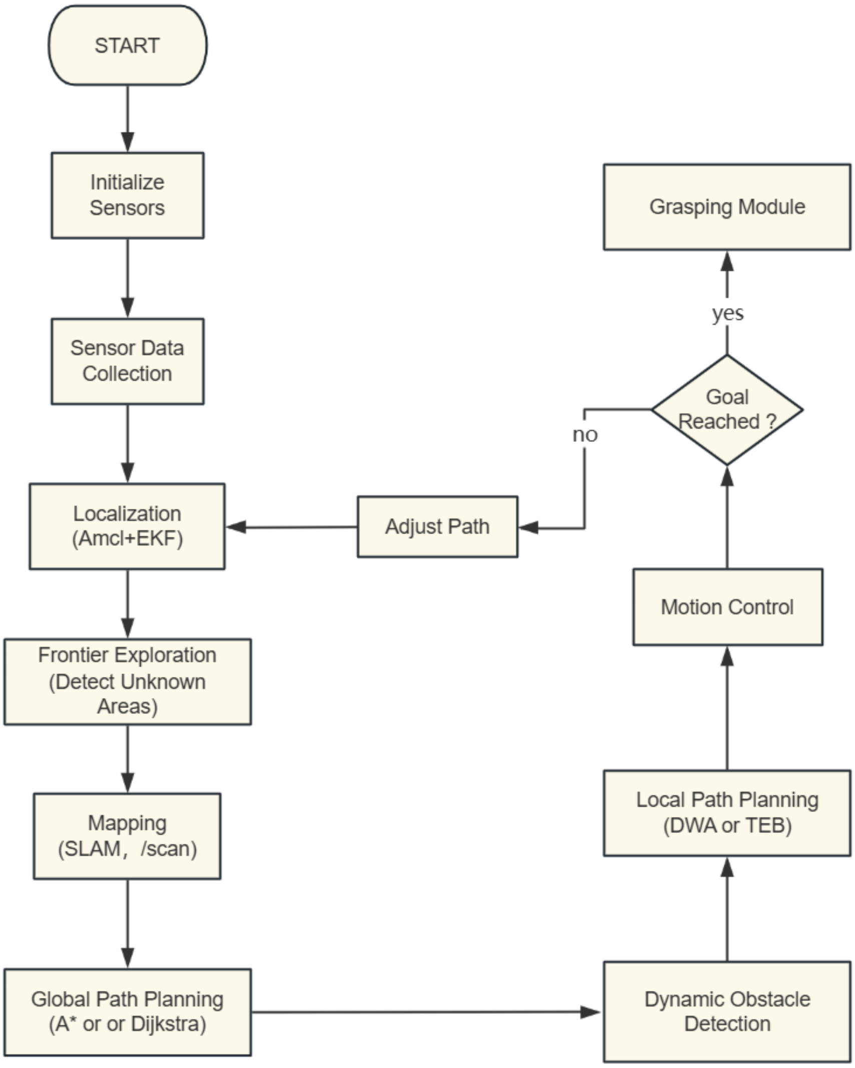


Fig. 16. Behavior Tree of the Navigation Module Implementation

Through Frontier Exploration, the boundaries of unexplored areas are identified, guiding the robot to navigate to uncharted regions, ultimately achieving global exploration of the environment. The flowchart is as follows:

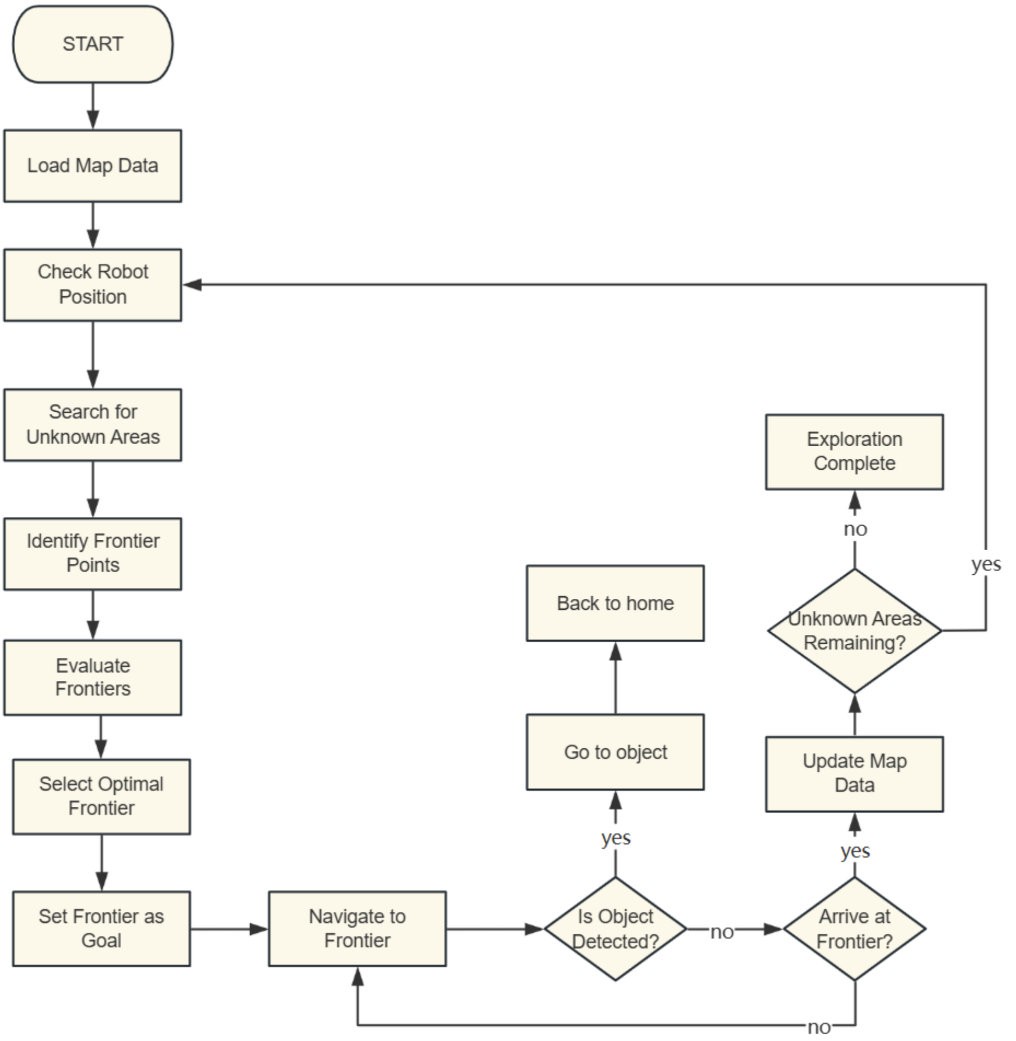


Fig. 17. Flowchart of Frontier Exploration Implementation

5.3.3 Grasping Module design

This module primarily corresponds to the gray section of the RQT graph. It is designed to implement the robotic arm's grasping and placement functions. By integrating the robotic arm's operational control, target detection via a depth camera, and grasping path planning, the system ensures that the robot can accurately identify, locate, and grasp target objects.

The core nodes include `/manipulator_node`, `/gripper_node`, and `/arm_controller`, which are responsible for controlling the robotic arm's movements, gripper operations, and their coordination.

In future plans, we aim to further refine and optimize the target detection algorithm, explore the potential of machine learning to enhance detection efficiency, and investigate force feedback-based grasping algorithms to improve the robot's adaptability to complex objects.

This module utilizes topics such as `/joint_states`, `/camera/depth`, and `/scan` to acquire the robotic arm's state and 3D information about objects. Coordination with the navigation module is achieved through `/cmd_vel`, ensuring that the grasping path avoids obstacles and completes grasping and placement tasks safely and efficiently.

Implementation Flowchart: We have preliminarily designed the implementation flowchart for the Grasping Module, as shown below. The flowchart illustrates the implementation process of robotic arm grasping, including target detection, 3D position calculation, grasp execution, and robot movement.

Please refer to the software section of the Git repository for more detailed and clear explanations about each node in the RQT graph of the grasping program.

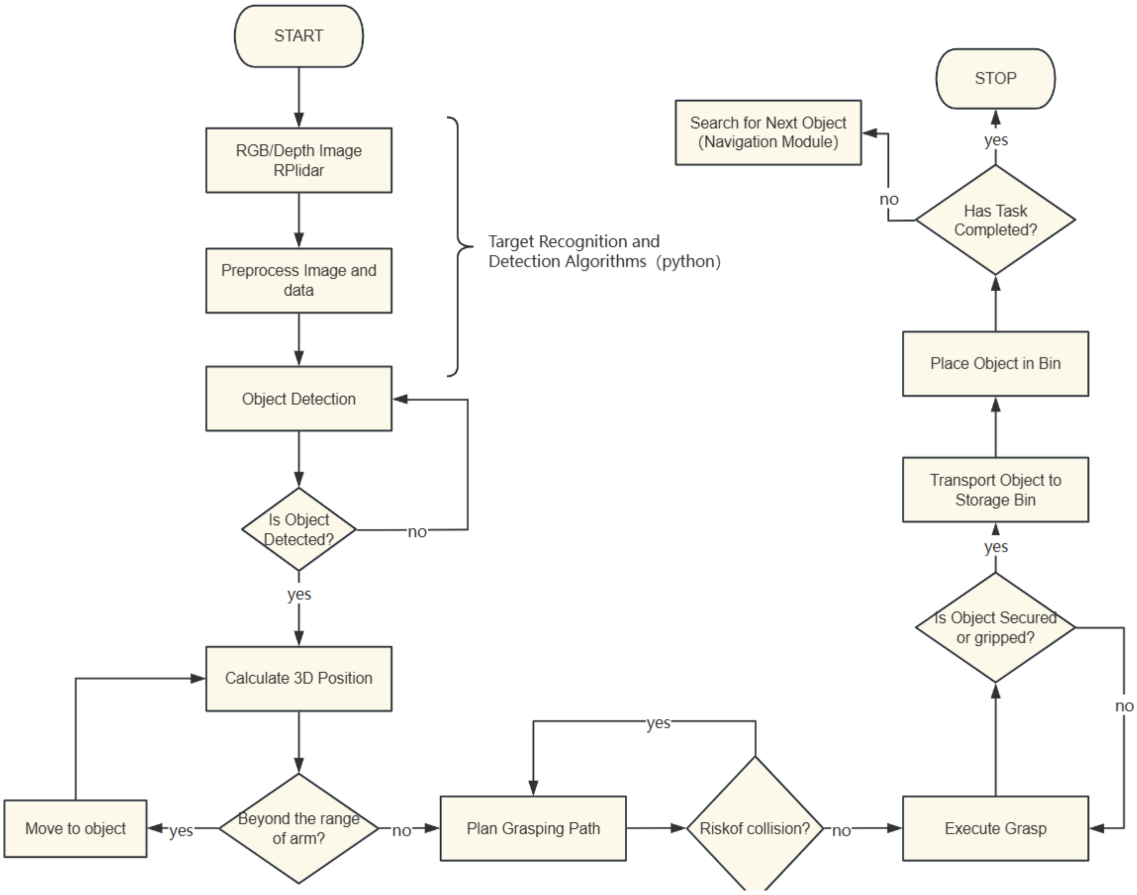


Fig. 18. Behavior Tree of the Grasping Module Implementation

5.3.4 Communication Module design

This module primarily corresponds to the orange and yellow sections of the RQT graph.

The orange section is responsible for enabling communication between the robot's internal modules through the ROS system, such as exchanging data in the form of topics and services. The core nodes include `/ros2_gz_bridge` and `/robot_state_publisher`.

The yellow section focuses on hardware communication, interacting with external devices (e.g., control consoles) via PCB boards, wireless communication interfaces, and other methods. The core nodes include `/firmware_lecore` and `/firm_message_converter`, which are used to receive and send data related to the chassis and sensors, ensuring detailed conversion between software and hardware.

Future work will focus on enhancing the stability and transmission speed of this module.

5.3.5 System Management Module design

This module primarily corresponds to the light purple section of the RQT graph, with some overlap with other modules.

We designed this module to be primarily responsible for the overall management and coordination of the robot system, including low-level hardware control, task scheduling, and status monitoring functionalities.

The core nodes include `/firmware_leo_core` for low-level hardware control, `/battery_node` for monitoring battery status, `/robot_state_publisher` for publishing robot state and posture information, and `/joint_state_publisher` for managing joint state data.
Through interaction with other modules, this module ensures the stable and reliable operation of the system.

5.3.6 User Interface Module design

This module primarily corresponds to the light blue section of the RQT graph.

We designed this module to be primarily responsible for providing human-machine interaction functionalities, including remote control, status monitoring, and task configuration.

In the RQT graph, nodes related to user interaction can be observed. For example, `/joystick_node` is used to receive joystick input, enabling manual control of the robot's motion and tasks. `/teleop_cmd_vel` converts remote control input into velocity control commands, which are published to the `/cmd_vel` topic to control the robot's movement. `/rviz` provides a visualization interface for the user. `/leo_ui` helps users monitor the robot's real-time status information, such as position, speed, and battery level.

- Start or shut down the robot
- Visualize the robot's trajectory and other details
- Monitor the robot's battery level, speed, etc.
- Steering the robot

GitHub: https://github.com/CMX-9/Team5-Mobile_Grasping_Robot

6 Analysis

This section analyzes the mechanical design, electrical design, and software design based on the project goals, in accordance with the Functional and Performance Requirements. It explains how the preliminary design meets these requirements.

6.1 The robot shall have mobility capability

Verification Success Criteria: The robot should be able to move forward, backward, and rotate on a dry, hard surface.

Verification Method: Various motion commands, such as moving forward, backward, and rotating in place, will be issued to the robot on a dry, hard surface.

Implementation Analysis: The LEO core controller serves as the central hub for motor control and sensor data processing, coordinating various modules to achieve precise motion control of the robot. The DC motors with encoders drive the robot's wheels and provide position feedback, ensuring precise control of speed and direction. The integrated IMU continuously monitors the robot's acceleration and angular velocity, supporting fine-tuned motion control. The WiFi adapter with an external antenna ensures a stable wireless connection for remote control and data transmission during testing and debugging.

6.2 The robot shall generate a map based on its surroundings

Verification Success Criteria: The map generated by the robot should match the actual environment with an accuracy of over 95

Verification Method: The robot will be placed in a relatively complex environment containing various obstacles. The generated map will be compared with the actual environment during the robot's operation.

Implementation Analysis: The RPLIDAR A2M12, along with the Intel RealSense depth camera, will be used to scan and capture the surrounding environment. These sensors provide essential data for mapping. The slam_toolbox will be employed to construct real-time maps of the dynamic environment, utilizing the LiDAR and depth camera data for precise localization and mapping. The SLAM algorithm processes the sensor information to build an accurate representation of the robot's surroundings, and continuous adjustments will be made based on the dynamic environment, ensuring an accurate map generation process.

6.3 The robot can autonomously navigate and perform obstacle avoidance

Verification Success Criteria: The robot should be able to navigate around all obstacles in its environment and autonomously plan a path to reach the target location.

Verification Method: The robot is provided with target coordinates, and its movement is observed to ensure successful navigation and obstacle avoidance.

Implementation Analysis: The robot's autonomous navigation and obstacle avoidance capabilities are achieved through a well-integrated system of sensors, algorithms, and computational resources. The IMU provides real-time data on the robot's direction, acceleration, and angular velocity, ensuring precise motion control. The Intel NUC serves as a high-performance platform for handling intensive tasks such as SLAM and advanced navigation algorithms. RPLIDAR A2M12 performs 360-degree environmental scanning, generating a detailed 2D map essential for path planning and obstacle detection, while the Intel RealSense depth camera captures 3D environmental data to enhance object detection and spatial analysis. To minimize sensor interference and maximize data accuracy, the LiDAR and depth camera are strategically positioned at the front of the robot's chassis. The A algorithm is employed for global path planning due to its efficiency in finding the shortest path, while TEB is used for local path planning to optimize navigation in dynamic environments. Key ROS nodes such as /planner_server, /controller_server, and /bt_navigator work together to ensure smooth, real-time navigation and obstacle avoidance, allowing the robot to autonomously reach its target.

6.4 The robot shall be capable of grasping and storing objects

Verification Success Criteria: The robotic arm should be capable of accurately grasping and transporting objects to a target location, with a drop probability of less than 10

Verification Method: Issue position coordinates to the robotic arm and test its ability to grasp an object at a specific location and transport it to another target position. Multiple tests will be conducted using objects of different shapes.

Implementation Analysis: The PincherX 150 robotic arm, with its high precision, multi-degree-of-freedom design, and flexibility, is well-suited for precise object grasping and placement tasks. Its core control system operates through software nodes such as /manipulator_node, /gripper_node, and /arm_controller, which are responsible for controlling the arm's motion, managing the gripper's operation, and ensuring coordination between these elements. Through these nodes, the robotic arm can precisely adjust its posture according to predefined paths and target positions, ensuring that objects are stably grasped and transported to the designated location. Additionally, the /joint_states and /camera topics provide real-time feedback on the arm's state and the 3D information of the object, enabling dynamic adjustments to prevent object drop during the grasping process and ensuring task completion with high efficiency. Furthermore, the PincherX 150's high payload capacity and durability allow it to maintain stability when handling objects of varying shapes and weights, ensuring reliable execution of tasks. Combining the advantages of hardware and software support, the PincherX 150 robotic arm can effectively and accurately complete object grasping and placement tasks.

6.5 The robot shall have a state tracking capability

Verification Success Criteria: The robot's operational status is continuously printed on the terminal. The robot shall provide real-time feedback on its operational status: gripping, recognizing, and avoiding obstacles.

Verification Method: During robot testing, record the terminal status information.

Implementation Analysis: The robot's state tracking capability is realized through the integration of hardware sensors, data processing algorithms, and the /leo_ui user interface, which allows real-time monitoring of the robot's status, including position, velocity, and battery level. The IMU provides data on the robot's orientation and movement, while wheel encoders track displacement and speed. Battery sensors continuously measure power levels. These data streams are collected and processed by the /leo_ui node, which visualizes the robot's state on the user interface. This enables users to monitor critical parameters in real time, ensuring that the robot's operational status is always available for adjustment or troubleshooting. The /leo_ui node plays a key role in integrating and displaying this information, ensuring the robot's state is accurately tracked and communicated.

6.6 Mechanical design requirements for the robot

Verification Success Criteria: The dimensions of the robot shall not exceed 500 mm x 500 mm x 500 mm, and the mechanical structure shall demonstrate strong stability.

Verification Method: Measure the final dimensions of the robot and conduct a structural strength analysis.

Implementation Analysis: The designed mechanical structure results in an overall robot size of 448.55 mm × 439.03 mm × 527.83 mm when the robotic arm is retracted. Based on the strength analysis in the mechanical design, it is determined that the structure will not undergo strain during movement, ensuring good stability.

6.7 The cost of the robot shall be controlled

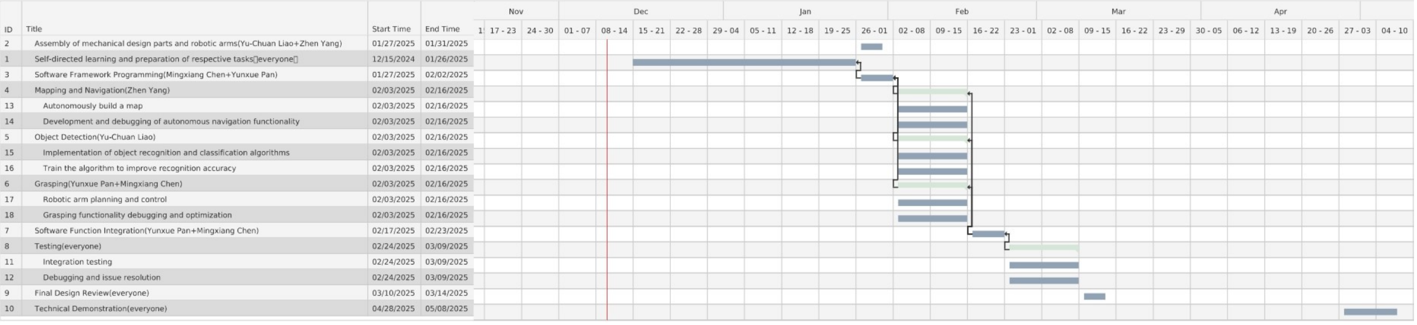
Verification Success Criteria: In subsequent designs, laser cutting and 3D printing shall not exceed the manufacturing budget.

Verification Method: Calculate the total cost.

Implementation Analysis: In addition to the already provided robot components, the subsequent costs will only involve the expenses for laser cutting or 3D printing to assemble the mechanical structure of various parts. This approach allows for effective cost control by minimizing the need for expensive or complex manufacturing processes, while still ensuring high precision and quality in the final assembly. The use of cost-effective manufacturing techniques like laser cutting and 3D printing offers flexibility in design and production, enabling rapid prototyping and modifications when necessary, which further enhances the overall cost-effectiveness of the project.

6.8 Safety

Verification Success Criteria: The maximum speed of the robot shall not exceed 0.5 m/s, and it shall be able to stop within 1 second after receiving a braking command.



ID	Title	Start Time	End Time
2	Assembly of mechanical design parts and robotic arms(Yu-Chuan Liao+Zhen Yang)	01/27/2025	01/31/2025
1	Self-directed learning and preparation of respective tasks□everyone□	12/15/2024	01/26/2025
3	Software Framework Programming(Mingxiang Chen+Yunxue Pan)	01/27/2025	02/02/2025
4	Mapping and Navigation(Zhen Yang)	02/03/2025	02/16/2025
13	Autonomously build a map	02/03/2025	02/16/2025
14	Development and debugging of autonomous navigation functionality	02/03/2025	02/16/2025
5	Object Detection(Yu-Chuan Liao)	02/03/2025	02/16/2025
15	Implementation of object recognition and classification algorithms	02/03/2025	02/16/2025
16	Train the algorithm to improve recognition accuracy	02/03/2025	02/16/2025
6	Grasping(Yunxue Pan+Mingxiang Chen)	02/03/2025	02/16/2025
17	Robotic arm planning and control	02/03/2025	02/16/2025
18	Grasping functionality debugging and optimization	02/03/2025	02/16/2025
7	Software Function Integration(Yunxue Pan+Mingxiang Chen)	02/17/2025	02/23/2025
8	Testing(everyone)	02/24/2025	03/09/2025
11	Integration testing	02/24/2025	03/09/2025
12	Debugging and issue resolution	02/24/2025	03/09/2025
9	Final Design Review(everyone)	03/10/2025	03/14/2025
10	Technical Demonstration(everyone)	04/28/2025	05/08/2025

Fig. 19. Gantt Chart

Verification Method: Test the robot's maximum speed and braking functionality.

Implementation Analysis: The robot's maximum speed and braking functionality are achieved through an integrated approach across electrical, software, and mechanical design. The electrical design utilizes motor controllers with encoders for precise speed regulation and rapid braking via reverse polarity or dynamic braking. The software ensures smooth deceleration by implementing control algorithms, such as PID, to meet the 1-second stop requirement. Mechanically, the lightweight structure and differential drive system, along with optimized wheel design, enable efficient deceleration and ensure the robot can stop quickly within the specified limits.

6.9 Reliability

Verification Success Criteria: The robot should be able to operate for more than 1 hour on a full charge.

Verification Method: The robot will be operated on a full charge, and the operating time will be recorded.

Implementation Analysis: The Leo Rover robot is powered by a lithium-ion battery with a built-in PCM (Protection Circuit Module) for short-circuit, overcurrent, and overdischarge protection. The battery operates at 12V DC with a capacity of 5800 mAh and a maximum current of 8A, which covers the total current consumption of the entire Rover. The robotic arm is powered by a separate 12V 5A battery, enabling extended operation times.

7 Project Plan

The tasks for the second semester can be broadly categorized into three primary modules: **Mapping and Navigation, Object Detection**, and **Grasping**. Each task requires substantial preliminary preparation and subsequent debugging and optimization. Based on the project timeline, the following work plan has been formulated.

8 Appendix

A Design Requirements Analysis

A.1 Stakeholder Engagement

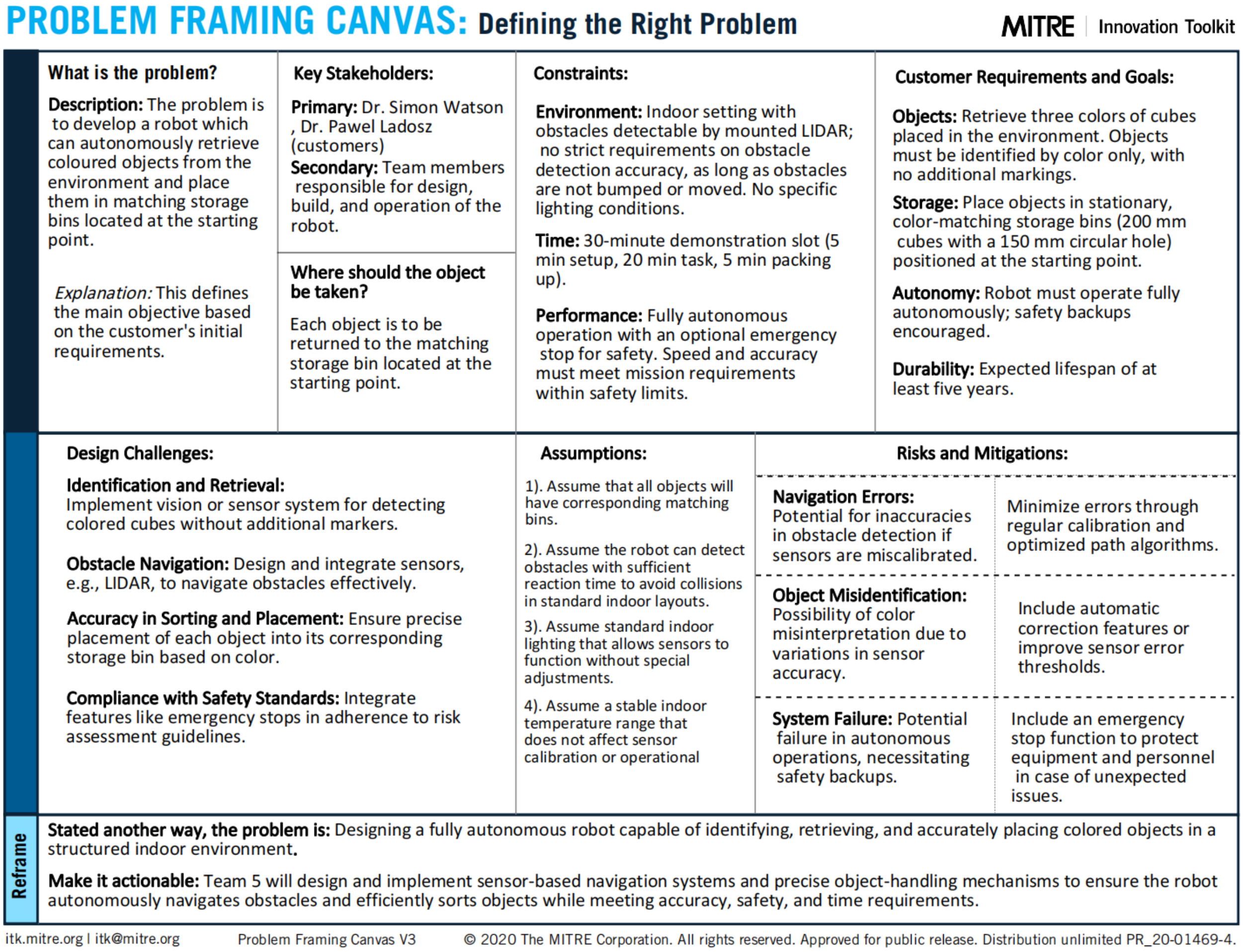
A.1.1 Introduction

In this project, Group 5 aims to understand and address the customer’s requirements effectively. The initial statement provided by the customer is as follows:

> **“Develop a robot which can autonomously retrieve coloured objects from the environment and place them in matching storage bins located at the starting point.”**

To further define the scope of the project, the team held two engagement sessions with the primary stakeholders, Dr. Simon Watson and Dr. Pawel Ladosz. The following sections summarize their responses to clarify key requirements and expectations for the robotic system design project.

A.1.2 Problem Framing Canvas



PROBLEM FRAMING CANVAS: Defining the Right Problem

MITRE | Innovation Toolkit

What is the problem?

Description: The problem is to develop a robot which can autonomously retrieve coloured objects from the environment and place them in matching storage bins located at the starting point.

Explanation: This defines the main objective based on the customer's initial requirements.

Key Stakeholders:

Primary: Dr. Simon Watson , Dr. Pawel Ladosz (customers)

Secondary: Team members responsible for design, build, and operation of the robot.

Where should the object be taken?

Each object is to be returned to the matching storage bin located at the starting point.

Constraints:

Environment: Indoor setting with obstacles detectable by mounted LIDAR; no strict requirements on obstacle detection accuracy, as long as obstacles are not bumped or moved. No specific lighting conditions.

Time: 30-minute demonstration slot (5 min setup, 20 min task, 5 min packing up).

Performance: Fully autonomous operation with an optional emergency stop for safety. Speed and accuracy must meet mission requirements within safety limits.

Customer Requirements and Goals:

Objects: Retrieve three colors of cubes placed in the environment. Objects must be identified by color only, with no additional markings.

Storage: Place objects in stationary, color-matching storage bins (200 mm cubes with a 150 mm circular hole) positioned at the starting point.

Autonomy: Robot must operate fully autonomously; safety backups encouraged.

Durability: Expected lifespan of at least five years.

Design Challenges:

Identification and Retrieval: Implement vision or sensor system for detecting colored cubes without additional markers.

Obstacle Navigation: Design and integrate sensors, e.g., LIDAR, to navigate obstacles effectively.

Accuracy in Sorting and Placement: Ensure precise placement of each object into its corresponding storage bin based on color.

Compliance with Safety Standards: Integrate features like emergency stops in adherence to risk assessment guidelines.

Assumptions:

1). Assume that all objects will have corresponding matching bins.

2). Assume the robot can detect obstacles with sufficient reaction time to avoid collisions in standard indoor layouts.

3). Assume standard indoor lighting that allows sensors to function without special adjustments.

4). Assume a stable indoor temperature range that does not affect sensor calibration or operational

Risks and Mitigations:

Navigation Errors: Potential for inaccuracies in obstacle detection if sensors are miscalibrated.	Minimize errors through regular calibration and optimized path algorithms.
Object Misidentification: Possibility of color misinterpretation due to variations in sensor accuracy.	Include automatic correction features or improve sensor error thresholds.
System Failure: Potential failure in autonomous operations, necessitating safety backups.	Include an emergency stop function to protect equipment and personnel in case of unexpected issues.

Reframe

Stated another way, the problem is: Designing a fully autonomous robot capable of identifying, retrieving, and accurately placing colored objects in a structured indoor environment.

Make it actionable: Team 5 will design and implement sensor-based navigation systems and precise object-handling mechanisms to ensure the robot autonomously navigates obstacles and efficiently sorts objects while meeting accuracy, safety, and time requirements.

itk.mitre.org | itk@mitre.org Problem Framing Canvas V3

Fig. 20. Problem Framing Canvas developed based on customer clarifications

A.2 Problem Statement

After discussion with the customers, Dr. Simon Watson and Dr. Pawel Ladosz, the requirements for the project were defined. The customer requested a fully autonomous robot equipped with sensors and a manipulator capable of retrieving specific objects from an unmapped environment and placing them into matching storage bins within a limited time. The environment will include several stationary obstacles.

This project aims to provide an opportunity to apply our robotics knowledge from lectures in a practical context while fostering teamwork skills, including communication, collaboration, and coordination.

A.2.1 Objectives

- Ensure the operation is safe, efficient, and fully autonomous.(3.1,3.2)

- Use a mobile robotic platform equipped with necessary hardware and sensors to achieve the project goals.
- Develop autonomous navigation capabilities, enabling the robot to map the environment and move without manual intervention.(3.2,3.3)
- Implement computer vision systems to detect, classify, and interact with objects accurately.(3.3,3.4)
- Ensure collision avoidance with both stationary obstacles in the environment.(3.2,3.4)
- Complete the task within a strict time limit of 20 minutes.
- Incorporate robust safety measures, such as an emergency stop function, to enhance operational safety. (3.1.4)

A.3 Functional and Performance Requirements

According to the above customer requirements, the robot to be designed shall be able to achieve autonomous navigation, avoiding obstacles, recognising objects and grasping and placing functions. The following is a detailed description of the function and performance.

A.3.1 The robot shall have mobility capability

3.1.1 The robot shall be capable of moving on dry, hard surfaces.

3.1.2 The robot shall operate in indoor environments.

3.1.3 According to the ISO/TS 15066 recommendations for safe speeds when interacting with humans, the maximum speed of the robot is specified to be no more than 0.5m/s. This speed significantly reduces the risk of injury from a collision, while still maintaining basic task execution efficiency

3.1.4 The robot shall have an braking function, enabling it to stop within 1 second to comply with safety standards.

A.3.2 The robot shall be able to autonomously plan its path

3.2.1 The robot shall be capable of generating a map of its surroundings in real-time during operation.

3.2.2 The robot shall be able to autonomously plan navigation paths based on the generated map.

A.3.3 The robot shall be able to avoid obstacles

3.3.1 The probability of the robot successfully navigating the path in the absence of moving obstacles shall be at least 90%.

3.3.2 The robot shall be capable of detecting obstacles within a one-meter diameter range, including those that may be lower than the robot’s frame, with a false detection rate of less than 5%.

3.3.3 The obstacles shall be fixed static objects.

A.3.4 The robot shall be able to detect objects

3.4.1 Under fixed lighting conditions and color temperature, the robot shall be able to recognize objects based on color, with a success rate exceeding 90%.

3.4.2 The robot shall be able to match recognized objects of the same color.

3.4.3 The robot shall be able to display the quantity and color of recognized objects on the laptop screen in real time.

A.3.5 The robot shall be capable of grasping and storing objects

3.5.1 The robotic arm shall be able to move to the target position in 10 seconds to ensure efficient execution of the task and meet the overall time constraints of the operation.

3.5.2 The robotic arm shall be capable of grasping polygonal objects, with the ability to grasp items weighing less than 50 grams at a time.

3.5.3 The robot shall be able to place the grasped objects at designated locations, with a drop rate of less than 5% during the process.

A.3.6 The robot shall have a state tracking capability

3.6.1 The robot shall provide real-time feedback on its operational status: gripping, recognizing, and avoiding obstacles.

A.3.7 Mechanical design requirements for the robot

3.7.1 The dimensions of the robot shall not exceed 500 mm x 500 mm x 500 mm.

3.7.2 The robot’s lifespan shall be at least 5 years to meet the client’s specified durability requirement, ensuring reliable performance and long-term value throughout its intended operational period.

3.7.3 The robot shall be able to operate for at least 1 hour when fully charged.

A.3.8 The cost of the robot shall be controlled

3.8.1 In subsequent designs, laser cutting and 3D printing shall not exceed the manufacturing budget.

A.4 Requirements Verification Matrix

After identifying all the specific requirements, it is necessary to identify how to validate that the requirements are realised. The following verification matrix is used to document the validation process. The verification matrix contains test items, test methods and testers. The final test results are recorded in a tabular form. The verification matrix is shown in the table 1 below.

Table 2. Requirements Verification Matrix Table

Requirement No.[A]	**Section**[B]	**Shall Statement**[C]	**Verification Success Criteria**[D]	**Verification Method**[E]	**Responsible Party**[F]	**Results**[G]
P-1	3.1.1	The robot shall be capable of moving on dry, hard surfaces.	The robot can move normally on a dry, hard indoor surface.	Test the robot's movement performance on dry, hard indoor surfaces.	Yunxue	
P-2	3.1.2	The robot shall be able to move forward and backward and to turn.	A. The robot can move forward and backward. B. The robot can rotate.	Test and verify in a laboratory environment, observing and recording the robot's movement performance.	YunXue	
P-3	3.1.3	The robot's movement speed shall comply with safety standards.	The maximum speed does not exceed 0.5 meters	Run the robot continuously for at least half an hour in a laboratory environment, recording the maximum speed during the test.	YunXue	

Requirement No.[A]	Section[B]	Shall Statement[C]	Verification Success Criteria[D]	Verification Method[E]	Responsible Party[F]	Results[G]
P-4	3.1.4	The robot shall have an emergency braking function, enabling it to stop.	Successful stopping.	Test if the robot can stop.	YunXue	
P-5	3.1.4	The robot shall can brake successfully in less than 1 second	The time from issuing the stop command to the robot completely stopping does not exceed 1 second.	Test whether the robot's average braking time exceeds 1 second.	YunXue	
P-6	3.2.1	The robot shall be capable of generating a map of its surroundings in real-time during operation.	Real-time map navigation updates.	Test, observe, and record whether the map remains active without any crashes or black screens during a 60-minute period.	MingXiang	
P-7	3.2.2	The robot shall be able to autonomously plan navigation paths based on the generated map.	The robot moves according to the planned path.	Test, observe and compare the robot's actual movement path with the planned path.	MingXiang	
P-8	3.3.1	The probability of the robot navigating the path without moving obstacles shall be greater than 90 percent.	Obstacle avoidance success rate reaches 90 per cent or higher.	Test the robot in different obstacle scenarios, recording the number of successful and failed obstacle avoidance attempts.	MingXiang	

Requirement No.[A]	Section[B]	Shall Statement[C]	Verification Success Criteria[D]	Verification Method[E]	Responsible Party[F]	Results[G]
P-9	3.3.2	The robot shall be capable of detecting obstacles within a one-meter diameter range, including those that may be lower than the robot's frame, with a false detection rate of less than 5 per cent.	A. The robot can detect all obstacles within a 1-meter diameter range. B. The false detection rate is lower than 5 per cent.	Test the robot in various one-meter obstacle scenarios, recording the number of false detections and missed obstacles.	MingXiang	
P-10	3.4.1	In conditions of fixed lighting and color temperature, the robot shall be able to recognize objects based on color.	Target recognition success rate reaches 90 percent or higher.	Use cubes of different colors, recording the number of successful color recognitions and calculating the recognition success rate.	MingXiang	
P-11	3.4.2	The robot shall be able to match recognized objects of the same color.	The robot can correctly match colors.	Conduct multiple experiments with objects of different colors to observe whether the robot can successfully match them.	Zhen Yang	
P-12	3.4.3	The robot shall be able to display the quantity and color of recognized objects on the laptop screen in real-time	The color and quantity of recognized objects are continuously printed on the terminal.	During robot testing, record the terminal status information.	Zhen Yang	

Requirement No.[A]	Section[B]	Shall Statement[C]	Verification Success Criteria[D]	Verification Method[E]	Responsible Party[F]	Results[G]
P-13	3.5.1	The robotic arm shall be able to move to the target position within 10 seconds	After recognizing the target, the robotic arm moves to the target coordinates within 10 seconds.	During the robot's target recognition and grabbing tests, record the time it takes for the robotic arm to move and reach the target position.	Zhen Yang	
P-14	3.5.2	The robotic arm shall be capable of grasping polygonal objects, with the ability to grasp items weighing less than 50 grams at one time.	The success rate of grabbing a multi-sided cube is greater than 90 percent	Test the robot by grabbing cubes of different shapes, recording the success rate of the grab.	Zhen Yang	
P-15	3.5.3	The robot shall be able to place the grasped objects at designated locations, with a drop rate of less than 5 per cent during the process	The drop rate when the robot places an object is less than 5 per cent	After grabbing cubes of different shapes, place them in the designated position, and record the placement success rate.	Zhen Yang	
P-16	3.6.1	The robot shall provide real-time feedback on its operational status: gripping, recognizing, and avoiding obstacles	The robot's operational status is continuously printed on the terminal.	During robot testing, record the terminal status information.	Yu-Chuan	

Requirement No.[A]	Section[B]	Shall Statement[C]	Verification Success Criteria[D]	Verification Method[E]	Responsible Party[F]	Results[G]
P-18	3.7.1	The dimensions of the robot shall not exceed 500 mm x 500 mm x 500 mm	The robot's size is less than 500 mm x 500 mm x 500 mm.	Measure the robot's length, width, and height.	Yu-Chuan	
P-19	3.7.3	The robot shall be able to operate for at least one hour when fully charged	The robot can operate for more than one hour on a full charge.	Test whether the robot can continuously run for more than one hour in the laboratory environment with a full charge.	Yu-Chuan	
P-20	3.8.1	In subsequent designs, laser cutting and 3D printing shall not exceed the manufacturing budget.	The cost does not exceed the minimum cost threshold.	Calculate the total cost.	Yu-Chuan	

A. Unique identifier for each Robotic System requirement.
B. Section number each unique Robotic System Requirement is contained within.
C. Text (within reason) of the Robotic System requirement, i.e., the "shall".
D. Success criteria for the Robotic System requirement.
E. Verification method for the Robotic System requirement (analysis, inspection, demonstration, or test).
F. Responsible person for performing the verification.
G. Indicate documents that contain the objective evidence that the requirement was satisfied.

B Updated Workplace Charter

Our Purpose: This Workplace Charter sets out the principles, rules, and conflict management processes that Team 5 will follow throughout the Robotic Systems Design Project. The Charter aims to ensure that all members contribute effectively while maintaining a respectful and inclusive environment.

Our Principles and Commitments:

- **Equity:**We are dedicated to ensuring that each team member is treated fairly and provided with the same opportunities for success.
- **Diversity:**Our team values the variety of backgrounds, experiences, and perspectives that each member brings. We encourage all team members to contribute their unique viewpoints, enriching the overall project.
- **Inclusion:**□Open dialogue will be encouraged so that everyone's opinions are acknowledged and respected, working collaboratively towards common goals.
- **Accessibility:**We are committed to ensuring that all team resources, meetings, and activities are accessible to every member.

Team Rules:

- All communication within the team will be conducted in English to ensure clarity and consistency.
- Standard working hours will be 9:00 AM to 6:00 PM on workdays. Work on weekends will only occur under special circumstances, with prior team agreement.
- The team will hold a regular weekly meeting every Wednesday from 3:00 PM to 4:00 PM to review progress, discuss tasks, develop a work plan, and resolve any issues. Attendance at all team meetings is required. If a member cannot attend, they should notify the team at least 24 hours in advance.
- All team members must adhere to agreed deadlines to maintain project momentum.
- Teams will be the primary platform for formal discussions, document sharing, and project updates. WhatsApp and WeChat will be used for quick and informal exchanges.
- The team will actively seek input from each team member throughout all stages of the project to ensure that everyone's thoughts and perspectives are considered.
- All major decisions will be made through a voting process, with a simple majority ruling. In the event of a tie, the team leader will have the final say.
- Respect for the politics, religion, culture and beliefs of each member. Discrimination is strictly prohibited.
- If someone break the rules, The first step is for the group leader to communicate privately with the group members. If the group member has a negative attitude and is extremely uncooperative with the group, seek help from the professor.
- The team will regularly assess and update the EDIA policies to ensure that our efforts remain effective, inclusive, and aligned with the project goals and values.

Daily Activity Conflict Avoidance:

In every meeting, each team member's tasks and responsibilities will be clearly defined, and the meeting minutes will be taken on a rotating basis to ensure all team members share responsibility for documentation and accountability. Regularly check the work progress of team members and supervise each other. A friendly and supportive communication style will be encouraged, promoting a positive atmosphere where issues can be discussed openly and resolved quickly before they escalate.

Conflict Resolution Flowchart:

When a conflict occurs, a judgement is first made about the type of conflict. Our team categorizes three types of conflict: communication and misunderstanding, work issues and, most seriously, issues such as discrimination and bullying. Different solutions apply to different types of conflicts. The process is illustrated in the figure below:

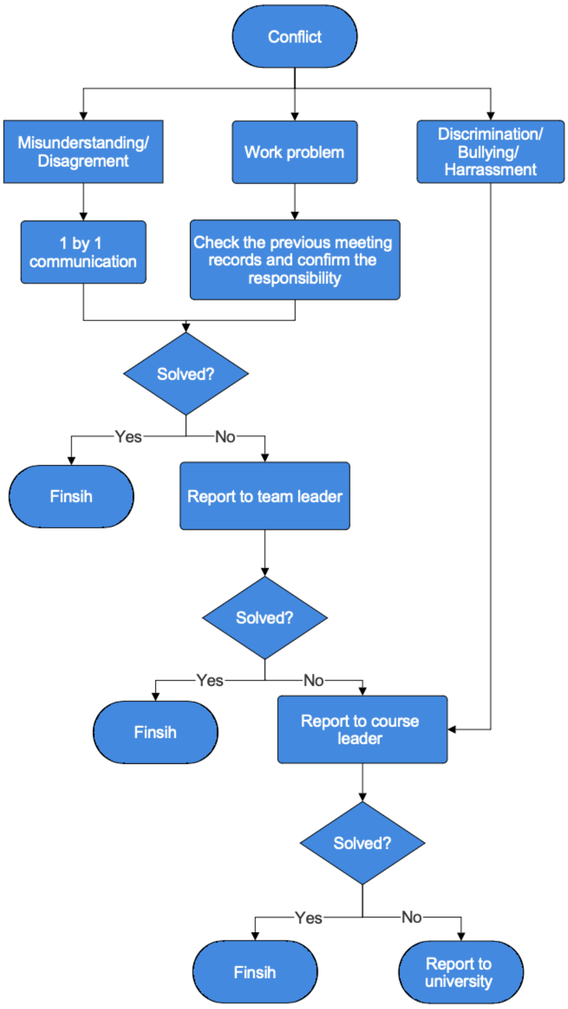


Fig. 21. Conflict Resolution Flowchart