

FINAL DESIGN REVIEW

AERO62520 - Robotic Systems Design Project

Version: 1.0
Date: March 19, 2026
Coursework ID: 06

PROJECT TEAM

Name	Student ID
Nadiya Ivahnenko	14367246
Bokai Qin	14137913
Jayant Bakolia	14158996
Mingyang Xu	14223377
Yinfeng Li	14187032
Allan Binoy Issac	10929893

Contents

1	Introduction	5
1.1	Problem Statement and Objectives	5
1.2	Response to PDR Feedback	5
1.3	Key Modifications Since the PDR	7
2	Sustainability Checklist	9
3	Cyber Security Considerations	10
3.1	Network and Access Security	10
3.2	ROS 2 Command and Control Protection	10
3.3	Secure Configuration and Recoverable Operation	10
3.4	Monitoring, Logging and Incident Response	10
3.5	Summary	10
4	System Overview	11
4.1	Leo Rover Primary System (Low-Level Control)	11
4.2	Secondary Peripheral Systems (High-Level Control)	12
5	Mechanical Design	14
5.1	Mechanical Design Objectives	14
5.2	Key Design Challenges and Decisions	14
5.2.1	Placement of the LiDAR Sensor	14
5.2.2	Placement of Manipulator Arm and Cameras with Respect to the LiDAR	15
5.2.3	Placement of the On-board Compute (NUC)	16
5.3	Payload Sled Architecture	16
5.4	Selection of Plate Thickness	17
5.5	LiDAR Adapter	18
5.6	Connectors Between Top and Bottom Plates	18
5.6.1	I-beam Connector Design	18
5.6.2	Connector Placement	19
5.7	NUC Holder / Adapter	20
5.8	Camera Adapter and Integration with the Manipulator	20
5.9	Sensor Obstructions and Field-of-View Considerations	23
5.10	Plate Designs	23
5.11	Full Assembly	27
5.12	Materials and Manufacturability	29
5.13	Structural Analysis of the Payload Architecture	29
5.14	Compliance with Mechanical Requirements	32
6	Electrical Design	33
6.1	Electrical Power Connections Diagram	33
6.2	Power Budget	33

6.3	Key Challenges	34
7	Software Design	36
7.1	RQT Graph & Software Architecture	36
7.1.1	Software block diagram	36
7.1.2	RQT Graph	36
7.1.3	Software Architecture	36
7.1.4	Trigger Conditions Introduction	38
7.2	Subsystem Detailed Design	38
7.2.1	Low-level Robot State & TF System (State & TF System)	38
7.2.2	Navigation Subsystem (Navigation Manager + SLAM & Nav2)	38
7.2.3	Perception Subsystem	39
7.2.4	TF Bridge Layer (Coordinate Transformation Layer)	40
7.2.5	Manipulation Subsystem (Manipulation Manager + ros2_control + MoveIt2)	41
7.2.6	Mission Manager	41
7.2.7	Mission Execution Logic — Pseudo-Code Specification	42
7.3	TF Tree	46
7.4	Key Design Decisions (KDD)	46
7.5	Compliance with Software Requirements	48
7.6	Git Repository Link	48
8	Analysis	49
9	Project Plan	52
9.1	Project Planning	52
9.2	Project Description	58
9.2.1	Work Package 1 – Project Management	58
9.2.2	Work Package 2 – Navigation	58
9.2.3	Work Package 3 – Perception	58
9.2.4	Work Package 4 – Manipulation	59
9.2.5	Work Package 5 – Hardware Design	59
9.2.6	Work Package 6 – Systems Integration	60
9.3	Milestones and Deliverables	61
Appendix A: Workplace Charter		63
Appendix B: Design Requirement Analysis		67

List of Figures

1	System-level hardware architecture of the autonomous object retrieval robot.	11
2	Conceptual layout of the front bottom plate, rear bottom plate, front top plate and rear top plate. . .	17
3	Customised RPLIDAR A2 adapter mounted on the front bottom plate.	19
4	Custom I-beam connector between bottom and top plates.	20
5	NUC holder / adapter.	21
6	RGB-D camera adapter mounted at the front of the manipulator assembly on the front top plate, with angular adjustment provided by the gimbal.	22
7	Front bottom plate with LiDAR adapter mount holes and front connector locations.	24
8	Rear bottom plate with rear connector locations and support interface for the rear top plate.	25
9	Front top plate with manipulator mounting pattern, camera support interface and connector locations.	25
10	Rear top plate with NUC mounting pattern and rear connector interfaces.	26
11	Four-plate payload assembly with front and rear upper modules, support connectors and adapters. . .	27
12	Full CAD assembly showing the Leo Rover with front-mounted LiDAR, front top manipulator and RGB-D camera, and rear top NUC.	28
13	Static stress analysis setup for the revised payload architecture.	30
14	Von Mises stress distribution in the revised payload architecture under the applied load case.	31
15	Displacement distribution in the revised payload architecture under the applied load case.	31
16	Power Connections Diagram indicating Power Connection Type and Rating for the Hardware Architecture	33
17	Software Block Diagram	36
18	ROS2 runtime communication graph of the software system	37
19	Mission Manager	37
20	Trigger conditions governing mission state transitions	39

List of Tables

1	Summary of how feedback from the Preliminary Design Review was addressed in the Final Design Review.	6
2	Verification success criteria, design approach, evidence and compliance assessment.	9
3	Pugh-style comparison of LiDAR vertical placement options (“+” = better than reference, “0” = similar, “-” = worse).	14
4	Pugh-style comparison of LiDAR / manipulator / camera placement concepts. Reference: Design B, the front elevated grasping configuration (chosen).	15
5	Summary of static stress analysis results for the revised payload architecture (PLA, plate thickness 5 mm).	30
6	Summary Specification for the Anker Prime Power Bank (26K, 300W).	34
7	Component Identification: Part Name and System Role	35
8	Leo Rover Component Power Budget (Refer to Table 1 for Names and Uses)	35
9	Description of TF System Nodes	38
10	Description of Navigation System Nodes	40
11	Perception System Nodes Summary	40
12	Description of TF Bridge Layer Nodes	41
13	Description of Manipulation System Nodes	41
14	Description of Mission Management System Nodes	42
15	Verification success criteria, design approach, evidence and compliance assessment.	49

1 Introduction

1.1 Problem Statement and Objectives

This project concerns the design, integration and verification of a complete autonomous mobile manipulation system based on the Leo Rover platform. The Leo Rover provides the primary mobile base and low-level locomotion system, onto which an additional peripheral subsystem has been integrated, including a robotic manipulator, onboard computation and vision sensors. The overall aim is to develop a robot capable of operating autonomously in an indoor arena environment.

In the target demonstration, coloured blocks are randomly distributed within the arena and must be detected, approached, grasped and transported to the correctly colour-matched storage bins within a constrained demonstration time. To achieve this, the robot must combine several interacting capabilities: online mapping and localisation, autonomous navigation, obstacle avoidance, colour and object detection, grasp-pose estimation, manipulation, mission supervision, and safe system operation.

The project is inherently interdisciplinary, requiring coordinated development across mechanical, electrical and software subsystems. This creates a systems-engineering challenge in which the performance of the final robot depends not only on the quality of each individual subsystem, but also on the quality of their integration. As a result, clear requirements, verification criteria and interface definitions are essential in order to reduce ambiguity and ensure that the work of different subteams contributes coherently to the final mission.

This Final Design Review (FDR) presents the evolved baseline design of the system following the Preliminary Design Review (PDR). It summarises the finalised mechanical, electrical and software architectures, explains how the design has developed since the PDR, and evaluates the current system against the functional and performance requirements defined in the Design Requirements Analysis (DRA). The report therefore serves both as a record of design maturity and as a justification of the engineering decisions taken prior to final manufacturing, integration and demonstration.

In addition, this review identifies the main design modifications introduced since the PDR, summarises how earlier feedback has been addressed, and highlights any remaining risks, dependencies or unresolved details that may affect the final implementation stage.

1.2 Response to PDR Feedback

The Final Design Review (FDR) was developed directly in response to the feedback received during the Preliminary Design Review (PDR). Rather than treating the comments as isolated corrections, the feedback was reviewed thematically and used to strengthen the design justification, improve traceability to requirements, clarify verification evidence, and refine the presentation of the overall system. In general, the feedback highlighted that the project concept and overall design direction were strong, but that several areas required clearer engineering justification, particularly in relation to requirement linkage, manufacturability, sensor obstruction, software architecture, and verification clarity.

A key aim of the FDR has therefore been not only to improve the quality of the report, but also to show how the design itself has matured since the earlier review. In some cases, this involved improving explanation and traceability, for example by linking component choices more explicitly to functional and performance requirements. In other cases, it led to more substantive refinement of the system design, such as revisiting aspects of the payload architecture, expanding the discussion of sensor visibility and obstruction, and providing stronger justification for manufacturing and structural decisions.

Accordingly, the response to the PDR has been organised into a set of focused improvements. Areas that had already been assessed strongly in the PDR—such as the system block diagram, design philosophy, CAD quality, language specificity, requirement fulfilment, Gantt planning, and workload allocation—were retained and carried forward with only minor updating. By contrast, areas that were assessed as less well justified or less clearly presented were revised more substantially so that the final review provides a more complete and better-supported account of the proposed system. A summary of the main feedback themes and the resulting actions is provided in Table 1.

Overall, the PDR feedback led not only to improvements in presentation, but also to several genuine design refinements. The FDR therefore represents both a clearer report and a more mature system definition, with stronger requirement traceability, improved justification of subsystem choices, clearer verification planning, and a more robust final payload architecture.

Feedback Theme	PDR Concern	Action Taken in the FDR	Status
Requirements traceability and component justification	Component choices were not always explicitly linked to the relevant functional and performance requirements.	Component justification sections were revised so that each major subsystem is now linked explicitly to the corresponding FR and/or PR identifiers, improving engineering traceability and demonstrating requirement-led design decisions.	Resolved
Mechanical design refinement	Sensor obstructions were not sufficiently discussed, and the use of 3D printing for flat structural plates was not well justified.	A dedicated sensor-obstruction discussion was added, including the effect of the manipulator base, camera mount and support members on LiDAR and camera fields of view. The manufacturability section was also expanded to justify 3D printing as an appropriate low-volume prototyping method for feature-integrated structural parts. In addition, the payload architecture was revised from the earlier concept to the final four-plate design, with the depth camera moved to the front of the manipulator and the NUC moved to the rear top plate.	Resolved / strengthened
Electrical and power clarification	Two components were missing current requirements, and the proposed power solution lacked sufficient detail to be actionable.	The electrical design was updated to include the missing current requirements and provide clearer explanation of the external power strategy and implementation assumptions. Planned battery sleeves were also identified as part of the integration approach, although their detailed dimensions remain subject to final battery availability.	Resolved
Software architecture and mission logic	The RQT graph alone did not clearly communicate software structure, and the mission controller required more detail regarding trigger conditions. GitHub documentation was also limited in scope.	A dedicated software block diagram was added alongside the RQT graph to clarify control flow and subsystem interaction. The mission controller description was expanded to explain state transitions and trigger conditions in more detail. Git-based documentation, branching, versioning and code structure were also improved across the software modules.	Resolved / strengthened
Verification clarity and evidence presentation	Some tests were ambiguous, there were arguably too many of them, and it was unclear which success criteria had already been achieved.	The verification success criteria were revised to reduce ambiguity and focus on the most important demonstrations. The total number of tests was reduced, pass criteria were made clearer, and visual status indication was added so that completed, in-progress and pending verification items can be distinguished more easily.	Resolved
Project planning and reporting continuity	Most planning elements were already strong, but required updating to reflect current project progress.	Milestones, deliverables and timelines were reviewed and updated so that the FDR reflects the current project status rather than the original PDR baseline.	Updated

Table 1: Summary of how feedback from the Preliminary Design Review was addressed in the Final Design Review.

1.3 Key Modifications Since the PDR

Since the Preliminary Design Review (PDR), the design has undergone several important modifications as the system has matured through continued integration, testing and analysis. These changes affect the mechanical architecture, subsystem packaging, software design, verification strategy and project planning. The most significant modifications are summarised below.

Mechanical Design

- **Payload architecture revised from three plates to four plates:** The earlier three-plate payload sled concept was replaced by a four-plate architecture consisting of a front bottom plate, rear bottom plate, front top plate and rear top plate. This provides clearer separation between the sensing/manipulation subsystem at the front and the compute subsystem at the rear.
- **Manipulator and RGB-D camera moved to the front top plate:** The manipulator and RealSense camera are now colocated on the front top plate so that perception and grasping occur on the same operational side of the robot. This improves the direct perception-to-grasp workflow and reduces the need for unnecessary rover reorientation before pickup.
- **NUC relocated to the rear top plate:** The Intel NUC was moved from the front upper region to the rear top plate. This reduces congestion near the LiDAR and manipulator, improves payload balance, and reduces the risk of LiDAR occlusion by the compute unit.
- **Camera placement refined:** The RGB-D camera is now fixed at the front of the manipulator assembly so that it observes the same workspace into which the arm reaches. Angular adjustment is still retained through the gimbal mount.
- **Sensor obstruction explicitly analysed:** A dedicated discussion of sensor obstructions caused by the manipulator base, camera support structure and other components has been added. This includes consideration of LiDAR occlusion and the camera field of view in both nominal and extreme arm poses.
- **Manufacturability justification strengthened:** The rationale for using FFF 3D printing has been expanded. The payload plates are now justified as integrated mounting structures with custom interfaces, recesses, cut-outs and connector features, rather than being treated as simple flat plates.
- **Battery sleeve provision identified:** Sleeves for the two batteries have been planned as part of the integration strategy, although the detailed design remains dependent on battery availability.

Structural Analysis

- **FEA updated for the final architecture:** The structural analysis now reflects the revised four-plate payload architecture rather than the earlier concept.
- **Conservative loading assumptions introduced:** The final analysis uses a 2 kg manipulator-equivalent load and a 1 kg NUC-equivalent load, both intentionally higher than the actual component masses to provide design margin.

Electrical Design

- **Power solution clarified:** The external power strategy has been explained in greater detail so that the proposed implementation is more actionable. The Anker Prime Power Bank (26K, 300W) has been selected as a secondary power source.
- **Missing current requirements added:** The electrical documentation has been updated to include the missing current specifications for previously incomplete components.

Software Design

- **Software block diagram added:** A dedicated software architecture diagram has been included alongside the RQT graph to explain subsystem interaction and control flow more clearly.
- **Mission controller description expanded:** The mission-planning section now gives clearer detail on trigger conditions, state transitions and supervisory logic.

- **Object-detection approach revised:** The design direction was updated from a YOLO-based approach to a classical computer vision / blob-detection pipeline, which is more appropriate for the constrained arena and target set of the project.
- **Manipulator software modifications:** The software has switched from the simple open-loop movement of pymycobot package to the smart path planning of MoveIt2. It also adds a task queue with a Finite State Machine, smart diagonal error correction with auto-retry, a two-step hover-and-grab method, and an active clamping system to prevent drops.
- **Code organisation and documentation improved:** Git-based documentation, branching, versioning and general code structure have all been strengthened across the relevant software modules.

Analysis and Verification

- **Verification success criteria revised:** The verification matrix has been refined to reduce ambiguity and improve clarity of pass conditions.
- **Total number of tests reduced:** The test plan has been streamlined so that it focuses more clearly on the most important and achievable verification activities.
- **Status indication added:** A visual indication of test progress has been introduced so that completed, ongoing and pending verification items can be distinguished more easily.

Project Planning

- **Milestones and deliverables updated:** The project plan has been revised so that timelines, milestones and deliverable status reflect the current state of the project rather than the earlier PDR baseline.

Overall, the modifications since the PDR represent both a refinement of the report and a genuine maturation of the design itself. The final design is therefore more clearly justified, better integrated, more rigorously analysed, and more transparently verified than the earlier PDR version.

2 Sustainability Checklist

Table 2: Verification success criteria, design approach, evidence and compliance assessment.

Topic	Current Design Notes	Possible Improvements / Actions
Materials	Key materials: aluminium, PLA printed parts, PCBs, copper wiring, rubber wheels, and plastic housings. Main material used for the robotic frame is aluminium, which is durable, lightweight and fit for use in the LEO Rover, however has a significant environmental impact due to energy-intensive mining and smelting. Generally, all materials used are recyclable, however this may not be practised at end-of-life due to disposal limitations.	Further parts designed for integration with the main LEO Rover body should be recyclable, with minimal environmental impact from production where possible. Recycled materials may also be used such as recycled aluminium or recycled polymers. Further 3D printed parts should be designed to minimise material waste and be easily disbanded. Current lab practices do not include hard plastic recycling of 3D printing waste, which should be amended in future.
Software	Throughout the project, computational overhead has been reduced through software design. As an example, computer vision approaches have evolved from a deep learning model-based approach to using traditional computer vision methods, reducing reliance on heavy machine learning models. Cloud-based workflow has not been implemented, further reducing system computational complexity.	Computational overhead reduction should be considered throughout system design. Data should be filtered, compressed and processed locally where possible, with data transmission being consistently monitored. Algorithm design should be lightweight with restricted computation.
Energy	The LEO Rover on-board Li-Ion battery typically operates at 11.1V DC 5000mAh, with an additional battery power supply (Anker Prime Power Bank 26kmAh, Total Output Power 300W). While these provide reliable and non-exhaust methods of power, electricity generation for battery charging has not been monitored for environmental impact.	In further systems, component selection should be optimised for energy efficiency. Where suitable, renewable battery charging should be implemented, such as through solar-power assistance. Later system iterations should be optimised for low power consumption, such as through lower-power modes or power use monitoring.
Waste	Consistent prototyping throughout the project creates consistent material waste, predominantly through hard plastics. While uncommon, additional material waste such as worn tyres and damaged printed parts may need to be accounted for. E-waste may also occur in the case of non-functioning electrical components (servos, wires etc.) or Li-Ion battery component failure.	Simulation and 3D modelling should be used to reduce physical prototyping. Where possible, damaged parts should be repaired as opposed to replaced. Parts should not be manufactured solely for testing purposes. E-waste, batteries and material waste should be disposed of responsibly following standard practices.
Emissions	The LEO Rover relies predominantly on battery power which has no direct emissions through exhaust. Indirect emissions come from manufacturing and waste products (such as carbon and toxic “red mud” waste), as well as delivery emissions. Life cycle emissions may be produced from end-of-life recycling/disposal and logistics. Noise pollution is minimal and not significant given predominantly large indoor use.	Where possible, use renewable electricity for on-board energy production or battery charging. Use lower-impact and local suppliers (e.g. suppliers with offer carbon offset measures). Consider noise production during electrical component selection and control design.
Communications	Data transmission and storage is predominantly completed through on-board RAM with no further external data storage requirement. Some data transmission such as video streaming and LIDAR logging require additional power and bandwidth consideration. Overall, no further data transmission frameworks such as Cloud Storage are required. Data is predominantly communicated live and on-board, with no or minimal data storage (e.g. navigation maps are not stored locally and are communicated online).	Future systems should restrict additional data requirements, with data processing occurring predominantly on-board. Data should be stored only when required, with unnecessary data being filtered and deleted through on-board processes. Data transmissions should reduce reliance on raw stream or continuous logging where possible (e.g. through consistent use of event-triggered logging).
Modularity	Current payload sled design consists of multiple, modular and individually manufactured parts. In the case of part damage this means that individual parts can be replaced without full sled re-manufacture. Software components (computer vision, navigation, manipulation) have been produced independently prior to full system integration, meaning sub-systems software changes can occur without full system reconstruction.	In further payload design, parts should be standardised, easily accessible and replaceable. Modularity should be similarly replicated in software design with independent software modules or ROS nodes.
Location Placement	Logistic emissions from part delivery increase the project environmental footprint. While upfront logistical emission of the LEO Rover unit to the university is minimal due to the multi-year lifespan, transportation of machining materials such as 3D printing filament results in long-term environmental impact. Movement of the whole unit is minimal given activities (testing, construction) are localised to a single building. Due to rover weight and limited transportability, additional transportation measures such as elevator use need to be implemented. While minimal, this creates a long-term environmental impact due to electricity use.	Site-wide transport should be minimised where possible (e.g. localised to single floor) and activities should continue to be restricted to a single working area. Local and low-environmental-impact suppliers should be used for long-distance logistics; transportation carbon offsetting should be implemented where possible. System weight/volume reduction measures should be considered from project start.
Maintenance	Due to project workflow restrictions, testing is restricted to simulation and software testing for the majority of projects. Thereby, there are no standardised consistent maintenance protocols in place for hardware maintenance. For software maintenance, industry standard progress logging, such as through GitHub version control, has been implemented to prevent loss of data.	Future systems should incorporate on-board sensor monitoring, such as through temperature and current monitoring. Battery health checks should be completed between sessions to detect potential damage. On-board hardware inspections should be intermittently scheduled to detect faults, in addition to diagnostic software tool use. Any hardware faults should be documented and reported.
Repurposing	Main LEO Rover unit, excluding further manufactured parts, will be reused in later project iterations. Further manufactured parts largely consist of PLA/ABS 3D printed parts which will be disposed of through recycling where possible. Additional fastenings such as nuts and bolts will be returned to a communal resource sharing area located in the main workspace.	Secondary purposes for new manufactured parts should be considered throughout the project and avoid over-specialisation. Where possible, single-use fastenings or parts should be avoided. The type and quantity of different additional fastenings should be reduced or standardised and documented for accountability.

3 Cyber Security Considerations

Although this project is a laboratory robotic systems integration exercise rather than a system deployed on a public network, cyber security remains relevant because the robot operates as a distributed cyber-physical system. In this context, the main concern is not the protection of highly sensitive data, but the preservation of control integrity, safe operation, and recoverable system behaviour.

3.1 Network and Access Security

The robot relies on a wireless connection between the laptop and the on-board computers for supervisory control, monitoring and maintenance access. If this network is open, shared or poorly controlled, unauthorised access or communication interference may occur. The principal network risk is loss of operator authority: if the communication link degrades, pause, abort or recovery commands may not reach the robot in time, allowing navigation or manipulation to continue under unsafe conditions.

To reduce this risk, the robot should operate on a dedicated and controlled network, with non-default user accounts, restricted device access and only the minimum necessary exposed services. If communication quality becomes unacceptable, the system should pause the mission or enter a safe stationary state.

3.2 ROS 2 Command and Control Protection

ROS 2 topics, services, actions and runtime parameters directly affect mission sequencing, navigation goals and manipulator behaviour. They must therefore be treated as security-relevant control interfaces rather than purely internal software mechanisms. If invalid commands or unsafe parameter changes are accepted, the robot may begin tasks in the wrong state, navigate to unsafe locations, or move the manipulator to incorrect poses, leading to collisions, failed grasp attempts or dropped objects.

Mission commands should therefore only be accepted through a validated supervisory control path. Invalid state transitions, conflicting requests and unsafe parameter changes should be rejected and logged rather than executed.

3.3 Secure Configuration and Recoverable Operation

The Raspberry Pi and Intel NUC are general-purpose computing devices, so weak device management or poor configuration control may alter the effective behaviour of the robot. This creates the risk of operating a demonstration system that no longer matches the validated baseline.

To control this risk, the on-board devices should use non-default credentials, restricted administrator access and only essential services. Source code, launch files, calibration data and configuration parameters should remain under version control, and any critical fault should trigger a controlled pause or safe-state transition.

3.4 Monitoring, Logging and Incident Response

Cyber security in this project also includes timely detection of abnormal conditions, including communication loss, invalid requests, node failure and unexpected state transitions. If such faults are not recognised promptly, the robot may continue operating after its control chain has already been compromised.

The system should therefore maintain structured, time-stamped logs linked to the software version, configuration and runtime state of each run. When an incident is detected, the robot should enter an appropriate safe response mode, and logs should be preserved before restart or reconfiguration.

3.5 Summary

Overall, the cyber-security approach adopted for this project is centred on four priorities: network and access control, protection of mission-critical control interfaces, secure configuration of on-board computing devices, and effective monitoring and incident response. The proposed controls therefore focus on operating the robot on a secured network, restricting command authority to validated supervisory pathways, maintaining software and configuration baselines under controlled management, and ensuring that abnormal conditions are detected, recorded and handled safely. Cyber security is thus treated not as a separate IT issue, but as a practical systems-engineering mechanism for protecting hardware, maintaining mission reliability, and supporting credible final-system verification.

4 System Overview

The robotic system uses a distributed computing architecture. It is divided into two main subsystems: the Leo Rover Base System (Low-Level Control) and the Secondary Systems (High-Level Control). This design choice helps to divide the real-time hardware control from the resource-intensive AI processing. This ensures that heavy computing tasks, such as SLAM or object detection, do not cause latency in the robot’s motor response or stability.

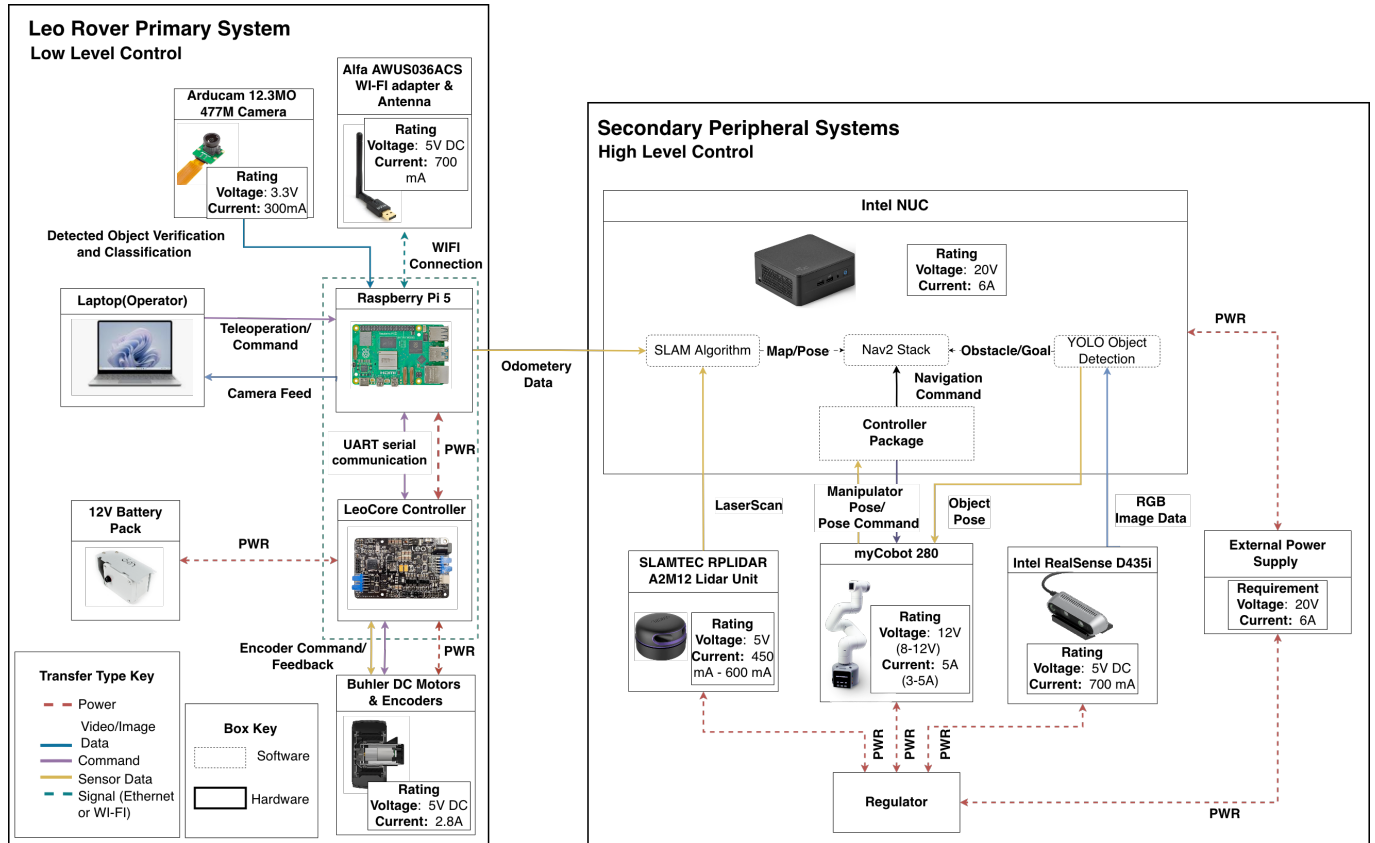


Figure 1: System-level hardware architecture of the autonomous object retrieval robot.

4.1 Leo Rover Primary System (Low-Level Control)

- **Laptop (Operator):** The Laptop serves as the remote command center and human-machine interface for the robotic system. Its main role is supervisory control. It receives a real-time video stream from the robot’s cameras. It also acts as the publisher for high-level state commands, allowing the operator to issue start, stop, and safety mode commands to the robot.

Justification: The inclusion of a remote operator laptop is essential for ensuring Human-in-the-Loop safety. It provides a safety override that allows the operator to remotely trigger an emergency stop ([FR-22], [PR-22.1]) and switch to safety or pause modes instantly ([FR-5], [PR-5.1], [FR-35]) if the autonomous system behaves unpredictably. Furthermore, it serves as the interface to issue the single start command to initiate the mission ([FR-34], [PR-34.1]).

- **Alfa AWUS036ACS Wi-Fi Adapter & Antenna:** This external USB adapter serves as the primary wireless network gateway for the entire system. It connects the internal network between the NUC and the Raspberry Pi, enabling real-time data transmission within the robotic system, and it also allows the external operator to access the system.

Justification: The Raspberry Pi’s internal antenna is insufficient to support the simultaneous transmission of heavy data, such as video streams and LaserScan data. Therefore, this adapter is necessary to provide a stable link capable of handling data transmission from multiple sub-components, which is critical for supporting the continuous export of run logs and diagnostics ([FR-32], [PR-32.1], [FR-33]).

- **Raspberry Pi 5:** The Raspberry Pi 5 acts as the essential bridge between the physical hardware and the high-level processor. It manages data transmission, encodes video streams from the Arducam, and communicates directly with the LeoCore controller.

Justification: The Raspberry Pi 5 has exclusive hardware interfaces, like CSI ports required by the Arducam. It is the only component capable of connecting to this camera to provide the operator with real-time video feedback, which is necessary for allowing safe operator intervention and repositioning ([FR-36]). Using the Raspberry Pi 5 also offloads Input/Output operations from the main NUC, ensuring the primary system runs efficiently.

- **LeoCore Controller:** The LeoCore acts as the dedicated low-level hardware driver for the mobile base. It sits between the high-level computing units Raspberry Pi and the physical actuators. It translates logical velocity commands received via the serial interface into specific pulse Width Modulation signals to drive the four Buhler DC motors.

Justification: Unlike main computers like NUC that can sometimes have small delays, the LeoCore reacts instantly and predictably, providing deterministic timing essential for high-frequency PID control loops. By checking the motor status every millisecond, it ensures the robot accurately tracks its velocity and enforces safe translational and rotational speed limits ([FR-24], [PR-24.1]). Additionally, it calculates raw wheel odometry data, which is a mandatory on-board sensor input for the localization subsystem ([FR-7]).

- **Buhler DC Motors & Encoders:** These units act as the propulsion system for the Leo Rover. Each wheel is driven independently by a Buhler DC motor. The integrated encoders provide locomotion and raw odometry data, which are indispensable for the mobile manipulation mission.

Justification: The integrated encoders are the primary source of proprioceptive feedback, mandatory for calculating wheel odometry used in on-board localization ([FR-7]). They enable the LeoCore to perform closed-loop velocity control, ensuring the robot strictly maintains the limited and safe operational speeds ([FR-24], [PR-24.1]). The motors ensure the robot can drive smoothly across the flat indoor arena floor ([FR-26], [PR-26.1]).

- **Arducam 12.3MO 477M Camera:** Connected via the CSI interface to the Pi 5, this camera is dedicated to route design and teleoperation. It provides a low-latency video stream to the operator, providing visual monitoring that is independent of the autonomous navigation system.

Justification: The Arducam's high resolution and clarity are essential for route design. They allow the operator to clearly distinguish details on the ground. Connected to the Raspberry Pi, the camera maintains visual awareness independently of the main NUC. This reliable backup ensures the operator has the necessary environmental feedback to safely intervene, reposition the robot, or drive it back manually if required ([FR-36]).

- **12V Battery Pack:** This serves as the primary energy source for the Leo Rover Base System. It powers the Leo Rover base and the Raspberry Pi by stepping down to 5V via the regulator.

Justification: The battery pack supplies a stable step-down voltage to the Raspberry Pi and maintains a consistent output even as battery capacity decreases. It is crucial for preventing system crashes and ensures the primary control system has sufficient, uninterrupted power to complete the full 20-minute autonomous mission window ([PR-1.1]).

4.2 Secondary Peripheral Systems (High-Level Control)

- **Intel NUC:** The NUC serves as the system's brain, enabling autonomous capabilities. It hosts the ROS 2 Nav2 stack, SLAM algorithms, YOLO object detection, and manipulator control commands.

Justification: The inclusion of the NUC is due to the requirement for substantial computational power to handle autonomous mission execution ([FR-1]), online map generation ([FR-8]), and real-time image inference for target detection ([FR-12]), which exceeds the capabilities of standard hardware. The NUC ensures that algorithms such as YOLO and SLAM run at a high frame rate without lag, which is critical for the system's overall performance.

- **SLAMTEC RPLIDAR A2M12 Lidar Unit:** The RPLIDAR A2M12 is a 360-degree 2D laser range scanner, which acts as the primary exteroceptive sensor for the autonomous navigation system. Connected to the Intel NUC, it sends high-frequency laser-scanning data to measure distances to surrounding objects. This data is the main input used by the SLAM algorithms to build maps.

Justification: Unlike the cameras, which have a limited field of view, the LiDAR provides a continuous 360-degree horizontal scan. This omnidirectional coverage is necessary for creating complete occupancy grid maps without a

pre-provided map ([FR-6]). It ensures the reliable detection of static obstacles within the sensing envelope ([FR-28], [PR-28.1]) and supports safe path planning to avoid contact ([FR-29], [PR-29.1]), especially when the robot moves backward or rotates.

- **Intel RealSense D435i:** It connects directly to the Intel NUC via USB 3.0; the RealSense D435i serves as the main sensor for the robot arm. It provides two streams of data at the same time: high-quality RGB images for the YOLO system to recognize objects, and Depth maps for calculating distances.

Justification: The inclusion of this RGB-D camera is mandatory for the pick-and-place task because it provides high-quality RGB images for color classification of objects and bins ([FR-13], [FR-19], [FR-20]) alongside 3D spatial localization via depth maps. This critical Z-axis information allows the system to accurately estimate a pickup pose for the target object ([FR-14], [PR-14.1]) and plan a valid trajectory for the mechanical arm.

- **myCobot 280:** The myCobot 280 serves as the main manipulation subsystem for the robot. It is a 6 degrees of freedom serial robotic arm attached to the Leo Rover's body. It serves as the physical component that executes joint trajectory commands sent by the Intel NUC.

Justification: The arm is mandatory for the action part of the mission: moving the end-effector to a defined pre-grasp pose ([FR-15], [PR-15.1]), and subsequently grasping, lifting, and carrying objects found by the vision system ([FR-16], [PR-16.1]) without causing permanent damage to them ([FR-17], [PR-17.1]). Its 6 degrees of freedom meet the system's requirement for arbitrary pose adjustment to successfully place or drop the target object through the bin opening ([FR-21], [PR-21.1]).

- **External Power Supply:** This is an independent power supply dedicated exclusively to the Intel NUC, myCobot, lidar and depth camera.

Justification: The standard 12V battery pack provided with the Leo Rover is designed only to support the energy needs of the DC motors and basic controllers. Therefore, an external power supply is required to handle the extra equipment (NUC, myCobot, lidar, camera), ensuring the secondary peripheral systems have sufficient power to execute the complete autonomous mission within the designated timeframe ([FR-1], [PR-1.1]).

5 Mechanical Design

5.1 Mechanical Design Objectives

The mechanical design has been driven by the following high-level objectives:

- Provide structurally sound, reversible mounting for all additional hardware (LiDAR, RGB-D camera, on-board compute) in line with FR-37–FR-44.
- Preserve, as far as possible, the original Leo Rover footprint and arena clearance limits (FR-37, PR-39.1).
- Maintain sufficient fields of view for perception sensors, particularly the LiDAR and RGB-D camera (FR-40, PR-40.1, PR-37.1).
- Keep the centre of mass low and well within the wheelbase to minimise the risk of toppling during manoeuvres.
- Allow non-destructive access to the Leo Rover internals (battery and power electronics), consistent with FR-38 and maintainability requirements.

To achieve this, a three-plate payload sled architecture has been developed, combined with a set of custom adapters, connectors and holders. The key design decisions and trade-offs are summarised in the following subsections.

5.2 Key Design Challenges and Decisions

5.2.1 Placement of the LiDAR Sensor

The LiDAR sensor is a primary source of range data for localisation, mapping and obstacle detection (supporting FR-7, FR-8, FR-28–FR-30 and PR-6.1, PR-8.1, PR-26.1, PR-28.1). Two constraints dominated the design:

- The LiDAR should be mounted as low as reasonably possible so that it can see close-range obstacles and the storage bins, reducing the risk of undetected collisions.
- The LiDAR rays should be minimally occluded by other robot components to retain the benefit of a near-360° scan.

Conceptually, there were two main options:

- (a) Mount the LiDAR above all other payloads (high position).
- (b) Mount the LiDAR directly on the Leo Rover base and place other payloads on a plane above it (low position).

A high mounting position would have simplified mechanical integration and kept the centre of mass low, but raised the issue of reduced sensitivity to low obstacles and bin edges due to the beam height.

Conversely, placing the LiDAR low on the base plate improves bin and obstacle visibility but forces other payloads (notably the manipulator and NUC) onto a higher plane, and introduces potential occlusion of the LiDAR scan by the arm if they share the same level.

Table 3 summarises this trade-off using a simple Pugh-style comparison. The low-mounted LiDAR configuration is used as the reference option.

Criterion	LiDAR high on top	LiDAR low on base (chosen)
Bin detection at close range	–	+
Detection of low obstacles	0	+
Risk of collision with bins	–	+
Centre of mass height	+	–
Mechanical integration complexity	0	0
Future payload mounting area	+	0

Table 3: Pugh-style comparison of LiDAR vertical placement options (“+” = better than reference, “0” = similar, “–” = worse).

Given the strong advantages in obstacle and bin detection, the LiDAR is mounted directly on the base. Other payloads are then handled via the dedicated payload sled above the LiDAR. This does sacrifice some potential mounting area, but the downstream sections show how the sled design mitigates this.

5.2.2 Placement of Manipulator Arm and Cameras with Respect to the LiDAR

With the LiDAR position fixed at the base level, the next design decision concerned the relative placement of the manipulator and cameras. In particular, the key question was whether the grasping subsystem should be positioned on the *opposite side* of the rover from the LiDAR, or whether it should be placed on the *same operational side* as the LiDAR but elevated onto an upper plate.

From a purely stability-driven perspective, one would ideally prefer to mount the manipulator as low as possible in order to keep the centre of mass low. However, the manipulator must also be positioned so that its workspace is useful for pickup, while the cameras must maintain continuous visibility of the target during the final stages of approach and grasping. The Leo Rover is geometrically and functionally symmetric, so either end can in principle be treated as the “front”; the main issue is therefore the relative placement of the perception and manipulation subsystems with respect to the LiDAR.

In parallel, the robot is equipped with two cameras:

- A forward-facing RGB camera (no depth) already provided on the Leo platform.
- A secondary Intel RealSense RGB–D camera providing colour and depth information.

The RGB camera can be used to coarsely detect coloured blocks, but depth information is required to estimate grasping distance and refine the final approach pose. Therefore, the RealSense camera should be colocated with the primary direction of object pickup so that the target remains in view during the final approach and grasp.

Based on this, two candidate configurations were considered.

Design A: Rear grasping configuration (earlier concept) In the earlier concept, the LiDAR is mounted low at the front of the rover, while the manipulator and RealSense camera are mounted at the rear. The existing front RGB camera is then used for coarse target spotting. If a target is detected at the front, the rover performs a 180° rotation to bring the object into the RealSense field of view and manipulator workspace at the rear.

This arrangement has the advantage that the manipulator does not share the same operational side as the LiDAR, thereby minimising the risk of arm-induced LiDAR occlusion. It also keeps the grasping subsystem mechanically separated from the front sensing stack. However, it introduces an extra operational step before pickup, since front-detected objects must first be reoriented into the rear grasping workspace.

Design B: Front elevated grasping configuration (chosen concept) In the revised concept, the LiDAR remains mounted low at the front of the rover, but the manipulator and RealSense camera are relocated on the same operational side as the LiDAR but on a higher level. The RGB–D camera is positioned in front of the manipulator arm, so that the target can be observed continuously during the approach and grasping process.

This arrangement creates a vertically separated front sensing and grasping stack: the LiDAR remains responsible for low-level ranging and obstacle detection near the ground, while the camera–manipulator pair operates above it for perception-guided pickup. The main benefit is that a target detected in front of the rover can now be approached and grasped directly, without first rotating the entire platform.

The main drawback is that the manipulator now occupies the same broad operational side as the LiDAR, so the mechanical packaging is more demanding and some care is required to minimise possible scan occlusion from the elevated arm structure and supporting members.

Table 4 summarises the trade-off between these two concepts.

Criterion	Design A	Design B (chosen)
LiDAR occlusion by arm	–	+
LiDAR isolation from manipulator	+	+
Direct front-side pickup workflow	–	+
Continuous RGB–D view during grasp	+	+
Use of existing front RGB camera	+	–
Need for 180° rover rotation before pickup	–	+
Mechanical packaging simplicity	+	–

Table 4: Pugh-style comparison of LiDAR / manipulator / camera placement concepts. Reference: Design B, the front elevated grasping configuration (chosen).

Although Design A provides better isolation between the manipulator and the LiDAR, Design B is ultimately preferred because it better supports the intended task sequence. In Design B, the robot can detect an object, approach it from the front, maintain depth-based visual feedback throughout the approach, and execute the grasp without first reorienting the entire rover. This reduces operational complexity during pickup and gives a more direct alignment between perception and manipulation.

For these reasons, the selected configuration is:

- **LiDAR at the front**, mounted low on the base.
- **Manipulator arm on the front top plate**, above the LiDAR.
- **RealSense RGB–D camera in front of the manipulator**, mounted on the same front top structure.

The consequence of this decision is that the front upper support structure must be sufficiently stiff to support the manipulator and RGB–D camera without excessive deflection, loss of alignment, or instability during arm motion and object pickup. This requirement is addressed in the thickness selection and structural analysis described later.

5.2.3 Placement of the On-board Compute (NUC)

Once the LiDAR, manipulator and cameras are accounted for, the remaining major payload is the on-board compute unit (Intel NUC). Under the chosen configuration, the manipulator arm and RealSense camera are mounted on the **front top plate**, directly above the LiDAR region. This makes the front upper structure the primary perception and manipulation module, and therefore it is undesirable to also place the NUC in the same region, as this would increase congestion, raise the front-side mass further, and reduce the flexibility of cable and component placement.

The base of the Leo Rover offers only a limited number of mounting points, many of which are already occupied by the LiDAR adapter, manipulator interfaces and existing hardware. Therefore, rather than relying exclusively on the native mounting points, a dedicated **rear top plate** is introduced to carry the NUC, with the rear bottom plate acting as its supporting lower structure.

Mounting the NUC on the rear top plate has several advantages:

- The front upper plate is reserved for the grasping subsystem, reducing packaging complexity around the manipulator and RGB–D camera.
- Moving the NUC rearward helps counterbalance the mass of the manipulator and camera at the front of the rover, thereby reducing excessive forward centre-of-mass shift.
- Relocating the NUC to the rear top plate avoids placing another bulky component above the LiDAR, thereby helping to preserve the LiDAR field of view and reducing the risk of scan occlusion.
- Cables from the NUC to the front sensing and manipulation hardware can still be routed cleanly along the payload structure, while keeping the compute module physically separated from the most crowded part of the system (supporting FR-42 and PR-39.1).
- Access to USB, power and Ethernet ports is preserved, and ventilation slots can be integrated directly into the adapter.
- The split front–rear top architecture improves modularity, since the compute unit can be mounted, removed or revised independently of the front manipulation module.

The consequence of this decision is that the rear upper support structure must be stiff enough to carry the NUC without excessive deflection, while the front upper support structure must be designed primarily for the more demanding manipulator and camera loading case. This load distribution is addressed in the thickness selection and structural analysis described later.

5.3 Payload Sled Architecture

To decouple payload mounting from the limited Leo Rover interface points, a four-plate payload architecture has been adopted:

- **Front bottom plate**: mounted over the front section of the rover and used to support the LiDAR adapter and the lower front structural interfaces.

- **Rear bottom plate:** mounted over the rear section of the rover and used primarily as the lower support structure for the rear top plate.
- **Front top plate:** mounted above the front bottom plate via custom connectors and used to carry the manipulator arm and RGB-D camera assembly.
- **Rear top plate:** mounted above the rear bottom plate via custom connectors and used to carry the Intel NUC and associated cable-routing interfaces.

This four-plate arrangement creates a clear functional separation between the main subsystems of the robot. The **front module** combines low-level ranging at the base with elevated perception-guided manipulation above it: the LiDAR remains mounted low for obstacle and bin detection, while the manipulator and RGB-D camera are positioned on the front top plate for direct front-side pickup. The **rear module** carries the on-board compute hardware, helping to balance the upper payload distribution while keeping the compute unit separated from the more congested sensing and grasping region.

The use of separate front and rear lower plates remains intentional. The Leo Rover layout requires periodic access to the battery and power electronics, and the split lower structure allows this to be achieved without dismantling the entire payload assembly. Similarly, dividing the upper structure into front and rear top plates improves modularity, reduces unnecessary spanning between front and rear subsystems, and allows the manipulation and compute modules to be designed, manufactured and revised more independently.

This arrangement is consistent with FR-38 (reversible, non-destructive modifications) and the maintainability requirements, while also supporting FR-39 to FR-42 through improved mounting flexibility, cable routing and subsystem packaging.

Figure 2 illustrates the overall arrangement of the four plates and their relationship to the rover chassis and payloads.

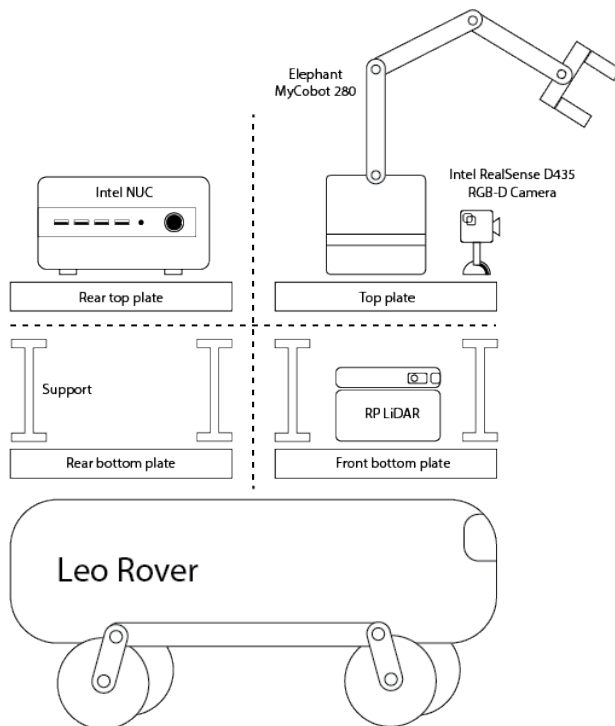


Figure 2: Conceptual layout of the front bottom plate, rear bottom plate, front top plate and rear top plate.

5.4 Selection of Plate Thickness

Plate thickness was driven by two coupled considerations:

1. Compatibility with the existing fasteners used on the Leo Rover.
2. Structural stiffness and deflection limits when loaded by the NUC and other payloads.

Fastener Compatibility

To minimise complexity and avoid introducing multiple fastener standards, the design reuses the screws already employed in the rover assembly: **M5** × **10** screws with a head height of approximately 2.75 mm, nominal diameter 5 mm and head diameter 9 mm.

For many payloads (e.g. the LiDAR adapter) it is desirable to insert screws from the underside of the bottom plates, as there are no factory holes at the required positions on the rover, and customised placement is needed. The plates themselves are then fixed to the rover using fasteners from above at dedicated mounting locations.

Simply resting the plates on exposed screw heads was rejected, as this would create local pressure points and lead to unpredictable deformation. Instead, the screw heads must be recessed within the plate thickness, while still leaving sufficient material above the recess to carry load.

The reasoning is as follows:

- Screw head height: ≈ 2.75 mm.
- Minimum material thickness above the head recess to avoid cracking and provide a stable contact surface: ≥ 2 mm.

This leads directly to a minimum plate thickness of roughly $2.75 \text{ mm} + 2 \text{ mm} \approx 4.75 \text{ mm}$. For simplicity, tolerance and print robustness, the plate thickness is standardised at **5 mm** for all sled plates and structural adapters.

Strength and Deflection Targets

In parallel with the fastener considerations, the plates must be stiff enough to support the mounted hardware without visible sagging or compromising sensor alignment. The most critical component in this regard is the **top plate**, which is supported on four discrete I-beam connectors and carries the NUC.

Design targets were set as:

- Minimum factor of safety (FoS) against material yield: **FoS** ≥ 4 .
- Maximum allowable vertical deflection at the top plate support surfaces: $\delta_{\max} \leq \mathbf{0.5 \text{ mm}}$.

A parametric model was created with the plate thickness as a design variable, and static stress analyses were performed for candidate thickness values using PLA properties. The final FEA results for the 5 mm design are summarised in Section 5.13; in brief, the achieved minimum FoS is approximately 15, significantly exceeding the target, and deflections remain well below the 0.1 mm limit.

5.5 LiDAR Adapter

The LiDAR is mounted using an adapter derived from the official Fictionlab design for the RPLIDAR A2, modified to suit the current payload sled. Non-essential details were removed to reduce print time and simplify the geometry, while preserving the critical mounting interfaces and clearances.

The adapter is designed to:

- Maintain the LiDAR unit as low as possible above the base plate, consistent with its scanning envelope.
- Present a flat interface to the front bottom plate with standardised M5 fastener patterns.
- Avoid occluding the LiDAR scan with unnecessary vertical structures.

The key dimensions of the adapter and its mounting pattern are shown in Figure 3.

5.6 Connectors Between Top and Bottom Plates

5.6.1 I-beam Connector Design

Four custom I-beam connectors link the front and rear bottom plates to the top plate. These connectors are responsible for transmitting the NUC load and any dynamic loads from the upper payloads into the base structure.

The connectors share the 5 mm wall thickness used throughout the sled to simplify manufacturing and analysis. Their remaining dimensions are driven by fastener geometry and required clearances:

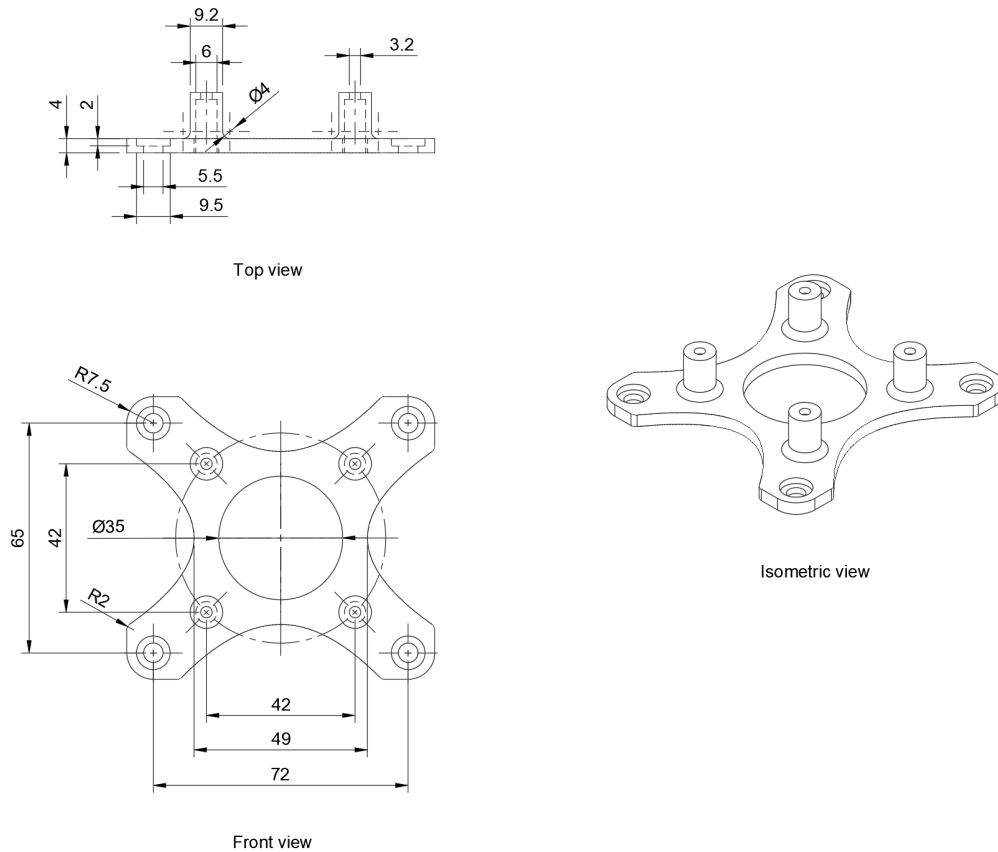


Figure 3: Customised RPLIDAR A2 adapter mounted on the front bottom plate.

- The through-hole for the M5 screw is modelled at 9.5 mm diameter to allow for a 0.5 mm tolerance over the 9 mm head diameter, accommodating 3D printing tolerances.
- A nominal clearance of 1 mm is provided on either side of the screw head within the connector profile, giving a total connector width of $9.5 \text{ mm} + 2 \times 1 \text{ mm} = 11.5 \text{ mm}$.
- The connector length in the lateral direction is sized similarly, resulting in an overall planform of approximately 11.5 mm by 27 mm to comfortably enclose the fastener and provide sufficient bearing area.

The connector height is determined by the required gap between the bottom and top plates. This gap is kept as small as practical: just enough to prevent the top plate from contacting the LiDAR body during operation, while still accommodating any necessary wiring.

Figure 4 shows the resulting I-beam connector geometry.

5.6.2 Connector Placement

To balance load distribution and preserve LiDAR visibility, four connectors are used:

- Two connectors are placed on either side of the LiDAR on the front bottom plate. This provides a clear 180° unobstructed sector for the LiDAR in the forward direction for obstacle detection, while still supporting the front portion of the top plate.
- Due to limited free space on the front bottom plate (occupied by the LiDAR and its adapter), the remaining two connectors are anchored to the rear bottom plate. This arrangement maintains adequate stiffness of the top plate while leaving the central region above the LiDAR largely unobstructed.

This layout provides reasonably symmetric support while prioritising sensor fields of view, in line with FR-40 and PR-40.1.

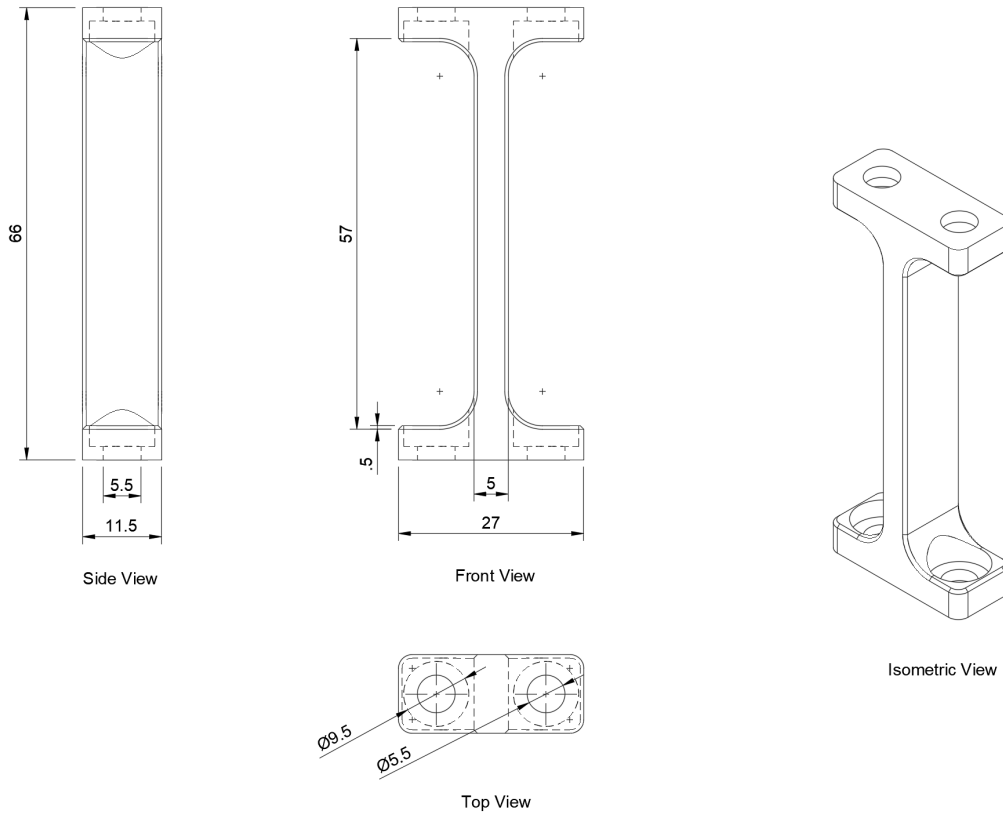


Figure 4: Custom I-beam connector between bottom and top plates.

5.7 NUC Holder / Adapter

The NUC holder is intentionally straightforward, reflecting the largely cuboidal shape of the compute unit. The adapter is designed as a lightweight cradle with:

- 5 mm wall thickness, matching the plates and connectors.
- Internal dimensions offset by ≈ 0.5 mm in both length and width relative to the measured NUC envelope to account for 3D printing tolerances and allow for easy installation.
- Ventilation slots and access cut-outs positioned to avoid blocking the NUC's existing airflow paths and to expose power, USB and network ports.

The NUC is mounted to the rear top plate via this adapter using M5 screws, and an evenly distributed load is assumed over the adapter's contact surface for the structural analysis in Section 5.13.

Figure 5 illustrates the NUC adapter and its measurements.

5.8 Camera Adapter and Integration with the Manipulator

In the final design, the RealSense RGB-D camera is mounted at the **front of the manipulator assembly** on the front top plate. Unlike the earlier concept, the camera mounting **position** is no longer treated as an open design variable; instead, it is intentionally placed at the front of the manipulator so that the camera observes the primary pickup direction and the target remains within view during the final stages of approach and grasping.

This placement is chosen so that the perception and manipulation subsystems are closely aligned. By locating the camera directly in front of the manipulator, the camera observes the same workspace into which the arm reaches, improving depth-based localisation of the target and reducing the likelihood that the arm base obscures the object during approach.

The camera is mounted using a dedicated front bracket integrated into the front top plate / manipulator support structure. The adapter is designed to:

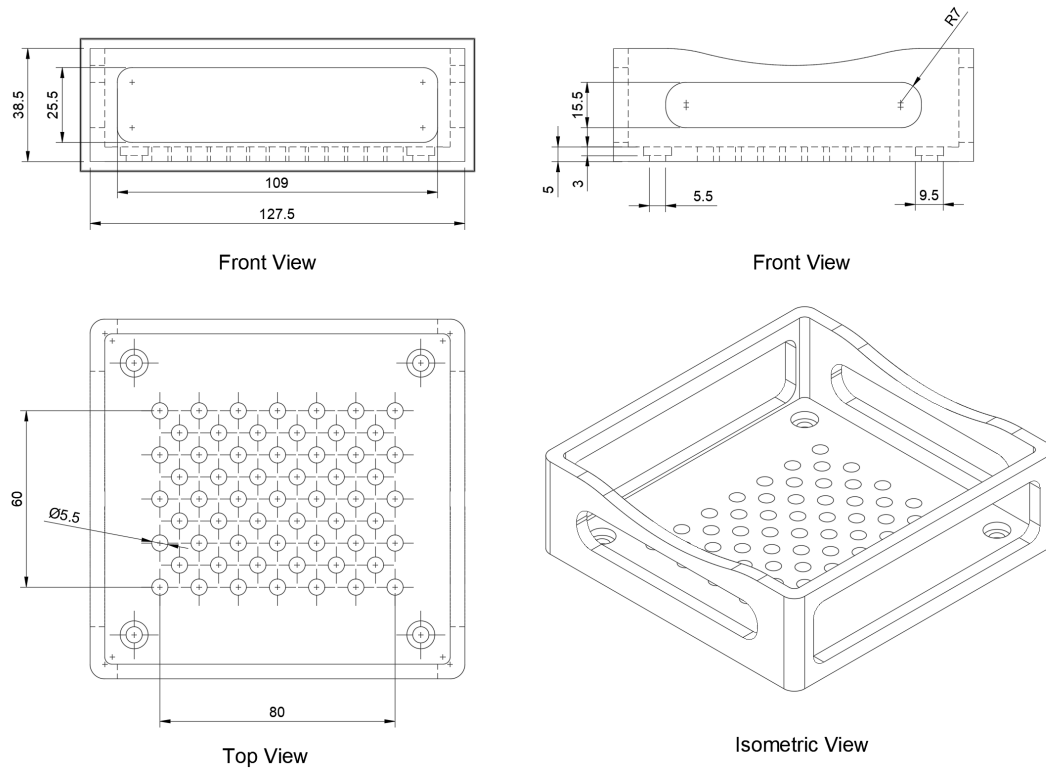


Figure 5: NUC holder / adapter.

- Position the RGB-D camera directly ahead of the manipulator along the primary grasping direction.
- Provide a rigid and repeatable mounting location so that the camera remains aligned with the manipulator workspace.
- Maintain sufficient clearance from the manipulator links and surrounding structures so that the camera field of view is preserved.
- Allow the camera cable to be routed cleanly back toward the rear compute module without interfering with the manipulator workspace.

Although the camera mounting location is fixed, angular adjustment is still retained through the gimbal mechanism provided with the camera tripod. This allows the camera tilt to be tuned experimentally to refine the field of view and optimise visibility of the grasping region, without changing the fundamental front-of-manipulator placement.

This combination of fixed translational placement and adjustable angular orientation is beneficial for both integration and calibration. The camera remains mechanically tied to the manipulator workspace in a repeatable way, while still allowing sufficient flexibility to optimise the viewing angle during testing.

Figure 6 shows the camera adapter concept and its integration at the front of the manipulator assembly.

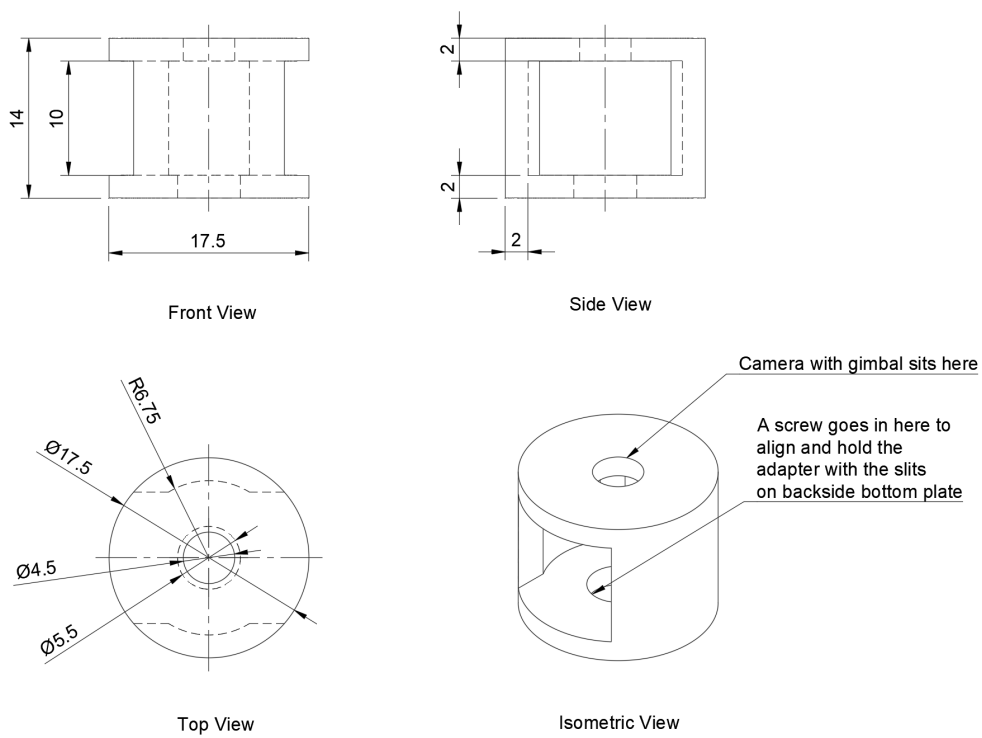


Figure 6: RGB-D camera adapter mounted at the front of the manipulator assembly on the front top plate, with angular adjustment provided by the gimbal.

5.9 Sensor Obstructions and Field-of-View Considerations

Because the final payload architecture places the manipulator and RGB–D camera on the front top plate above the front-mounted LiDAR, potential sensor obstruction by structural and functional components was treated as an explicit design consideration.

LiDAR Obstruction Considerations

The principal risk to the LiDAR field of view arises from components mounted above or near its scan region, particularly the manipulator base, camera support bracket, upper support connectors and any bulky payloads located in the front upper module. If not considered carefully, these components could block portions of the LiDAR scan and reduce its effectiveness for low obstacle detection, bin-edge detection and local navigation.

To reduce this risk, the following measures were adopted:

- The LiDAR is mounted as low as possible on the front bottom plate, so that its scan plane remains close to the ground and below most of the upper payload structure.
- The manipulator is mounted on the front top plate rather than directly on the base, introducing vertical separation between the LiDAR and the manipulation subsystem.
- The NUC is relocated to the rear top plate, preventing an additional bulky component from being placed in the field of view of the LiDAR and thereby reducing unnecessary scan occlusion.
- The front support connectors are positioned away from the most critical forward sensing sectors as far as practicable.

Although some localised LiDAR occlusion from the manipulator support structure and associated hardware is unavoidable, this was judged acceptable because the most important forward-facing low-level sensing region is preserved to a useful extent.

RGB–D Camera Obstruction Considerations

The RealSense RGB–D camera must retain a clear view of the pickup region during final approach and grasping. The main possible obstructions to this camera are the manipulator base, the arm links themselves, the camera mounting bracket, and nearby structural members.

To minimise these effects, the camera is mounted at the **front of the manipulator assembly**, so that it observes the grasping region from ahead of the arm rather than from behind it. This placement reduces the likelihood that the manipulator base obstructs the target during approach. In addition, the gimbal mount allows the camera angle to be adjusted experimentally so that the field of view can be tuned to the most useful grasping region without changing the fundamental camera location.

Some partial occlusion by the manipulator links may still occur during certain arm poses, particularly near full extension or during object pickup. However, this is an inherent trade-off in compact integrated manipulation systems and was considered acceptable given the benefits of maintaining a direct front-side perception-to-grasp workflow.

Overall Assessment

The final design does not eliminate all sensor obstructions completely; rather, it manages them through careful packaging and subsystem placement. The LiDAR retains useful low-level coverage for navigation and obstacle detection, while the RGB–D camera maintains a direct view of the grasping workspace. On this basis, the residual obstruction effects were considered acceptable within the constraints of the Leo Rover platform and the selected four-plate payload architecture.

5.10 Plate Designs

The four plates—front bottom, rear bottom, front top and rear top—are each customised with cut-outs, holes and connector interfaces to accommodate the payloads while remaining compatible with the existing Leo Rover mounting constraints:

- **Front bottom plate:** features mounting holes for the LiDAR adapter, holes matching the rover’s factory mounting pattern, and connector interfaces for the front upper structure, while preserving the required clearances around the LiDAR and access to the rover body.

- **Rear bottom plate:** acts primarily as the support interface for the rear top plate and includes the required mounting holes for the rear connector set, while still preserving access to the rover battery and power electronics.
- **Front top plate:** includes mounting points for the manipulator base, the front-mounted RGB-D camera support structure, and the connector interfaces linking it to the front bottom plate.
- **Rear top plate:** provides mounting points for the NUC adapter and the rear support connectors, with sufficient edge clearance so as not to exceed the original rover footprint (satisfying PR-39.1).

Together, these four plates create a modular payload architecture in which the front pair of plates supports the sensing and manipulation subsystem, while the rear pair supports the compute subsystem. This division reduces congestion between hardware groups and allows each module to be designed around its own functional and structural requirements.

Figures 7–10 present the final plate designs with principal dimensions.

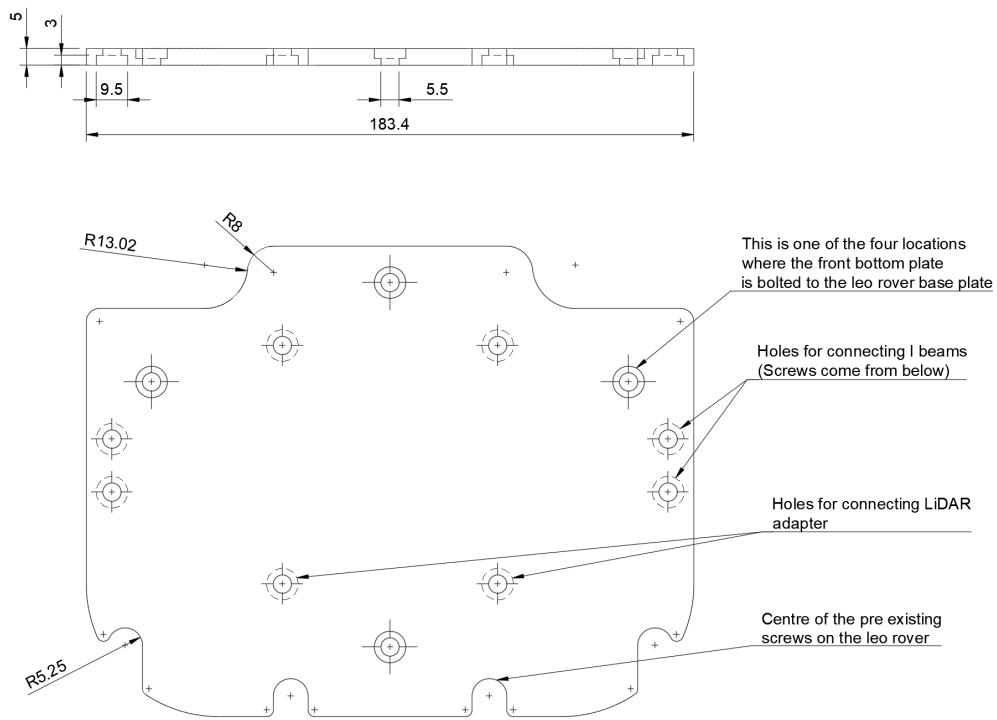


Figure 7: Front bottom plate with LiDAR adapter mount holes and front connector locations.

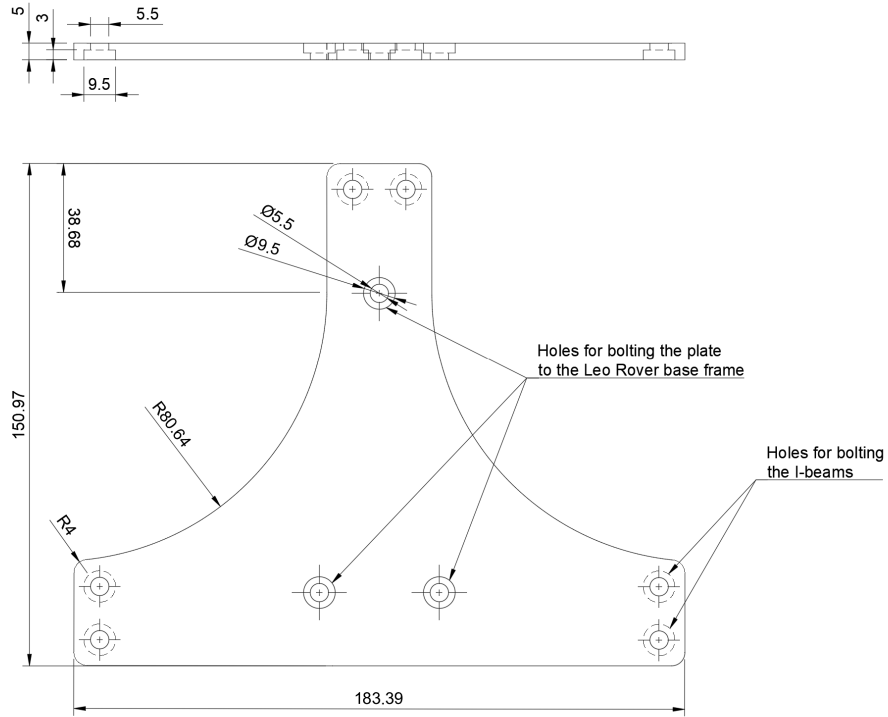


Figure 8: Rear bottom plate with rear connector locations and support interface for the rear top plate.

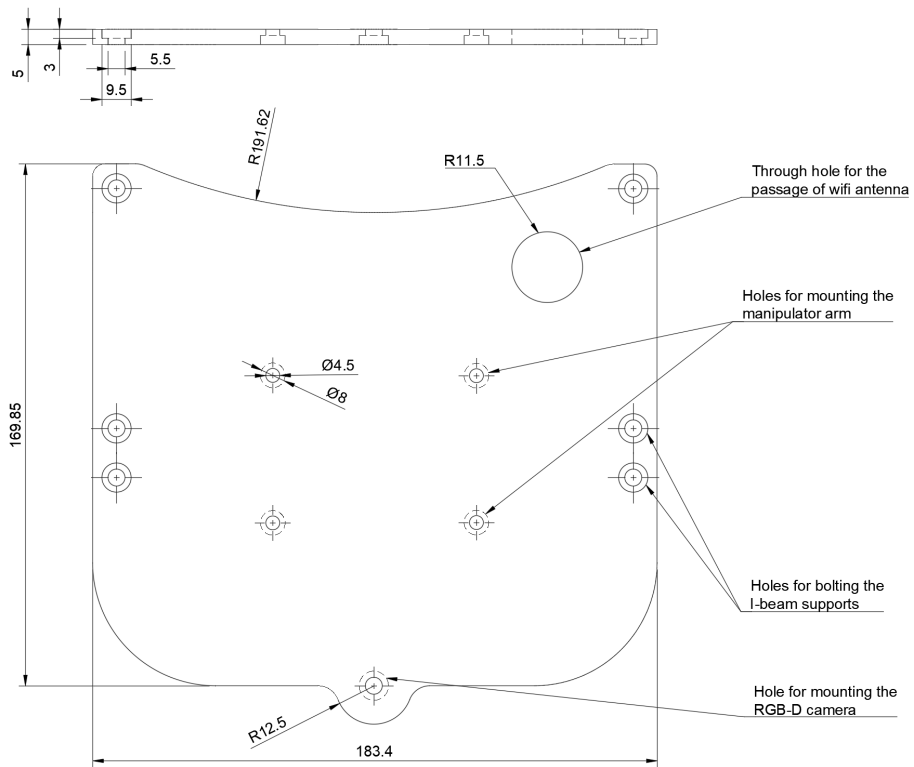


Figure 9: Front top plate with manipulator mounting pattern, camera support interface and connector locations.

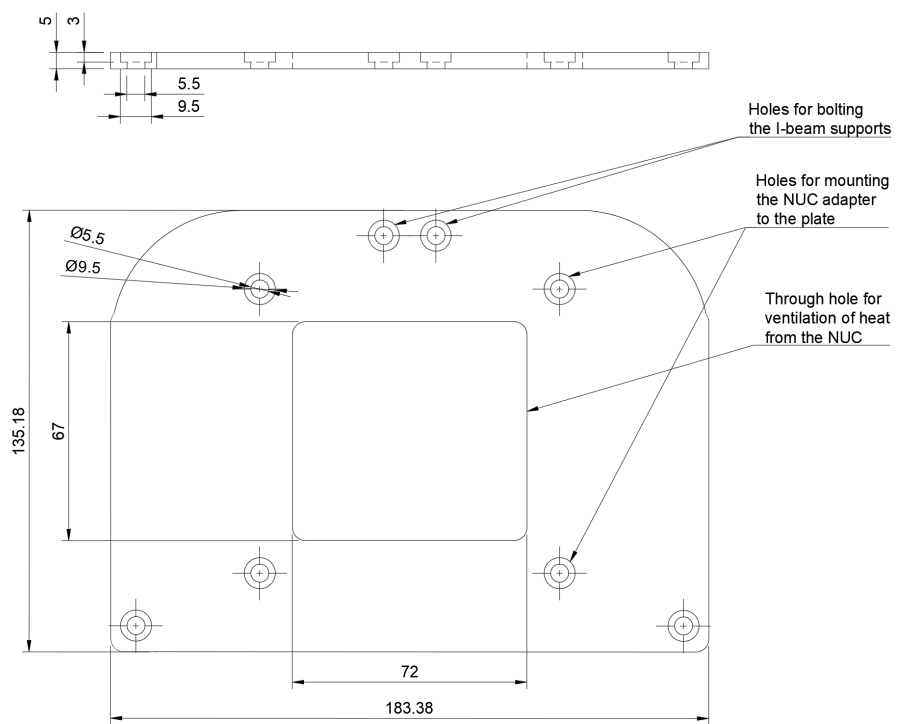


Figure 10: Rear top plate with NUC mounting pattern and rear connector interfaces.

5.11 Full Assembly

The full payload assembly, including the front bottom plate, rear bottom plate, front top plate, rear top plate, support connectors, LiDAR adapter, NUC adapter and camera adapter, is shown in Figure 11. In the final arrangement, the LiDAR is mounted low at the front, the manipulator and RGB-D camera are mounted on the front top plate, and the NUC is mounted on the rear top plate.

When combined with the Leo Rover base, the resulting structure forms a compact and modular payload architecture. The front module integrates low-level ranging with elevated perception-guided manipulation, while the rear module supports the compute hardware and helps balance the payload distribution. The complete assembly remains within the arena clearance limits and preserves access to safety-critical interfaces such as the emergency stop button and power controls.

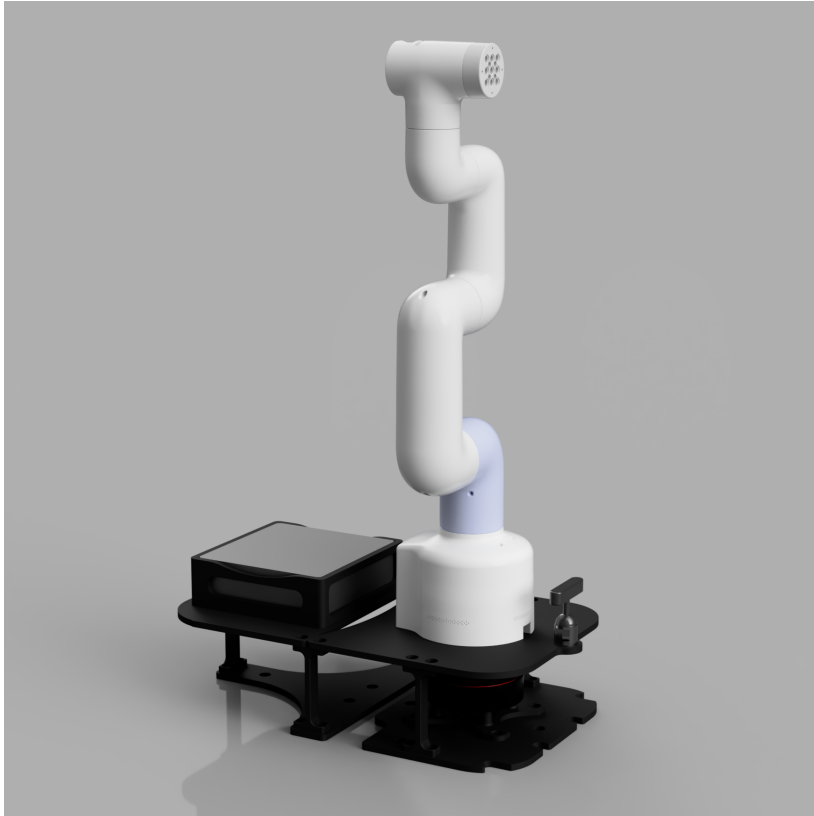


Figure 11: Four-plate payload assembly with front and rear upper modules, support connectors and adapters.

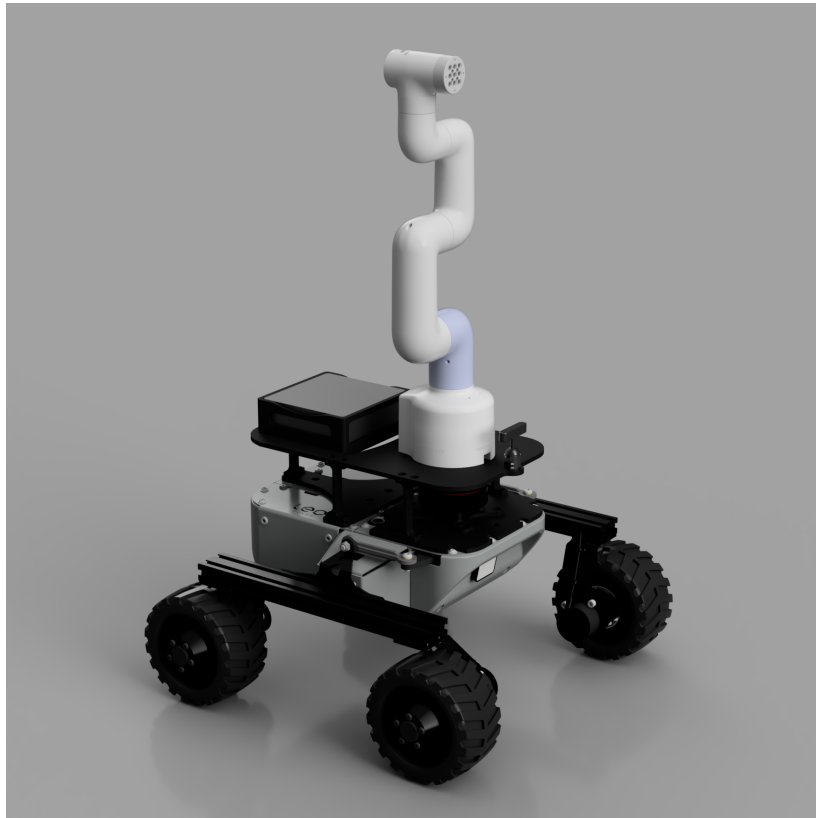


Figure 12: Full CAD assembly showing the Leo Rover with front-mounted LiDAR, front top manipulator and RGB-D camera, and rear top NUC.

5.12 Materials and Manufacturability

The plates and structural adapters are manufactured using fused-filament fabrication (FFF) 3D printing with PLA. This choice was made not only for convenience, but because it offers a practical balance between manufacturability, customisation, structural adequacy and speed of iteration for the current prototype stage.

The justification is as follows:

- **Integration of non-standard features:** although the main plate geometry is broadly flat, the plates do not function as simple flat sheets. They incorporate recessed fastener pockets, localised mounting bosses, connector interfaces, slot features, clearance cut-outs and customised hole patterns matched to the Leo Rover, manipulator, LiDAR and NUC interfaces. Producing all of these features as a single printed part reduces secondary operations and simplifies assembly.
- **Low-volume prototyping and rapid iteration:** the payload architecture is a one-off prototype rather than a mass-manufactured product. In this context, 3D printing is more suitable than sheet-metal manufacture or machining because design changes can be implemented and re-fabricated quickly without requiring new tooling, external workshop support or complex post-processing.
- **Sufficient structural performance:** the structural analysis presented below shows that the printed PLA parts provide adequate strength and stiffness for the expected loading conditions. Under conservative loading assumptions of 2 kg for the manipulator and 1 kg for the NUC, the minimum factor of safety is 12.082 and the maximum total displacement is 0.422 mm, which remains below the adopted allowable deflection limit of 0.5 mm. The use of PLA is therefore justified structurally for this application.
- **Ease of customisation:** the mounting interfaces are highly specific to the geometry of the Leo Rover and the selected payloads. Additive manufacture allows these interfaces to be tailored directly in CAD without being constrained to standard stock sections or requiring multiple joined parts.
- **Availability and process familiarity:** PLA is readily available in the lab and is well characterised in existing slicer profiles, reducing process uncertainty and supporting reliable fabrication within the available project timescale.

For this reason, 3D printing was selected even for the plate-like components: not because they are merely flat parts, but because each plate also acts as a *custom integrated mounting structure* with multiple embedded features that would be less convenient to realise using simple sheet cutting alone.

All parts have been designed with FFF constraints in mind, including consistent wall thicknesses, allowance for dimensional tolerances, and avoidance of unsupported overhangs where possible. In addition, the 5 mm plate thickness provides sufficient section stiffness while remaining practical for printing and assembly.

5.13 Structural Analysis of the Payload Architecture

In the revised four-plate architecture, the primary structural concern is the stiffness of the upper support structures under payload loading, particularly the front upper module carrying the manipulator and RGB-D camera, and the rear upper module carrying the NUC. A static stress simulation was therefore performed to assess the structural adequacy of the revised payload architecture under conservative loading assumptions.

The analysis was carried out with the following assumptions:

- **Material:** PLA with nominal elastic modulus and yield strength from standard datasheets.
- **Manipulator load:** conservatively assumed as 2 kg, applied to the front upper support structure. This is intentionally higher than the actual manipulator mass, so that the analysis represents a worst-case loading condition.
- **NUC load:** conservatively assumed as 1 kg, applied to the rear upper support structure. This is also higher than the actual NUC mass, again making the structural assessment conservative.
- **Boundary conditions:** the lower plates were assumed rigidly fixed to the rover chassis through their bolted interfaces.
- **Connector joints:** the connectors between upper and lower plates were modelled as ideal bolted joints for the purposes of first-order static analysis.

Thus, the structural model intentionally uses payload masses that are higher than the true component masses, meaning that the resulting stress and displacement values represent a conservative upper-bound estimate for normal operation.

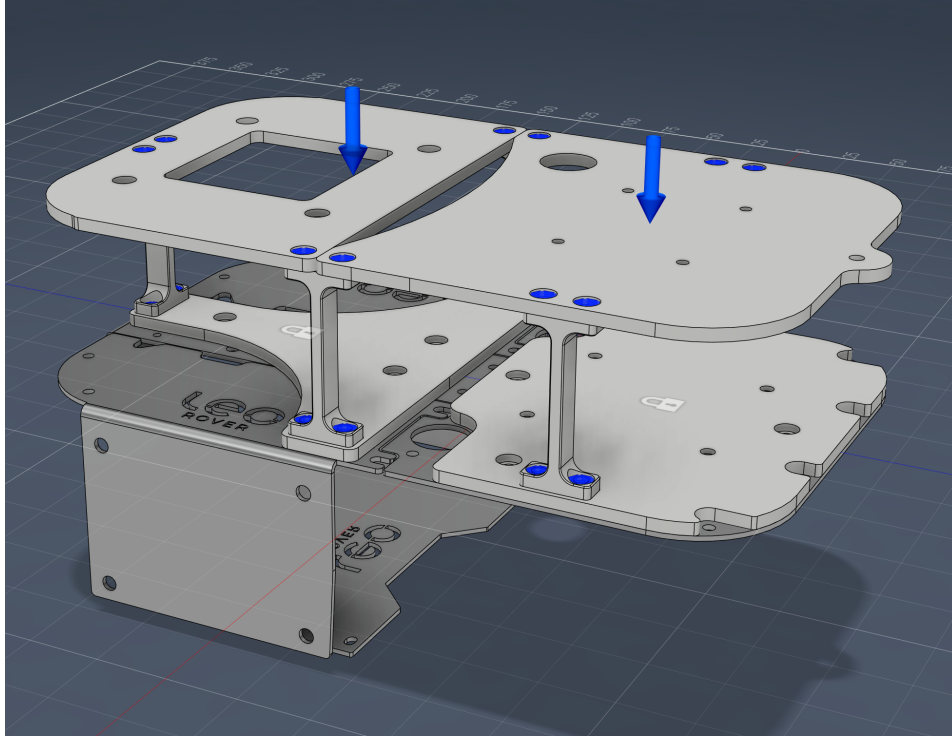


Figure 13: Static stress analysis setup for the revised payload architecture.

A representative pre-processing model is illustrated in analysisFigure 13.

The key output metrics are summarised in Table 5. The most important results are the minimum factor of safety, the maximum von Mises stress, and the maximum displacement components.

Table 5: Summary of static stress analysis results for the revised payload architecture (PLA, plate thickness 5 mm).

Quantity	Result	Comment
Minimum safety factor	12.082	> 4; structurally acceptable with good margin
Maximum safety factor	6.934×10^{13}	Occurs in near-unloaded regions; not design-critical
Maximum von Mises stress	1.655 MPa	Well below typical PLA yield strength
Minimum von Mises stress	0 MPa	In low-stress regions away from the load path
Maximum total displacement	0.422 mm	Global maximum deformation
Maximum displacement in X	0.019 mm	Small lateral deformation
Maximum displacement in Y	0.421 mm	Dominant displacement component
Maximum displacement in Z	0.032 mm	Small out-of-plane deformation

Figures 14 and 15 illustrate the von Mises stress distribution and displacement contours respectively.

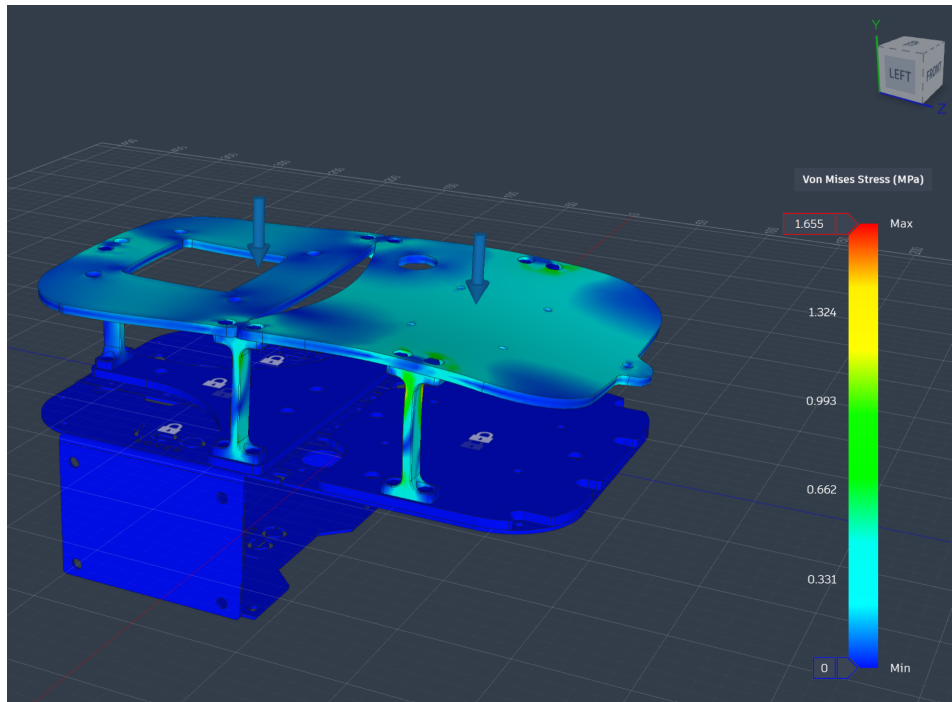


Figure 14: Von Mises stress distribution in the revised payload architecture under the applied load case.

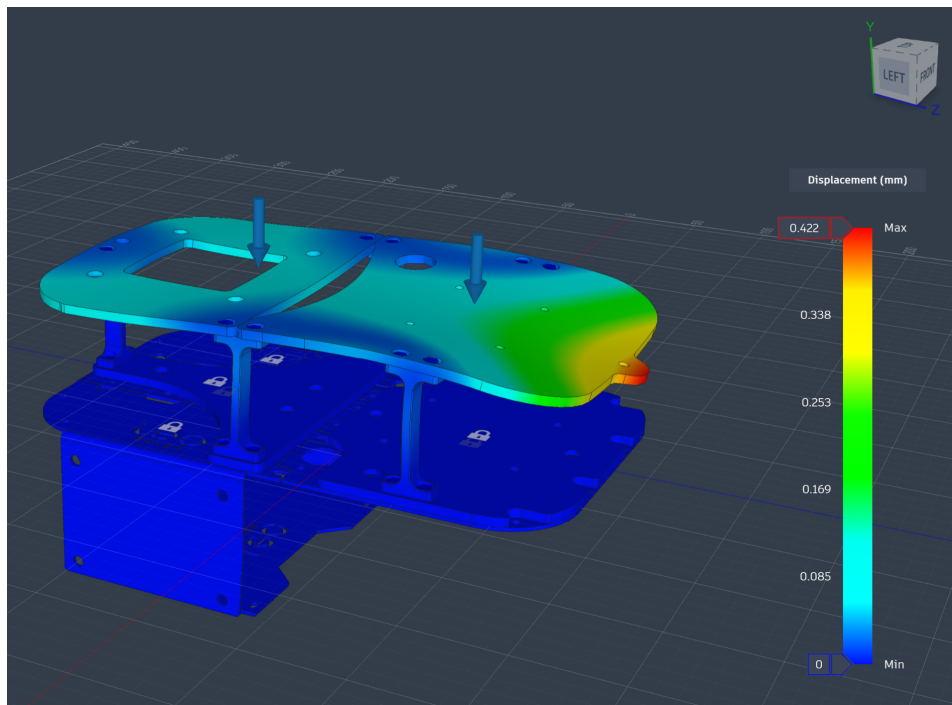


Figure 15: Displacement distribution in the revised payload architecture under the applied load case.

The simulation results show that the revised payload architecture is structurally safe from a stress point of view. The minimum factor of safety is 12.082, which is comfortably above the design target of 4, and the maximum von Mises stress of 1.655 MPa remains far below the yield strength typically associated with PLA.

It is important to note that these results were obtained using deliberately conservative payload masses: 2 kg for the manipulator and 1 kg for the NUC, both of which are higher than the actual component masses. The real in-service stresses and deflections are therefore expected to be lower than those reported here.

In terms of stiffness, the maximum total displacement is 0.422 mm, with the deformation dominated by the global Y-direction component (0.421 mm), while the Z-direction displacement remains comparatively small at 0.032 mm. Since the maximum total displacement remains below the revised allowable deflection limit of 0.5 mm, the structure also satisfies the stiffness requirement under the applied conservative loading case.

Overall, the revised payload architecture is considered structurally adequate for the intended application, with a strong safety margin, low stress levels, and acceptable deformation under conservative loading.

5.14 Compliance with Mechanical Requirements

The final mechanical design addresses the mechanical / structural requirements identified in the Design Requirements Analysis as follows:

- **FR-37, PR-37.1:** Added structures remain within the arena vertical and horizontal clearance limits. The front and rear top plates are sized so that the overall payload architecture does not exceed the original Leo Rover footprint, and the combined height of the LiDAR, manipulator, camera and NUC remains within the permitted vertical envelope.
- **FR-38, PR-38.1:** All modifications are reversible and non-destructive. The payload architecture is bolted to existing mounting points or custom interfaces without drilling or permanently modifying the Leo Rover chassis, and the split front/rear plate layout preserves access to internal components for maintenance.
- **FR-39, PR-39.1:** The four-plate payload architecture provides dedicated mounting locations for sensing, manipulation and compute hardware without increasing the base robot footprint. The front module supports the LiDAR, manipulator and RGB-D camera, while the rear module supports the NUC and associated cable-routing interfaces.
- **FR-40, PR-40.1:** Sensor positioning and connector placement are selected to preserve the required fields of view as far as practicable. The LiDAR is mounted low at the front for improved detection of low obstacles and bin edges, while the RGB-D camera is mounted at the front of the manipulator so that the pickup region remains in view during approach and grasping. Relocating the NUC to the rear top plate also helps avoid unnecessary LiDAR occlusion by the compute unit.
- **FR-41, PR-41.1:** Structural analysis confirms that the revised payload architecture can support the applied payload loads with an adequate safety margin. Under conservative assumptions of 2 kg manipulator loading and 1 kg NUC loading, the minimum factor of safety is 12.082 and the maximum total displacement is 0.422 mm, which remains below the adopted allowable deflection limit of 0.5 mm.
- **FR-42, PR-42.1:** The split front/rear plate arrangement and connector layout allow cables to be routed in a controlled manner between the front sensing/manipulation module and the rear compute module. This reduces congestion around the manipulator workspace and helps minimise snagging risk while maintaining adequate clearance from the ground and moving components.
- **FR-43, PR-43.1:** The layout avoids obstructing the emergency stop, power switches and other safety-critical interfaces. The arrangement of the upper plates and payload modules preserves access for manual operation and emergency intervention.
- **FR-44, PR-44.1:** All exposed structural edges are modelled with chamfers or rounded transitions to remove sharp corners and reduce the risk of cable abrasion, handling injury or damage during installation and maintenance.

Overall, the final mechanical concept satisfies the identified structural and packaging requirements while also providing a more task-aligned arrangement of the sensing, manipulation and compute subsystems. On this basis, the design is considered sufficiently mature to proceed to manufacturing and physical integration on the Leo Rover platform.

6 Electrical Design

6.1 Electrical Power Connections Diagram

A key challenge in system design is to anticipate the power distribution needs in both the primary LEO Rover base system and the secondary peripheral system, as distinguished in Figure 16.

The following electrical design diagram aims to provide an overview of key power distribution and communication architecture for the LEO Rover and mission system. Through mapping the integration of core electrical components, Figure 16 serves as a reference point for the electrical dependencies, power pathways, and interface standards used throughout the system.

Key elements shown in the diagram include:

- **Voltage Regulation Identification** addresses voltage regulation requirement points for power conversion between battery power into subsystem-specific operating voltages.
- **Sub-System Identification** highlights key sub-architectures such as the Raspberry Pi 5 and LeoCore motor controller unit responsible for core computation and mobility control.
- **Power Pathway Definition** Labels Voltage and Current differentiation points between peripherals and electrical components
- **Differentiated Connection Interfaces** depiction of the power, UART, CSI, USB, and RF connections used to interface all modules.

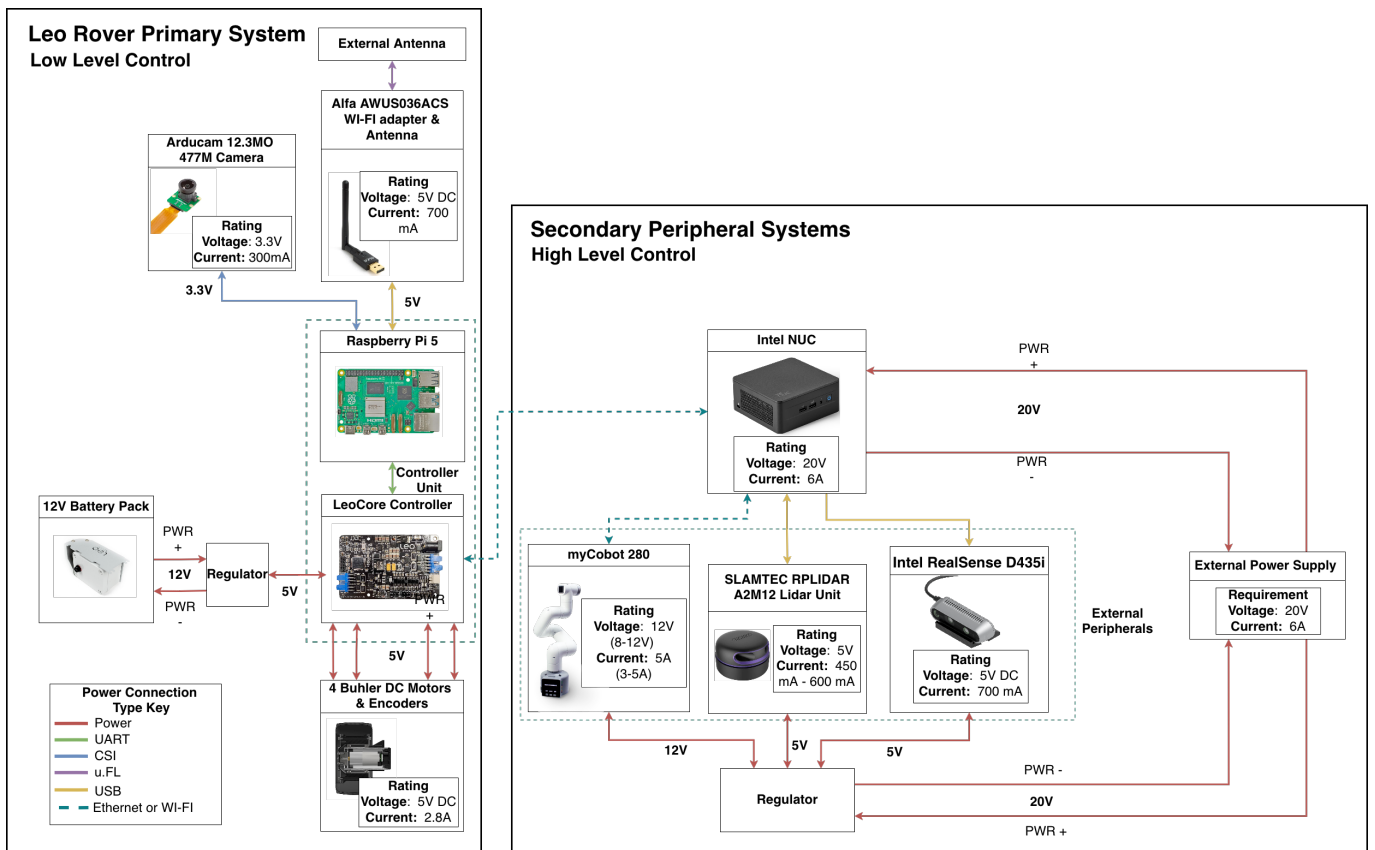


Figure 16: Power Connections Diagram indicating Power Connection Type and Rating for the Hardware Architecture

6.2 Power Budget

The Power Budget depicted in Tables 7 and 8 aims to define the power requirements of the electrical system. To ensure the reliable operation of the LEO Rover throughout all mission phases, in addition to testing and demonstration. To address this, the following power budget identifies and quantifies the electrical demands of each major subsystem, namely the primary low-level system and the secondary peripheral system, enabling effective energy allocation.

6.3 Key Challenges

1. Graded Peripheral Power Requirement 8

Indicates the Current, Voltage and Power Requirements for hardware components of the systems. A traffic light system identifies the low, medium and high power demands of the system, thereby highlighting areas of the electrical system which may require external power or energy management. Cataloguing of individual components in the context of subsystems ensures the power system can support nominal operations and mission-critical activities.

2. External Power or Additional Battery Requirement

The Power Budget identifies the need for an external battery to support the cumulative power load of each subsystem. Leo Rover is powered by a 3S Li-Ion battery with a capacity of 7000 mAh, which is able to meet the combined 3733mAh and 2000mAh requirements of each subsystem for the duration of a 20 minute mission. However, this is unsuitable for long duration running, testing and demonstration of the system.

The proposed solution, depicted Figure 16 is to incorporate an additional Li-Ion battery, which is able to meet the nominal rating of the Intel NUC which is the highest rated electrical component. The Anker Prime Power Bank (26K, 300W) has been selected as a suitable secondary power source, with the following specification:

Specification / Feature	Details
Product	Anker Prime Power Bank (26K, 300W)
Description	High-capacity portable charger with 3 ports (2 USB-C, 1 USB-A), capable of high total power output.
Total Output	300W max
Capacity	26K
Ports	2× USB-C, 1× USB-A
USB-C Output	Max PD 3.1
USB-A Output	Max output not specified in source text
Recharging Speed	Up to dual USB-C input, or single USB-C input (exact values not specified in source text)
Recharge Time	~48–62 minutes for full charge
Dimensions	62.7 x 159.9 x 38 mm
Weight	~600 g
Features	Smart screen for real-time data, Bluetooth app connectivity, ActiveShield 4.0 temperature monitoring
Fast Charging Performance	Charges a 16-inch MacBook Pro from 0–79% in under 50 minutes, or an iPhone 16 Pro to 50% in 26 minutes
Simultaneous Charging	Supports high-speed charging for up to three devices at once
Recharge Performance	Partial recharge in 17–23 minutes; full recharge in under an hour with adequate input

Table 6: Summary Specification for the Anker Prime Power Bank (26K, 300W).

3. Regulation Requirement

This Power overview, when combined information depicted in Figure 16, identifies key changes in nominal voltage and current rating. For the Raspberry Pi, an onboard linear regulator allows the LEO Rover Battery power to safely decrease from 12V to 5V to meet the voltage required by the controller unit and connected peripherals. This would need to be similarly replicated in the Secondary Peripheral System where rated voltages vary from 20V to 5V. This may be achieved through the use of a higher efficiency switching regulator, such as a buck converter or switched-mode power supply.

Part Identifier	Part Name	Part Use
U1,U2,U3,U4	Buhler DC Motors & Encoders	Wheel Motors + Encoders
U5	Arducam 12.3MP 477M Camera	Leo Rover On-Board Camera
U6	Alfa AWUS036ACS Wi-Fi Adapter	Leo Rover On-Board Wi-Fi Adapter
U7	Intel NUC	Secondary On-Board Computer
U8	myCobot 280	Manipulator Arm
U9	Intel RealSense D435i	Peripheral Depth Camera

Table 7: Component Identification: Part Name and System Role

Part Identifier	Qty	I/device (mA)	I/system (mA)	Voltage (V)	Power (W)
U1,U2,U3,U4	4	2800	11200	5	56
U5	1	300	300	3.3	0.99
U6	1	700	700	5	3.5
U7	1	6000	6000	20	120
U8	1	5000	5000	12	60
U9	1	700	700	5	3.5

Table 8: Leo Rover Component Power Budget (Refer to Table 1 for Names and Uses)

Leo Rover Primary System	Value
Minimum Subsystem Voltage	56
Minimum Subsystem Current (mA)	11200
Minimum Subsystem Current (A)	112
Power Total (W)	60.49
Est. Efficiency	50.00%
Input Power Needed (W)	90.735
Mission Duration (min)	20
Energy Requirement (Wh)	30.245
Energy Requirement (mAh)	3733

Primary System Summary

Secondary Peripheral System	Value
Minimum Subsystem Voltage	120
Minimum Subsystem Current (mA)	6000
Minimum Subsystem Current (A)	60
Power Total (W)	183.5
Est. Efficiency	50.00%
Input Power Needed (W)	275.25
Mission Duration (min)	20
Energy Requirement (Wh)	91.75
Energy Requirement (mAh)	2000

Secondary System Summary

7 Software Design

7.1 RQT Graph & Software Architecture

7.1.1 Software block diagram

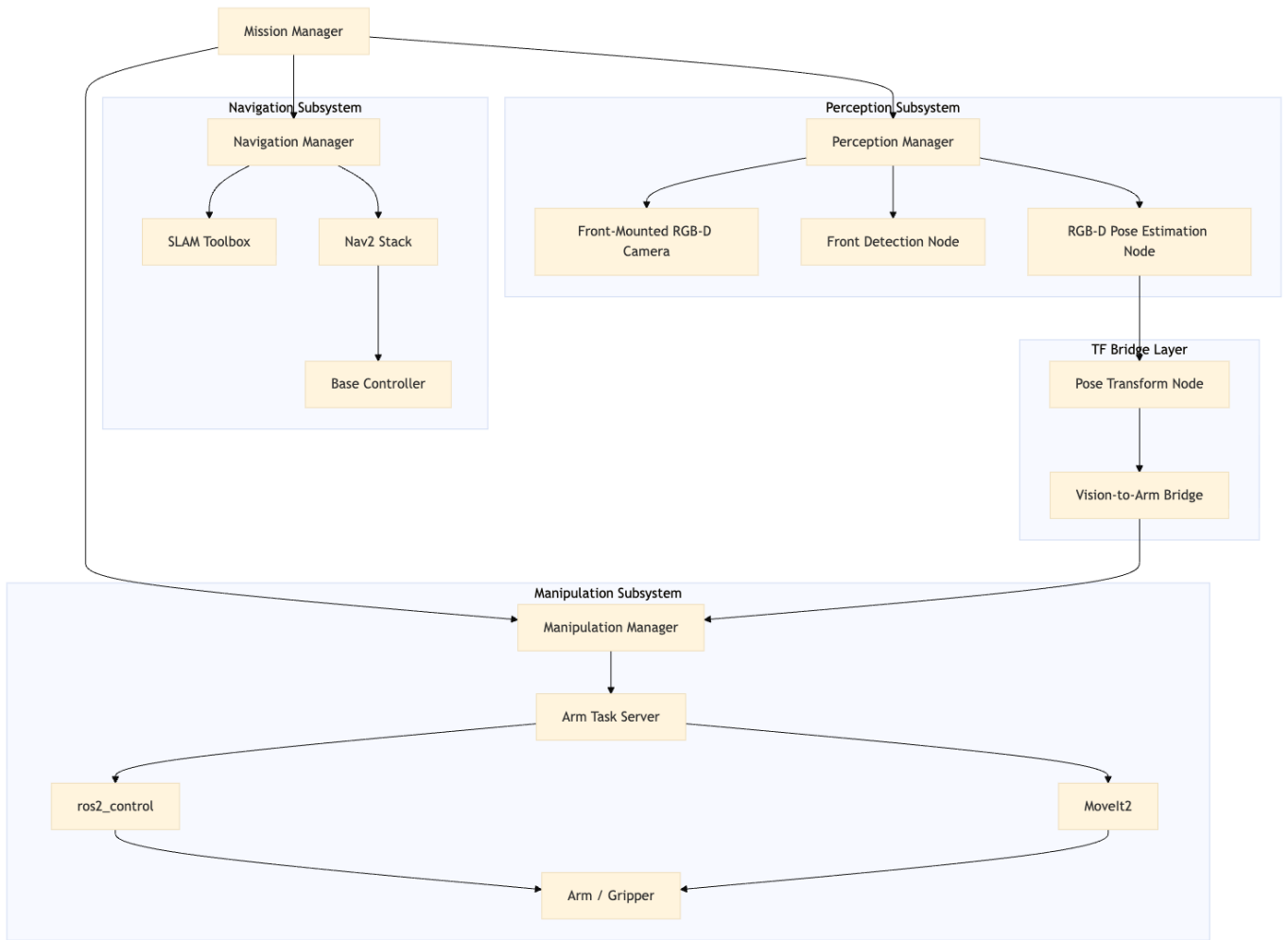


Figure 17: Software Block Diagram

The mission_manager acts as the global orchestrator, coordinating the navigation, perception, and manipulation managers. The TF bridge layer supports direct perception-to-action integration by transforming perception outputs into manipulator-referenced target poses.

7.1.2 RQT Graph

Figure 18 illustrates the node- and topic-level interfaces linking the mission, navigation, perception, TF bridging, and manipulation subsystems. A unified front-mounted RGB-D perception pipeline provides object detection, colour classification, and 3D pose-estimation outputs for both grasping and placement tasks.

7.1.3 Software Architecture

The software architecture developed for this project is strongly inspired by principles and best practices from **modern software engineering**, particularly those related to modular design, layered system decomposition, and responsibility encapsulation. Although the initial design motivation came from solving practical challenges encountered during software development—such as reducing code complexity, isolating subsystem failures, and improving maintainability—the resulting structure aligns remarkably well with architectural patterns used by leading robotics companies and advanced research platforms.

Through further research, we found that similar hierarchical and manager-oriented architectures are widely adopted in state-of-the-art robotic systems, including industrial manipulation frameworks, commercial autonomous mobile robots,

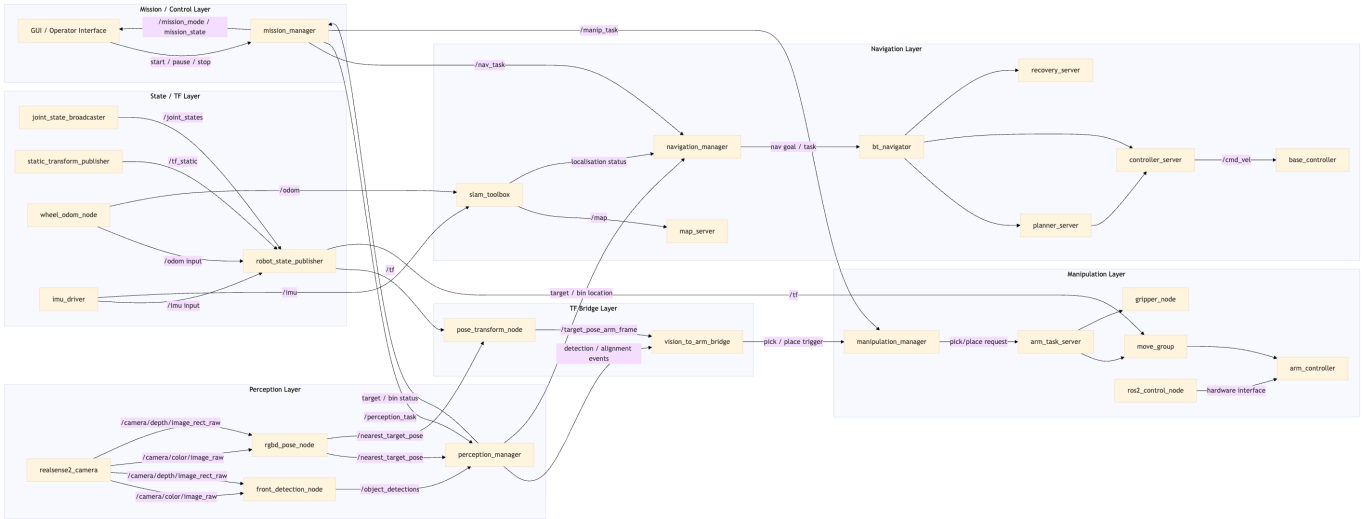


Figure 18: ROS2 runtime communication graph of the software system

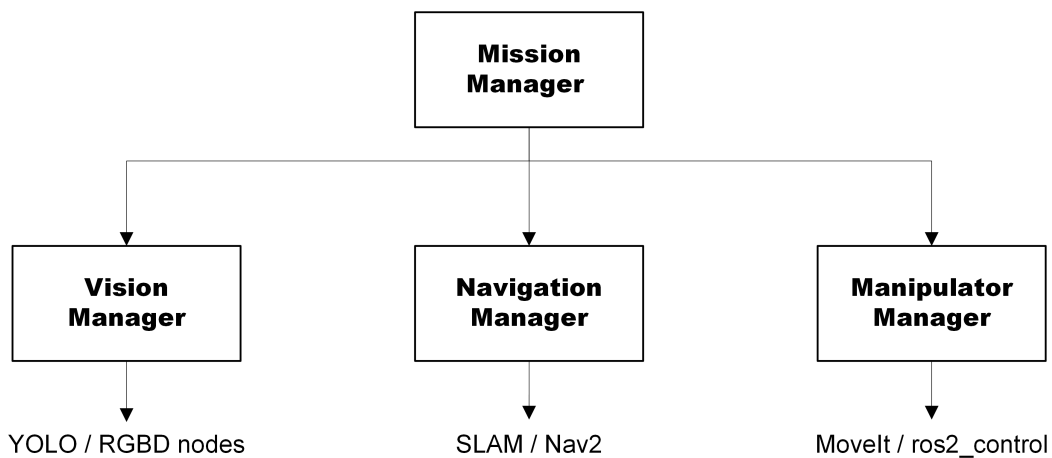


Figure 19: Mission Manager

and large-scale ROS 2-based deployments. These systems consistently emphasize **clear separation of concerns**, **high cohesion and low coupling**, and **capability-oriented modularization**. Our architecture naturally reflects these principles: a **Global Mission Manager** orchestrates the robot’s overall task execution, while independent subsystem managers govern perception, navigation, and manipulation. This structure resembles established patterns such as the **Facade Pattern** in software engineering and the **Capability Layer approach** used in high-end robotics middleware.

By applying these software engineering techniques to robotics, the architecture achieves multiple goals simultaneously:

- it provides a clean abstraction boundary between mission logic and subsystem implementation;
- it enables scalable development where each subsystem can evolve independently;
- it allows well-defined error handling and recovery within each capability module;
- and it supports flexible runtime configuration without modifying source code.

In addition, this layered layout mirrors the architectural philosophy behind frameworks such as **Nav2’s Behavior Tree executor**, **MoveIt’s Task Constructor**, and the **modular autonomy stacks** deployed in commercial robots. The similarity indicates that the approach is not only conceptually sound but also **industry-validated and future-proof**.

Although the architecture originated from practical software engineering thinking, it stands firmly on recognized theoretical foundations and aligns with how modern robotic systems are engineered at scale. It provides a robust, extensible, and maintainable structure capable of supporting complex, multi-stage autonomous tasks.

7.1.4 Trigger Conditions Introduction

To make the mission logic explicit at the software level, the main state transitions are defined through a set of trigger conditions associated with perception outputs, navigation thresholds, pose-validation results, manipulation outcomes, and operator commands. These triggers are evaluated by the relevant subsystem managers and coordinated by the `mission_manager`, ensuring that mission progression is based on observable software events rather than only high-level task descriptions.

The table (Figure 20) summarizes the conditions used to advance between execution states, including subsystem readiness, detection events, navigation distance thresholds, 3D pose validity checks, manipulation outcomes, and operator control commands.

7.2 Subsystem Detailed Design

7.2.1 Low-level Robot State & TF System (State & TF System)

The low-level state and TF subsystem forms the **infrastructure layer** of the entire system. It continuously provides unified spatial and kinematic information for all higher-level modules. Joint states, wheel odometry, and IMU data are fused and propagated through the TF tree via `robot_state_publisher` and the static TF configuration derived from the URDF model. The Table 9 shows the mentioned subsystem.

This subsystem establishes a consistent coordinate framework linking the chassis → body → sensors → arm end-effector. As a result, upper-layer modules such as navigation, perception, and manipulation can operate without manually handling sensor-specific coordinate transforms. This ensures accurate localization of the robot at its starting position and provides reliable spatial information throughout all movement phases.

Node Name	Primary Function	Input Source	Output / Affected Modules
<code>joint_state_broadcaster</code>	Publishes joint states (<code>/joint_states</code>)	<code>ros2_control</code> hardware interface	<code>robot_state_publisher</code> , MoveIt2
<code>robot_state_publisher</code>	Publishes full TF tree based on URDF + joint states	<code>joint_state_broadcaster</code> , IMU, <code>wheel_odom_node</code>	All modules requiring TF (navigation, perception, arm)
<code>static_transform_publisher</code>	Publishes static TFs (camera poses, arm base, etc.)	—	<code>robot_state_publisher</code> , TF chain
<code>wheel_odom_node</code>	Provides odometry (<code>/odom</code>)	Wheel encoders / chassis motors	<code>robot_state_publisher</code> , <code>slam_toolbox</code>
<code>imu_driver</code>	Publishes IMU data and angular velocity	Onboard IMU	<code>robot_state_publisher</code> , <code>slam_toolbox</code>

Table 9: Description of TF System Nodes

7.2.2 Navigation Subsystem (Navigation Manager + SLAM & Nav2)

The navigation subsystem is encapsulated by the **navigation_manager**, which exposes high-level navigation capabilities such as *autonomous mapping*, *approach target*, *return home*, and *approach bin*. Upper layers (e.g., the Mission Manager) do not directly interact with Nav2 internals, but simply issue abstract commands. The Table 10 shows the mentioned subsystem.

Inside the subsystem:

- `slam_toolbox` handles mapping and localization, maintaining the critical `/map` → `/odom` transform.
- `Nav2` (`planner_server`, `controller_server`, `bt_navigator`, `recovery_server`) executes global path planning, local motion control, and recovery actions.
- `navigation_manager` acts as a mediator, coordinating SLAM and Nav2 based on mission state.

This architecture enables reliable autonomous navigation for three key tasks:

1. approaching detected objects,
2. returning to the start area using SLAM-based localization,
3. navigating to the correct bin for object placement.

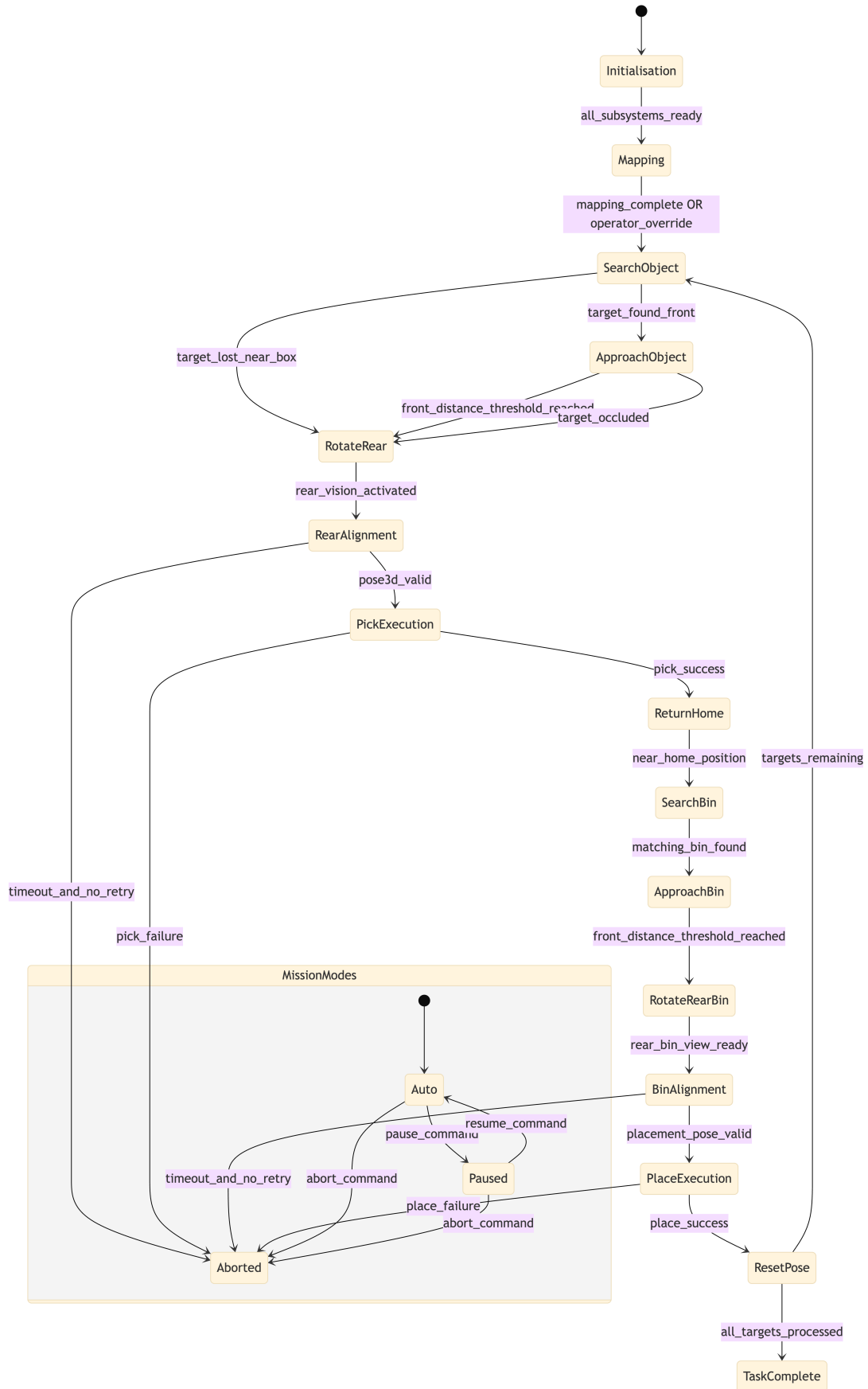


Figure 20: Trigger conditions governing mission state transitions

7.2.3 Perception Subsystem

The perception subsystem is centralized under **perception_manager**, which unifies front-view coarse detection and rear-view precise 3D perception:

Node Name	Primary Function	Input Source	Output / Affected Modules
navigation_manager	Manages navigation tasks: mapping, approach, return, bin approach	mission_manager, slam_toolbox, perception events	bt_navigator, slam_toolbox
slam_toolbox	Online mapping & localization; maintains /map → /odom	wheel_odom_node, imu_driver	navigation_manager, map_server
map_server	Provides global map to planners	slam_toolbox	planner_server
planner_server	Computes global paths	map_server, bt_navigator	controller_server
controller_server	Generates velocity commands for local control	planner_server, bt_navigator	base_controller
bt_navigator	Executes Behavior Trees for navigation	navigation_manager, slam_toolbox	planner_server, controller_server
recovery_server	Provides recovery behaviors when navigation fails	bt_navigator	bt_navigator
base_controller	Sends movement commands to the chassis	controller_server	Robot hardware

Table 10: Description of Navigation System Nodes

The perception subsystem is centralized under the perception_manager and is implemented as a unified front-mounted RGB-D perception pipeline. Object detection, colour classification, bin identification, and 3D pose estimation are performed using a single Intel RealSense RGB-D camera mounted at the front of the robot and co-located with the manipulator workspace. This arrangement supports direct perception-to-action alignment during both grasping and placement.

The perception subsystem provides two tightly related capabilities. First, during search and approach, the RGB-D pipeline detects candidate target objects and storage bins, classifies their colour, and provides coarse directional and positional information to support navigation-level decisions. Second, once the robot reaches the configured pre-grasp or pre-placement distance threshold, the same RGB-D pipeline refines the estimate of the nearest valid object or bin pose in 3D, allowing precise alignment within the manipulator workspace. In this way, the system progresses from global search to local precision alignment within a single sensing framework.

The perception_manager abstracts these perception processes and provides stable outputs to the mission_manager, navigation_manager, and TF bridge layer. It manages perception according to mission context, including object search, object alignment, bin search, and bin alignment. This design supports the project requirements for autonomous search, detection, pose estimation, and object-to-bin matching, while also enabling configurable thresholds, logging, and robust runtime coordination. Table 11 shows the mentioned subsystem.

Node Name	Primary Function
perception_manager	Coordinates object/bin perception, target selection, and stable perception outputs
realsense2_camera	Publishes RGB-D image streams and depth data
front_detection_node	Performs object/bin detection and colour classification
rgbd_pose_node	Extracts nearest valid 3D object/bin pose from RGB-D data
/object_detections	Candidate object/bin detections
/nearest_target_pose	Refined 3D pose for manipulation

Table 11: Perception System Nodes Summary

7.2.4 TF Bridge Layer (Coordinate Transformation Layer)

This layer forms the **interface between perception and manipulation**:

- **pose_transform_node** ensures that vision outputs are expressed in coordinates meaningful to the robotic arm.
- **vision_to_arm_bridge** interprets perception events and mission conditions to decide when manipulation should be triggered.

By placing all transformation and bridging logic here, the system avoids contaminating the manipulation subsystem with visual logic and keeps the Mission Manager clean from coordinate-frame details. The Table 12 shows the mentioned Layer.

Node Name	Primary Function	Input Source	Output / Affected Modules
<code>pose_transform_node</code>	Converts 3D poses from camera frame → <code>base_link</code> / <code>arm_base_link</code>	<code>/nearest_object_pose</code> , TF tree, <code>robot_state_publisher</code>	<code>vision_to_arm_bridge</code>
<code>vision_to_arm_bridge</code>	Determines grasp/place trigger timing; sends transformed target pose to manipulation subsystem	<code>perception_manager</code> , <code>pose_transform_node</code>	<code>arm_task_server</code> / <code>manipulation_manager</code>

Table 12: Description of TF Bridge Layer Nodes

7.2.5 Manipulation Subsystem (Manipulation Manager + `ros2_control` + MoveIt2)

The manipulation subsystem provides a unified manipulation capability through `manipulation_manager`, which coordinates:

- Safe approach to grasp poses
- Grasp and lift motions
- Bin placement procedures
- Recovery behaviors
- Arm safety poses

MoveIt2 handles motion planning and collision checking, while `ros2_control` executes joint trajectories. Through `arm_task_server`, complex multi-step sequences (e.g., pre-grasp → descend → grasp → lift → move → place) are abstracted into simple service calls for upper layers. The Table 13 shows the mentioned subsystem.

Node Name	Primary Function	Input Source	Output / Affected Modules
<code>manipulation_manager</code>	Manages pick/place tasks; ensures safety; error handling	<code>mission_manager</code> , <code>vision_to_arm_bridge</code> , <code>rgbd_vision_node</code>	<code>arm_task_server</code> , MoveIt2
<code>ros2_control_node</code>	Hardware interface for the arm; publishes joint states	Arm hardware	<code>arm_controller</code> , MoveIt2
<code>arm_controller</code>	Executes joint trajectories generated by MoveIt2	<code>move_group</code> , <code>ros2_control_node</code>	arm hardware
<code>move_group</code>	Motion planning, collision checking (MoveIt2)	<code>arm_task_server</code> , <code>robot_state_publisher</code>	<code>arm_controller</code>
<code>gripper_node</code>	Controls end-effector opening/-closing	<code>arm_task_server</code>	Gripper actuator
<code>arm_task_server</code>	High-level pick/place execution logic	<code>manipulation_manager</code> , <code>vision_to_arm_bridge</code>	<code>move_group</code> , <code>gripper_node</code>

Table 13: Description of Manipulation System Nodes

7.2.6 Mission Manager

The Mission Manager is now a **pure orchestrator**. It does not interact with raw ROS nodes; instead, it commands three subsystem managers:

1. `navigation_manager` (movement tasks)
2. `perception_manager` (detection & perception fusion)
3. `manipulation_manager` (pick & place execution)

Its finite-state-machine (FSM) drives the entire mission:

Start → Mapping → Object Search → Front-Vision Approach → Pick → Return Home → Bin Search → Approach Bin → Precision Placement → Reset Pose → Task Complete

Mission Manager also handles:

- Four operational modes: Auto / Pause / Resume / Abort
- Runtime parameter configuration
- Complete logging and export (CSV/JSON/TXT)
- Multi-object vs. single-object strategies

Because the subsystem managers encapsulate complex behaviors, the Mission Manager remains clean, readable, and easy to extend. The Table 14 shows the structure of Mission Manager.

Node Name	Primary Function	Input Source	Output / Affected Modules
mission_manager	Global task state machine: mapping → search → approach → alignment → pick → return → bin search → bin approach → alignment → place → reset	navigation_manager, perception_manager, manipulation_manager	navigation_manager, perception_manager, manipulation_manager
Control Interface	Runtime configuration & mode switching (Auto / Pause / Resume / Abort; multi-object / single-object mode; parameter tuning)	Operator terminal input	mission_manager internal state, logging system

Table 14: Description of Mission Management System Nodes

The mission execution logic is implemented as a node-level orchestration workflow centred on the `mission_manager`, which acts as the global finite-state controller of the robot. At system start, the `mission_manager` loads runtime configuration parameters, initializes logging, resets the manipulator to a safe pose through the manipulation stack, and activates the three subsystem managers: `navigation_manager`, `perception_manager`, and `manipulation_manager`. In parallel, the localisation and state infrastructure is brought online through `slam_toolbox`, `wheel_odom_node`, `imu_driver`, `joint_state_broadcaster`, `robot_state_publisher`, and the required static TF publishers.

During the mapping and exploration phase, the `navigation_manager` coordinates `slam_toolbox` and the Nav2 stack to generate and maintain a usable map of the arena while preserving obstacle-aware autonomous motion. Once mapping completion criteria are satisfied, the system transitions to object search. At this stage, the `perception_manager` processes RGB-D data from `realsense2_camera` through the front detection and pose-estimation pipeline to identify candidate target objects, classify their colour, and publish stable detection outputs. If a valid target is detected, the `navigation_manager` receives the target direction or navigation objective and commands the robot to approach the object while maintaining the required navigation and safety constraints.

When the robot reaches the configured pre-grasp distance threshold, the system enters front precision alignment. The refined target pose is computed by the RGB-D perception pipeline, transformed into the manipulator reference frame by `pose_transform_node`, and passed through `vision_to_arm_bridge` to the `manipulation_manager`. The `manipulation_manager` then invokes `arm_task_server`, `move_group`, `arm_controller`, `ros2_control_node`, and `gripper_node` to execute the grasp sequence. After a successful grasp, control returns to the `navigation_manager`, which directs the robot toward the home region and the bin area. The perception subsystem then performs bin search and colour matching using the same front-mounted RGB-D sensor, followed by front bin alignment and refined placement pose estimation. Once a valid placement pose is available, the transformed pose is delivered to the manipulation subsystem, which executes the placement sequence and subsequently resets the arm to a safe pose. The `mission_manager` then determines whether additional targets remain; if so, the system returns to the object-search state, otherwise it transitions to task completion. Throughout the mission, the `mission_manager` also supervises timeout handling, failure recovery, logging, and human-machine interaction modes including Auto, Pause, Resume, and Abort.

7.2.7 Mission Execution Logic — Pseudo-Code Specification

(This section provides a structured pseudo-code representation of the mission execution workflow coordinated by the Global Mission Manager and its three subordinate managers: Perception Manager, Navigation Manager, and Manipulation Manager. The pseudo-code abstracts system behavior without relying on implementation details.)

Algorithm 1 System Initialization

```
1: on system_start:  
2:   load runtime configuration parameters  
3:   set global_mode = AUTO  
4:   initialize logging system  
5:   reset_arm_to_safe_pose()  
6:  
7:   start SLAM module  
8:   start all subsystem managers:  
9:     perception_manager.start()  
10:    navigation_manager.start()  
11:    manipulation_manager.start()  
12:  
13:   transition_to(MAPPING)
```

Algorithm 2 Mapping & Exploration (Navigation Manager)

```
1: State: MAPPING  
2:  
3: navigation_manager.perform_exploration()  
4:  
5: if mapping_complete() then  
6:   save_map()  
7:   transition_to(SEARCH_OBJECT)  
8: end if  
9:  
10: if operator_command_received() then  
11:   transition_to(SEARCH_OBJECT)  
12: end if
```

Algorithm 3 Object Search (Perception Manager → Navigation Manager)

```
1: State: SEARCH_OBJECT  
2:  
3: detection = perception_manager.scan_front_camera()  
4:  
5: if detection.target_found then  
6:   navigation_manager.set_target_direction(detection.approx_direction)  
7:   transition_to(APPROACH_OBJECT)  
8: end if  
9:  
10: if detection.lost_near_white_box then  
11:   transition_to(ROTATE_FOR_REAR)  
12: end if
```

Algorithm 4 Approaching the Object (Front Vision Guidance)

```
1: State: APPROACH_OBJECT  
2:  
3: navigation_manager.approach_target()  
4:  
5: if target_lost_occluded() then  
6:   transition_to(ROTATE_FOR_REAR)  
7: end if  
8:  
9: if front_distance_threshold_reached() then  
10:   transition_to(ROTATE_FOR_REAR)  
11: end if
```

Algorithm 5 Pick Execution (Manipulation Manager)

```
1: State: PICK_EXECUTION
2:
3: result = manipulation_manager.execute_pick(arm_pose)
4:
5: if result.success then
6:   transition_to(RETURN_HOME)
7: else
8:   log_failure()
9:   abort_task()
10: end if
```

Algorithm 6 Return to Home (Navigation Manager)

```
1: State: RETURN_HOME
2:
3: navigation_manager.navigate_to_home()
4:
5: if near_home_position() then
6:   transition_to(SEARCH_BIN)
7: end if
```

Algorithm 7 Bin Search (Front Vision Again)

```
1: State: SEARCH_BIN
2:
3: detection = perception_manager.detect_bin_front()
4:
5: if detection.bin_found then
6:   transition_to(APPROACH_BIN)
7: end if
```

Algorithm 8 Approaching the Bin

```
1: State: APPROACH_BIN
2:
3: navigation_manager.approach_bin()
4:
5: if front_distance_threshold_reached() then
6:   transition_to(ROTATE_FOR_REAR_BIN)
7: end if
```

Algorithm 9 Final Depth Alignment Before Placement

```
1: State: BIN_ALIGNMENT
2:
3: placement_pose = perception_manager.get_bin_3d_pose()
4:
5: if placement_pose.valid then
6:   arm_pose = transform_to_arm_frame(placement_pose)
7:   transition_to(PLACE_EXECUTION)
8: end if
9:
10: if timeout then
11:   retry_or_abort()
12: end if
```

Algorithm 10 Placement Execution

```
1: State: PLACE_EXECUTION
2:
3: result = manipulation_manager.execute_place(arm_pose)
4:
5: if result.success then
6:   transition_to(RESET_POSE)
7: else
8:   retry_or_abort()
9: end if
```

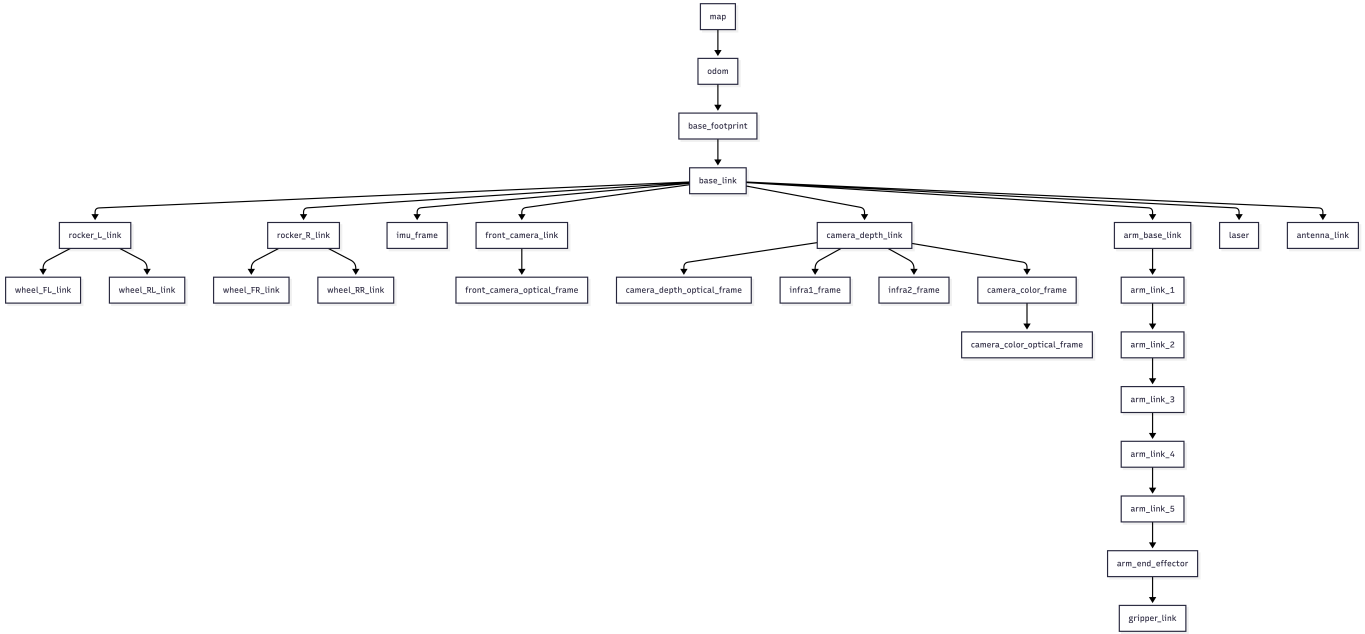
Algorithm 11 Reset and Task Loop

```
1: State: RESET_POSE
2:
3: manipulation_manager.reset_arm_safe()
4:
5: if all_targets_processed() then
6:   transition_to(TASK_COMPLETE)
7: else
8:   transition_to(SEARCH_OBJECT)
9: end if
```

Algorithm 12 System Modes: Auto / Pause / Resume / Abort

```
1: on pause_command:
2:   record_current_state()
3:   freeze_all_subsystems()
4:   transition_to(PAUSED)
5:
6: on resume_command:
7:   restore_previous_state()
8:   unfreeze_all()
9:   continue_execution()
10:
11: on abort_command:
12:   navigation_manager.stop_motion()
13:   manipulation_manager.reset_arm_safe()
14:   save_logs_to_file()
15:   transition_to(ABORTED)
```

7.3 TF Tree



The TF tree used in this project consists of four functional layers that collectively support the robot’s core capabilities in navigation, perception, and manipulation. At the highest level, the chain `map` → `odom` → `base_footprint` → `base_link` establishes the global spatial reference, ensuring that SLAM mapping, path planning, and autonomous return-to-home behaviors operate consistently within a unified coordinate framework.

With `base_link` as the central reference frame, the chassis structure is represented through the rocker joints (`rocker_L_link` / `rocker_R_link`) and the four wheel links (`wheel_*_link`), providing an accurate kinematic description of the mobile platform. The perception system is anchored to `base_link` through the front-mounted Intel RealSense D435 RGB-D camera, whose sensor frame cluster includes `camera_depth_link` and its associated optical, infrared, and colour frames. These frames provide the reference for target and bin detection, colour classification, and high-precision 3D pose estimation within the front manipulator workspace. Independent IMU (`imu_frame`) and LiDAR (`laser`) frames additionally support attitude estimation, localisation, and obstacle/environment perception.

The manipulation subsystem attaches to the robot via `arm_base_link`, which serves as the mounting reference for the manipulator. The kinematic chain from `arm.link.1` to `arm_end_effector` fully describes the robot arm’s degrees of freedom, enabling the MoveIt2 motion planning framework to perform inverse kinematics and to execute grasping, lifting, and placement operations. Through this layered TF structure, multimodal perception and motion control pipelines are efficiently fused within a consistent coordinate system, ensuring accurate and robust task execution across all robot subsystems.

7.4 Key Design Decisions (KDD)

This section summarizes the major architectural and algorithmic decisions that shape the robot’s software, perception, navigation, and manipulation strategy. Each decision is driven by practical engineering constraints, theoretical considerations, and the characteristics of the robot platform.

KDD-1: Adoption of a Hierarchical Manager-Based Software Architecture

Decision: The system adopts a hierarchical control structure consisting of:

- a **Global Mission Manager**, and
- three capability-oriented **Subsystem Managers** (Perception, Navigation, Manipulation).

Rationale: The initial design motivated by software engineering practice showed that managing all functionalities through a single monolithic task node leads to excessive coupling, limited scalability, and low maintainability. After further investigation, we found that modern robotics companies (e.g., autonomous mobile robot vendors, industrial manipulators, and ROS2-based commercial platforms) adopt similar *capability-layered* architectures.

This structure offers several advantages:

- Clear separation of responsibilities (“perception computes, navigation moves, manipulation acts”).
- Failures can be isolated within subsystems without collapsing the entire mission logic.
- Modules can be replaced, upgraded, or extended without modifying other parts.
- The Mission Manager only orchestrates behavior and no longer interacts with low-level nodes directly.

This design enables the system to scale beyond the current demonstration and evolves naturally toward a production-ready robotics stack.

KDD-2: Use of SLAM Toolbox and Nav2 for Autonomous Exploration, Target Approach, and Return-to-Home

Decision: The system uses **SLAM Toolbox** for online mapping and localization, and **Nav2** (with Behavior Trees) for global navigation, goal-following, and recovery behaviors.

Rationale: Custom path planners were intentionally avoided to reduce development overhead and increase reliability. SLAM Toolbox provides mature, stable, ROS2-native mapping and loop closure, while Nav2 offers industrial-grade navigation features such as obstacle avoidance, costmap fusion, failure recovery, and path replanning.

This combination supports the three mission-critical movements:

1. Autonomous exploration during mapping;
2. Approaching the target during object collection;
3. Returning to the starting zone before bin detection and placement.

The behavior tree execution model further supports modular task composition and seamless integration with the Mission Manager.

KDD-3: Use of MoveIt2 + ros2_control for Manipulation and Grasp Execution

Decision: The system uses MoveIt2 for planning, `ros2_control` for hardware execution, and a high-level `arm_task_server` to implement grasp and place primitives.

Rationale: MoveIt2 provides advanced algorithms for collision-aware planning, inverse kinematics, and trajectory optimization, which are essential for safe and accurate grasping in constrained spaces. `ros2_control` abstracts hardware interactions, while `arm_task_server` offers a high-level API (pick/place) that decouples manipulation algorithms from perception and navigation systems.

This separation aligns with industry patterns (e.g., “primitive-based manipulation pipelines”), ensuring extensibility and maintainability.

KDD-4: Unified TF Tree for Cross-Subsystem Spatial Consistency

Decision: All perception, navigation, and manipulation tasks rely on a single coherent TF tree describing:

- global frames (`map`, `odom`, `base_link`),
- sensor frames (RGB, RGB-D, IMU, LiDAR),
- mechanical arm kinematic chain (`arm_base` → `end_effector` → `gripper`).

Rationale: A consistent TF tree ensures that navigation, detection, depth estimation, and manipulation all reason about objects in the same spatial coordinate space. This enables:

- Camera-to-arm coordinate transformations,
- Accurate distance thresholds for state transitions,
- Precise object localization relative to the workspace.

The TF tree is unchanged by the new software architecture because it reflects hardware geometry, not software wiring.

KDD-5: Runtime-Configurable Task Parameters and Human-in-the-Loop Control Modes

Decision: The Mission Manager supports four modes (Auto, Pause, Resume, Abort) and allows all mission parameters—thresholds, timeouts, allowed object colors, number of objects—to be configured at runtime.

Rationale: This design provides operational flexibility, safe recovery procedures, and adaptability to variable test environments. It also aligns with user-study requirements and real-world robotics practices, where operators must intervene to reposition robots, tune thresholds, or modify task objectives without recompiling software.

7.5 Compliance with Software Requirements

The updated hierarchical software architecture—consisting of the Global Mission Manager and the Perception, Navigation, and Manipulation Managers—meets all functional requirements defined in the system specification.

The **Low-Level State & TF System** provides consistent localization, start-pose recording, and global frame alignment, supporting the robot’s ability to return to its initial position (**FR-2, FR-10**).

The **Navigation Subsystem** (SLAM Toolbox + Nav2), managed by the Navigation Manager, enables autonomous mapping, safe exploration, dynamic replanning, obstacle avoidance, narrow-gap handling, movement to pre-grasp positions, and safe traversal of the environment (**FR-2, FR-6, FR-8, FR-9, FR-10, FR-11, FR-15, FR-25, FR-26, FR-27, FR-28, FR-29, FR-30**).

The **Perception Subsystem**, under the **Perception Manager**, is organized as a unified front-mounted RGB-D perception pipeline. The Intel RealSense camera provides RGB-D data for target and bin detection, colour classification, and 3D pose estimation, enabling both search-level perception and manipulation-level alignment within the front workspace. These perception functions collectively satisfy object detection, color-bin matching, grasp-pose refinement, placement verification, and re-localization after operator intervention (**FR-12, FR-13, FR-14, FR-15, FR-16, FR-19, FR-20, FR-21, FR-36, FR-46, FR-47**).

The **Coordinate Transformation Layer** (TF + `vision_to_arm_bridge`) provides reliable camera-to-base and base-to-arm pose conversion required for generating graspable target coordinates (**FR-16**).

The **Manipulation Subsystem** (`ros2_control` + MoveIt2), managed by the Manipulation Manager, supports grasp-pose generation, motion planning, approach trajectories, lifting to transport height, and stable object retention during movement (**FR-14, FR-15, FR-16, FR-18**).

7.6 Git Repository Link

Git repository: <https://github.com/allanbissac/V.I.S.O.R.>

8 Analysis

Table 15: Verification success criteria, design approach, evidence and compliance assessment.

Verification Success Criteria	Design Approach	Verification Evidence	FR ID	PR ID	Compliance Assessment	Risks & Notes
System must locate and approach the first target in ≥ 9 of 10 demonstration runs within the arena.	The system controller executes a state machine (Start \rightarrow Search \rightarrow Approach \rightarrow Pick \rightarrow Return \rightarrow Deposit \rightarrow End). The system controller initiates the robot to move around the arena using the navigation map while the vision system identifies the nearest target block.	Trial runs in the arena and ROS 2 logs show the correct sequence from the start pose with randomly placed targets; objects are accurately identified in 90% of testing scenarios.	FR-2	PR-2.1	Expected to meet the set threshold with the complete integrated system.	End-to-end integration testing is still ongoing to confirm the 9/10 success rate.
Run trials with 1, 2 and 3 objects and verify that multi-object run time stays within $\pm 10\%$ of the one-object baseline, and that the mission terminates within 5 s of timeout or final deposit.	The trial run is repeated for a variable object count; a global timer and deposit counter are monitored to ensure that the mission executes correctly and terminates properly.	Data from repeated runs show the loop repeating for multiple targets and transitioning cleanly to the terminal state when the object count or time limit is reached.	FR-3, FR-4	PR-3.1, PR-4.1	Software architecture supports multi-object mission execution while obeying time constraints.	Full-stack timing still depends on navigation and manipulation load.
The start command initiates autonomous operation within 3 s and the system supports Auto, Pause, Resume and Abort modes. Manual intervention for safe pause and resume is supported within the demonstration window.	Operator commands from the NUC are implemented as ROS 2 services/actions. The mission manager coordinates start, pause, resume and abort.	Arena trials show a single start action, responsive mode changes, and successful re-localisation after manual repositioning in Pause mode.	FR-5, FR-34, FR-35, FR-36	PR-5.1, PR-34.1, PR-35.1, PR-36.1	Responsive supervisory control is implemented and demonstrated.	During startup, sensor initialisation and arm state transitions may add latency in some runs.
Generate the map online using only onboard sensors and compare the resulting map/localisation accuracy against arena measurements, targeting < 50 mm positional error.	SLAM performs online mapping. SLAM Toolbox fuses onboard RPLIDAR data, wheel odometry and IMU data; no external sensors or devices are used during operation.	Nav2 and bagged SLAM runs show online map growth and confirm that only onboard LiDAR, odometry and IMU topics feed into mapping.	FR-6, FR-7	PR-6.1	Navigation architecture satisfies the onboard-only localisation requirement; map accuracy is to be validated experimentally.	Sensor placement and arena geometry strongly affect final mapping accuracy; sensor positioning has been chosen to maximise useful view.
Introduce environmental changes during operation and verify that mapping and path planning update within 5 s.	SLAM maintains a live occupancy grid and the navigation costmaps support continuous replanning as objects or obstacles change.	RViz visualisations and navigation bags show concurrent map growth and replanning during arena changes.	FR-8	PR-8.1	Design supports the required update latency.	High compute load may degrade replanning performance.
From varied target locations, verify that the robot reaches the goal pose and then returns autonomously to the start area across repeated runs.	Object detections are converted into <code>base_link</code> approach goals for Nav2, and the mission planner later sends the robot back to the stored start area in the map frame.	RViz views, Nav2 logs and demonstration runs show successful approach to pickup locations and reliable return to start area from varied starting points.	FR-9, FR-10	PR-9.1, PR-10.1	Navigation architecture supports both target approach and autonomous return.	Final pose accuracy depends on map quality.
In demonstration runs, place objects within decreasing distance of each other across 10 runs. The system must detect and reroute appropriate circumvention or traversal within ≤ 60 s.	Nav2 costmaps use the Leo Rover footprint and inflation radius so passages narrower than the robot width are treated as blocked.	Scenario tests with closely spaced obstacles show the planner routing around narrow gaps instead of attempting unsafe traversal.	FR-11	PR-11.1	Replanning behaviour is consistent with the configured footprint model.	Extreme layouts may sometimes approach the 60 s limit if few alternative paths exist.
Operate on the nominal flat arena floor and verify that the robot remains within the mapped boundary without scraping, grounding or trespassing the boundary.	The Leo Rover retains its required ground clearance and planar motion model, while arena walls are represented as occupied boundaries in the SLAM map.	Arena-floor trials and recorded trajectories show stable motion on flat surfaces and paths that stay inside the mapped perimeter without trespassing beyond it.	FR-26, FR-27	PR-26.1	Base platform and mapping approach are suitable for indoor arena operation.	None identified.
Place obstacles at distances of up to 3 m from the robot and verify obstacle detection, path clearance ≥ 100 mm, and stop/recovery response for collision or near-collision events.	LiDAR data populates Nav2 obstacle/voxel layers, while inflation settings enforce path clearance and safety logic commands zero velocity when collision conditions are detected.	Tests with obstacles show mapped obstacles within the sensing envelope, collision-free routed paths with clearance margins, and prompt stop/recovery behaviour during deliberate near-contact cases.	FR-28, FR-29, FR-30	PR-28.1, PR-29.1, PR-30.1	Sensor fusion and navigation safety configuration address detection, clearance and stop-response requirements.	Small or thin obstacles may be harder to detect.
Detect coloured target objects across the specified size range and verify colour classification under representative arena lighting.	The RealSense RGB-D stream and front-view vision pipeline publish detected object centroids and colour labels for the wooden blocks used in the mission.	Live arena trials and offline evaluation with the depth camera show correct detection/classification performance for the target colours.	FR-12, FR-13	PR-12.1, PR-13.1	Perception pipeline supports size-range detection and colour classification.	Severe shadows may reduce confidence.

Continued on next page

Verification Success Criteria	Design Approach	Verification Evidence	FR ID	PR ID	Compliance Assessment	Risks & Notes
Measure the error between estimated pickup pose and the true object centre across repeated runs, targeting the required grasp-pose accuracy.	The <code>rgbd_vision_node</code> fuses RGB and depth data to publish a grasp-suitable object centroid that is transformed through TF into the robot frame.	Measured object positions compared against published poses show accuracy within the expected close-range sensing envelope.	FR-14	PR-14.1	Design supports precise 3D target localisation.	Depth accuracy degrades beyond 10 m, but pickup occurs at close distance within the arena.
Measure time from approach start to pre-grasp pose and verify successful grasp plus lift to transport height in repeated object-collection trials. This time should not exceed 30 s.	MoveIt 2 plans the arm from a transport pose to a pre-grasp pose relative to the estimated object position, then commands grasp closure and lift.	Time-stamped logs, video and arena trials show the robot reaching pre-grasp quickly and lifting blocks to the transport pose while obeying the 30 s time limit.	FR-15, FR-16	PR-15.1, PR-16.1	Manipulation sequence supports the required timing and lift behaviour; tuning continues.	Collision-avoidance complexity and object orientation may affect success rate.
Verify that the grasp does not damage the wooden blocks and that no uncommanded release occurs while carrying an object.	The gripper uses compliant pads, limited-torque actuation and controlled acceleration limits while the arm holds a stable transport pose.	Post-test inspections show no splintering or permanent deformation, and drive-while-carrying tests show continued retention of grasped objects.	FR-17, FR-18	PR-17.1, PR-18.1	Current design supports safe grasping and secure transport.	Sudden base acceleration or inconsistent grip friction could challenge object retention if not tuned carefully.
Present the set of coloured bins to the robot and verify both correct bin identification and correct object-to-bin colour matching.	The perception stack detects and labels bins in the start area, and mission logic maps each detected object colour to the corresponding bin.	Start-area perception tests and mission replay logs show the bins being identified and selected consistently according to object colour.	FR-19, FR-20	PR-20.1	Software and perception architecture support correct bin selection.	Accurate bin labelling is critical; additional handling is needed if a matching bin is missing or misdetected.
Perform repeated deposit trials and verify that objects remain inside the correct bin after release.	The manipulator executes a deposit trajectory above the bin and opens the gripper; an optional RealSense-based check compares depth before and after release to confirm deposition.	Video, mission logs and optional depth snapshots show successful drop-in-bin events and demonstrate that fast deposit confirmation is feasible.	FR-21, FR-47	PR-21.1	Baseline placement is supported; rapid deposit confirmation remains optional.	Placement accuracy still depends on bin-pose estimation and manipulator repeatability.
Trigger both hardware emergency-stop and software safety-hold conditions during runs and verify prompt motion stop plus transition to a safe state.	The physical E-stop removes drive power independently of software, while a supervisory safety node halts base and arm motion and forces Safe Hold on critical faults.	Timed E-stop tests and fault-injection experiments show immediate motion stop and correct mission-state transition when hardware or software safety events occur.	FR-22, FR-23	PR-22.1, PR-23.1	Critical safety-stop behaviour is implemented in both hardware and software.	Fault thresholds and detection coverage work are still ongoing.
Verify that commanded speed never exceeds the indoor testing limit and that final inspection shows no damage to the robot, arena, bins or university equipment.	Velocity limits are enforced in the rover controller/Nav2 configuration, and mechanical design choices plus safe-stop logic are intended to minimise impact energy.	<code>cmd_vel</code> logs stay within the configured limits, and post-run inspections confirm no damage after representative demonstrations.	FR-24, FR-25	PR-24.1, PR-25.1	Speed limiting and conservative operation support the no-damage requirement.	Safe execution depends on careful software implementation and operator availability as a last resort.
Change mission parameters via an external configuration file and verify that autonomous behaviour remains consistent without recompilation after making complex code changes.	Mission-critical values such as object count, colour set, thresholds and timeouts are stored in YAML files loaded at launch instead of being hard-coded.	Configuration reviews and runtime parameter dumps confirm that the mission uses external configuration files for all critical settings.	FR-31	PR-31.1	Full parameterisation is implemented.	Input validation is still needed to guard against invalid configuration values.
Review mission logs for pose, perception and action completeness, and verify that exported log archives are produced in the expected format after a run.	Selected ROS 2 topics are recorded to rosbag alongside high-level event logs, and post-run scripts compress and transfer logs for offline analysis.	Sample rosbag/CSV outputs and archive files contain time-stamped pose, perception and mission-event data and can be evaluated later.	FR-32, FR-33	PR-32.1	Logging and export workflow meet the analysis requirement.	Extended testing will require disk-space management and archive data handling.
Confirm that the completed payload architecture and mounted hardware stay within the Leo Rover footprint envelope and can be removed without permanent modification; added mechanical structures should remain within arena vertical and horizontal boundaries.	A four-plate PLA payload architecture (front bottom, rear bottom, front top and rear top plates) uses existing fixing points only, keeps mounted sensors and compute hardware within the base footprint, and avoids drilling, glue or irreversible fasteners. These structures also obey the arena vertical and horizontal boundaries.	CAD checks, dimensional measurements, assembly photographs and a documented disassembly sequence verify footprint control and reversibility.	FR-37, FR-38, FR-39	PR-37.1, PR-38.1, PR-39.1	Mechanical design meets dimensional, footprint and reversibility requirements.	Repeated assembly/disassembly can still cause cosmetic wear; local cable-routing overhangs must be checked.

Continued on next page

Verification Success Criteria	Design Approach	Verification Evidence	FR ID	PR ID	Compliance Assessment	Risks & Notes
Verify that the LiDAR 360° scan and the primary RGB-D camera field of view are not persistently occluded in nominal operating configurations.	The LiDAR is mounted low at the front; the manipulator and RealSense are mounted on the front top plate with packaging chosen to minimise mutual occlusion. The NUC is relocated to the rear top plate to avoid unnecessary LiDAR occlusion by the compute unit. Except during manipulator motion at extreme poses, the RealSense and Leo Rover front camera views remain unobstructed.	RViz field-of-view checks with the arm at extreme poses show that LiDAR coverage and the grasping camera view are maintained in normal operating postures.	FR-40	PR-40.1	Sensor placement supports the intended field of view.	Some temporary occlusion remains possible in extreme arm poses and must be managed in motion planning.
Under worst-case payload loading, verify that the payload architecture shows no permanent deformation and that structural deflection stays below the 0.5 mm design target.	The 5 mm PLA plates and support connectors were sized using conservative FEA with a 2 kg manipulator load and a 1 kg NUC load, both above the actual component masses, to provide design margin.	FEA predicts a high factor of safety, low von Mises stress and maximum total displacement below the 0.5 mm design limit; physical assembly checks confirm adequate stiffness.	FR-41	PR-41.1	Structural design appears adequate with significant margin.	Dynamic loading during aggressive arm motion should still be validated experimentally.
Inspect final cable routing, ground clearance, emergency-stop/power-switch access and external edge finishing on the assembled structure.	Cables are routed through the payload architecture with strain relief, a clearance zone is reserved around the control interfaces, and all exposed edges are modelled with ≥ 1 mm chamfers/fillets and deburred after printing.	Final inspection and measurements confirm cable clearance above the floor, unobstructed access to safety controls, and rounded or chamfered external edges.	FR-42, FR-43, FR-44	PR-42.1, PR-43.1, PR-44.1	The mechanical integration supports the build safety and finish requirements.	Any rework or additional brackets/cables must be rechecked so clearance and edge-finish assumptions remain valid.
Stretch goal: when multiple targets are visible, demonstrate prioritisation based on defined factors and make the selection within 5 s.	The mission logic includes a basic prioritisation framework, although the current implementation still defaults to nearest-target selection.	Early simulations and design notes show how alternative prioritisation strategies could be integrated; full validation remains optional.	FR-46	PR-46.1	Stretch goal only; not required for baseline mission success.	Lower priority than core retrieval and may not be completed before final demonstration.

9 Project Plan

9.1 Project Planning

Work Package1--Project Management																													
ID	Project	Project Type	Lead Team Member	Activity Type	Schedule	2025.09.22--2025.12.19 (Semester1)												2026.02.02--2026.05.15 (Semester2)											
						W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
1.1	Workplace Charter	General	Nadiya Ivahnenko	Documentation & Reporting	Plan Time	3 weeks																							
					Real Time	3 weeks																							
1.2	Client Needs	General	Bokai Qin	Management & Coordination	Plan Time				2 weeks																				
					Real Time				2 weeks																				
1.3	Design Requirement Analysis	General	Jayant Bakolia	Documentation & Reporting	Plan Time				3 weeks																				
					Real Time				3 weeks																				
1.4	Preliminary Design Review – Planning & Drafting	General	Yinfeng Li	Documentation & Reporting	Plan Time							4 weeks																	
					Real Time							4 weeks																	
1.5	Final Design Review – Planning & Production	General	Mingyang Xu	Documentation & Reporting	Plan Time												4 weeks												
					Real Time															3 weeks									
1.6	Technical Demonstration Videos Planning & Production	General	Allan Binoy Issac	Documentation & Reporting	Plan Time																4 weeks								
					Real Time																								

Work Package2—Navigation																													
ID	Project	Project Type	Lead Team Member	Activity Type	Schedule	2025.09.22--2025.12.19 (Semester1)												2026.02.02--2026.05.15 (Semester2)											
						W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
2.1	LiDAR Setup	General	Jayant Bakolia	Implementation & Experimentation	Plan Time				2 weeks																				
					Real Time				2 weeks																				
2.2	Navigation Requirements Review	General	Mingyang Xu	Analysis & Design	Plan Time				2 weeks																				
					Real Time					1 week																			
2.3	SLAM Stack Selection and Configuration	Concept	Jayant Bakolia	Analysis & Design	Plan Time				2 weeks																				
					Real Time				2 weeks																				

ID	Project	Project Type	Lead Team Member	Activity Type	Schedule	2025.09.22--2025.12.19 (Semester1)												2026.02.02--2026.05.15 (Semester2)											
						W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
5.1	Leo Rover Assembly	General	Nadiya Ivahnenko	Implementation & Experimentation	Plan Time	3 weeks																							
					Real Time	3 weeks																							
5.2	Base Platform Testing and Calibration	Test	Mingyang Xu	Implementation & Experimentation	Plan Time				1 week																				
					Real Time				1 week																				
5.3	Sensor Mount Concept	Concept & Implementation	Jayant Bakolia	Implementation & Experimentation	Plan Time					2 weeks																			
					Real Time					1 week																			
5.4	Payload Sled Design	Concept & Implementation	Yinfeng Li	Implementation & Experimentation	Plan Time												2 weeks												
					Real Time																2 weeks								
5.5	Sensor Installation and test	Test	Allan Binoy Issac	Implementation & Experimentation	Plan Time						1 week												1 week						
					Real Time							1 week														1 week			
5.6	Mechanical Installation and test	Test	Bokai Qin	Implementation & Experimentation	Plan Time																		2 weeks						
					Real Time																								
5.7	Power & Wiring Layout Design	Concept & Implementation	Allan Binoy Issac	Implementation & Experimentation	Plan Time												2 weeks												
					Real Time																	2 weeks							
5.8	Power System Testing and Validation	Test	Nadiya Ivahnenko	Implementation & Experimentation	Plan Time																		2 weeks						
					Real Time																								
5.9	Hardware Integration & Stability Testing	Test	Mingyang Xu	Implementation & Experimentation	Plan Time																		2 weeks						
					Real Time																								
5.10	Final Hardware Preparation	Test	Jayant Bakolia	Implementation & Experimentation	Plan Time																		2 weeks						
					Real Time																								

Work Package6—Systems Integration

ID	Project	Project Type	Lead Team Member	Activity Type	Schedule	2025.09.22--2025.12.19 (Semester1)												2026.02.02--2026.05.15 (Semester2)											
						W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
6.1	System Integration Plan & Interface Review	Concept	Nadiya Ivahnenko	Analysis & Design	Plan Time									2 weeks															
					Real Time																								

9.2 Project Description

9.2.1 Work Package 1 – Project Management

- ID-1.1 Draft and approve the Team Workplace Charter defining shared purpose, behavioural standards, working practices and conflict-resolution processes for the project team.
- ID-1.2 Engage with teaching staff to clarify problem statement, operating environment assumptions, success criteria. Capture discussions in structured notes for Design Requirements Analysis.
- ID-1.3 Draft and iterate Design Requirements Analysis document including problem statement, functional and performance requirements, and initial Requirements Verification Matrix.
- ID-1.4 Plan and produce Preliminary Design Review by mapping Design Requirements to system structure. Draft system, mechanical, electrical, software sections and project plan chapters.
- ID-1.5 Update and extend PDR into Final Design Review incorporating feedback, design changes, and updated requirements verification matrix before submission.
- ID-1.6 Plan, record, and edit Technical Demonstration videos including promotional video and requirements evaluation video. Create clear narrations with simulations and data plots.

9.2.2 Work Package 2 – Navigation

- ID-2.1 Mount 2D LiDAR sensor to the robot, connect to the Intel NUC, install ROS2 drivers, and verify live scan data availability. Design and 3D-print a dedicated LiDAR mount for secure platform installation.
- ID-2.2 Extract all navigation-related functional and performance requirements from the DRA, define measurable metrics, and summarise them in a checklist used to guide later navigation design and testing.
- ID-2.3 Evaluate candidate 2D LiDAR SLAM packages/stacks, compare map quality and robustness, select a baseline stack, and configure frames, topics, and parameters for the target arena to enable reliable mapping and localisation.
- ID-2.4 Configure the SLAM node for online operation, tune update rates and map parameters, and test live mapping while teleoperating the robot through representative trajectories to ensure stable map growth without excessive drift.
- ID-2.5 Arbitrarily place obstacles, quantify drift and loop-closure behaviours, and determine whether the selected SLAM configuration meets navigation requirements.
- ID-2.6 Compare the estimated attitude with the actual motion. Use statistical error data to verify whether the stack meets the required accuracy and stability.
- ID-2.7 Install and configure the Nav2 stack and set up basic launch files. Verify that costmaps, planners, and controllers start correctly. Test simple point-to-point navigation in an empty environment.
- ID-2.8 Configure global planner parameters including resolution, planning frequency, and heuristic weights. Tune the local DWA-style controller with velocity limits, acceleration constraints, and scoring weights to obtain smooth collision-free motion that reaches global-path goals efficiently.
- ID-2.9 Configure global and local costmaps, including inflation radius, obstacle layers, and footprint settings, and validate that occupied and unsafe regions are correctly represented for both planning and control.
- ID-2.10 Implement navigation-level collision-avoidance behaviours using costmaps and local planner outputs. Ensure emergency stop or re-planning is triggered on imminent collisions.
- ID-2.11 Review candidate coverage patterns and algorithms suitable for the arena, estimate their efficiency and robustness, and select a baseline coverage strategy compatible with the chosen SLAM, costmaps, and robot kinematics.
- ID-2.12 Implement the chosen coverage planner as a Nav2-compatible node or behaviour tree, generating waypoints or virtual lanes that the global and local planners can follow to sweep the working area.
- ID-2.13 Evaluate coverage runs on test arenas, measure visited-area percentage, time, and path length, and tune coverage parameters to meet the required efficiency and robustness targets from the navigation requirements.
- ID-2.14 Design and implement a reliable behaviour that returns the robot from its last working location to the starting zone, handling map updates, dynamic obstacles, and safe final docking within the demonstration area.
- ID-2.15 Reserve time for final system-level debugging of navigation, resolving occasional failures, tuning edge-case behaviours, and confirming robustness before the Technical Demonstration and final assessment.

9.2.3 Work Package 3 – Perception

- ID-3.1 Physically mount the RGB-D camera to the Intel NUC, install and configure ROS 2 camera drivers. Verify colour, depth, and point-cloud streams in RViz for use by object detection and 3D perception modules.
- ID-3.2 Extract performance requirements and functional requirements from the Design Requirements Analysis to study colour-classification accuracy and detection robustness. Collate these in a checklist to guide dataset generation and algorithm evaluation.
- ID-3.3 Collect RGB-D data from the arena, define labelling rules, annotate cubes and bins, and perform dataset split into train/validation/test sets.
- ID-3.4 Apply colour jitter, noise, brightness changes, and rotations to training images to improve YOLO robustness to lighting change, motion blur, and unseen arena configurations.
- ID-3.5 Train a YOLO-based detector on the augmented dataset, monitoring training curves to achieve reliable cube and bin detection on validation data.

- ID-3.6 Evaluate the trained detector on the test set, computing precision and comparing performance against the minimum perception requirements.
- ID-3.7 Develop a ROS 2 point-cloud processing pipeline that subscribes to depth data, converts it into organised point clouds, applies filtering, and transforms them into the robot frame for later segmentation and 3D feature extraction.
- ID-3.8 Use RANSAC plane fitting and DBSCAN to isolate cube-sized obstacles above the floor while rejecting noise and distant clutter in the arena.
- ID-3.9 From segmented point-cloud clusters, compute 3D centroids, principal axes, sizes, and surface normals, transforming features into the arm base frame to provide grasp-relevant pose information for planners.
- ID-3.10 Fuse YOLO 2D detections with depth-based clusters to obtain consistent 3D cube and bin poses, resolving mismatches and occlusions.
- ID-3.11 Generate oriented 2D and 3D bounding boxes aligned with cube edges or principal axes, providing more accurate position and orientation estimates for grasp planning than simple axis-aligned boxes.
- ID-3.12 Perform final perception regression tests against key scenarios. Freeze configuration and model files. Hand over code, calibration data, and documentation to the team repository for long-term maintenance after the project.

9.2.4 Work Package 4 – Manipulation

- ID-4.1 Initialize and bench-test the standalone manipulator arm including firmware and ROS 2 driver setup, joint homing, limit verification, and basic motion commands.
- ID-4.2 Review all PR and FR about manipulation, check the real Leo arm payload and reach, and define a realistic grasping strategy that the hardware can actually achieve in the arena.
- ID-4.3 Implement forward kinematics using DH parameters to calculate the end effector attitude from joint angles.
- ID-4.4 Develop analytical or numerical IK solvers for 6-DOF robotic arms. Implement singularity avoidance, joint-constraint checking, and optimise computational efficiency.
- ID-4.5 Test the IK solver using diverse target attitudes within the workspace. Verify the solution’s accuracy, convergence speed, and positioning accuracy.
- ID-4.6 Design a top-down grasp-approach strategy for cube-shaped objects. Define pre-grasp offset distance, approach trajectory, and gripper orientation constraints to maximise grasping success rate.
- ID-4.7 Detect contact via position stall and error, apply a bounded preload offset (Δw), and adapt Δw on slip to hold objects safely without damage, and define the optimal transport posture.
- ID-4.8 Conduct extensive gripping tests on wooden blocks of different sizes and colours. Measure success rates under various conditions and optimise parameters.
- ID-4.9 Achieve smooth trajectory generation using trapezoidal velocity profiles. Ensure collision-free paths while considering robot base, arm links, and obstacles in the workspace.
- ID-4.10 Develop collision-avoidance behaviours between arm links, robot body, and other sensors. Implement emergency-stop triggers and safe recovery behaviours, ensuring compliance with safety requirements.
- ID-4.11 Develop bin-approach and object-release strategies. Calculate the placement trajectory considering bin size and opening geometry.
- ID-4.12 Conduct placement tests on objects and bin configurations. Measure placement accuracy success rate, and optimise release time and approach angle to improve reliability.
- ID-4.13 Allow a buffer period to resolve critical operational issues, perform final calibration, and conduct comprehensive system testing. Ensure the final demonstration is ready once all requirements have been verified.

9.2.5 Work Package 5 – Hardware Design

- ID-5.1 Assemble the Leo Rover base platform and perform initial software bring-up, including Leo installation, motor control, and basic ROS workspace setup.
- ID-5.2 Conduct comprehensive base-platform tests, including wheel calibration, odometry verification, motor performance, and basic mobility tests.
- ID-5.3 Design rigid mounts for the RGB camera and LiDAR on the Leo base, ensuring both have unobstructed fields of view and are compatible with the chosen payload sled and arm layout.
- ID-5.4 Design and dimension the payload sled for the Leo rover, supporting the arm and NUC while preserving the base footprint, sensor fields of view, and safe cable routing.
- ID-5.5 Physically install RGB-D camera and LiDAR sensors on the Leo Rover. Verify mechanical stability, unobstructed sensor FOVs, protected cable routing, and strain relief.
- ID-5.6 Mount the calibrated manipulator arm and gripper onto the payload sled on the Leo base. Align mechanical interfaces, install all fasteners, and verify joint clearances and rigidity before connecting power or data cables.
- ID-5.7 Plan battery and wiring for the base, NUC, sensors, and arm. Define connectors, fuses, and cable routes, avoiding moving joints. Ensure compliance with project safety and electrical guidelines.
- ID-5.8 Test the complete power system under operational loads. Measure voltage drops, current consumption, and battery runtime. Verify all components receive adequate power and emergency-stop functionality.
- ID-5.9 With full payload installed, perform tilt and braking tests to check static and dynamic stability. Confirm nothing fouls wheels and arm, and record any mechanical issues for redesign and adjustment.
- ID-5.10 Carry out final hardware maintenance, verification, and demonstration preparation, and ensure the robot is in optimal condition.

9.2.6 Work Package 6 – Systems Integration

- ID-6.1** Review all ROS 2 interfaces between navigation, perception, manipulation, and hardware. Define the final node graph and topics. Produce an integrated system architecture and test order for each subsystem.
- ID-6.2** Establish the complete ROS 2 communication framework, configure launch files, parameter servers, and TF transformation trees. Verify communication channels between all subsystems and ensure proper node initialisation sequences.
- ID-6.3** Define standardised message protocols, service interfaces, and action definitions for inter-subsystem communication, including navigation commands, perception outputs, and grasp requests.
- ID-6.4** Implement bidirectional communication pathways connecting navigation, perception, and manipulation modules. Handle data synchronisation, coordinate-frame transformations, and establish timing protocols to maintain system coherence during operation.
- ID-6.5** Design system-level fault-detection mechanisms and recovery procedures. Address scenarios including sensor failures, unsuccessful grasp attempts, navigation obstacles, and communication timeouts.
- ID-6.6** Validate integrated system performance on the physical hardware platform. Test sequential operation of navigation, perception, and manipulation subsystems under realistic conditions.
- ID-6.7** Execute complete mission scenarios from start to finish: autonomous navigation to the search area, object detection and localisation, grasping operation, return navigation, and colour-based sorting.
- ID-6.8** Document success rates, completion times, and failure modes. Identify system limitations and refine handling strategies.
- ID-6.9** Analyse logs from integrated trials to compute mission time, navigation accuracy, grasp success, and sorting success, and check that all system-level requirements and performance targets defined in the DRA are met.
- ID-6.10** Perform final system optimisation addressing all identified issues from the testing phases. Refine technical documentation and prepare deliverables for project handover.

9.3 Milestones and Deliverables

ID	Description	Type	Responsible	Due Date	Status	Note
D 1.1	Workplace Charter	Document	Nadiya Ivahnenko	S1-W3	Completed	
D 1.2	Design Requirements Analysis	Document	Jayant Bakolia	S1-W7	Completed	
D 1.3	Preliminary Design Review	Document	Yinfeng Li	S1-W11	Completed	
D 1.4	Final Design Review	Document	Mingyang Xu	S2-W7	Completed	
D 1.5	Technical Demonstration Videos	Video	Allan Binoy Issac	S2-W11	Ongoing	
D 2.1	SLAM Configuration Package	Algorithm	Jayant Bakolia	S1-W12	Completed	
D 2.2	NAV2 Configuration Package	Algorithm	Jayant Bakolia	S2-W5	Completed	
D 2.3	Obstacle Detection and Avoidance System	Functionality	Mingyang Xu	S2-W7	Completed	
D 2.4	Coverage Path Planning Algorithm	Algorithm	Mingyang Xu	S2-W9	Completed	
D 3.1	Raw RGB-D Dataset	Document	Nadiya Ivahnenko	S1-W10	Completed	
D 3.2	3D Perception Pipeline Module	Functionality	Nadiya Ivahnenko	S2-W9	Completed	
D 3.3	Integrated Detection System	Functionality	Allan Binoy Issac	S2-W11	Ongoing	
D 4.1	Forward and Inverse Kinematics Module	Algorithm	Yinfeng Li	S1-W12	Completed	
D 4.2	Top-Down Grasp Strategy	Algorithm	Bokai Qin	S2-W3	Completed	
D 4.3	Motion Planning	Algorithm	Bokai Qin	S2-W9	Completed	
D 4.4	Bin Placement System	Functionality	Bokai Qin	S2-W10	Completed	
D 5.1	Leo Rover Base Platform	Gadget	Nadiya Ivahnenko	S1-W3	Completed	
D 5.2	Radar and camera mount	Gadget	Jayant Bakolia	S1-W6	Completed	
D 5.3	Payload Sled	Gadget	Yinfeng Li	S2-W2	Completed	
D 6.1	Core ROS2 Infrastructure	Functionality	Mingyang Xu	S1-W12	Completed	
D 6.2	Error Detection & Recovery Module	Functionality	Bokai Qin	S2-W4	Completed	
D 6.3	End-to-End Mission Demonstration	Video	Jayant Bakolia	S2-W9	Pending	
D 6.4	Final Integrated System Release Package	Functionality	Allan Binoy Issac	S2-W11	Pending	
M 2.1	LiDAR Operational	Gadget	Mingyang Xu	S1-W8	Completed	
M 2.2	SLAM Mapping Functional	Functionality	Jayant Bakolia	S1-W12	Completed	
M 2.3	Localization Validated	Algorithm	Mingyang Xu	S2-W2	Completed	
M 2.4	NAV2 Stack Configured	Functionality	Jayant Bakolia	S2-W5	Completed	
M 2.5	Obstacle Avoidance Ready	Functionality	Mingyang Xu	S2-W7	Completed	
M 2.6	Coverage Algorithm Complete	Algorithm	Mingyang Xu	S2-W9	Completed	
M 3.1	Camera Operational	Gadget	Allan Binoy Issac	S1-W8	Completed	
M 3.2	YOLO Model Trained	Algorithm	Nadiya Ivahnenko	S1-W12	Completed	
M 3.3	3D Pipeline Complete	Functionality	Nadiya Ivahnenko	S2-W7	Completed	
M 3.4	2D-3D Fusion Complete	Algorithm	Nadiya Ivahnenko	S2-W9	Ongoing	
M 4.1	Hardware Operational	Gadget	Yinfeng Li	S1-W8	Completed	
M 4.2	IK Solver Validated	Algorithm	Yinfeng Li	S1-W12	Completed	

M 4.3	Grasp Strategy Implemented	Algorithm	Bokai Qin	S2-W5	Completed	
M 4.4	Motion Planning Complete	Algorithm	Bokai Qin	S2-W8	Completed	
M 4.5	Placement System Ready	Functionality	Bokai Qin	S2-W10	Completed	
M 5.1	Base Platform Ready	Gadget	Nadiya Ivahnenko	S1-W3	Completed	
M 5.2	Payload Sled Complete	Gadget	Yinfeng Li	S2-W2	Completed	
M 5.3	Sensors Mounted	Gadget	Jayant Bakolia	S1-W6	Completed	
M 5.4	Manipulator Integrated	Gadget	Bokai Qin	S2-W8	Ongoing	
M 5.5	Power System Ready	Gadget	Allan Binoy Issac	S2-W5	Completed	
M 5.6	Hardware Integration Done	Gadget	Mingyang Xu	S2-W9	Ongoing	
M 6.1	ROS2 Core Skeleton Running	Functionality	Mingyang Xu	S1-W12	Completed	
M 6.2	Subsystems Integrated on Common Interface	Functionality	Jayant Bakolia	S2-W3	Completed	
M 6.3	Integration Tests Passing	Functionality	Allan Binoy Issac	S2-W6	Completed	
M 6.4	End-to-End Mission Achieved	Functionality	Jayant Bakolia	S2-W8	Pending	
M 6.5	Final Integrated System Delivered	Functionality	Allan Binoy Issac	S2-W11	Pending	

Appendix A: Workplace Charter (version: 2.0)

Workplace Charter

Statement of Purpose

This document aims to establish the core principles and code of conduct required of an effective, inclusive, and respectful workplace environment. It emphasizes the need for the team to welcome and value diverse perspectives whilst maintaining a positive team dynamic. This document highlights professional behavioural standards that the team must abide by and introduces various conflict resolution mechanisms. It serves as a reference point for professional communication, collaboration, and project management. In cases of differing opinions or interpersonal conflicts, team members should follow the established procedures outlined herein to ensure smooth cooperation and prevent exclusion or misunderstanding. Any amendments to this document can be made only upon mutual agreement of all team members.

Statement of Principles and Commitments

We, as a team, will strive to maintain a collaborative, inclusive, and respectful environment that promotes development, learning, and innovation. We will uphold the below mentioned moral and professional standards.

1. **Inclusivity:** We recognise cultural and educational diversity as strengths that enhance our project outcomes.
2. **Teamwork:** We prioritise collective success over individual preference. We take collective responsibility for the outcomes and learn from our mistakes.
3. **Equality:** We are committed to providing equal access to opportunities, resources, and participation for all team members, ensuring that everyone can contribute fully and fairly to the project.
4. **Respect:** We respect the identity, background, and opinions of each member. We value diverse perspectives and treat all members fairly.
5. **Accountability:** We take responsibility for our individual tasks and contributions.
6. **Integrity:** We uphold honesty and transparency in all communication and decisions. Our decision-making process is collaborative (reflecting collective discussion and agreement) rather than individual authority.
7. **Neurodiversity:** We acknowledge that team members may have different cognitive profiles and ways of processing information, including neurodivergent conditions. We will respect these differences and, where needed, adjust how we collaborate (for example, meeting formats or information presentation) to support equitable participation and performance.

Team rules

We abide by the below mentioned set of rules to yield a systematic and productive work schedule:

1. **Working Hours**
 - Each member is expected to contribute approximately 5 hours per week (outside scheduled lectures and labs).
 - Each member is expected to attend and come prepared for two core team meetings per week, scheduled as follows:
 - o Tuesday: 11:30 - 12:30 (Information exchange and planning)
 - o Friday: 13:00 - 14:00 (Review of lab work, data discussion and coursework preparation)These meeting times apply to Semester 1 and will be reviewed and adjusted at the start of Semester 2 to reflect changes in the timetable.
 - If a member must miss a meeting for a legitimate reason, they should notify the team at least 12

hours in advance; in exceptional circumstances (e.g. an overlapping assessment deadline, sudden illness, or serious personal emergency), they should do so as early as reasonably possible.

- Minutes of Meeting is to be shared through email after each scheduled meeting.
- Project work is normally carried out during daytime on weekdays. Work outside these agreed hours (for example, in the evening or at weekends) should only occur by mutual consent.

2. Language

- The working language of the team must be English for all meetings, documentation, and correspondence.
- Members are encouraged to ask for clarification if communication barriers arise.
- All communication must be professional, and should be free from discriminatory, aggressive, or offensive language.

3. Modes of Communication

- **Microsoft Teams** – for daily coordination, and for instant messaging.
- **OneDrive** – for file sharing and documentation.
- **Weekly Meeting (in-person or online)** – to discuss progress and challenges.
- **Email** – for formal communication.
- **Notion**- for progress and task management.

4. Decision Making

- Decisions are made through consensus; if consensus is not reached, majority vote will apply.
- Major design changes must be discussed in a scheduled meeting with all members present.
- All major decisions are to be documented in shared project logs.

5. Task Allocation and Accountability

- Tasks are tracked using a shared task board on Notion.
- Members are expected to update progress before each weekly meeting.
- If a member is unable to complete a task on time, they must inform the team at least 24 hours in advance.

6. Professional Conduct

- We will respect deadlines and provide constructive feedback.
- Personal disagreements must not affect professional collaboration.
- All members will adhere to university guidelines on equality, diversity, and harassment prevention.
- Discriminatory behaviour (verbal or physical) will not be tolerated, and such instances will be seriously dealt with.

Daily Activity Conflict Avoidance

To minimize team conflicts, the following preventive measures will be implemented:

1. **Regular discussions:** Weekly meetings will be held to review progress and address issues identified by members, helping to identify potential friction points in advance. Project logs will be kept, discussed, and updated every meeting.
2. **Clear roles and responsibilities:** Clarify the responsibilities and division of labour of each member of the team. Tasks will be assigned based on individual strengths and expertise to avoid overlapping duties and conflicts arising from duplicated efforts.
3. **Work plan establishment:** For each team task, phase-specific goals and timelines will be defined at the outset to ensure efficiency and reduce disagreements caused by mismatched expectations.
4. **Respect of differences:** Allowing the difference between working methods and personal working style without affecting the team's goals. Pay attention to the cultural differences of the international team for example, festivals, and time concepts.
5. **After-work/activities:** Team Members are encouraged to prioritise personal non-work commitments. Members can participate in relevant clubs or school activities together, manage and release pressure reasonably, and check in with team member's wellbeing.

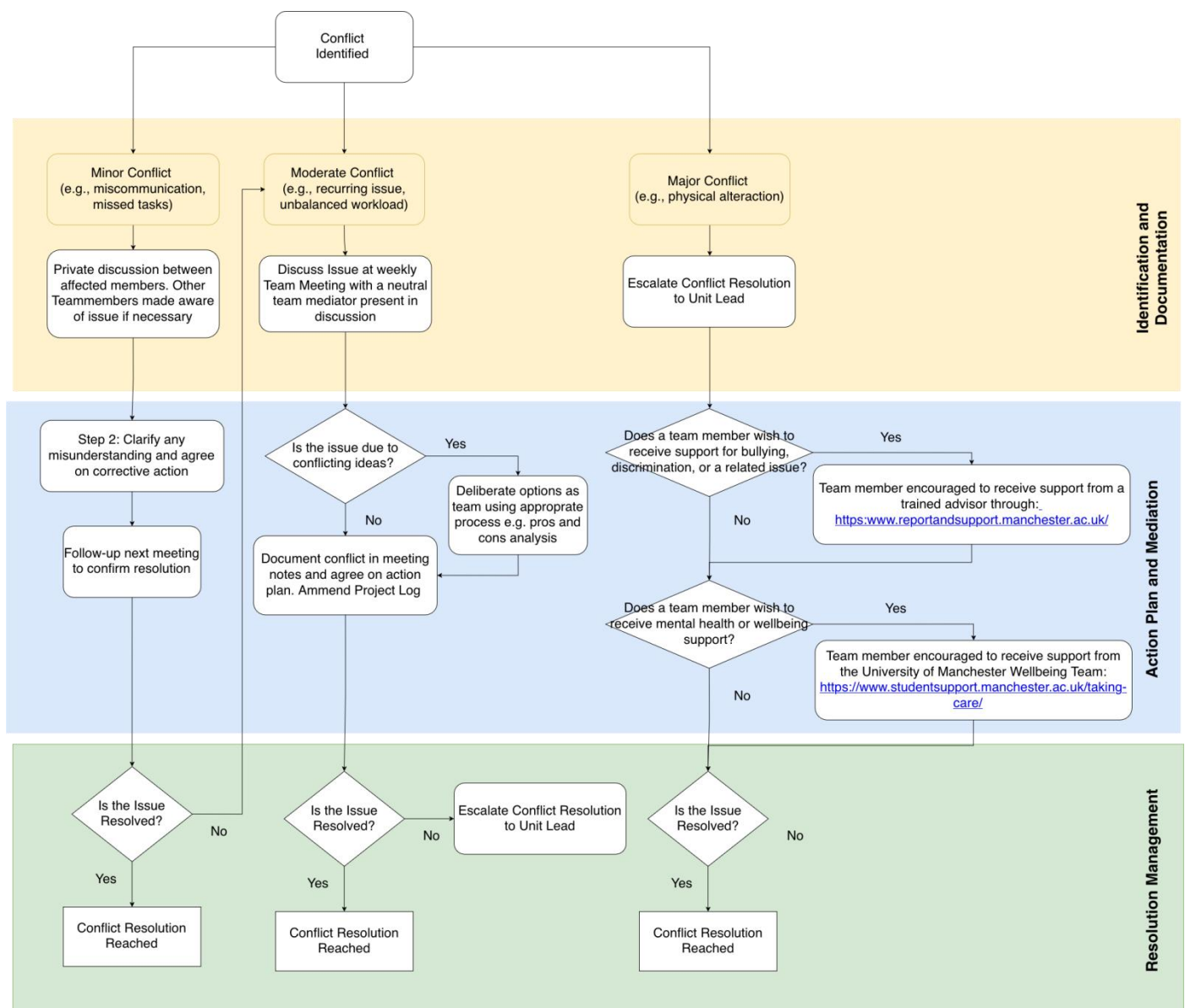
6. **Additional measures:** Team Members are encouraged to discuss personal issues which may affect their ability to work on the project, such as wellbeing or mental health concerns. These issues will be treated respectfully, and a resolution pathway will be discussed as a team.
7. **Pause and perspective:** If a mitigating circumstance or sudden issue arises, team meetings can be paused or rescheduled e.g. allowing for a 15-minute break. Any personal issues or conflicts will be deliberated with a neutral mediator in line with the conflict resolution flowchart.
8. **Further information** is available on the University's EDI webpages:
<https://www.manchester.ac.uk/connect/jobs/equality-diversity-inclusion/>
<https://www.manchester.ac.uk/connect/jobs/equality-diversity-inclusion/initiatives/>.

Conflict Resolution Flowchart

The flowchart below provides a structured framework for managing interpersonal or task-related conflicts within the team. It ensures that issues are addressed in a fair, transparent, and constructive manner while maintaining a positive and inclusive working environment.

The process categorizes conflicts into minor, moderate, and major levels, with corresponding escalation and mediation steps. It emphasizes early communication between affected members, encourages team-based deliberation for recurring or complex issues, and provides clear pathways for seeking external support in cases involving discrimination or wellbeing concerns.

By integrating documentation, follow-up actions, and escalation mechanisms, this process ensures accountability, consistency, and timely resolution of conflicts, thereby supporting effective collaboration and team harmony throughout the project lifecycle.



Appendix B: Design Requirement Analysis (version: 2.0)

DESIGN REQUIREMENTS ANALYSIS

AERO62520 - Robotic Systems Design Project

Version: 2.0
Date: December 12, 2025
Coursework ID: 03

PROJECT TEAM

Name	Student ID
Nadiya Ivahnenko	14367246
Bokai Qin	14137913
Jayant Bakolia	14158996
Mingyang Xu	14223377
Yinfeng Li	14187032
Allan Binoy Issac	10929893

Contents

Contents	1
List of Figures	2
List of Tables	2
1 Problem Statement	3
2 Concept of Operations	3
2.1 Operational environment	3
2.2 Operational phases	4
2.3 Safety, hardware-protection, and operability constraints	5
2.4 Operations Flow	5
3 Functional Requirements & Performance Requirements	8
3.1 Functional Requirements	9
3.1.1 Mission Management and Autonomy	9
3.1.2 Localization and environment understanding	10
3.1.3 Navigation and mobility	10
3.1.4 Target object perception	11
3.1.5 Grasping and pickup	11
3.1.6 Payload	12
3.1.7 Bin identification and deposition	12
3.1.8 Safety and compliance	13
3.2 Environmental requirements	14
3.2.1 Arena and Obstacles	14
3.2.2 Obstacle detection and avoidance	14
3.3 Maintainability & Operability requirements	15
3.3.1 Configuration and tuning	15
3.3.2 Logging and diagnostics	15
3.3.3 Human-machine interface	15
3.4 Mechanical / structural requirements	16
3.4.1 Robot structure	16
3.4.2 Payload sled	16
3.5 Other requirements (stretch / optional)	17
4 Requirements Verification Matrix	19
References	23

List of Figures

1	Flowchart of anticipated mission workflow and operation plan	7
---	--	---

List of Tables

1	Shared assumptions for functional requirements	8
2	Sources for functional requirements	8

1 Problem Statement

Autonomous mobile robots (AMRs) are engineering platforms that integrate perception, navigation, and manipulation to perform useful tasks with minimal human intervention. Building such systems provides an effective vehicle for teaching system-level robotics, as it forces students to design, implement, and validate interacting subsystems rather than isolated algorithms. This project uses that context to develop an autonomous mobile manipulator that can retrieve coloured objects in a previously unmapped indoor arena under strict time and safety constraints, serving as a capstone learning exercise for graduate robotics engineers.

The motivation for this work stems from a recognised gap in robotics education between modular algorithm development and full-system integration. While students frequently implement individual components—such as path planners, object detectors, or grasp controllers—in isolation, few academic exercises demand the coordinated operation of perception, planning, and actuation subsystems under real-world constraints of time, safety, and hardware limitations. Industrial deployment of AMRs for warehouse automation, manufacturing logistics, and service applications requires precisely this system-level competence, yet traditional coursework rarely provides opportunities to develop it. This project addresses that gap by requiring students to architect, implement, and validate a complete autonomous system capable of closed-loop mission execution in an uncertain environment.

The task excludes outdoor navigation, dynamic obstacle avoidance, multi-robot coordination, and unstructured object manipulation, focusing instead on the integration challenges inherent in combining mobile-base navigation with manipulator control in a perception-limited setting. Successful completion of this project will yield a functional autonomous mobile manipulator capable of unsupervised mission execution, demonstrating closed-loop perception-action cycles, adaptive behaviour in response to environment variability, and graceful handling of operational contingencies.

This Design Requirements Analysis (DRA) formalises that problem by translating the client’s conceptual brief into a structured set of functional requirements and associated performance requirements, each traceable to assumptions, sources, and verification methods. Together, they define a complete, measurable specification against which the autonomous object retrieval robot can be designed, implemented, and systematically qualified.

2 Concept of Operations

This section outlines the workflow of the Autonomous Object Retrieval Robot system, detailing the sequence of operations for localisation, mapping, object identification, navigation, robotic arm manipulation, object retrieval and deposition.

2.1 Operational environment

The operational environment consists of a bounded arena in which obstacles, target objects and storage bins are placed. Obstacles and target blocks are randomly placed and storage bins located adjacent to a designated starting point. All objects inside the arena are static. The arena is further characterised by consistent lighting conditions and a dry, flat surface. No map of the arena is available prior to robot deployment.

The target blocks possess the following characteristics:

- Three pre-selected colours from the set of red, green, blue, purple and yellow
- Each block is monochromatic with uniform shading
- Potentially different shapes

Which three target colours are included in the operation are to be determined by further testing and verification of the colours that are most accurately identifiable by the computer vision system.

The storage bins possess the following characteristics:

- Five storage bins, one each of red, green, blue, purple and yellow colour
- Each bin is monochromatic with uniform shading
- Cubical structure: 150 mm × 150 mm × 150 mm
- Open-top design for block deposition
- Positioned adjacent to the designated starting point

2.2 Operational phases

Within an approximately 20-minute demonstration window, the robot is required to autonomously search for target objects, detect and classify their colour, navigate to feasible pickup poses, grasp and transport them, return to the start area, and deposit each object into the bin whose colour matches the object. The system repeats this cycle until mission termination (e.g., all targets processed, operator abort, or demonstration time limit reached).

Phase 1: System initialisation

The robot is placed at the starting point. As the robot is initially turned off, it is to be turned on manually at the start. Sensors and the manipulator undergo calibration routines. If the calibration routine is unsuccessful for any electronic/mechatronic component, the calibration re-starts until all subsystems report ready status for the next phase to begin.

Phase 2: Environmental mapping

The robot performs a scan of the environment to generate a map to locate the targets, obstacles and storage bins. Boundaries of the arena are also identified during the scan. If the boundaries are not clear, the robot executes exploratory patterns inside the arena and performs scans iteratively until clear boundary definition is achieved. Mapping is maintained online and updated throughout subsequent navigation; the initial scan is intended to seed localisation, identify boundaries, and detect any immediately visible targets.

Phase 3: Target Selection and Target Approach

Next, distance calculation to each detectable target block is made. The nearest target object is selected to be retrieved and route planning is initiated. The optimal path is generated taking into account both the robot-target distance and the necessity to avoid collision with any objects. If the robot detects that the planned route involves navigation between two (or more) objects whose separating distance is less than the width of the robot, the route is adapted to find an alternate route that can avoid a collision. Successful route planning leads on to navigation. The robot autonomously navigates towards the target block along the planned path. The robot

stops at a preset safe distance from the target block to prevent collision and/or the target block moving underneath the robot. It is also ensured that the robot positions itself so that the target block lies within the working radius of the manipulator.

Phase 4: Transfer of target object

Once the robot stops, the manipulator commences operation and reaches out to the target block. The manipulator picks the target block using the end-effector (gripper) and the robot verifies that the target block is secured. The manipulator moves to a preset position so that the secured target block is outside the field of view (FOV) of the depth camera. This is done to avoid false detections and consequent unintended behaviours. Knowing that all storage bins are positioned adjacent to the designated starting point, the robot navigates towards the start area. The robot performs object detection to locate the storage bin that matches the colour of the secured target block. Bins of other colours are ignored and the robot autonomously navigates towards the target bin. The robot stops at a preset safe distance from the storage bin while ensuring that the bin is within the effective working radius of the manipulator. The manipulator moves to position the secured target block within the perimeters of the storage bin opening. At this point, the gripper releases the block. Visual verification is performed to ensure successful deposition and the manipulator is reset to a preset position. The robot inspects for additional target blocks, if any. If a target block is identified, the procedure for retrieval and deposition repeats. The mission is successful if at least one target block is correctly retrieved and deposited. It demonstrates accuracy in individual subsystems and an effective integration of all the subsystems.

2.3 Safety, hardware-protection, and operability constraints

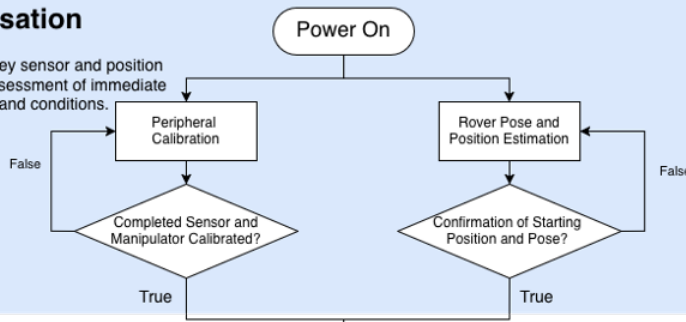
Parallel to mission performance, the system must satisfy safety, hardware-protection, and operability constraints. The robot shall provide an immediately accessible emergency stop that removes drive power, enforces safe translational and rotational speed limits, and avoid damaging any supplied hardware (base robot, sensors, bins, arena). Any mechanical modifications (e.g., payload sled and gripper) shall be structurally sound, reversible, and remain within arena clearance limits. All mission parameters (colour sets, object counts, thresholds, timeouts) shall be configurable without code changes. Time-stamped logging of robot pose, perception outputs, and high-level actions shall be recorded to support verification, diagnostics, and assessment.

2.4 Operations Flow

A flowchart of the anticipated mission workflow is provided below. This identifies key mission targets and sub operation, interconnecting the expected mission plan across the Navigation, Object Detection and Object Manipulation research areas.

I. Initialisation

System runs key sensor and position calibration. Assessment of immediate environment and conditions.



II. Environmental Navigation & Mapping

Environmental Survey - generation of navigation map of obstacles, objects, and storage bins

Initial survey of environment, identifying presence of arena features through sensor feedback: boundary, goal bins, obstacles and targets

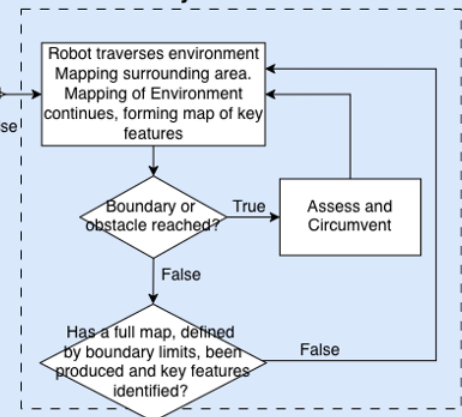
Are boundaries identified and has a full map been produced within boundary limits?

Key Feature Identification

Robot traverses environment Mapping surrounding area. Mapping of Environment continues, forming map of key features

Boundary or obstacle reached?

Has a full map, defined by boundary limits, been produced and key features identified?



III. Path Planning & Navigation

System plans path implementing position data drawn from previous environment mapping and optimal route is drawn. Further steps address navigation challenges and approaches

Input: Colour of objects to be identified and retrieved

Route Planning

Path plan a route of least resistance - maximise area traversed while minimising potential collisions with large obstructions

Robot moves autonomously across route

Survey known (required size) object locations from environment map

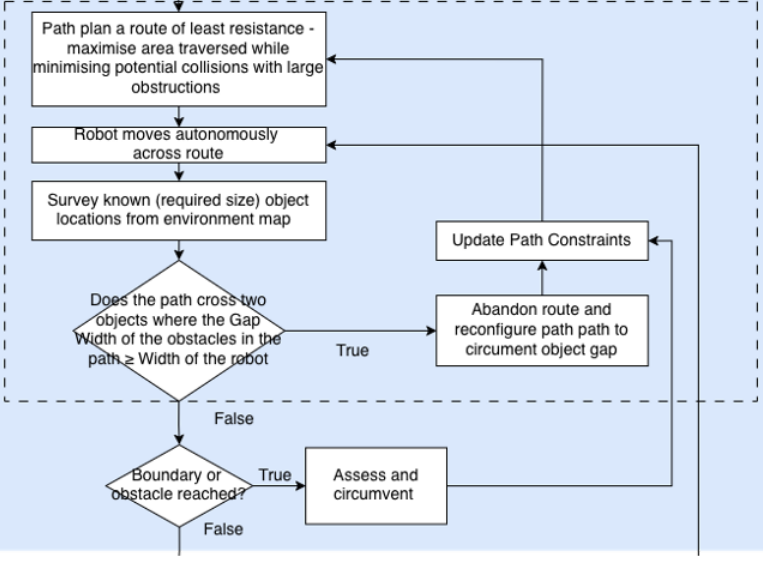
Does the path cross two objects where the Gap Width of the obstacles in the path \geq Width of the robot

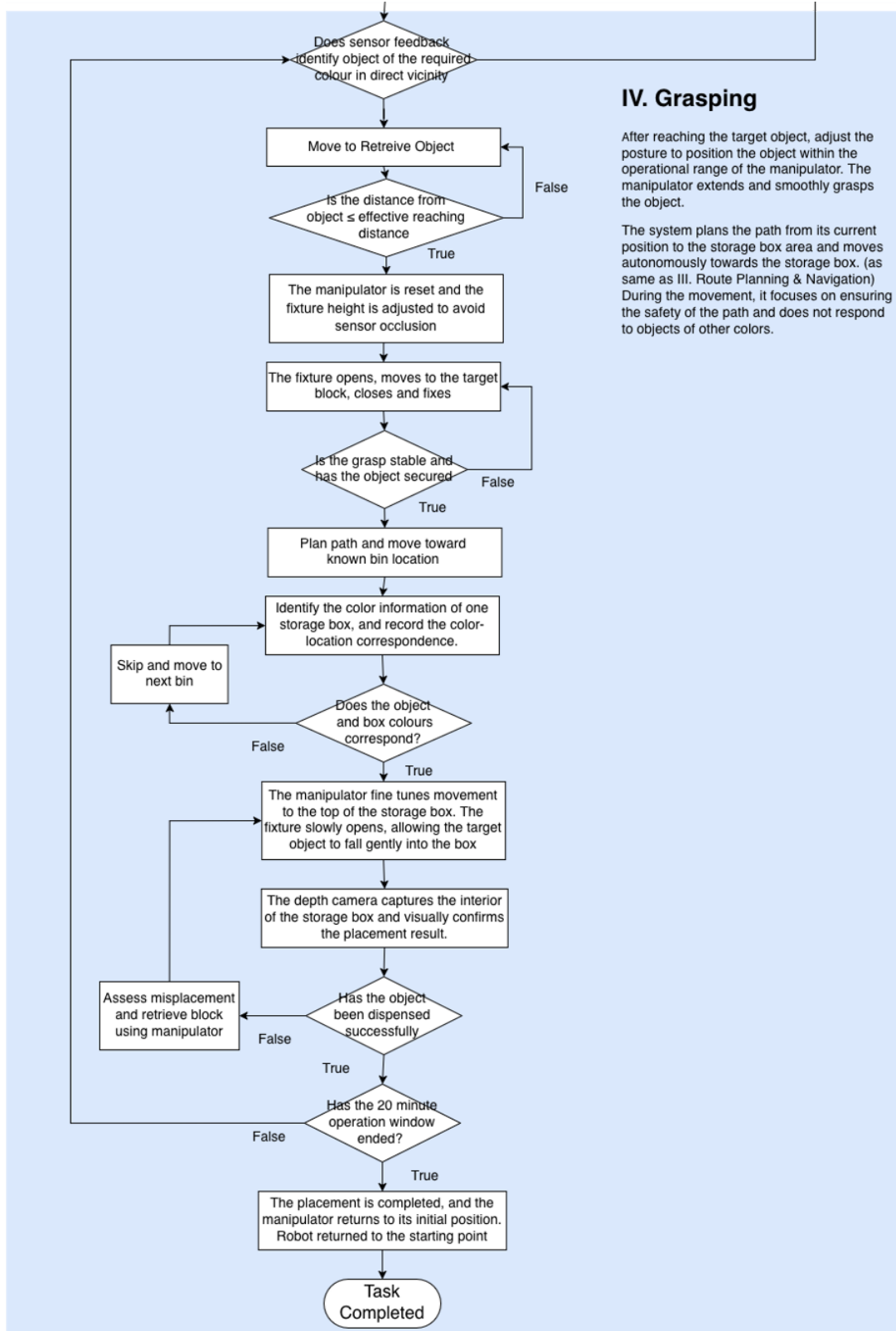
Update Path Constraints

Abandon route and reconfigure path path to circumvent object gap

Boundary or obstacle reached?

Assess and circumvent





IV. Grasping

After reaching the target object, adjust the posture to position the object within the operational range of the manipulator. The manipulator extends and smoothly grasps the object.

The system plans the path from its current position to the storage box area and moves autonomously towards the storage box. (as same as III. Route Planning & Navigation) During the movement, it focuses on ensuring the safety of the path and does not respond to objects of other colors.

Figure 1: Flowchart of anticipated mission workflow and operation plan

3 Functional Requirements & Performance Requirements

This section defines the Functional Requirements for the Autonomous Object Retrieval Robot system for governing autonomous search, collection, and deposit of target objects.

The following programme-level assumptions scope the environment and demonstration conditions against which requirements are written and verified.

Table 1: Shared assumptions used across all functional requirements.

ID	Assumption
A1	Demonstration window is approximately 20 minutes; manual pause/reset is permitted.
A2	Three different coloured wooden objects; bins at start with top opening; bin colour matches object colour.
A3	Flat indoor arena; no ramps; rigid, approximately uniform-colour perimeter walls; static obstacles; at least one valid path; obstacle contact penalised.
A4	Fixed start pose; random target locations; no pre-provided obstacle map or locations.
A5	On-robot sensors only by default; any “external” sensor must be deployed by the robot itself.
A6	Variable indoor lighting; no fixed external lighting rigs provided.
A7	University hardware must be protected; an emergency stop (E-stop) is required.
A8	Gripper/payload reconfiguration is allowed if reversible and non-damaging.

Requirements are derived from the customer brief and clarifications, and are authored for clarity and verifiability per good-practice guidance.

Table 2: Sources and interpretation notes used to derive the requirements.

ID	Source	Scope
S1	Customer Requirements Statement	Autonomous retrieval and deposit into matching bins.
S2	Q&A sessions with teaching staff (Weeks 4/5)	Random object placement; bins at start; static obstacles; 20-minute window; manual intervention allowed; no prior obstacle placement info.
S3	University safety guidance (implied)	E-stop; protect hardware; safe speeds; operator protocols.
S4	NASA SE Handbook [1], Appendix C	Writing style and verification quality: clarity, testability, traceability.

Functional requirements are grouped into topical subsections (e.g. mission, navigation, perception) and presented as sub-sub-section heading [FR-#]. Each entry contains exactly one testable statement, a brief rationale, and inline references to assumptions and sources using A# and S#. The IDs follow the stable format FR-# (e.g., FR-12) to enable bidirectional traceability.

3.1 Functional Requirements

3.1.1 Mission Management and Autonomy

3.1.1.1 [FR-1] The system shall execute the mission autonomously.

Rationale: Proves end-to-end autonomy aligned to customer requirements and the demo expectation “navigate–detect–retrieve–return–deposit.”

Assumptions: A1, A2 *Sources:* S1, S2

[PR-1.1] The system shall complete the full autonomous mission by retrieving and depositing target objects within 20 minutes of initiation.

Rationale: The demonstration time is 20 minutes.

3.1.1.2 [FR-2] The system shall operate from a fixed starting position with target objects at random locations in the arena.

Rationale: Establishes a deterministic initial condition for localization and return-to-start behaviour (bins colocated at the start area) while enforcing robustness to unknown environments via random target placement.

Assumptions: A4 *Sources:* S1, S2

[PR-2.1] The system shall operate from a fixed starting position with target objects at random locations within 10 minutes.

Rationale: The 10-minute threshold is derived from a strategic allocation of the total 20-minute mission window, this duration accounts for worst-case search scenarios. So allocating 50% of the total time to the initial approach ensures that the robot can reach the target reliably even under suboptimal conditions.

3.1.1.3 [FR-3] The system shall handle at least one but may handle multiple target objects per demonstration run.

Rationale: Multiple objects reflect S2/S3 and demonstrate mission loop robustness.

Assumptions: A2 *Sources:* S2

[PR-3.1] The system shall handle between one and three target objects within 20 minutes per demonstration, without incorrect operation.

Rationale: The demonstration time is 20 minutes.

3.1.1.4 [FR-4] The system shall terminate the mission autonomously when all detected targets are deposited or when the configured time limit is reached.

Rationale: Deterministic mission end supports safe operations and grading within time limit.

Assumptions: A1 *Sources:* S2

[PR-4.1] The mission termination mechanism shall trigger reliably within 5 seconds of the time limit being reached or the final deposit being confirmed.

Rationale: This 5-second limit allows the robot to safely stop moving and save data, ensuring nothing breaks and no information is lost before the mission ends.

3.1.1.5 [FR-5] The system shall support safe pause and resume after manual intervention during the demonstration window.

Rationale: Manual intervention is allowed; pause/resume ensures safety and continuity.

Assumptions: A1 *Sources:* S2

[PR-5.1] Within 3 seconds of the operator issuing a pause/resume command, the system shall complete the corresponding pause/resume mode transition.

Rationale: This 3-second limit ensures the robot responds quickly enough to feel safe to the user, and also gives the system time to stop or start the motors smoothly without damage.

3.1.1.6 [FR-6] The system shall operate without a pre-provided map of obstacle locations.

Rationale: Forces online perception/coverage and generalisation beyond prior maps.

Assumptions: A4 *Sources:* S2

[PR-6.1] The system shall build a map based on the site conditions in each run with a positional error of no more than 50 mm.

Rationale: This 50 mm accuracy limit ensures the created map is precise enough for the robot to safely avoid hitting obstacles and to find targets.

3.1.2 Localization and environment understanding

3.1.2.1 [FR-7] The system shall localize using only sensors carried on the robot.

Rationale: Matches constraint of on-robot sensing.

Assumptions: A5 *Sources:* S2

3.1.2.2 [FR-8] The system shall support online map-building during navigation.

Rationale: Online mapping supports unknown arenas (no prior map).

Assumptions: A4 *Sources:* S2

[PR-8.1] The online map update shall be available to the path planner with a maximum latency of 5 seconds from sensor data acquisition.

Rationale: This 5-second limit ensures the robot updates its path quickly enough to avoid hitting new obstacles, and also gives the computer enough time to process the map data without freezing.

3.1.3 Navigation and mobility

3.1.3.1 [FR-9] The system shall navigate to a goal pose near a detected target.

Rationale: Goal approach is necessary to enable grasping after detection.

Assumptions: A2, A4 *Sources:* S1

[PR-9.1] The system shall stop within 50 mm and 10 degrees of the calculated goal position near the target object.

Rationale: These accuracy limits are based on the map accuracy, which ensure the robot stops close enough and faces the right way for the gripper, in order to successfully reach and grab the object, while a larger error would cause the grasp to fail.

3.1.3.2 [FR-10] The system shall navigate from any pickup location back to the start area to access the storage bins.

Rationale: Return-to-start is required because bins are at the start area.

Assumptions: A2 *Sources:* S2

[PR-10.1] The system shall successfully navigate from any pickup location back to the start area with in 10 minutes.

Rationale: Allocating 50% of the total time to the final dispose ensures that the robot can reach the target reliably even under suboptimal conditions.

- 3.1.3.3 [FR-11] The system shall detect when the current planned path becomes untraversable due to obstacles or insufficient free space and shall adjust its motion to avoid the obstruction while continuing to seek the goal.

Rationale: Handling untraversable regions prevents the robot from attempting to move into spaces it cannot safely traverse, while preserving the ability to reach the goal when a valid path exists.

Assumptions: A3 *Sources:* S2

[PR-11.1] When the system determines that the current path is untraversable, it shall initiate an avoidance response (such as re-planning an alternative route or coming to a controlled stop) and commence safe motion or reach a controlled stop within 1 minute.

Rationale: This one-minute limit ensures the robot has enough time to safely react to obstructions without getting stuck or making errors due to rushing.

3.1.4 Target object perception

- 3.1.4.1 [FR-12] The system shall detect target objects by their colour in the arena.

Rationale: Detectability of target objects is fundamental to mission success.

Assumptions: A2, A6 *Sources:* S1, S2

[PR-12.1] The system shall detect target objects whose minimum width along any of its axis lies between 20 mm and 45 mm.

Rationale: The minimum dimensions along any of the target object's axis should range from 20 mm to 45 mm, and the detection range of the system shall cover all sizes within this interval.

- 3.1.4.2 [FR-13] The system shall classify each detected target's colour as one of the allowed set used in the arena.

Rationale: Colour classification is required to choose the correct bin.

Assumptions: A2 *Sources:* S1, S2

[PR-13.1] The system shall correctly classify the every target colour under typical lab lighting conditions, which ranges from 300 to 1000 lux.

Rationale: the system must make zero classification errors on the specific objects handled during the demonstration, to prove its perception system is functional. 300 to 1000 lux is a typical indoor lighting condition.

- 3.1.4.3 [FR-14] The system shall estimate a pickup pose for the target object suitable for grasping.

Rationale: Estimating a pickup pose enables reliable grasping given cube geometry.

Assumptions: A2 *Sources:* S1

[PR-14.1] The deviation between the pickup pose calculated by the system and the center of the target object shall not exceed within 10 mm.

Rationale: This 10 mm limit ensures the object is centered enough for a secure and successful hold.

3.1.5 Grasping and pickup

- 3.1.5.1 [FR-15] The system shall approach the target object to a defined pre-grasp pose before actuating the gripper.

Rationale: Pre-grasp staging improves success and avoids collisions.

Assumptions: A3 Sources: S1

[PR-15.1] The entire process from initiating approach to the target object to completing the approach and pre-grasp sequence shall not exceed 30 seconds

Rationale: This 30-second limit allows the robot to move slowly for a precise and safe alignment, and ensures it doesn't waste too much time needed for the rest of the mission.

- 3.1.5.2 [FR-16] The end-effector shall secure and lift the target object to a minimum height for transporting the object.

Rationale: Secure lift to transport height prevents loss during transit.

Assumptions: A3 Sources: S1

[PR-16.1] During the demonstration, the system shall grasp the target object, lift it to a height above the robot platform, and hold it securely without dropping until the object is placed into the box of the corresponding color.

Rationale: Lifting above the platform prevents dragging on the floor. Holding it until the box ensures the object stays safe during the ride, proving the grip works for the full task.

- 3.1.5.3 [FR-17] The gripper shall avoid causing permanent damage to the target objects during normal operation.

Rationale: Non-destructive handling protects university hardware (objects/fixtures).

Assumptions: A7 Sources: S3

[PR-17.1] During the single demonstration run, the gripper shall apply a clamping force of 1.5 N to the target object, resulting in 0 mm of permanent surface indentation or visible damage.

Rationale: According to the official documentation of myCobot, the max gripping force of the gripper is approximately $\leq 1.5N$.

3.1.6 Payload

- 3.1.6.1 [FR-18] The payload system shall retain the target object during mission motions without dropping them.

Rationale: Retention under motion prevents drops and retries.

Assumptions: A3 Sources: S1

[PR-18.1] The payload system shall retain the target object when the speed is less than or equal to 1 km/h.

Rationale: The robot shall maintain system stability within the specified speed range, with its maximum operating speed being 1 km/h.

3.1.7 Bin identification and deposition

- 3.1.7.1 [FR-19] The system shall detect the storage bins at the start area and identify each bin's colour.

Rationale: Bin recognition at start area is essential to complete the loop.

Assumptions: A2 Sources: S2

- 3.1.7.2 [FR-20] The system shall select the bin whose colour matches the carried target object's colour classification.

Rationale: Colour-matching deposit is explicitly required.

Assumptions: A2 *Sources:* S1, S2

[PR-20.1] The system shall correctly classify the coloured bin that matches the color of the target object under typical lab lighting conditions, which ranges from 300 to 1000 lux.

Rationale: The system must make zero classification errors on the coloured bin during the demonstration, to prove its perception system is functional. 300 to 1000 lux is a typical indoor lighting condition.

3.1.7.3 [FR-21] The system shall place or drop the target object through the bin opening.

Rationale: Successful placement through the opening is the measurable completion criterion.

Assumptions: A2 *Sources:* S1

[PR-21.1] Ideally, the system shall place the object accurately at the center of the bin, while a tolerance range under 70 mm shall be allowed.

Rationale: The tolerance corresponds to the opening size of the bin. This tolerance is determined according to the 150 mm opening size of the target box; so, the offset of the placement point relative to the bin's center coordinates can be limited within a 70 mm.

3.1.8 Safety and compliance

3.1.8.1 [FR-22] The system shall have an immediately accessible emergency stop that removes drive power.

Rationale: E-stop is a mandatory university safety control.

Assumptions: A7 *Sources:* S3

[PR-22.1] Actuation of the emergency stop shall result in the power cut from the drive actuators within 5 seconds.

Rationale: This five-second limit ensures power is cut quickly for safety, while allowing a short interval to prevent the sudden shock from breaking the robot's hardware.

3.1.8.2 [FR-23] The system shall transition to a defined safe state on critical faults (loss of localization, sensor failure, collision detection).

Rationale: Safe state on critical faults mitigates risk to people/hardware.

Assumptions: A7 *Sources:* S3

[PR-23.1] The safe state transition shall activate within 3 seconds when critical fault occurrences.

Rationale: This three-second limit ensures the robot stops moving almost immediately to prevent collisions or damage, while allowing the software just enough time to process the error signal and shut down motors safely.

3.1.8.3 [FR-24] The system shall have a limit on translational and rotational speed for safe operation.

Rationale: Speed limiting allows safe demos in constrained indoor arenas.

Assumptions: A3, A7 *Sources:* S2, S3

[PR-24.1] The maximum movement and rotation speeds of the robot shall be of 1 km/h and shall be adjustable, which meets the safety requirements specified in the operational environment.

Rationale: This velocity limit of 1 km/h prevents accident by giving enough time to react. The adjustability allows the robot to slow down further in crowded areas, ensuring it is always safe for the specific environment.

- 3.1.8.4 [FR-25] The system shall avoid damaging anything including university hardware (base robot, sensors, bins, obstacles, arena).

Rationale: Non-damage requirement protects shared assets.

Assumptions: A7 *Sources:* S2, S3

[PR-25.1] The system shall complete the demonstration runs with zero recorded instances of damage to any provided hardware

Rationale: This PR strictly protecting hardware and following lab equipment usage rules.

3.2 Environmental requirements

3.2.1 Arena and Obstacles

- 3.2.1.1 [FR-26] The system shall be designed to operate on the provided flat indoor arena floor without ramps.

Rationale: Matches arena's flat indoor floor.

Assumptions: A3 *Sources:* S2

[PR-26.1] The system shall be designed to operate on flat indoor surfaces and shall be adaptable to slightly uneven floor conditions.

Rationale: Defining the system's ability to adapt to minor changes in the floor while staying within the intended indoor arena conditions.

- 3.2.1.2 [FR-27] The system shall recognize arena boundaries of approximately uniform colour and avoid breaching them.

Rationale: Boundary recognition prevents leaving the arena or scraping walls; marks are penalised.

Assumptions: A3 *Sources:* S2

3.2.2 Obstacle detection and avoidance

- 3.2.2.1 [FR-28] The system shall detect only static obstacles within the sensing envelope.

Rationale: Static obstacle detection is required due to ad-hoc arena objects. There are no moving objects in the arena.

Assumptions: A3 *Sources:* S2

[PR-28.1] The system shall detect obstacles presenting a minimum vertical cross-section of 150 mm × 150 mm facing the robot at a distance of up to 3 m.

Rationale: The vertical cross-section of 150 mm × 150 mm represents a standard solid object that could damage the robot. This requirement ensures the perception system ignores negligible pieces but reliably catches real threats.

- 3.2.2.2 [FR-29] The system shall plan paths that avoid contact with detected obstacles.

Rationale: Avoiding contact aligns with penalties and hardware protection.

Assumptions: A3, A7 *Sources:* S2, S3

[PR-29.1] A minimum clearance distance of 100 mm shall be maintained between the planned path and obstacles.

Rationale: The minimum clearance distance of 100 mm makes sure there is enough space from obstacles to help avoid collision damage.

- 3.2.2.3 [FR-30] The system shall stop motion and trigger a recovery behaviour if a collision or near-collision is detected.

Rationale: Safe stop/recovery mitigates near-misses or unintended contact.

Assumptions: A7 *Sources:* S2, S3

[PR-30.1] When detecting any obstacle within a static proximity radius of 100 mm, the system shall stop within 5 second.

Rationale: This requirement ensures the robot stops immediately whenever it gets too close to any object, which guarantees safety.

3.3 Maintainability & Operability requirements

3.3.1 Configuration and tuning

- 3.3.1.1 [FR-31] All mission parameters (colour set, number of objects, thresholds, timeouts) shall be configurable without code changes.

Rationale: Parameterisation enables rapid tuning across arenas without code changes; improves testability (Appx C).

Assumptions: A4, A6 *Sources:* S4

[PR-31.1] During the demonstration run, the system shall read and apply 100% of the mission parameters from an external configuration file within 5 seconds.

Rationale: Using an external file allows us to change the robot's settings easily without rewriting complex computer code. The 5-second limit ensures the robot starts the mission quickly without making the user wait.

3.3.2 Logging and diagnostics

- 3.3.2.1 [FR-32] The system shall log time-stamped pose, perception outputs, and high-level actions during each run.

Rationale: Logging provides evidence for final video and verification.

Assumptions: A1 *Sources:* S2, S4

[PR-32.1] During each operation, the system shall record the robot's pose with time-stamps, sensory outputs, and high-level operations, and the data loss rate over time should be under 5%. In other words, the information recorded by the system shall cover more than 95% of the robot's runtime.

Rationale: This requirement makes sure there are sufficiently complete records to verify behaviour and fix problems.

- 3.3.2.2 [FR-33] The system shall provide a means to export run logs in a documented format after a run.

Rationale: Exported logs aid assessment, debugging, and traceability.

Assumptions: A1 *Sources:* S4

3.3.3 Human-machine interface

- 3.3.3.1 [FR-34] The system shall have a single operator command to start the autonomous mission.

Rationale: Single-command start simplifies demos and reduces operator error.

Assumptions: A1 Sources: S4

[PR-34.1] The start command shall result in the initiation of autonomous movement within 3 seconds.

Rationale: 3 seconds ensures the system starts quickly when commanded and making operation more efficient.

3.3.3.2 [FR-35] The system shall support the modes Auto, Pause, Resume, and Abort.

Rationale: Mode control aligns with permitted manual intervention and safety.

Assumptions: A1 Sources: S2, S3

[PR-35.1] The start command shall result in the initiation of autonomous movement within 3 seconds, ensuring the system starts quickly when commanded and making operation more efficient.

Rationale: This requirement makes sure switching modes is fast enough to keep operation easy and efficient.

3.3.3.3 [FR-36] The system shall allow safe operator intervention to reposition the robot and then resume autonomous operation.

Rationale: Safe operator repositioning is allowed and often necessary.

Assumptions: A1 Sources: S2

3.4 Mechanical / structural requirements

3.4.1 Robot structure

3.4.1.1 [FR-37] Added mechanical structures shall remain within arena vertical and horizontal boundaries.

Rationale: Mechanical structures must fit within arena boundaries to avoid collisions and penalties.

Assumptions: A3 Sources: S2, S3

[PR-37.1] No additional mechanical structure can alter the robot's existing maximum lateral and longitudinal dimensions, which is of 450 mm × 440 mm.

Rationale: This requirement makes sure modified parts fit the arena's space limits so the robot can move normally.

3.4.1.2 [FR-38] Mechanical modifications shall be reversible and shall not permanently alter/damage the base robot or the peripherals.

Rationale: Reversibility protects university assets and supports maintenance.

Assumptions: A7, A8 Sources: S2, S3

3.4.2 Payload sled

3.4.2.1 [FR-39] The team shall design and deploy a payload sled that shall provide mounting locations for all sensors and on-robot compute units.

Rationale: Sled must host sensors/compute while respecting base footprint.

Assumptions: A3, A8 Sources: S2

[PR-39.1] The payload sled shall not increase the base robot footprint of 450 mm × 440 mm.

Rationale: This requirement makes sure modifications do not increase the robot's footprint and keeping it within space constraints.

3.4.2.2 [FR-40] The payload sled shall position each sensor to avoid occluding required fields of view.

Rationale: Avoiding occlusions is necessary for sensor performance.

Assumptions: A5, A6 *Sources:* S1

3.4.2.3 [FR-41] The payload sled shall support the combined mass of mounted sensors and compute units with adequate stiffness.

Rationale: Stiffness prevents misalignment under motion loads.

Assumptions: A3 *Sources:* S1

[PR-41.1] The payload sled shall be capable of supporting the total weight of all sensors and computing units, which is 3 kg, shall possess sufficient stiffness, and shall maintain structural integrity during operation

Rationale: This requirement makes sure the structure is stable enough to avoid deformation that would harm sensor performance.

3.4.2.4 [FR-42] The payload sled shall provide protected cable routing and strain relief to prevent snagging.

Rationale: Protected routing prevents snags with obstacles/floor.

Assumptions: A3 *Sources:* S3

[PR-42.1] All cable routing shall maintain a minimum distance of 50 mm from the ground.

Rationale: This requirement prevents cables from colliding or tangling with other things and helping keep equipment safe.

3.4.2.5 [FR-43] The payload sled shall not obstruct access to the emergency stop, power switches, or other safety interfaces.

Rationale: Access to E-stop/power is a safety requirement.

Assumptions: A7 *Sources:* S3

3.4.2.6 [FR-44] The payload sled shall use materials and finishes that avoid sharp edges and burrs.

Rationale: Safe edges/finishes reduce injury and cable damage risks.

Assumptions: A7 *Sources:* S3

[PR-43.1] A minimum clearance of 15 millimeters shall be maintained around the emergency stop button to facilitate manual operation.

Rationale: Since the robot doesn't have much assembly space and the layout is pretty tight, we leave a 15 mm gap around the button, this is to make sure we've got enough room to press it manually even with the limited space.

3.5 Other requirements (stretch / optional)

3.5.0.1 [FR-45] The system may retrieve multiple target objects in one trip or one-by-one, selectable via configuration.

Rationale: Batch vs one-by-one retrieval is permitted and can improve throughput.

Assumptions: A2 *Sources:* S2

[PR-44.1] All exposed edges of the payload sled shall be smoothly polished with a minimum 1 mm radius or chamfer, with no sharp corners.

Rationale: This requirement reduces the risk of injury and equipment damage by removing sharp edges.

- 3.5.0.2 [FR-46] The system may prioritize which target to retrieve next based on distance, visibility, or colour.

Rationale: Target prioritisation (distance/visibility/colour) can reduce time-to-success in the 20 min window.

Assumptions: A1, A2 *Sources:* S2

- 3.5.0.3 [FR-47] The system may confirm deposit via a feedback method (e.g. vision).

Rationale: Deposit confirmation reduces false positives and rework

Assumptions: A2 *Sources:* S2, S4

[PR-46.1] The system shall evaluate all detected target objects and compute a priority score based on distance, visibility, and colour within 5 seconds of each decision cycle.

Rationale: This five-second rule ensures the robot thinks fast. By checking every option quickly, it picks the easiest target to grab next without wasting time sitting still, ensuring target objects are prioritised in time and making the mission more efficient.

5 Requirements Verification Matrix

Project Area	Theme	Functional Requirement ID	Functional Requirement Statement	Performance Requirement ID	Performance Requirement Statement	Verification Success Criteria	Verification Method	Facility or Lab	Phase	Acceptance Requirement?	Pre-Performance Acceptance Requirement?
Functional	Mission Management and Autonomy	FR-2	The system shall operate from a fixed starting position with target objects at random locations in the arena.	PR-2	The system shall successfully locate and approach the first target object in $\geq 95\%$ of demonstration runs.	System must locate and approach the first target in ≥ 9 of 10 demonstration runs within the arena.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	Yes	No
Functional	Mission Management and Autonomy	FR-3	The system shall handle at least one but may handle multiple target objects per demonstration run.	PR-3	The system shall handle between one and three target objects per demonstration without performance degradation.	Run trials with 1, 2, and 3 objects placed. For each configuration: navigation, object detection and manipulation - success rates and completion time must remain within $\pm 10\%$ of baseline one-object performance.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Mission Management and Autonomy	FR-4	The system shall terminate the mission autonomously when all detected targets are deposited or when the configured time limit is reached.	PR-4	The mission termination mechanism shall trigger reliably within 5 seconds of the time limit being reached or the final deposit being confirmed.	Perform demonstration runs, with two mission end criteria. 1. Time Limit Expiration 2. Final Object Placed. Termination event must occur within ≤ 5 s of end criteria being met.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Mission Management and Autonomy	FR-5	The system shall support safe pause and resume after manual intervention during the demonstration window.	PR-5	Within 3 seconds of the operator issuing a pause/resume command, the system shall complete the corresponding pause/resume mode transition.	During operation testing measure response time across >10 commands. System must respond within ≤ 3 s of operator input.	System Design Analysis	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	No	Yes
Functional	Mission Management and Autonomy	FR-6	The system shall operate without a pre-provided map of obstacle locations.	PR-6	The system shall build a map based on the site conditions in each run with a positional error of no more than 50mm.	Comparison of generation map visualisation with real-world environment measurement. Error must be ≤ 50 mm across 5-10 runs.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Localisation and Environment Understanding	FR-7	The system shall localise using only sensors carried on the robot.	n/a	n/a	In localisation testing and payload sled construction, only onboard sensors should be used. No additional interaction with environment or objects should occur.	System Structure Inspection	Workshop/Lab Workspace	System Design	Yes	Yes
Functional	Localisation and Environment Understanding	FR-8	The system shall support online map-building during navigation.	PR-8	The online map update shall be available to the path planner with a maximum latency of 5 seconds from sensor data acquisition.	Perform environmental changes during testing such as new object placement or movement. Navigation and Planning update latency shall be ≤ 5 s.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Navigation and Mobility	FR-9	The system shall navigate to a goal pose near a detected target.	PR-9	The system shall accurately navigate to the target pose adjacent to the target object in accordance with the specified distance requirements.	During demonstration runs change object pick-up locations across >10 runs. Robot must reach goal pose within a test determined distance margin.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Navigation and Mobility	FR-10	The system shall navigate from any pickup location back to the start area to access the storage bins.	PR-10	The system shall successfully navigate from any pickup location back to the start area.	During demonstration runs change robot starting locations across >10 runs. Robot must return to starting point autonomously in 8 out of 10 runs.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Navigation and Mobility	FR-11	The system shall identify untraversable gaps narrower than the robot's minimum safe width and re-plan a path.	PR-11	After detecting an impassable narrow gap, the system shall re-plan the route and commence movement within 1 minute.	In demonstration runs, place objects within decreasing distance of each other across 10 runs. System must detect and reroute appropriate circumvention or traversal within ≤ 60 s.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Target Object Perception	FR-12	The system shall detect wooden target objects in the arena.	PR-12	The system shall be capable of detecting target objects ranging in size from 20 to 150 millimeters.	Through demonstration testing collect objects with widths varying 20 - 150mm range for collection. System should be capable of detecting items in 8 out of 10 runs.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Target Object Perception	FR-13	The system shall classify each detected target's colour as one of the allowed set used in the arena.	PR-13	The system shall correctly classify the target colour with an accuracy of $\geq 90\%$ under typical lab lighting conditions.	With same typical lighting variation/lightning conditions of the arena. Show 10-50 samples of each colour. Colour classification must have a success rate of $\geq 90\%$ overall.	System Design Analysis	Workshop/Lab Workspace	System Design	Yes	No
Functional	Target Object Perception	FR-14	The system shall estimate a pickup pose for the target object suitable for grasping.	PR-14	The deviation between the pickup pose calculated by the system and the center of the target object shall not exceed ± 10 millimeters.	Measure difference between estimated pickup pose and object center across 10 demo runs. Mean error must be ≤ 10 mm across test runs.	Demonstration	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	No	No
Functional	Grasping and Pickup	FR-15	The system shall approach the target object to a defined pre-grasp pose before actuating the gripper.	PR-15	The entire process from initiating approach to the target object to completing the approach and pre-grasp sequence shall not exceed 30 seconds.	Time from approach start to pre-grasp completion must be ≤ 30 s.	Demonstration	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	Yes	No
Functional	Grasping and Pickup	FR-16	The end-effector shall secure and lift the target object to a transport height.	PR-16	In $\geq 90\%$ of attempts, the system shall securely grasp the target object and lift it to the required transportation height without dropping it.	During object collection demonstration runs 9 out of 10 runs must succeed - lifting object to transport height and maintain grasp.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes

Functional	Grasping and Pickup	FR-17	The gripper shall avoid causing permanent damage to wooden target objects during normal operation.	PR-17	The maximum gripper force applied to the target object shall not exceed the target object's tolerance to prevent indentation damage.	Video recording of demonstration, as well on-board sensor acceleration reading should indicate no sudden drops in acceleration mid-run that would indicate collision. Video record shall verify if collision occurred in such cases or was due to an external fault	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	No	Yes
Functional	Grasping and Pickup	FR-18	The payload system shall retain the target object during mission motions with out uncommanded loss.	PR-18	Once the robot has successfully grasped and is transporting the target object, the object shall not accidentally drop during the entire transportation process.	During a demonstration run, uncommanded release of object should not occur at any point.	Performance Arena Demonstration Run	Testing Arena	On-board Systems Testing	Yes	No
Functional	Bin Identification And Deposition	FR-19	The system shall detect the storage bins at the start area and identify each bin's colour.	n/a	n/a	In object detection training and demonstration runs, bins of full colour variety should be presented. Correct colour identification must occur.	System Design Analysis	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	Yes	No
Functional	Bin Identification And Deposition	FR-20	The system shall select the bin whose colour matches the carried target object's colour classification.	PR-19	The system shall accurately identify the coloured bin that matches the color of the target object.	In object detection training and demonstration runs, bins of full colour variety should be presented alongside objects. Correct colour matching process must occur.	System Design Analysis	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	Yes	Yes
Functional	Bin Identification And Deposition	FR-21	The system shall place or drop the target object through the bin opening.	PR-20	In $\geq 90\%$ of attempts, the placement process shall ensure that the target object remains inside the bin.	Perform 10 demonstration runs of object bin placement. ≥ 9 must remain inside bin after release.	System Design Analysis	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	No	No
Functional	Safety And Compliance	FR-22	The system shall provide an immediately accessible emergency stop that removes drive power.	PR-21	Actuation of the emergency stop shall result in the power cut from the drive actuators within 5 seconds.	Simulate an emergency stop during a demonstration run. Robot should pause movement in ≤ 5 s	Performance Arena Demonstration Run	Demonstration Arena Testing	On-board Systems Testing	Yes	Yes
Functional	Safety And Compliance	FR-23	The system shall transition to a defined safe state on critical faults (loss of localization, sensor failure, collision detection).	PR-22	The safe state transition shall activate when critical fault occurrences.	Simulate an emergency stop during a demonstration run. Robot should pause movement and activate safe state transition.	Performance Arena Demonstration Run	Demonstration Arena Testing	On-board Systems Testing	Yes	Yes
Functional	Safety And Compliance	FR-24	The system shall have a maximum translational and rotational speed of 1 km/hr for safe operation.	PR-23	The maximum movement and rotation speeds of the robot shall be adjustable and shall comply with the safety requirements specified in the operational environment.	On-board sensors or speed recording during demonstration should not exceed more than 1km/hr during initial testing.	Performance Arena Demonstration Run	Demonstration Arena Testing	On-board Systems Testing	Yes	No
Functional	Safety And Compliance	FR-25	The system shall avoid damaging university hardware (base robot, sensors, bins, obstacles, arena).	PR-24	The system shall complete the demonstration runs with zero recorded instances of damage to any provided hardware.	For Final robot inspection, no damage should be recorded	Inspection	Workshop/Lab Workspace	System Design	Yes	Yes
Environmental	Arena and Obstacles	FR-26	The system shall be designed to operate on the provided flat indoor arena floor without ramps.	PR-25	The system shall be designed to operate on flat indoor surfaces and shall be adaptable to slightly uneven floor conditions.	System shall be designed and tested on nominal flat surfaces in testing area. Floor irregularities should not impact mission operations.	Performance Arena Demonstration Run	Demonstration Arena Testing	On-board Systems Testing	Yes	No
Environmental	Arena and Obstacles	FR-27	The system shall recognize arena boundaries of approximately uniform colour and avoid breaching them.	n/a	n/a	In video recording of demonstration testing, robots should not be recorded trespassing or colliding with boundary at any point.	Performance Arena Demonstration Run	Demonstration Arena Testing	On-board Systems Testing	Yes	No
Environmental	Obstacle detection and avoidance	FR-28	The system shall detect static obstacles within the sensing envelope.	PR-26	The system shall detect obstacles with a horizontal dimension ≥ 150 mm at a distance of up to 3 m.	Demonstration runs should feature obstacles at 1 m, 2 m and 3 m. horizontal dimension should also be varied in separate tests. Must detect and avoid obstacles during testing.	Performance Arena Demonstration Run	Demonstration Arena Testing	On-board Systems Testing	No	No
Environmental	Obstacle detection and avoidance	FR-29	The system shall plan paths that avoid contact with detected obstacles.	PR-27	A minimum clearance distance of 100mm shall be maintained between the planned path and obstacles.	Through path planning output analysis. All planned path routes must keep ≥ 100 mm distance	System Design Analysis	Workshop/Lab Workspace	System Design	Yes	No
Environmental	Obstacle detection and avoidance	FR-30	The system shall stop motion and trigger a recovery behaviour if a collision or near-collision is detected.	PR-28	The robot shall stop all motion within 5 seconds of detecting a near-collision event (The distance between the obstacle and the robot is less than 100 mm and the distance continues to decrease in the current direction of travel).	Through simulation of collision event during testing - speed log should indicate robot should stop movement within ≤ 5 s.	Demonstration Run	1. Workshop/Lab Table Based Testing 2. Demonstration Arena Testing	System Design	Yes	Yes
Maintainability & Operability	Configuration and tuning	FR-31	All mission parameters (colour set, number of objects, thresholds, timeouts) shall be configurable without code changes.	PR-29	All mission parameters, including colour set, number of objects, thresholds, and timeouts, shall be configurable without code changes.	Modify parameters using a configuration interface during programming Mission should run autonomously with consistent onboard programming	System Design Analysis	Workshop/Lab Workspace	System Design	Yes	No
Maintainability & Operability	Logging and Diagnostics	FR-32	The system shall log time-stamped pose, perception outputs, and high-level actions during each run.	PR-30	During each operation, the system shall record the robot's pose with timestamps, sensory outputs, and high-level operations, with a data loss rate of less than 5%.	Review log completeness for pose, perception, and actions during demonstration testing Data loss or corruption should be $\leq 5\%$.	System Design Analysis	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	No	No
Maintainability & Operability	Logging and Diagnostics	FR-33	The system shall provide a means to export run logs in a documented format after a run.	n/a	n/a	During testing run, export logs should be produced with expected format and interface.	Demonstration	Demonstration Arena Testing	On-board Systems Testing	Yes	No
Maintainability & Operability	Human-machine interface	FR-34	The system shall provide a single operator command to start the autonomous mission.	PR-31	The start command shall result in the initiation of autonomous movement within 3 seconds.	In testing, initial autonomous motion must begin within ≤ 3 s.	Demonstration	Demonstration Arena Testing	On-board Systems Testing	Yes	No

Maintainability & Operability	Human-machine interface	FR-35	The system shall support the modes Auto, Pause, Resume, and Abort.	PR-32	Mode transition latency (time between operator input and system response) shall be less than 1 seconds.	Issue and switch mode command during 5 runs. Transition latency must be ≤ 1 s	System Design Analysis	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	No	No
Maintainability & Operability	Human-machine interface	FR-36	The system shall allow safe operator intervention to reposition the robot and then resume autonomous operation.	PR-33	The total time required for a manual intervention shall not exceed 2 minute.	During a demonstration run, human intervention should not exceed 2 minutes	Demonstration	Demonstration Arena Testing	On-board Systems Testing	No	No
Mechanical / Structural	Robot Structure	FR-37	Added mechanical structures shall remain within arena vertical and horizontal clearances.	PR-34	No additional mechanical structure can alter the robot's existing maximum lateral and longitudinal dimensions.	Final dimension measurements of completed robot payload and structure, should be within an appropriate test-defined clearance margin.	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Mechanical / Structural	Robot Structure	FR-38	Mechanical modifications shall be reversible and shall not permanently alter or damage the base robot.	PR-35	Modifications to the robot shall not cause permanent damage, shall utilize commonly available laboratory tools, and shall be fully reversible.	No damage should be recorded during final robot inspection	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Mechanical / Structural	Payload sled	FR-39	The team shall design and deploy a payload sled that shall provide mounting locations for all sensors and on-robot compute units.	PR-36	The payload sled shall not increase the base robot footprint (length \times width).	Final dimension measurements of completed robot payload should be equal to the original footprint	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Mechanical / Structural	Payload sled	FR-40	The payload sled shall position each sensor to avoid occluding required fields of view.	PR-37	The horizontal field of view of the primary vision sensor shall not be obstructed by the robot or payload sled components.	Through sensor feedback and camera logs, occlusion should not occur in mission-critical events.	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Mechanical / Structural	Payload sled	FR-41	The payload sled shall support the combined mass of mounted sensors and compute units with adequate stiffness.	PR-38	The payload sled shall be capable of supporting the total weight of all sensors and computing units, shall possess sufficient stiffness, and shall maintain structural integrity during operation.	For final payload sled mounting, no deformation or decrease in structural support should be inspected.	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Mechanical / Structural	Payload sled	FR-42	The payload sled shall provide protected cable routing and strain relief to prevent snagging.	PR-39	All cables shall be routed with a minimum ground clearance of 50 mm.	For final structure, measure lowest-hanging cable. Clearance must be ≥ 50 mm	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Mechanical / Structural	Payload sled	FR-43	The payload sled shall not obstruct access to the emergency stop, power switches, or other safety interfaces.	PR-40	A minimum clearance of 10 millimeters shall be maintained around the emergency stop button to facilitate manual operation.	During final inspection, clearance around emergency stop button must be ≥ 10 mm for full access.	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Mechanical / Structural	Payload sled	FR-44	The payload sled shall use materials and finishes that avoid sharp edges and burrs.	PR-41	All exposed edges of the payload sled shall be smoothly polished with a minimum 1-millimeter radius or chamfer, with no sharp corners.	During final inspection, All edges ≥ 1 mm radius/chamfer.	System Structure Inspection	Workshop/Lab Workspace	Final Inspection	Yes	Yes
Other (stretch/optional)	Advanced Behaviours	FR-45	The system may retrieve multiple target objects in one trip or one-by-one, selectable via configuration.	PR-42	The system shall be capable of switching between multi-object and single-object picking modes by adjusting configurations without the need for software redeployment.	During autonomous operation testing, multi-object and single-object collection modes should be demonstrated.	System Design Analysis	1. Workshop/Lab Workspace Testing 2. Testing arena	System Design	Yes	No
Other (stretch/optional)	Advanced Behaviours	FR-46	The system may prioritize which target to retrieve next based on distance, visibility, or colour.	PR-43	The system shall evaluate all detected target objects and compute a priority score based on distance, visibility, and colour within 5 seconds of each decision cycle	Through operation testing, Multiple targets objects should be placed and detected System should indicate prioritisation process based on provided factors, with execution time of ≤ 5 s.	Demonstration	Testing Arena	On-board Systems Testing	No	No
Other (stretch/optional)	Advanced Behaviours	PR-47	Within 1 second of placing the target object into the bin, the system shall confirm successful placement through the selected feedback method.	n/a	n/a	During run, record time taken from object release to confirmation Should be ≤ 1 s.	Demonstration	Testing Arena	On-board Systems Testing	Yes	No

References

- [1] S. R. Hirshorn, L. D. Voss, and L. K. Bromley, "Nasa systems engineering handbook," tech. rep., NASA, 2017.