Critical Design Review Report

Lunar ROADSTER

Team I

Author: **Ankit Aggarwal**
Andrew ID: ankitagg
E-mail: *ankitagg@andrew.cmu.edu*

Author: **Deepam Ameria**
Andrew ID: dameria
E-mail: *dameria@andrew.cmu.edu*

Author: **Bhaswanth Ayapilla**
Andrew ID: bayapill
E-mail: *bayapill@andrew.cmu.edu*

Author: **Simson D'Souza**
Andrew ID: sjdsouza
E-mail: *sjdsouza@andrew.cmu.edu*

Author: **Boxiang (William) Fu**
Andrew ID: boxiangf
E-mail: *boxiangf@andrew.cmu.edu*

Supervisor: **Dr. William "Red" Whittaker**
Department: Field Robotics Center
E-mail: *red@cmu.edu*

May 2, 2025

Carnegie Mellon University
Robotics Institute

Abstract

The Lunar ROADSTER (Robotic Operator for Autonomous Development of Surface Trails and Exploration Routes) project develops an autonomous rover designed to groom traversable trails on the Moon. These trails support sun-synchronous mobility for future lunar missions, improving energy efficiency and operational longevity. At the core of the system is a custom-fabricated dozer blade assembly, developed through iterative CAD design and manufacturing, enabling crater grooming and backblading via a linear actuator.

A dedicated transport planner processes terrain data to assign grooming tasks by identifying optimal source and sink regions. These assignments are then translated into waypoints, which the navigation stack uses to generate feasible trajectories, tuned for Ackermann steering and lunar terrain constraints.

During the Spring Validation Demonstration (SVD), the system successfully showcased autonomous crater identification, grooming, and navigation with less than 10% path deviation. Remaining challenges include resolving compute bottlenecks, refining localization accuracy, and improving mechanical reliability. Planned upgrades include implementing a ZED 2i stereo camera, SkyCam-based localization, navigation tuning, and expanding the tool planner to groom multiple craters in sequence.

Lunar ROADSTER sets the foundation for autonomous surface preparation, providing critical infrastructure for sustained lunar operations, logistics, and future colonization.

Contents

1 Project Description

Humanity is preparing to return to the Moon, with the Artemis missions focusing on exploring the South Pole—a region rich in sites of interest. Establishing a circumnavigating route around the lunar pole will serve as a critical "highway" connecting these sites and enabling key activities such as transportation, human settlement, and resource extraction.

A solar-powered rover capable of sun-synchronous circumnavigation could achieve perpetual operation by avoiding lunar sunsets. At high latitudes, this is feasible at low speeds, as shown in Table 1. However, these assumptions rely on the terrain being flat and traversable, free from major topographical challenges. A mission to manipulate the lunar regolith in the circumnavigating path to make it more traversable for future missions is thus, a clear step forward. A robotic system can be designed to conduct these operations efficiently for extended durations.

Table 1: Average Speed Required to Circumnavigate at Different Latitudes on the Moon

Latitude	Distance (km)	Speed (kph)
Equator	11,000	16
50°	7,040	10
60°	5,500	8
70°	3,700	6
75°	2,800	4
80°	1,870	3
81°	1,529	2.5

The Lunar Robotic Operator for Autonomous Development of Surface Trails and Exploration Routes (Lunar ROADSTER) is an autonomous moon-working rover, capable of finding exploration routes and grooming the lunar surface to develop traversable surface trails. These groomed trails will become the backbone for the colonization of the Moon by enabling transportation, logistics, and enterprise development.

2 Use Case

The conceptual use case for the Lunar ROADSTER system is illustrated in Figure 1. The process begins with the system receiving detailed maps of the user-specified latitude from prior exploration missions, such as orbiters or exploratory rovers. Due to the natural irregularities of the lunar surface, this initial reference latitude (white dotted line) often intersects with craters, dunes, and rough terrain, resulting in a non-traversable original path (blue line). These irregularities make the path non-traversable for standard solar-powered rovers, which rely on relatively flat terrain for safe and efficient motion. Traversing such terrain would not only consume excessive energy but also increase the risk of mechanical failure or mission interruption.

To address this, Lunar ROADSTER autonomously navigates this original path and identifies areas requiring terrain conditioning. Using onboard perception and terrain analysis, it classifies craters into two categories: those that are too deep or wide to groom (red), and gradable craters (orange) that fall within the rover's manipulation capabilities. The rover uses a custom-designed dozer blade to push regolith from the rim into the crater, smoothing the terrain and forming a groomed, traversable path (green line). This path is shorter, safer, and more energy-efficient than the original route.

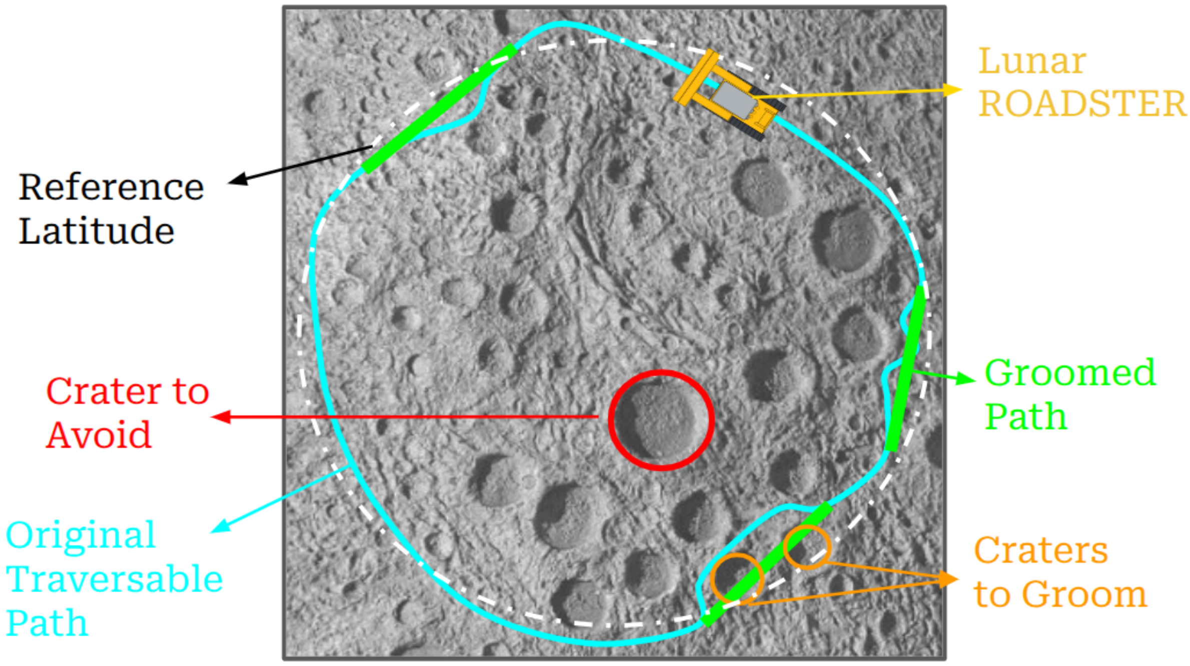


Figure 1: Conceptual System Graphic Representation

After completing the initial grooming operation at a designated crater, the Lunar ROADSTER does not immediately continue along its path. Instead, it retreats slightly to a vantage point where it can scan and assess the modified terrain using its onboard perception system, such as a depth camera. This evaluation step is critical to ensure that the crater has been adequately filled and leveled to meet defined traversability criteria. If the surface still exhibits irregularities such as steep inclines, or dips, the rover autonomously initiates a re-grooming cycle. It re-plans its manipulation trajectory and performs another round of dozing or backblading, adjusting the terrain further. This closed-loop grooming-evaluation cycle continues iteratively until the rover confirms, through sensor feedback, that the modified surface is suitable for safe traversal. Once validated, the rover marks the crater as complete and advances to the next waypoint, progressively transforming the rugged trail into a continuous, traversable path.

This use case demonstrates Lunar ROADSTER's ability to autonomously transform hazardous lunar terrain into a continuous, navigable highway, enabling long-duration sun-synchronous missions and supporting future lunar exploration and infrastructure.

3 System-Level Requirements

The system requirements for the Lunar ROADSTER project are derived from a comprehensive understanding of the problem statement, its use cases, and the high-level objectives. These objectives shown in Figure 2, informed by inputs from stakeholders, provide a clear framework for defining the system requirements.

The requirements are organized into mandatory and desirable categories, further classified into functional, performance, and non-functional requirements. The mandatory requirements form the core functionalities essential for the project's success, while the desirable requirements, though initially out of scope, aim to enhance the system's overall performance. The requirements may evolve as the system develops, further research is conducted, and tests refine the design. The team will focus on meeting all mandatory requirements by project deadlines while working to implement desirable ones as resources permit.

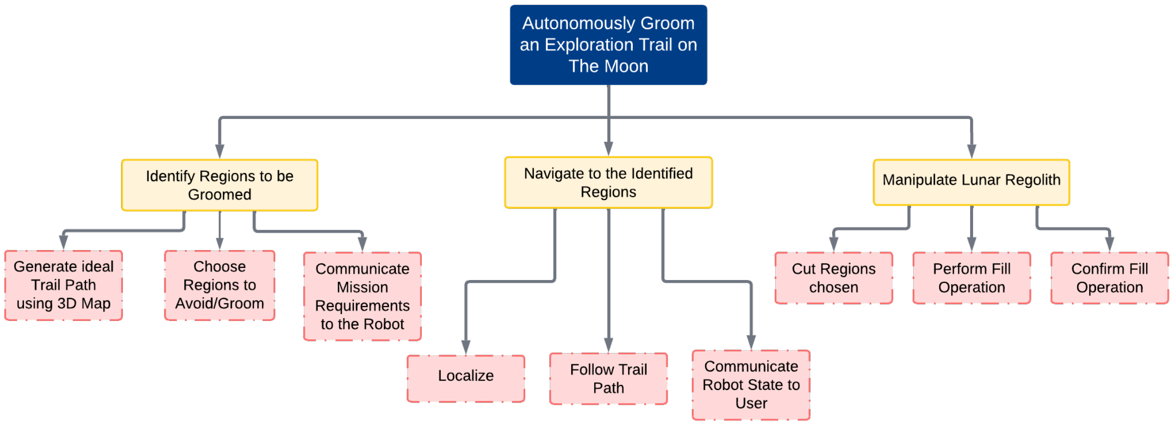


Figure 2: Objectives Tree

3.1 Mandatory Requirements

3.1.1 Mandatory Functional Requirements

Table 2: Mandatory Functional Requirements

Sr.No.	**Mandatory Functional Requirement (Shall)**
M.F.1	Perform **trail path planning**
M.F.2	Operate **autonomously**
M.F.3	**Localize** itself in a GPS denied environment
M.F.4	**Navigate** the planned path
M.F.5	Traverse **uneven terrain**
M.F.6	Choose craters to **groom and avoid**
M.F.7	**Grade** craters and **level dunes**
M.F.8	**Validate grading** and trail path
M.F.9	**Communicate** with the user

3.1.2 Mandatory Performance Requirements

Table 3: Mandatory Performance Requirements

Sr.No.	Performance Metrics (Will)
M.P.1	Plan a path with **cumulative deviation of** $\leq 25\%$ from chosen latitude's length [2]
M.P.2	**Follow planned path** to a maximum deviation of **10%**
M.P.3	**Climb gradients up to** $15°$ and have a **contact pressure of less than 1.5 kPa** [3]
M.P.4	**Avoid craters** ≥ 0.5 **metres** and **avoid slopes** $\geq 15°$
M.P.5	Fill craters of **up to 0.5 meters** in diameter and **0.1 meter** in depth [1]
M.P.6	Groom the trail to have a **maximum traversal slope of** $5°$

3.1.3 Mandatory Non-Functional Requirements

Table 4: Mandatory Non-Functional Requirements

Sr.No.	Parameter	Description
M.N.1	Weight	The rover must weigh **under 50 kg**
M.N.2	Cost	The cost for the project must **be under $5000**
M.N.3	Computing Capacity	The onboard computer should be able to **run all required tasks**
M.N.4	Size/Form Factor	The rover should measure **less than 1 meter** in all dimensions

3.2 Desirable Requirements

3.2.1 Desirable Non-Functional Requirements

Table 5: Desirable Non-Functional Requirements

Sr.No.	Parameter	Description
D.N.1	Technological Extensibility	The system will be **well documented** and designed so that future teams can easily access and build on the work
D.N.2	Aesthetics	Requirement from sponsor, the rover must **look presentable and lunar-ready**
D.N.3	Modularity	To enable **tool interchangeability**, the tool assemblies must be modular and easy to assemble/disassemble
D.N.4	Repeatability	The system will **complete multiple missions without the need of maintenance**

4 Functional Architecture

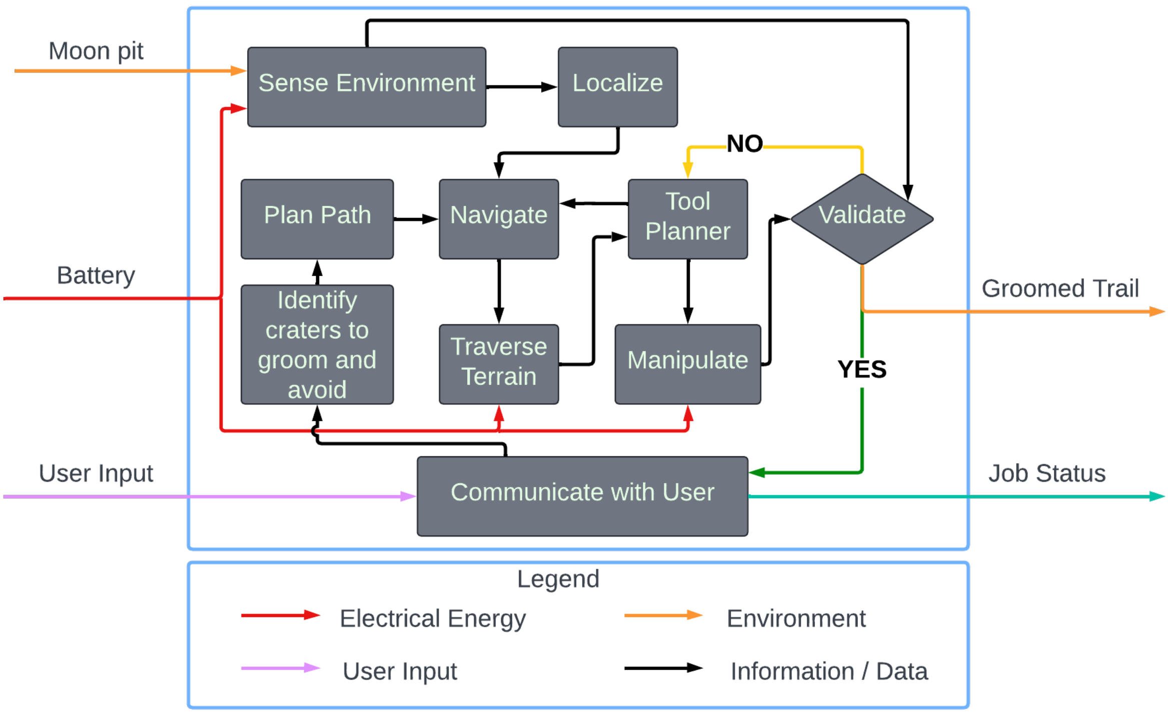


Figure 3: Functional Architecture

Figure 3 illustrates the functional architecture of our system. The system receives three types of input: user input in the form of a map of the environment from the operational terminal, battery input as the electrical energy that powers the components, and environment input from the moonpit (worksite).

The `Communicate with User` block serves as the critical interface between the user and the system. It transmits the map to the `Plan Path` algorithm and updates the user with real-time job status information for active monitoring. The `Plan Path` algorithm processes environmental information from the map to identify craters to groom and avoid. These constraints are defined in the performance requirements (M.P.4 and M.P.5). Based on this analysis, it generates precise waypoints near the craters requiring grooming and sends them to the `Navigate` block. Before initiating navigation, the robot undergoes localization through the `Localize` block, using information from the sensor stack to accurately determine its position within the environment. The `Navigate` block translates the planned waypoints into motor commands for the chassis, which are executed by the `Traverse Terrain` block, enabling the robot to maneuver through the moonpit and approach the target crater effectively.

Once positioned near the crater, the `Tool Planner` is activated, sending motor commands to the `Manipulate` block to initiate tool operations, such as excavation and grading. The grooming process is then evaluated by the `Validate` block to determine if the crater meets the specified grooming criteria, as defined in the performance requirements (M.P.6). If validation fails, the system repeats the cycle, navigating the robot back to the position near the crater and restarting the tool operation. If grooming is successful, the `Communicate with User` block updates the user with the job status, and the system outputs a groomed trail. This iterative and modular workflow ensures precise grooming operations while maintaining active user monitoring and operational reliability.

5 Cyberphysical Architecture

The Cyberphysical architecture, depicted in Figure 4, shows how our Lunar rover is physically realized. It integrates a network of the following major subsystems: Sensors, Computations, External Infrastructure, Mechanical subsystem, Actuation and Electronics, and Electrical Power. Each component plays a specific role, with all of them working together in unison to meet the unique demands of lunar surface operations.

5.1 Sensors

The rover relies of the following set of sensors for the essential data, which are crucial for navigating and executing tasks:

- Wheel motor encoders
- Mast depth camera (RealSense D435i)
- Inertial Measurement Unit (IMU)
- Actuator Feedback Sensor

These sensors provide critical feedback on the rover's position, orientation, and material to manipulate.

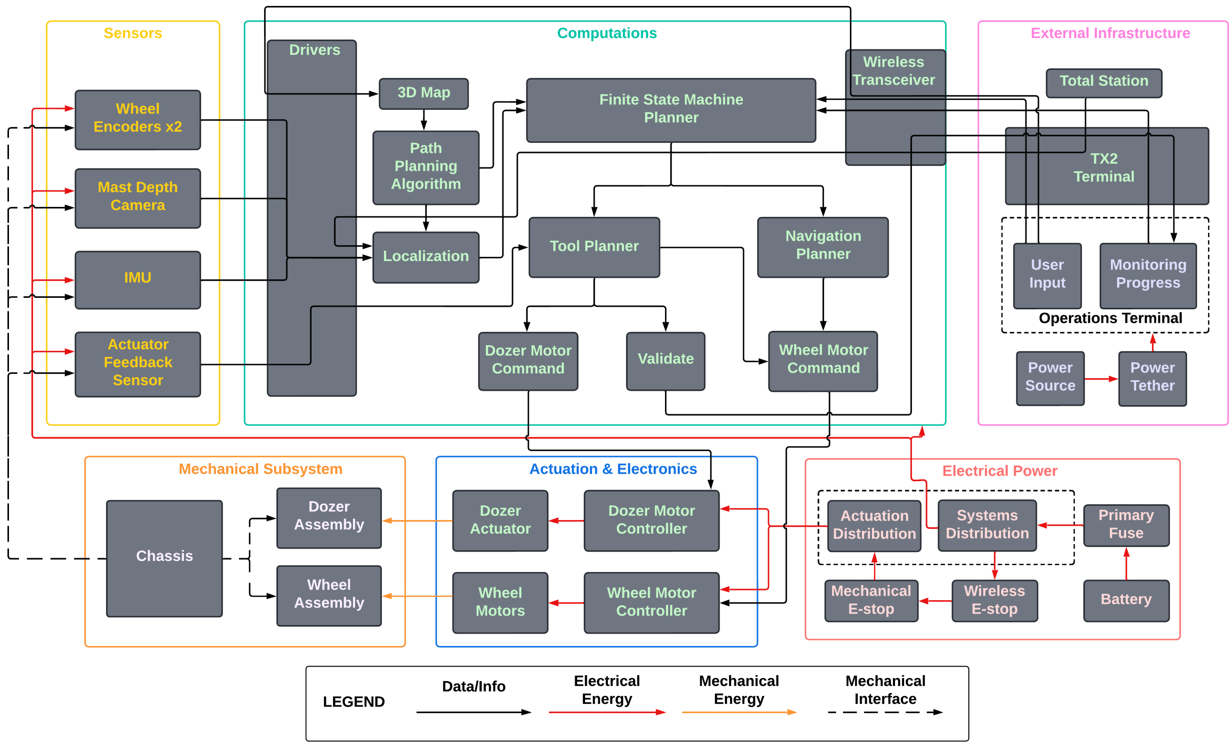


Figure 4: Cyberphysical Architecture

5.2 External Infrastructure

The extrernal infrastructure comprises the Total Station, a Wireless Transceiver, and an Operations Terminal. The robotic total station provides precise robot pose estimates, and the operations terminal allows for seamless communication between the rover and mission control through the wireless transceiver, by providing a user interface to monitor progress and receive updates.

5.3 Computations

The computations subsystem is the processing powerhouse of the rover, where data from sensors are transformed into actions. It includes the following components:

1. The drivers form the interface between sensors and processing units.
2. A 3D map is fed into the 3D Map block through a wireless transceiver from the operations terminal. The path planning algorithm runs directly on the Jetson AGX Xavier that serves as the brain of the rover, which processes the map and generates a navigation path for the rover to follow.
3. A robotic total station in the external infrastructure, which provides precise robot pose, sends its data to the localization block.
4. The data from the total station is fused with the data from wheel encoders, and IMU, and sent to the localization block, which keeps track of the rover's position on the lunar surface.
5. The FSM Planner manages high-level decision-making and receives inputs from the localization block and the path that the robot has to follow. The planner chooses between 2 states - the Tool Planner and Navigation Planner.
6. Tool Planner and Navigation Planner are used to coordinate tool operations and

rover movement respectively. The Tool Planner sends commands to the dozer motors and interfaces with the validation module, while the Navigation Planner generates wheel motors commands based on the path generated.

7. The Tool Planner generates commands for terrain manipulation and passes them directly to the Dozer Motor Command block, which actuates the dozer blade for grooming operations. Simultaneously, it interfaces with the Validate block to assess whether the terrain meets the required criteria after each grooming action.
8. The validate block passes the data to the operations terminal through the wireless transceiver. At the operations terminal, the progress of grading is monitored, and is sent back into the FSM Planner as feedback, i.e., whether the surface has been excavated or graded satisfactorily.

5.4 Actuation and Electronics

This subsystem translates electrical signals into physical movements. The dozer motor controller receives commands from the dozer motor command block in the computations subsystem. Similarly, the wheel motor controller receives commands from the wheel motor command block in the computations subsystem. These controller blocks provide signals to the respective dozer actuator and wheel motors, which then make the respective assemblies connected to them in the mechanical subsystem move.

5.5 Mechanical Subsystem

It forms the structural backbone of the rover. The main components include the Chassis, Dozer Assembly, and Wheel Assembly. This subsystem provides both the physical support required for the rover and the mechanisms needed to interact with the lunar surface. All the sensors and hardware sit on the chassis of the rover, and the dozer assembly is used to groom the trail on the Moon.

5.6 Electrical Power

This subsystem is responsible for supplying energy to the entire rover. In the operations terminal, a power source supplies power to the robotic total station as well as the operations terminal via a tether. On the rover, a battery provides the electrical power and is connected to the Power Distribution Board (PDB) through a primary fuse for safety. The PDB allocates power to the Systems Distribution block and the Actuation Distribution block. The systems distribution block supplies power to all the subsystems on the rover, and the Actuation Distribution block supplies power to the actuators – motor controllers and motors. We also have a wireless emergency stop (E-stop), which translates to a mechanical E-stop to cut off all power to the actuators in case of an emergency.

6 Current System Status

6.1 Targeted Requirements in Spring

Table 6: Targeted Requirements in Spring

Requirement	Description	Status
M.P.1	Will plan a path with cumulative deviation of <= 25% from chosen latitude's length	Achieved
M.P.2	Will follow planned path to a maximum deviation of 10%	Demonstrated
M.P.4 (Part 1)	Will avoid craters >= 0.5 meters	Demonstrated
M.P.5	Will fill craters of up to 0.5 meters in diameter and 0.1 meters in depth	Demonstrated
M.N.1	Weight - The rover must weigh under 50kg	Achieved
M.N.4	Size - The rover should measure less than 1m in all dimensions	Missed
D.N.1	Technological Extensibility - The system will be well documented and designed so that future teams can easily access and build on the work	Achieved
D.N.4	Aesthetics - Rover must look presentable and lunar-ready	Demonstrated

6.2 Overall System Depiction

Figure 5 shows the overall system hardware depiction and Figure 6 shows the overall system software depiction.

6.3 Subsystem Completion Status

The following section presents a comprehensive overview of all subsystems developed during the spring semester. It also summarizes the modeling, analysis, and testing conducted to meet design specifications and validate subsystem functionality. Appendix A.1 shows the completion status of each subsystem.

6.3.1 Sensors Subsystem

The sensors subsystem is responsible for capturing data from the environment and the internal state of the rover to support various computational tasks. It includes four wheel encoders, an Intel Realsense D435i depth camera, a VectorNav IMU, and a linear actuator.

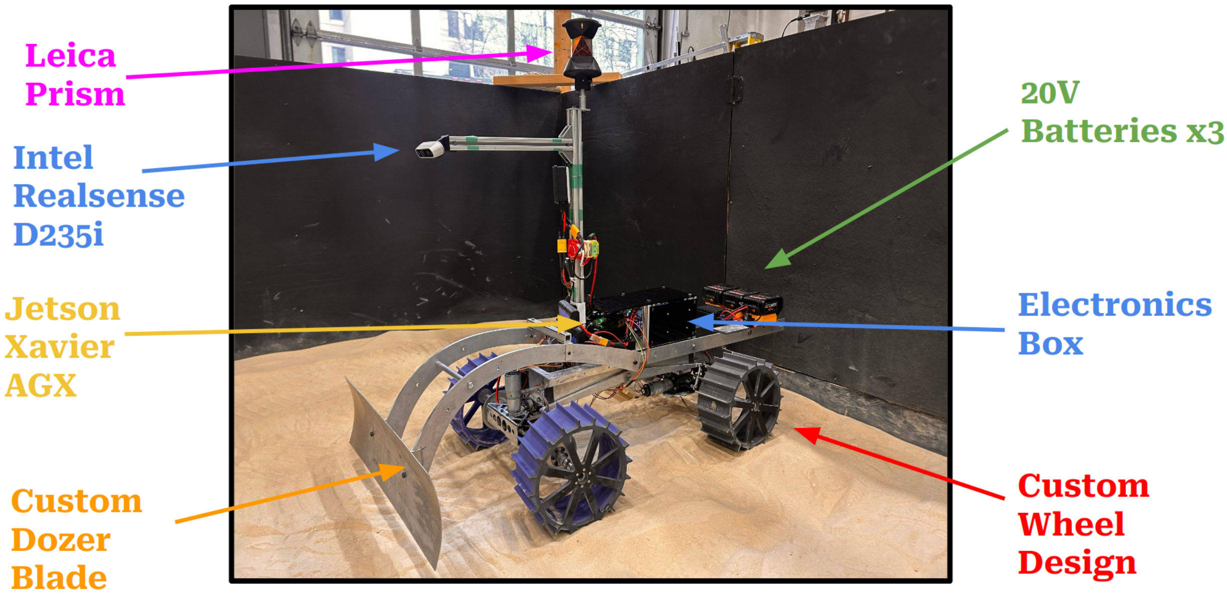


Figure 5: Overall System Depiction - Hardware

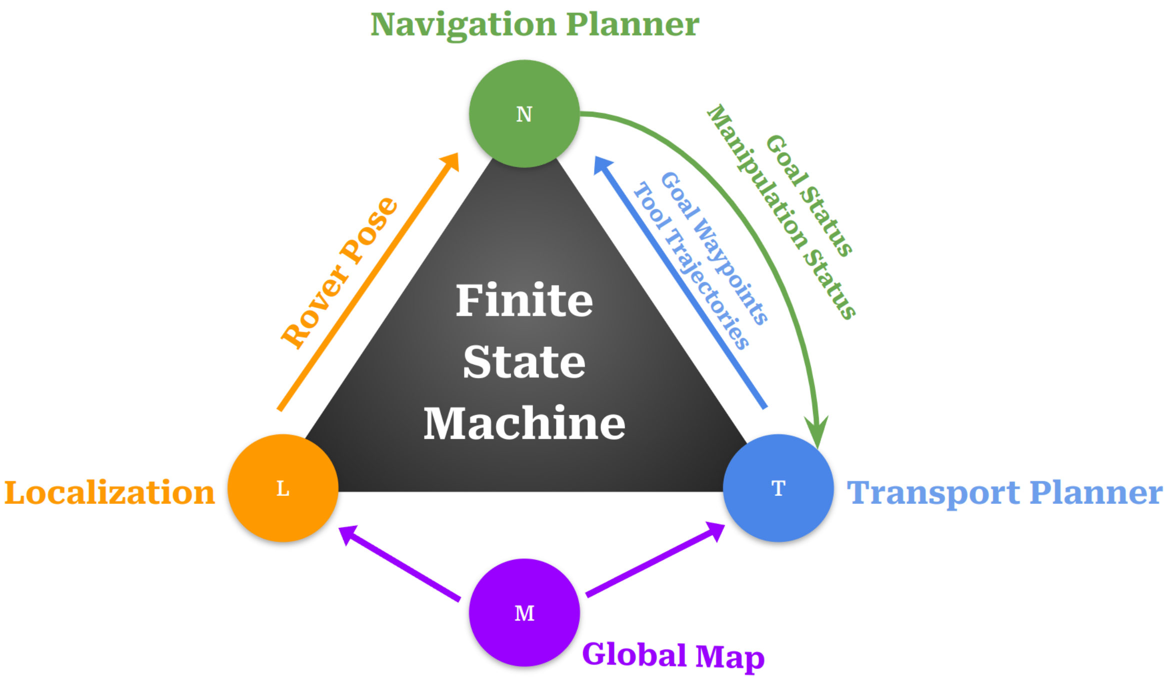


Figure 6: Overall System Depiction - Software

The IMU is integrated using the VectorNav ROS2 package, and the wheel encoders are interfaced via micro-ROS. The linear actuator came with an onboard potentiometer for feedback. The depth camera is interfaced using RealSense ROS wrappers, which provided point cloud data for perception. We initially had plans to use the ZED 2i stereo camera, but were faced with CUDA compatibility issues within the Docker environment. We will be integrating this during the Fall semester.

We 3D printed the camera mount and explored optimal mounting angles for depth perception. Experimental analysis showed that the 50° mount provided the best crater visibility, though occlusions increased when the mounted tool height exceeded 15%. The functionality of this subsystem was verified through targeted unit tests, including Optimal Mast Depth Camera Placement (T10) and a Maintenance and QA Test (T14).

6.3.2 Computations Subsystem

Jetson & Docker:

The Jetson and Docker unit hosts all key system packages and manages device drivers through a Docker-based ROS2 framework running on the NVIDIA Jetson AGX Xavier.

We built a custom Docker container to run ROS2 Humble and micro-ROS, along with all necessary system packages and device drivers. We also configured the Jetson with a static IP for remote SSH access. While conducting the Depth Camera Connectivity Test (T03), we faced challenges is integrating the ZED 2i stereo camera with the Jetson due to CUDA compatibility issues. Additionally, increased computational demands projected for the Fall semester will require an upgrade to a more powerful Jetson.

Localization:

The localization unit enables the rover to estimate its pose within the Moon Yard. It combines data from the Leica TS16 Total Station, VectorNav IMU, and wheel encoders, to support both global and local localization. Detailed explanation regarding the implementation is provided in Appendix A.2.

While conducting the Localization Test (T09), we observed continuous drift in the total station frame, which was resolved by correcting the coordinate frame transformations and fine-tuning the EKF. We also calibrate the rover's yaw to ensure consistent orientation w.r.t the `map` frame. An issue that currently persists is that each time the total station battery is replaced, a minor frame offset is introduced, leading to localization inaccuracies. This was identified during out Spring Validation Demo Test (T15). To mitigate this, we plan to implement resection-based frame correction and also explore alternative localization methods that eliminate dependency on the total station, both during the Fall semester.

Transport Planner:

The transport planner unit is responsible for planning sand manipulation. It takes the global map as an input and outputs a set of goal poses, which will be used by the navigation stack as targets to plan a path. This is illustrated in Figure 7. Detailed explanation regarding the transport assignments is given in Appendix A.3.

We previously faced challenges in processing the dense elevation maps and properly modeling tool height in the planner. The current implementation handles single craters, but needs to be extended for multiple-crater scenarios, which we will implement in the Fall semester.

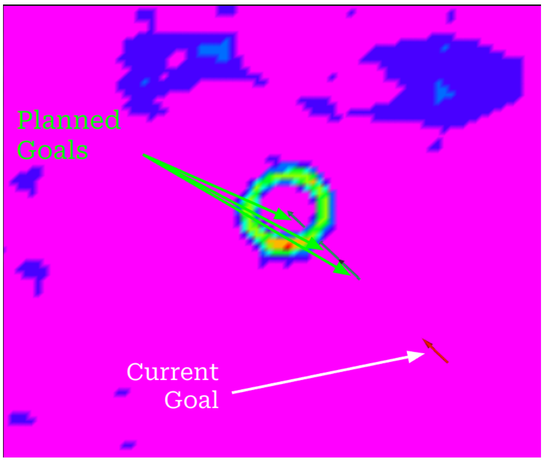


Figure 7: Tool Planner Ouput

Navigation Planner:
The navigation planner unit is responsible for identifying gradable craters, obtaining their coordinates, and planning paths for the rover to reach them while avoiding obstacles.

We use a FARO laser scanner to obtain a high resolution map of the environment, using which we categorize craters as gradable or obstacles based on their size. This was done as part of our Mapping the Moon Yard Test (T08). We then used the ROS2 Navigation2 stack to plan a path for the rover based on target locations provided by the transport planner. We fine-tuned the Nav2 parameters to suit our rover's Ackermann steering model and used RViz to debug frame alignment and planned paths. Our Autonomous Navigation Validation Test (T11) validated that the rover could successfully reach target craters while avoiding obstacles. This unit, however, needs to be further tuned in the Fall semester, and is hence 90% complete. Detailed explanation regarding the implementation details is provided in Appendix A.4.

Validation:
The validation unit verifies whether the groomed crater satisfies M.P.6, which is to groom the trail to a maximum traversal slope of 5°. We use the Intel Realsense to evaluate the terrain by computing metrics such as mean elevation and elevation RMSE. We calibrated the camera to ensure that flat ground is perceived as having zero elevation and no tilt. However, the Realsense does not provide a very dense point cloud, and hence we plan to switch to the ZED 2i stereo camera in the Fall.

Finite State Machine (FSM):
The FSM Unit governs the high-level autonomy logic for the entire system, orchestrating sensing, planning, navigation, and tool actions. Each stage is implemented as a state with well-defined transitions and service calls to respective subsystem functions, ensuring that autonomy flows smoothly from start to mission end.

We implemented the FSM with lightweight callbacks at 2 Hz, and heavy computations were offloaded to parallel threads to prevent blocking. We validated the autonomy behavior based on our Integration Test (T13). The current version is approximately 95%

Figure 8: Left, Leica TS16 Total Station connected to TX2 Relay. Right, LAN Router and NVIDIA TX2 Relay

complete, with the final goal being to integrate the validation stack in the Fall. Detailed implementation details are provided in Appendix A.5.

6.3.3 External Infrastructure Subsystem

The external infrastructure subsystem includes the off-board components required for localization and network communication between the rover and the operations terminal. It comprises the Leica TS16 Total Station, an NVIDIA TX2 relay chip, a LAN router, and the team's laptop, as shown in Figure 8. The total station continuously tracks the Leica prism mounted on the mast of the rover and sends this data to the TX2 relay chip. The relay forwards these messages over the LAN to the Jetson onboard for localization. We validated the subsystem through the External Infrastructure Test (T05), ensuring reliable message transmission.

6.3.4 Mechanical Subsystem

Dozer Assembly:

The dozer assembly is the primary terrain manipulation component of the rover, designed to push and grade sand in the Moon Yard. It consists of a front-mounted blade, support arms, actuator mounts, and a linear actuator that receives commands from the transport planner. Figure 5 shows the custom dozer blade mounted on the front.

We conducted extensive research on standard dozing and grading tools to design the blade, arms, and actuator yoke. We machined all the components from lightweight aluminum for strength and durability. We also tested multiple actuator configurations for optimal force transfer. We demonstrated successful autonomous dozing and backblading during the Spring Validation Demo.

Wheel Assembly:

The wheel assembly, shown in Figure 5, enables the rover's mobility by serving as the interface between the drivetrain and the lunar terrain. The wheels are 3D printed using PLA and are mounted directly to the drivetrain. We went through several iterations and tested them in the Moon Yard, with design refinements made based on visual wear, deformation, and observed rover performance. The current wheel design successfully provides sufficient traction for pushing sand and allows for steering on loose terrain, as demonstrated in the Spring Validation Demo (SVD). In the Fall semester, we plan the make the wheels lighter and also integrate torque feedback to estimate wheel slip.

6.3.5 Actuation Subsystem

The actuation subsystem is responsible for delivering motion to the rover's drivetrain and controlling the tool for sand manipulation. It includes four DC motors with encoders and a feedback-enabled linear actuator.

We selected high torque planetary gear motors and interfaced them with RoboClaw motor controllers. We also integrated the linear actuator, used for lifting and lowering the dozer blade, with the Actuonix LAC board and Arduino Due. A major issue that we faced during spring tests was that the pinion gears wore out or broke off of the motor shaft frequently. We addressed this by drilling a hole through the motor shaft and fixing a dowel pin to secure it. We also observed oscillations near the linear actuator's setpoint under load. We plan to further tune the PID or replace the component during the Fall.

6.3.6 Electrical Power Subsystem

The electrical subsystem manages power distribution across all components of the rover. It includes three 20V batteries, a DC-DC buck converter, a custom Power Distribution Board (PDB), and controllers for motors, actuators, and sensors. We designed a system-level circuit diagram and manufactured a compact electronics box.

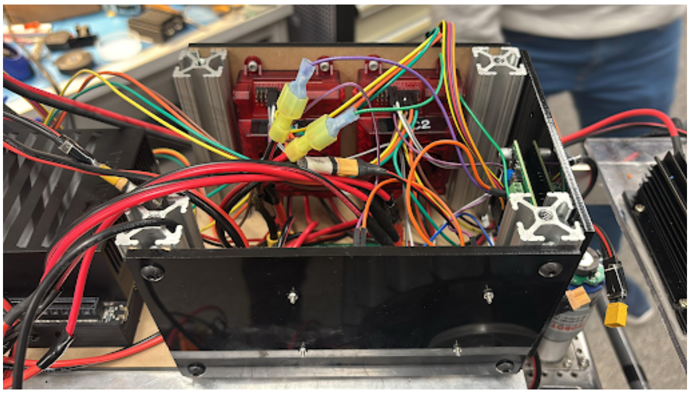


Figure 9: Electronics Box

As part of the Complete Hardware Test (T07), we integrated and tested the PDB, which distributes power to all modules while meeting updated current and voltage demands. We also ensured that the electronics box minimizes space while maintaining accessibility and cooling, as shown in Figure 9. The subsystem is currently 90% complete. Fall plans include integrating torque feedback sensing and updating the battery mounts. Full implementation details, design schematics, and hardware validation results are provided in the Appendix A.6.

6.4 SVD Performance Evaluation

We structured our performance evaluation around 3 main questions - what went well? what didn't go well? what needs to improve?. Further, based on our SVD demonstration, we scored how well we were able to demonstrate our targeted requirements and identified the required improvements. Finally, we scored our entire spring project at a high level.

6.4.1 Self-Score

At a high-level, this is how we scored our system,

1. **Hardware:** 9/10
2. **Software:** 6/10
3. **Presentation/Management:** 9/10
4. **Overall: 8/10**

6.4.2 Qualitative Evaluation

What went well?

1. Mechanical Design: The drive system and tool assembly were sturdy and well-built.
2. Electronics Box Design: The design is compact, reliable and integrates the PCB. It also satisfies D.N.2 by providing a more finished look to the ROADSTER.
3. Transport Planner: The planner was deterministic and outputted optimal solutions.
4. Navigation Planner: The navigation stack found the shortest paths to goal while avoiding obstacles.
5. Finite State Machine: The behavior tree worked well at 2Hz and switched between the various states well.

What did not go well?

1. Reliability of Autonomy: Due to trying to add new features for SVD Encore, our autonomy stack was not reliable. This led to our autonomy failing during the Encore demonstration.
2. Demonstrating performance metrics: We did not demonstrate some performance requirements that we achieved during the SVD.
3. System Improvement for Encore: We were unable to show improvement for Encore as we did not have a plan in place. Instead of showing all capabilities during SVD, we should have left some for Encore.

What needs to improve?

1. Codebase: The current codebase is built as a minimal viable product, meaning that it has not been optimized. It is currently bloated and requires heavy compute which we will need to resolve to add additional features.
2. Localization Accuracy: When using the TS16 total station, we noticed that the static frame shifts everytime the battery is changed, causing offsets in localization. We will explore new methods of setting the static frame to eliminate this issue.
3. Wiring: Despite being in much better shape than the beginning of the semester, some wiring is still visible and untidy. We plan to clean this up in the fall semester.

4. Pre-demo setup methodology: Our current pre-demo setup is convoluted and time-consuming as it involves mapping, calibration and external infrastructure setup. This needs to be improved to allow us to test more frequently and efficiently.

6.4.3 Evaluation of Requirements

Table 7: Requirement Evaluation

Req.	Description	Status	Score
M.P.1	Cumulative Deviation from Latitude	**Achieved** - Global path planning methodology has been set up. We were only able to demonstrate a straight line for a single crater.	5/10
M.P.2	Navigation Accuracy	**Demonstrated** - The ROADSTER followed the generated global path to an average deviation of 8.16%. This metric will also improve with localization accuracy.	9/10
M.P.4	Crater Avoidance	**Demonstrated** - Methodology for identification of craters to avoid has been set up. This was shown in simulation during SVD.	8/10
M.P.5	Crater Grooming	**Demonstrated** - Groomed craters in both autonomous and tele-operation modes. We built a strong, capable and specialized machine for crater grooming.	10/10
M.N.1	Weight	**Achieved** - The rover weighs 25kg. This number may increase with additions during the Fall. We will setup a way to showcase this during FVD.	8/10
M.N.4	Size	**Missed** - The size of the dozing arms were increased to enable backblading. The battery mounting extension arms need to be removed to meet this requirement. This is planned for the Fall.	3/10
D.N.1	Technological Extensibility	**Achieved** - We have maintained an Engineering Wiki and extensively documented our entire design process. We need to demonstrate this during Fall.	8/10
D.N.2	Aesthetics	**Demonstrated** - Sponsor requirement. The design look is satisfactory but will require minor changes in cable management and color schemes.	9/10

6.5 Strong/Weak Points

6.5.1 Strong Points

1. **Robust Tool Subsystem:** The dozer assembly is well-built and effective.
2. **Compact E-Box:** The new E-box is an efficient design with an accessible casing that integrated the PDB.
3. **Wheels:** The custom wheels reliably enable superior traversability and increased drawbar pull.
4. **Power Distribution Board:** The PDB is a well-organized circuit that reliably protects and powers all subsystems.
5. **Management and Presentation:** The spring semester was well-managed and on schedule. Our presentations during SVD and Encore were well received.
6. **External Infrastructure:** All off-board components worked well and reliably.
7. **Software Integration:** The behaviour tree, tool planner and navigation stack were integrated and functioned seamlessly.

6.5.2 Weak Points

1. **Compute Bottleneck:** With heavy onboard computations from the navigation stack and tool planners, we ran into a compute bottleneck while trying to implement the validation unit. We will address this by both code optmization and upgrading compute.
2. **Reliability of Autonomy:** A combination of localization errors and compute bottlenecks caused our autonomy to be unreliable. By addressing the above challenges and further testing, we hope to make our system much more reliable.
3. **Mechanical Vulnerability:** As our inherited chassis is old, wear and tear was an issue throughout the semester. However, we were able to minimize this risk by SVD.
4. **Electrical Connectivity:** Loose jumper wire connections were an ongoing issue through the Spring. We found a temporary solution by taping these down, however, we are still looking for a better solution.
5. **Demonstrating Performance Metrics:** As our scope for SVD was limited to one crater in a straight path, we were unable to properly showcase some of our targeted metrics. We will setup better methodologies to clearly show results for FVD.
6. **Scheduling Mistakes:** We underestimated the time it would take to get the ROADSTER to a state where it is strong and reliable enough to perform our desired tasks. Additionally, we also did not take into account project course assignments and the time they would take to finish. This led to backlogs and the team overworking to meet deadlines. We have planned our Fall semester more conservatively and hope to be able to follow the schedule.

7 Project Management

7.1 Work Plan and Tasks

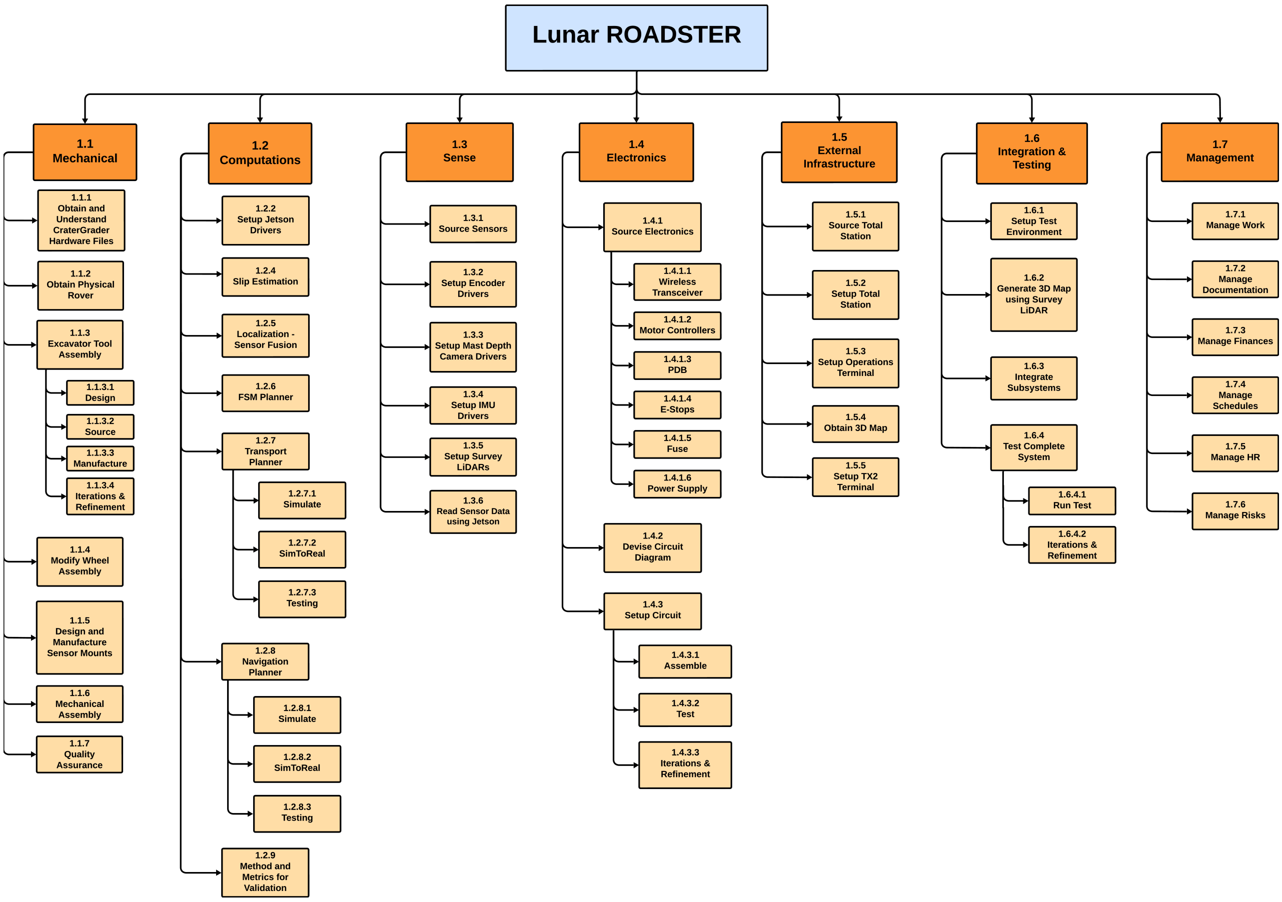


Figure 10: Work Breakdown Structure

Figure 10 shows the Work Breakdown Structure of the Lunar ROADSTER Project. It is a product-based WBS that is derived from the cyber-physical architecture. The work blocks were created by breaking down the individual blocks in the cyber-physical architecture into tasks required to be completed. A subsystem-wise breakdown is shown in Appendix A.7.

Some key highlights are:

- The Excavator Assembly task is split into a general methodology of design, source and manufacture. The team added Iterations and Refinement blocks based on the advice of our sponsors to ensure that we can enable an iterative design process.
- The Localization task will consist of several interdependent tasks. To maintain flexibility, the team has kept this task as an umbrella term directly.
- The 2 planners - Transport and Navigation, follow a uniform Simulate, Sim2Real, and Testing process to allow effective scheduling and progress tracking.
- The Validation task involves novel work regarding the method and metrics. This task will require extensive brainstorming and multiple team meetings.

7.2 Schedule

Several capability milestones are planned to meet the deliverables for the Fall Semester. Bi-weekly reviews are aligned with Progress Reviews to enhance progress tracking and support the timely achievement of critical external and internal milestones. The key system development milestones and their respective bi-weekly schedule is tabulated below in Table 8:

Table 8: Subsystem Development Milestones

Date	Event	Milestones
09/10	PR7	• Hardware and Software **refinement**
09/24	PR8	• **Validation stack** setup • **Wheel torque** measurement • **Navigation** tuning
10/08	PR9	• Autonomous grading of **multiple craters**
10/29	PR10	• **SkyCam-based localization** for improved global positioning
11/12	PR11	• **Full system** integration • **Quality** assurance testing
11/12 & 11/24	PR12 (FVD and Encore)	• **Final System Demonstration** involving autonomous grading of multiple craters

We are generally on schedule, having met all major milestones set through the Critical Design Review (CDR). While there were some delays earlier in the semester, we were able to recover and meet our targets ahead of the Spring Validation Demo (SVD). We had planned to introduce an additional feature—Active Mapping for Validation—as part of the SVD Encore, but integration and compute challenges with the existing software stack prevented optimal functionality during the demo. We will be addressing this early in the Fall Semester by upgrading to a higher compute and slimming and optimizing the codebase.

7.3 Test Plan

Following is the high-level test plan for the subsystems for the Fall Semester:

- **Sensors**:
  - Integrate and Test ZED 2i Camera for Validation
  - Integrate and Test SkyCam for localization
- **Computation**:
  - Optimize codebase to improve system reliability
  - Upgrade to better compute and integrate all the existing codebase
  - Test global positioning accuracy with updated localization methodology
  - Test autonomous avoidance of non-gradable craters with a fine-tuned Navigation stack
  - Integrate and test updated transport planner to target multiple craters
- **Mechanical**:
  - Test entire mechanical hardware for wear and tear
  - Test new iterations of wheels
- **Actuation**:
  - Test different actuators for better performance
  - Measure wheel torque to test stall estimation of the rover
- **Electrical**:
  - Test entire electrical hardware for wear and tear

These high-level tests are mapped to capability milestones and system requirements in Table 9.

7.3.1 Fall Validation Demonstration

The FVD test conditions are as follows:

- **Location**: Planetary Robotics Lab Moon Yard
- **Key Equipments**:
  - Lunar ROADSTER Rover
  - Leica TS16 Total Station (subject to change)
  - Operaations Terminal
  - TP-Link Router
  - Jetson TX2 Relay
- **Operating Area**: Moon Yard of approximately 7 m X 7 m area, with the Total Station at one corner (subject to change)

- **Pre-Demo Setup**:
  1. Prepare the Moon Yard with several craters and dunes in a circular path.
  2. Scan the Moon Yard with a FARO Scanner to obtain a global map for navigation.
  3. Attach and connect all the components and subsystems of the rover.
  4. Place the rover in the Moon Yard and calibrate its localization using a star-sun tracker, visual odometry, and/or total station.
- **During Demo**:
  1. Switch the rover to autonomous mode and use goal poses with offsets to plan the path.
  2. Observe the rover autonomously grade craters and level dunes in a circular path.
  3. After each dozed crater, use the ZED camera to validate whether the dozing satisfies the performance requirements.
  4. If anything unexpected occurs press the emergency stop button.
- **Quantitative Performance Metrics**:
  - M.P.1: Will plan a path with a cumulative deviation of <= 25% from chosen latitude's length.
  - M.P.2: Will follow planned path to a maximum deviation of 10%
  - M.P.3: Will climb gradients up to 15° and have a contact pressure of less than 1.5 kPa
  - M.P.4: Will avoid craters >= 0.5 metres and avoid slopes >= 15°
  - M.P.5: Will fill craters of up to 0.5 meters in diameter and 0.1m in depth
  - M.P.6: Will groom the trail to have a maximum traversal slope of 5°

The graphical illustration of the demo is shown in Figure 11.

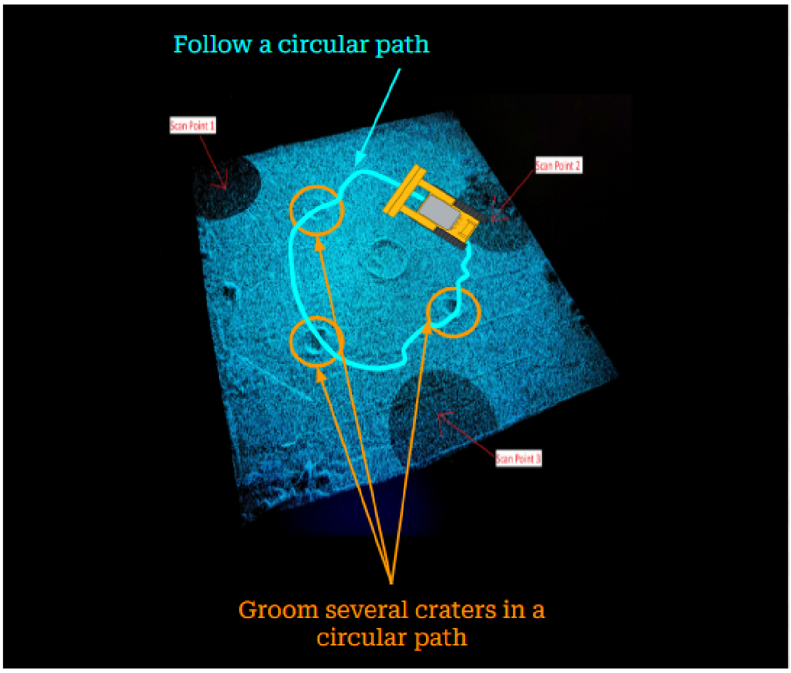


Figure 11: Fall Validation Demo - Graphical Illustration

Table 9: Subsystem Capability Milestones, Tests, and Requirements

Date	Event	Capability Milestones	Tests	Requirements
09/10	PR7	Hardware and software **refinement**	Validate **hardware upgrades**, **software fixes**, and **system stability** improvements	M.F.5, M.F.9, M.N.3
09/24	PR8	**Validation stack** setup. **Wheel torque** measurement. **Navigation** tuning.	Detect **robot stalling** through torque measurement, adjust tool height, and verify **smooth navigation** through path execution tests	M.F.2, M.F.3, M.F.4, M.F.8, M.F.9
10/08	PR9	Autonomous grading of **multiple craters**	Verify **autonomous grading performance** across multiple craters	M.F.2, M.F.3, M.F.4, M.F.5, M.F.6, M.F.7, M.F.9
10/29	PR10	SkyCam-based localization for **improved global positioning**	Test SkyCam-based localization by checking rover's ability to **self-localize accurately with/without external infrastructure**	M.F.3
11/12	PR11	**Full system integration Quality assurance testing**	Check all **subsystems and units are functioning correctly**	M.F.2–M.F.9
11/17 & 11/24	PR12 (FVD and Encore)	**Final system demonstration** involving autonomous grading of multiple craters	Demonstrate **full autonomous operation** by detecting, avoiding ungradable craters, and grading multiple suitable craters according to mission specs	M.F.1–M.F.9, M.P.1–M.P.9

7.4 Parts Lists and Budget

As of the Critical Design Review at the end of the Spring semester, \$2,309.07 of the \$5,000 MRSD budget has been spent. This constitutes approximately 46.2% of the MRSD budget. This leaves us with \$2,690.93 for the subsequent Fall semester, which will be mainly spent on a new Jetson compute device to manage higher real-time computational requirements. Including items provided to us free of charge by our supervisor and items inherited from the Crater Grader MRSD team from 2022, our total budget spent is \$5,379.07. This figure is not incorporated into the \$5,000 MRSD budget allocated to us. Table 10 outlines a refined parts list of major big-ticket item purchases. Figure 12 shows item purchases under the MRSD budget segregated by functionality. Our full parts list includes 42 entries and is available on our website:

`https://mrsdprojects.ri.cmu.edu/2025teami/parts-list/`

Table 10: Refined Budget Statement with Big-Ticket Purchases

ID	Part Name	Description	Unit	Quantity	Total
P01	NVIDIA Jetson	Computing board	$800	2	$1600*
P02	VN-100 IMU	IMU	$800	1	$800*
P03	Planetary Gear Motor	Motor	$60	4	$240*
P04	RoboClaw 2x30A	Motor Controller	$135	2	$270*
P08	ZED 2i	Stereo Camera	$562	1	$562
P15	Planetary Gear Motor	Motor	$60	4	$240
P25	Linear Actuator	Actuator	$97	3	$290
P34	DC/DC Power Converter	Power Converter	$316	1	$316
				MRSD Budget Total:	**$2309.07**
				Grand Total:	**$5379.07**
					*Excluded from MRSD budget

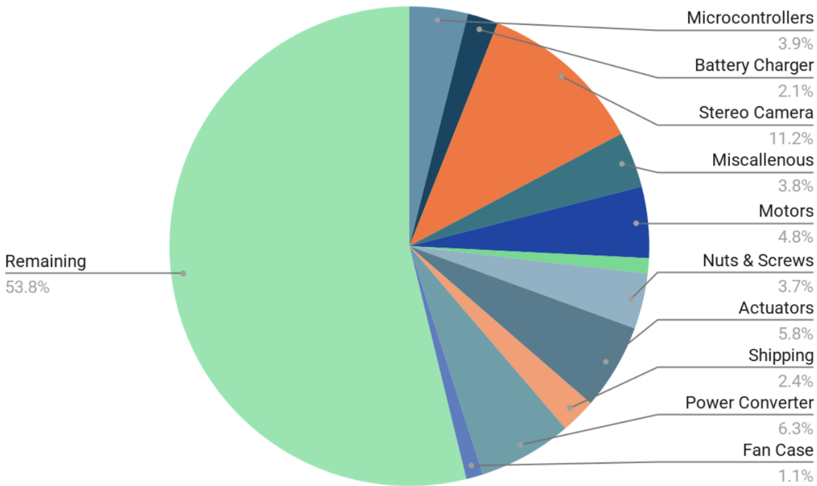
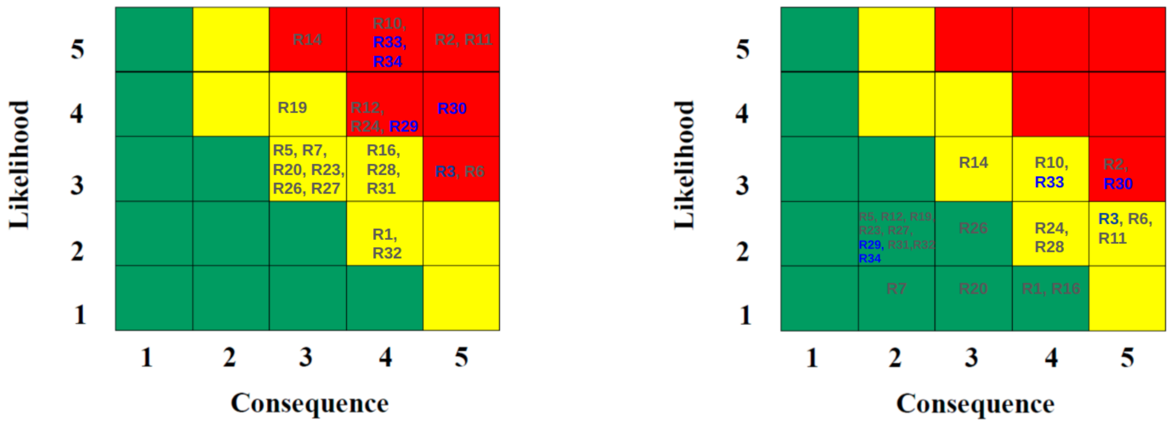


Figure 12: Pie Chart of Budget Expenditures

7.5 Risk Management

By the time of the Preliminary Design Review (PDR), several risks had been identified, some of which have since been mitigated. Following the PDR, additional risks R31, R32, R33 and R34 were identified during system integration and testing. This section highlights the top five highest-priority risks, based on their current likelihood and impact. The full risk management table is provided in Appendix A.8. For each listed risk, we include a preliminary mitigation plan along with any actions taken to address or reduce the associated threat.

The Figure 13 shows the Likelihood-Consequence tables for all risks identified so far, with (a) representing the unmitigated risks and (b) the risks after mitigation actions. The highlighted entries correspond to the top five risks actively tracked and addressed in the project.



(a) Unmitigated Risks **(b)** Mitigated Risks

Figure 13: Likelihood-Consequence Tables

7.5.1 Integration Issues between Subsystems

This risk had an unmitigated **likelihood of 3 and consequence of 5**. Due to the complexity of the system, individual subsystems may work correctly but fail to communicate properly when integrated, leading to schedule delays, requirement changes, and potential demo failure.

To address this, we adopted a common ROS2-based communication framework, performed continuous unit testing, and integrated subsystems incrementally. These actions were implemented successfully in Spring, reducing the **likelihood to 2 while the consequence remains at 5**. As several subsystem improvements are planned for the Fall, this risk persists and will require careful management to prevent reintegration delays.

7.5.2 Access to FRC Workshop

This risk had an unmitigated **likelihood and consequence of 4**. Since access to the FRC Workshop was essential for hardware fabrication and repairs, lack of access, especially in the absence of key personnel, posed a serious threat to meeting hardware deadlines.

To mitigate this, the team successfully utilized alternative fabrication labs on campus and coordinated with professor Red Whittaker, Tim and other stakeholders to gain temporary access to the FRC Workshop. These actions were implemented by early April and effectively reduced both the **likelihood and consequence to 2**.

7.5.3 No spares available

This is a critical risk with an unmitigated **likelihood of 4 and consequence of 5**. It arose after discovering that one of the rear axles had broken and disengaged from the driveline. As the rover model is discontinued, spare parts are no longer available, putting the entire project at risk.

With support from professor Red and Chuck, the team located a twin rover and salvaged compatible parts. Additional mitigations, including stocking similar components and maintaining mechanical subsystems, have reduced the **likelihood to 3, though the consequence remains high at 5**. This remains a top concern moving forward.

7.5.4 Localization frame shifts after total station battery swap

This risk was identified after the PDR during system testing for the Spring Validation Demo (SVD). It had an unmitigated **likelihood of 5 and consequence of 4**. Battery replacement in the total station introduced small frame shifts, resulting in localization inaccuracies that affected navigation and posed a risk of missing craters during grading.

To mitigate this, the team plans to implement a resection method using three known prism locations instead of the orientate-to-line approach, and explore alternative localization solutions such as SkyCam. These actions are expected to reduce the **likelihood to 3, while the consequence remains at 4** due to the critical role of accurate localization in mission success.

7.5.5 Arduino requires reset before teleoperation

This risk was identified after the PDR during system testing for the Spring Validation Demo (SVD). It had an unmitigated **likelihood of 5 and consequence of 4**. The Arduino required a manual reset each time before starting autonomy or switching between autonomy and teleoperation modes, which slowed down setup and impacted operational readiness.

To address this, the team planned actions including checking USB port permissions on the Jetson, ensuring the Arduino was connected via USB 3.0, and resolving ROS node frequency mismatches. These mitigations were implemented, resulting in a consistent and stable connection without the need for manual resets. The **likelihood and consequence have both been reduced to 2**, effectively minimizing the impact of this issue.

8 Conclusions

8.1 Key Lessons

8.1.1 Scheduling Lab Access and Resource Availability

During the Spring semester, one of our key blockers was gaining access to the FRC machine workshop. This was because we were only allowed access to the FRC when Tim was physically present in the workshop. Additionally, the lead times of hardware manufacturing is restricted by machine and resource availability. We had to wait for weeks for all four custom wheels to be 3D printed as other jobs also required access to the 3D printer. Through this experience, our team learned the lesson to schedule system dependencies that incorporates this lead time. We have even identified it as a top risk (Risk ID 29) and have taken action to mitigate this risk by using other fab-labs on campus such as CMU TechSpark.

8.1.2 Checking for Common Issues Online

One issue that we faced was interfacing the ZED 2i stereo camera with the Jetson docker. We spent considerable time and attention to resolving this issue to no avail. This is because the Jetson is too outdated and the CUDA drivers required by the ZED camera is not supported. We only realized this issue by going through common issues documentation online. Through this experience, our team learned the lesson to first check for potential common issues online before spending considerable time and attention to debugging. If we knew sooner that the issue cannot be resolved, we would not have wasted so much time and cause delays in our schedule.

8.1.3 Plan for Progress Review Goals Early

Although we have developed a Spring test plan to align our development milestones with the progress reviews, we still found ourselves scrambling to meet the milestones at the last minute. We have learned the lesson to plan early for what we want to achieve for each progress review and develop a more coherent plan during each test day to gather data and videos to present during the review.

8.1.4 Plan for Demo Split between SVD and SVD Encore

During the Spring semester, we have focused most of our attention on delivering the performance requirements for SVD and left very little flexibility for SVD Encore improvements. This led to rushed demonstrations for the Encore as we have already shown all deliverables during SVD itself. Going forward, we have learned the lesson to plan early for what to show during FVD and what extra to show for FVD Encore. This allows us more time to plan and test for the extra requirements that we want to show for FVD Encore.

8.1.5 Plan for Hardware and Software Integration

Our project involved building most of the hardware and software stack by ourselves. We did not use off-the-shelf fully integrated rovers nor completed software stacks. This meant that extra time commitment and debugging was necessary to integrate the hardware and software aspects of the project. We did not anticipate this integration to take this long. Going forward, we have learned to allocate more time for testing and integration so as to not rush the final product days before the demonstration.

8.2 Key Fall Activities

8.2.1 Allocate More Time for Integration

From the key lessons learned, one of our key fall activities is to plan and allocate more time for integration and testing. For the Fall semester, we plan on allocating time to integrate and test continuously for each unit and subsystem. Additionally, we plan on leaving at least 2 weeks between PR11 and FVD to work exclusively on full system integration and quality assurance testing. We plan on having a strict hardware and software deadline at PR11, and will not implement new functionalities after this deadline.

8.2.2 Improving and Refining Subsystems

Based on the deficiencies identified during SVD and SVD Encore, we have identified a TODO list of subsystems that requires improvement and refining. In particular, we will improve the tool actuator to remove the jittering motion and implement a wheel torque feedback monitor to identify slippage. We will also work on optimizing the overall software stack to reduce time complexity and upgrade the compute hardware to a new Jetson device to increase computational power and allow the integration with the ZED stereo camera. Finally, to solve the issue of localization frame shifts after total station battery swaps (Risk ID 31), we plan on implementing a new SkyCam-based localization method so that the rover can self-localize accurately without external infrastructure.

8.2.3 Grooming Multiple Craters

Our capability milestone for FVD is to demonstrate our final system autonomous grading multiple craters in a circular path around the Moon Yard. Our main task in the Fall semester is to achieve this milestone. This will involve extending our existing tool planner architecture and our FSM software stack. Additionally, a new validation stack will need to be implemented to verify the grading quality of each crater. Finally, the navigation stack will also need to be extended to handle curved trajectories.

9 References

[1] Teng Hu et al. "Population of Degrading Small Impact Craters in the Chang'E-4 Landing Area Using Descent and Ground Images". In: *Remote Sensing* 14.15 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14153608. URL: https://www.mdpi.com/2072-4292/14/15/3608.

[2] P. Muñoz, P. Bellutta, and M. D. R-Moreno. "Proposing new path-planning metrics for operating rovers on Mars". In: *Scientific Reports* 13 (2023), p. 22256. DOI: 10.1038/s41598-023-49144-8. URL: https://doi.org/10.1038/s41598-023-49144-8.

[3] Laurent Sibille et al. *Lunar Regolith Simulant Materials: Recommendations for Standardization, Production, and Usage*. Tech. rep. NASA/TP-2006-214605. NASA Marshall Space Flight Center, 2006. URL: https://ntrs.nasa.gov/api/citations/20060051776/downloads/20060051776.pdf.

A Appendices

A.1 Subsystem Completion Status

Table 11: Subsystem Completion Status

Subsystem	Completion	Future Work
Sensors	75%	Upgrade RealSense D235i to ZED 2i stereo camera
Computations	60%	
Jetson & Docker	95%	Integrate ZED drivers
Localization	85%	Address frame shifts caused by battery replacements
Transport Planner	60%	Implement unit
Navigation Planner	90%	Tune navigation stack
FSM Planner	95%	Integrate validation unit into FSM
Validation	10%	Implement unit
External Infrastructure	100%	None
Mechanical	90%	
Dozer Assembly	90%	Refinement of dozer
Wheel Assembly	90%	Wheel design iterations, final manufacturing
Actuation	80%	Linear actuator upgrade/tuning, torque sensing
Electrical Power	90%	Change battery placement

A.2 Localization Unit Implementation Details

A.2.1 EKF-based Sensor Fusion

Localization is achieved using an Extended Kalman Filter (EKF) implemented via the ROS2 `robot_localization` package. We implement two EKF nodes, as shown in Figure 14:

1. **Local EKF**: Fuses data from the VectorNav IMU and wheel encoders.
2. **Global EKF**: Fuses the data from the VectorNav IMU, wheel encoders along with absolute position data from the Leica TS16 Total Station.

We require both global and local localization nodes because they complement each other's weaknesses. The local localization node provides fast and smooth updates on the rover's motion, and is good for real-time control. However, it drifts over time because it is dead-reckoning. The global localization node helps correct any long-term drift and keeps the rover anchored to the `map` frame in the Moon Yard. However, it is too sluggish for steering control. By using both nodes together, we get the benefits of smooth short-term motion tracking and long-term position accuracy.

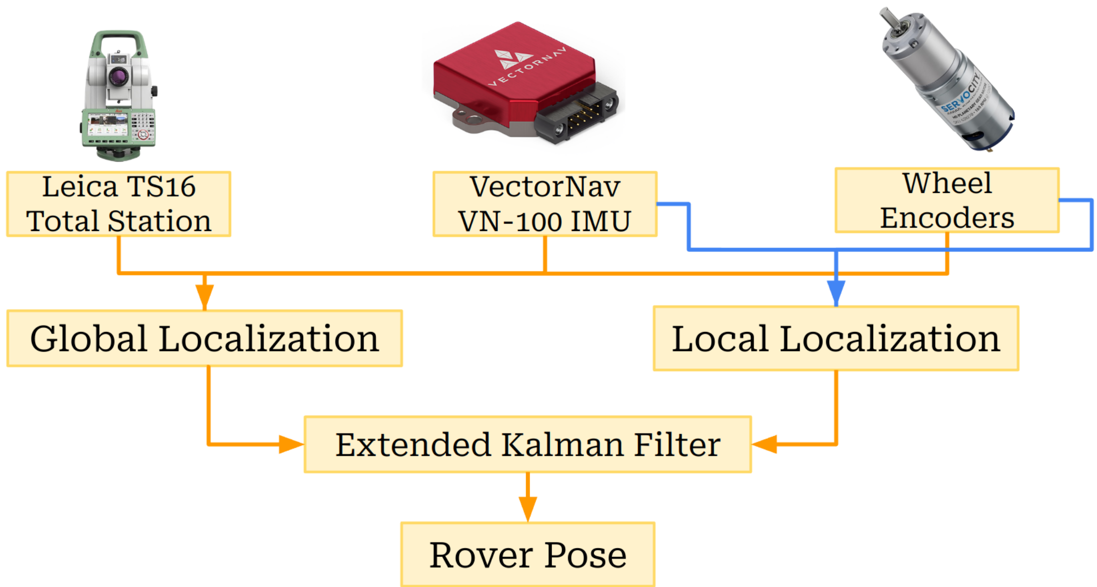


Figure 14: Localization Method

A.2.2 Coordinate Frame Alignment

The following coordinate frames are defined and broadcasted to ensure proper alignment of all sensor inputs:

- `map` : This is the global frame with origin at the corner of the Moon Yard.
- `odom` : This is the local frame origin that offers a continuous homogeneous transform to the `base_link` frame.
- `base_link` : This is the frame of the rover at its Center of Mass.
- `total_station_prism` : This is the frame of the Leica Prism mounted on the mast of the rover.
- `imu_link` : This is the frame of the IMU.

A.3 Transport Planner Unit Implementation Details

The global map is processed to identify source and sink locations, and assignments are generated to move regolith based on a cost-minimization strategy. These assignments are filtered to compute both tool actions and rover paths. The transport planner gives out 6 goal poses:

1. Offset pose: This is the initial pose, a certain distance away from the crater.

2. Source pose: It is positioned at the rim of the crater and the tool is first lowered here. It marks the starting point for forward dozing, where sand is pushed into the crater.

3. Sink pose: It is located at the center of the crater. The rover pushes the sand from the source to this sink while keeping the blade lowered. Upon reaching this pose, the blade is raised.

4. Backblading source pose: The rover moves to the opposite side of the crater for a return trip. At this position, the tool is lowered again to begin backblading.

5. Backblading sink pose: The rover moves back toward the beginning of the crater, dragging the sand in the backward direction. This grades the crater by backblading. Once complete, the dozer blade is raised.

6. Offset pose: The rover returns to the initial offset position, and this terminates the grading operation.

A.4 Navigation Planner Unit Implementation Details

A.4.1 Mapping and Costmap Generation

To enable autonomous navigation, the Moon Yard was mapped using the FARO Laser Scanner. We took three scans each time the environment was prepared and stitched them into a dense point cloud. This point cloud was converted into a ROS-compatible format and aligned with the rover's global map frame. Key steps in processing included:

1. Conversion of FARO's native `.fls` format into usable point cloud data.

2. Filtering and downsampling to reduce noise and computational load.

3. Applying RANSAC plane fitting to extract a ground plane and generate a 2D elevation-based costmap.

4. Setting the map origin to align with the localization frame for consistency across modules.

Figure 15 shows the Moon Yard scan visualization after stitching and the global costmap.

A.4.2 Crater Identification and Classification

We identified gradable craters from the processed map based on geometric features. Each crater was classified by:

- Diameter: Only craters within a 0.5 m diameter, as per our performance requirements, were marked as gradable.

- Depth: Only craters within 0.1 m depth were marked as gradable.

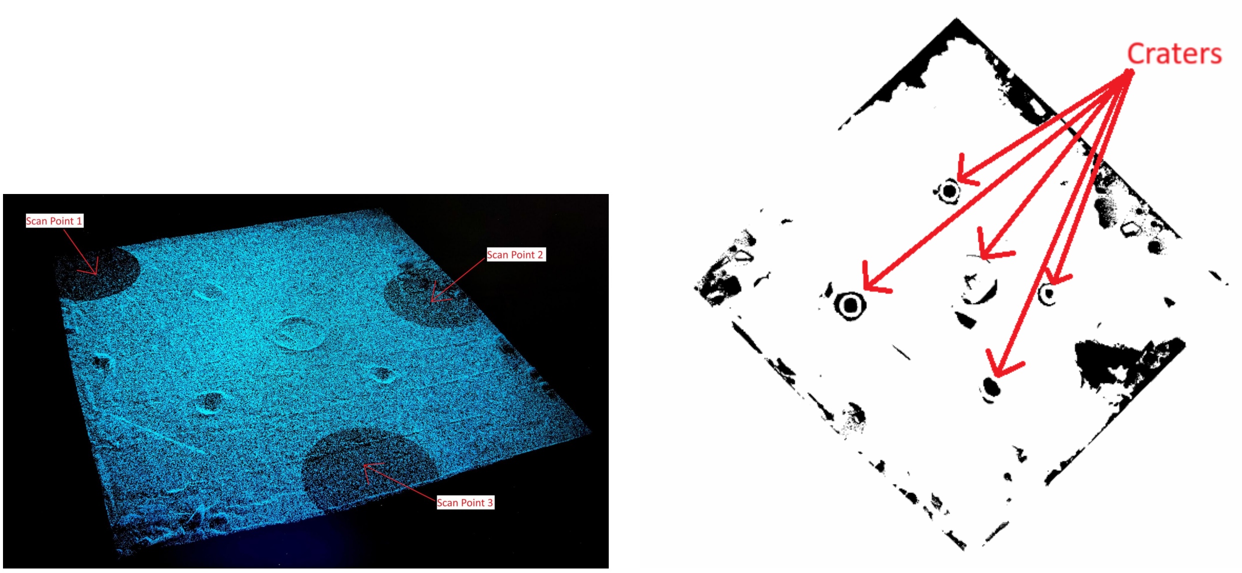


Figure 15: Left, Moon Yard Scan Visualization. Right, Global Costmap

A.5 Finite State Machine Unit Implementation Details

- We implemented the FSM as a behavior executive node running at 2 Hz to control high-level autonomy flow.
- States include: `UPDATE_MAP`, `PLAN_TRANSPORT`, `GET_TRANSPORT_GOALS`, `FOLLOW_TRAJECTORY`, `GET_TOOL_GOAL`, and `END_MISSION`, among others. This is illustrated in Figure 16.
- Each FSM transition triggers service callbacks such as `updateMap()`, `planTransport()`, `getGoalPose()`, `handleToolTrajectory()`, and `sendNavigationGoal()`.
- Heavy computations (e.g., trajectory generation, path planning) were parallelized using ROS2 multithreading to avoid blocking the FSM loop.
- Status messages from the FSM node were logged to monitor state transitions, goal completions, and command execution.

A.6 Electrical Subsystem Implementation Details

- We use three 20V batteries to power the system.
- A buck converter steps down voltage to 12V for logic-level components.
- Power is routed through an Emergency Stop, wireless switch (FOB), and monitored using two battery sensors before reaching the custom Power Distribution Board (PDB).
- The PDB supports connection to RoboClaw motor controllers, actuator controllers, Jetson, Arduino Due, and other peripherals.

Figure 17 shows the detailed circuit schematic. We modeled the PDB, shown in Figure 18, with protective features, including over-current, reverse-voltage, and indicator LEDs. We also evaluated the voltage and current demands of all subsystems to ensure load balancing and stable PDB output.

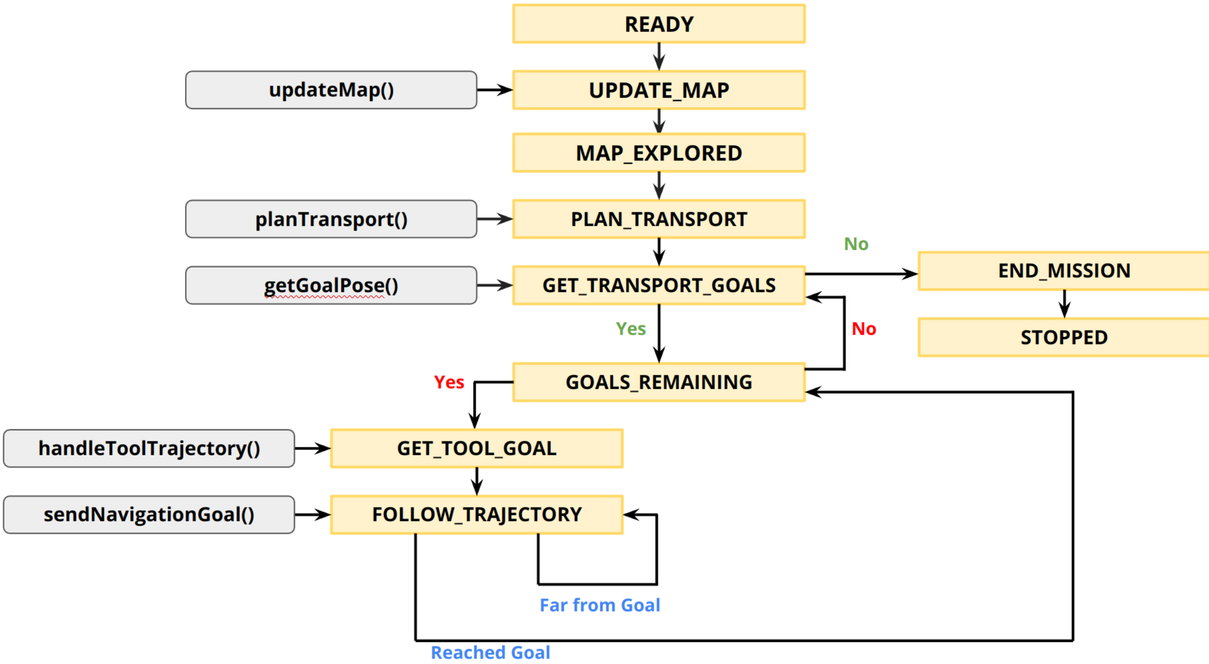


Figure 16: Finite State Machine Behavior Tree

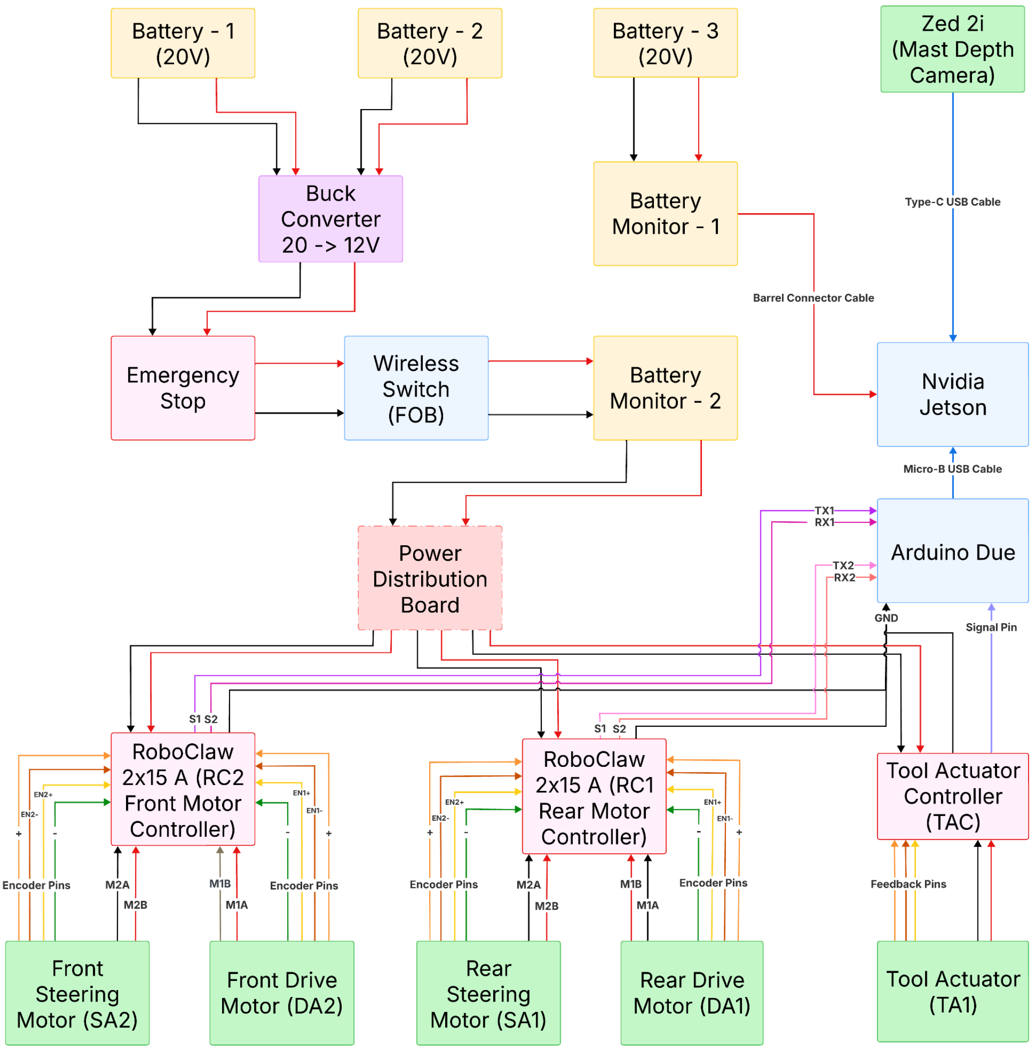


Figure 17: Circuit Schematic

Figure 18: Power Distribution Board

A.7 Subsystem-Wise Detailed WBS

A.7.1 Mechanical

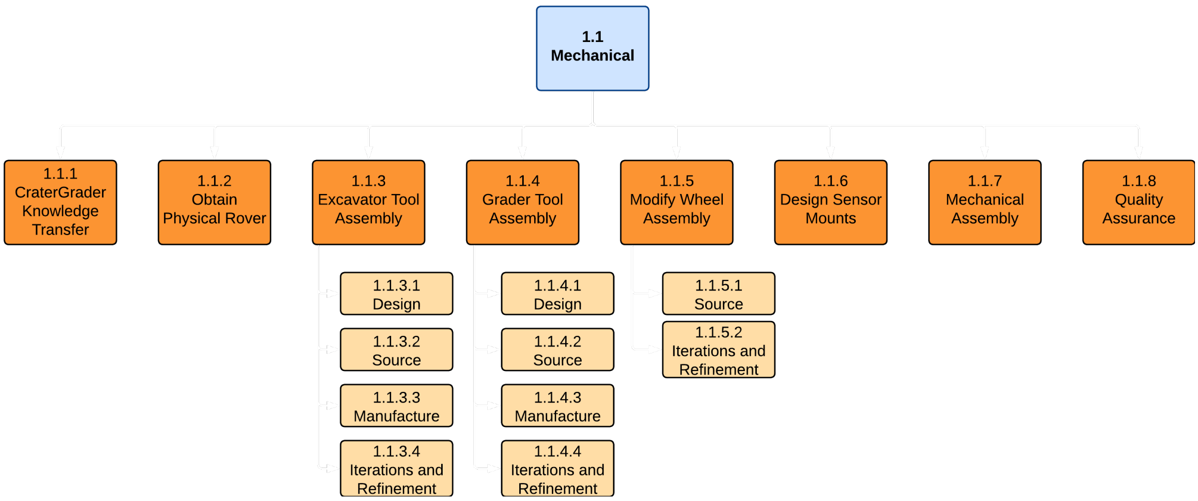


Figure 19: Level 4 WBS for Mechanical Subsystem

Figure 19 shows the Level 4 Work Breakdown Structure for the Mechanical Subsystem. The tasks shown are:

- Crater Grader Knowledge Transfer and obtaining Physical Rover involve reading through the documentation, going through CAD files and assessing the condition of the robot.
- The Modify Wheel Assembly task will involve sourcing new wheels and working on integrating them with the current drive system of the rover.
- The Mechanical Assembly task will involve mounting all created sub-assemblies onto the main rover chassis.
- The Quality Assurance block will involve unit testing to ensure that the created assemblies meet the desired standards of functionality and robustness.

A.7.2 Computations

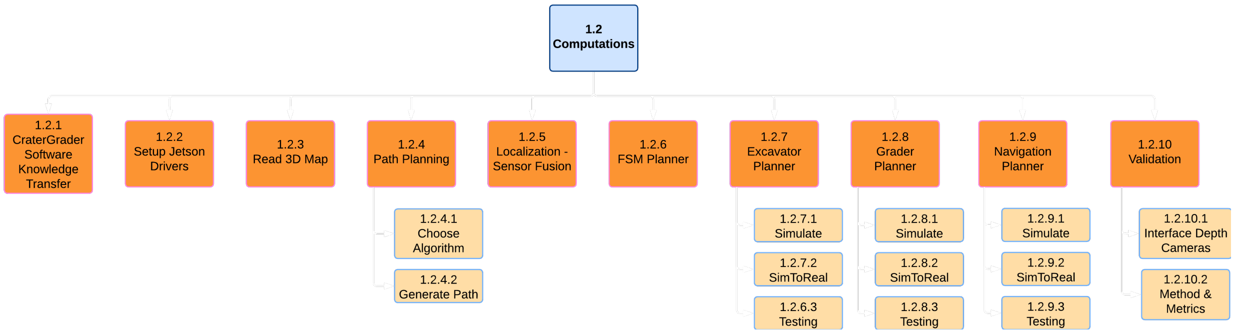


Figure 20: Level 4 WBS for Computations Subsystem

Figure 20 shows the Level 4 Work Breakdown Structure for the Computations Subsystem. The tasks shown are:

- The first 3 tasks are initial setup tasks involving research, setting up drivers and obtaining initial data.
- The Path Planning task will follow a regular pipeline of choosing an algorithm and generating a path

A.7.3 Sense

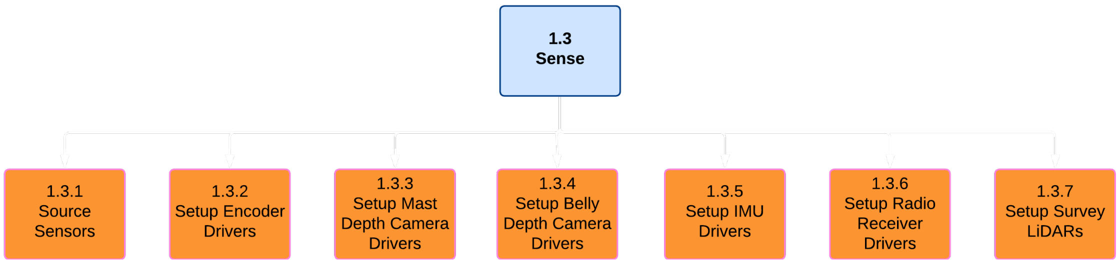


Figure 21: Level 4 WBS for Sense Subsystem

Figure 21 shows the Level 4 Work Breakdown Structure for the Sense Subsystem. The tasks shown are:

- This subsystem involves sourcing our sensor stack - Encoders, Depth Cameras, IMU, Radio Receivers and Survey LiDARs.
- The tasks also include setting up the required drivers and obtaining data using the desired communication method.

A.7.4 Electronics

Figure 22 shows the Level 4 Work Breakdown Structure for the Electronics Subsystem. The tasks shown are:

- The Source Electronics task involves sourcing electronic components - Wireless Transceiver, Motor Controllers, Power Distribution Board, E-Stops, Fuses and Power Supply.

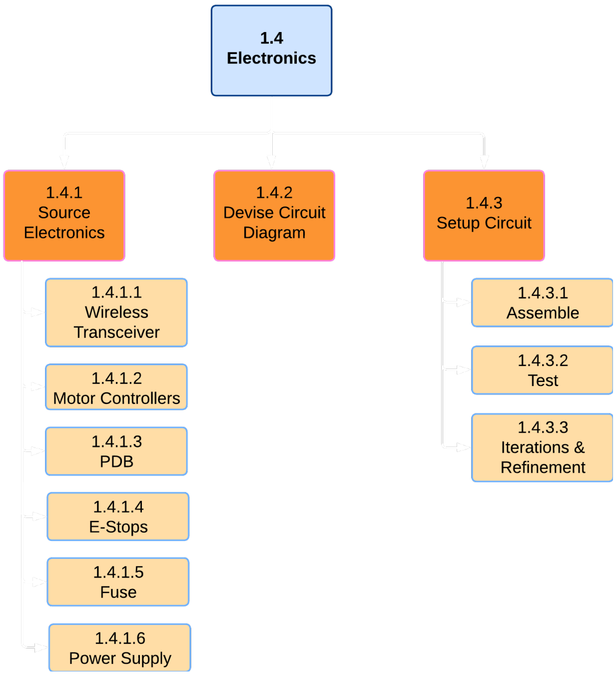


Figure 22: Level 4 WBS for Electronics Subsystem

- The Devise Circuit Diagram involves creating and optimal circuit to connect all components to their required inputs and outputs.
- Setup Circuit will consists of assembling, testing and iterating the circuit to ensure robustness.

A.7.5 External Infrastructure

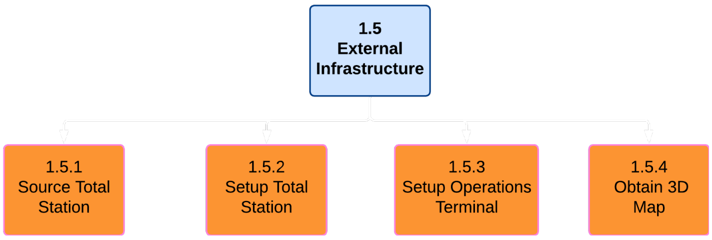


Figure 23: Level 4 WBS for External Infrastructure Subsystem

Figure 23 shows the Level 4 Work Breakdown Structure for the External Infrastructure Subsystem. This consists of all components of the project that are not on the rover. The tasks shown are:

- Sourcing and Setting up total station will require training with FRC Technicians. It will involve calibrating and tracking the rover's postion during the mission.
- The Operations Terminal will be the team's laptops, emulating the Moon Station. This station will be used to monitor the mission and check for the validation conditions.
- Obtain 3D Map involves obtaining a map of the Moon Yard using the Survey LiDAR. This map will be used to plan the trail path to be groomed by the ROADSTER.

A.7.6 Integration and Testing

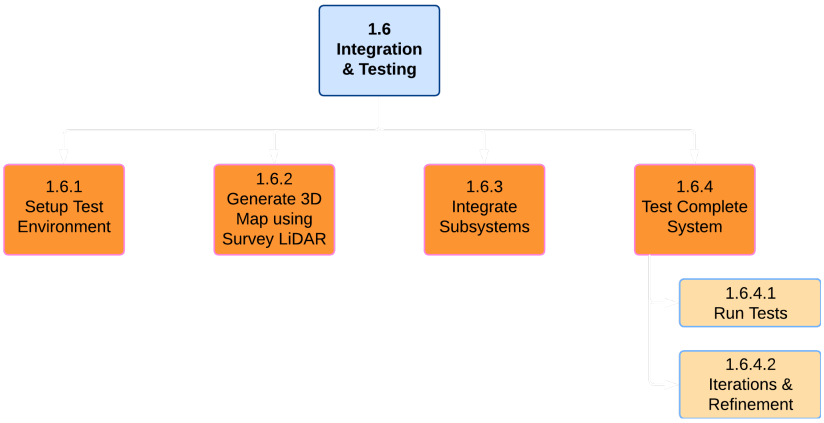


Figure 24: Level 4 WBS for Integration and Testing Subsystem

Figure 24 shows the Level 4 Work Breakdown Structure for the Integration and Testing Subsystem. The tasks shown are:

- All tasks involve ensuring that all subsystems can work together as a cohesive unit.
- The team will setup concrete testing plans with varying environments to test the ROADSTER
- The Iterations and Refinement block will allow the team to improve overall functionality of the rover, focusing on robustness and repeatability.

A.7.7 Management

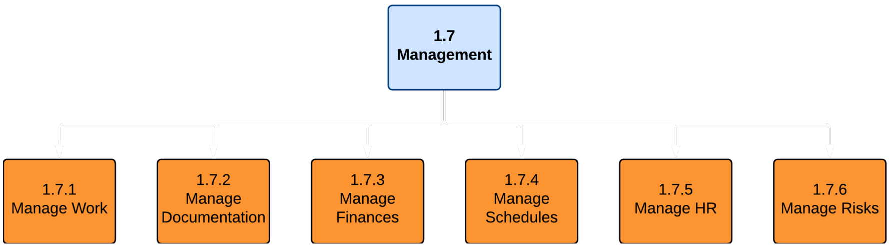


Figure 25: Level 4 WBS for Management Subsystem

Figure 25 shows the Level 4 Work Breakdown Structure for the Integration and Testing Subsystem. The Program Manager will be responsible for all tasks in the Management Subsystem. The tasks shown are:

- Manage Work involves tracking progress of the assigned tasks to each member.
- Manage Documentation will cover all reports, presentations and process documentations through the work period.
- Manage Finances involves managing purchases and allocating adequate budget to every subsystem.
- Manage Schedules involves tracking the overall timeline of the project and ensuring that the team follows the planned schedule.

- Manage HR is a task where the team will allocate some time every week to uplift team morale and maintain motivation.
- Manage Risks involves identifying and mitigating any potential risks in the project.

A.8 Identified Risks

The following risks are identified during our preliminary risk management analysis. Potential mitigation actions are put in place to minimize these risks.

Risk ID	**Risk Title**	**Risk Owner**	**Date Added**	**Date Updated**
R1	PRL Testbed Scheduling	Ankit	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
PRL Testbed unavailable due to scheduling conflicts with other high priority projects			**2**	**4**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
No testbed available for testing and/or SVD			**1**	**4**
Risk Reduction Plan Summary			**Risk Type:**	Scheduling
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Devise and discuss a testing and demo plan with Red and other stakeholders of the PRL testbed beforehand and reserve slots		Successfully reserve slots for using the PRL Testbed	11/30/2024	
Reach out to external testing facilities like Astrobotic or CAT for a backup testing facility		Communicate and discuss possibility of using other testbeds if PRL falls through		
Schedule tests at night		Schedule tests at off-hours to avoid clashes		
Comments				

Figure 26: PRL Testbed Scheduling

Risk ID	**Risk Title**	**Risk Owner**	**Date Added**	**Date Updated**
R2	Excavator and grader tool planner takes longer than expected to deliver	Simson	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Integration of the excavator and grader software with hardware takes longer than expected			**5**	**5**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Unable to meet SVD deadline and potential requirements change			**2**	**5**
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Shift requirements for SVD		Working prototype for SVD	11/28/2024	11/28/2024
Integrate the grader during Fall semester		Working excavator and grader for FVD	11/28/2024	
Potentially use off-the-shelf code if avaliable, preferably from CraterGrader		Successful integration of off-the-shelf components		
Comments				
Decided to move delivery of grader tool planner to the Fall semester				

Figure 27: Excavator and Grader Tool Planner takes Longer than Expected to Deliver

Risk ID	Risk Title		Risk Owner	Date Added	Date Updated
R3	Integration issues between subsystems		Deepam	11/27/2024	11/27/2024
Description				**Original Likelihood**	**Original Consequence**
Subsystems work individually, but integration and communication between the subsystems are flawed				**3**	**5**
Consequence				**Mitigated Likelihood**	**Mitigated Consequence**
Delay in integration causing scheduling overruns, requirements change and failure of the demo				**2**	**5**
Risk Reduction Plan Summary				**Risk Type:**	Technical
Action/Milestone			**Success Criteria**	**Date Planned**	**Date Implemented**
Perform unit testing and subsystem validation continuously			Successful testing of all major subsystems	11/30/2024	04/04/2025
Integrate one subsystem at a time			Successful integration of all major subsystems	11/30/2024	04/04/2025
Use a common framework (e.g. ROS2 interfaces) for communication between subsystems to reduce bugs			Adoption of common framework for communications	11/30/2024	01/18/2025
Keep to planned schedule and have at least 5 weeks for testing and integration			Successful integration of all major subsystems	11/30/2024	03/25/2025
Comments					

Figure 28: Integration Issues between Subsystems

Risk ID	Risk Title		Risk Owner	Date Added	Date Updated
R4	Belly depth sensor is not suitable for validation		Bhaswanth	11/27/2024	11/27/2024
Description				**Original Likelihood**	**Original Consequence**
The belly depth camera is used to validate if a groomed crater is satisfiable. The sensor may not be able to adequately determine depth variations suitable for validation				**4**	**3**
Consequence				**Mitigated Likelihood**	**Mitigated Consequence**
Will result in major revision and changes to the validation architecture and functional requirement, causing delays in scheduling				**2**	**2**
Risk Reduction Plan Summary				**Risk Type:**	Technical
Action/Milestone			**Success Criteria**	**Date Planned**	**Date Implemented**
Mount the depth camera at another location on the rover (e.g. on a mast)			Acceptable validation specifiied by performance requirement		
Use another sensor to determine depth variations (e.g. LIDAR, visual odometry, IR sensor)			Acceptable validation specifiied by performance requirement		
If all else fails, use the total station for validation			Acceptable validation specifiied by performance requirement		
Comments					

Figure 29: Belly Depth Sensor is not Suitable for Validation

Risk ID	Risk Title		Risk Owner	Date Added	Date Updated
R5	Unable to get Crater Grader to perform autonomous crater filling		Bhaswanth	11/27/2024	11/27/2024
Description				**Original Likelihood**	**Original Consequence**
Our rover builds on top of the work acomplished by Crater Grader. If we cannot get Crater Grader to perform autonomous crater filling, we may need to spend time on the navigation stack and design the entire pipeline				**3**	**3**
Consequence				**Mitigated Likelihood**	**Mitigated Consequence**
Extra time commitment to start from scratch or obtaining a suitable replacement				**2**	**2**
Risk Reduction Plan Summary				**Risk Type:**	Technical
Action/Milestone			**Success Criteria**	**Date Planned**	**Date Implemented**
Thoroughly go through Crater Grader's code and the mechanical schematics provided			Thoroughly understand Crater Grader's operations	11/27/2024	
Test each component and wiring to see if they are working			Validate all components and replace broken ones	11/28/2024	
If it is still not working, inherit only the software component from Crater Grader and build hardware ourselves			Working prototype for SVD		
Comments					

Figure 30: Unable to get Crater Grader to Perform Autonomous Crater Filling

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R6	Delay in arrival and manufacture of hardware components	William	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Shipping delays of components ordered and/or manufacturing delays on custom made components			3	5
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Delays in hardware integration, causing pushbacks in schduling and software development			2	5
Risk Reduction Plan Summary			**Risk Type:**	Scheduling
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Use off-the-shelf components that are avaliable on hand (e.g. from CMU labs or Red's workshop)		Obtain components before end of December		
Start ordering and designing components during Winter break so there is adequate leeway for delivery and manufacturing before Spring semester starts		Order components before end of December	11/27/2024	12/09/2024
Use simulations to work on software components while we wait for the components to be delivered and/or manufactured		Successful integration of all subsystems on schedule		
Implement other subsystems that are independent from the subsystem that is missing parts		Successful integration of all subsystems on schedule		
In case of delay in wheels, work with the existing wheels and proceed with the timeline while waiting for the new ones to arrive		Successful integration of all subsystems on schedule		
Comments				

Figure 31: Delay in Arrival and Manufacture of Hardware Components

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R7	Lack of proper simulation environment	Simson	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Inability to accurately simulate the rover in a Lunar-like environment can lead to suboptimal performance			3	3
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
The rover's performance in the Moon Pit may be compromised, leading to inefficiencies, mission delays, or potential failure in achieving key objectives			1	2
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Ask CraterGrader how they ran all their simulations and gather resources		Meet with CraterGrader team	11/28/2024	12/2/2024
Explore LunarSim - https://github.com/PUTvision/LunarSim and check how useful this will be, during the winter break		Working simulation	12/12/2024	
Develop Gazebo environment		Working simulation		
Comments				

Figure 32: Lack of Proper Simulation Environment

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R8	Mast depth camera field of view (FOV) is blocked	William	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Mast depth camera's FOV can be blocked, partially or completely, due to dust, misalignment of camera, or interference from the rover's own excavator assembly.			5	4
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Hinders the rover's ability to perceive its surroundings accurately, resulting in navigation errors and inefficiencies in excavation tasks			3	4
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Conduct field tests to choose an optimal height to place the depth camera such that dust does not reach it and it can clearly see in front of the rover, despite the excavator assembly. Ensure that visual data such as depth perception and object detection should not be compromised		Working mast depth camera with a clear FOV		
Comments				

Figure 33: Mast Depth Camera Field of View (FOV) is Blocked

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R9	Too many performance requirements	Ankit	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
We have a lot of performance requirements and we may not be able to meet all of them by April for SVD			**5**	**5**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Delays in testing and validation, impacting project timelines and April SVD Demo results			**2**	**5**
Risk Reduction Plan Summary			**Risk Type:**	Technical, Scheduling
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Have revised performance requirements separately for SVD and FVD (focus more on SVD)		Achievable Performance Requirements	11/28/2024	12/4/2024
Talk to CraterGrader and discuss what is feasible and what is not in the given time		Meeting conducted	11/28/2024	12/2/2024
PM should track schedule properly and team members have to push to meet the timeline		Project follows the schedule	11/28/2024	
Comments				

Figure 34: Too Many Performance Requirements

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R10	Drive system wear-and-tear causes malfunction	Deepam	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
The transmission and steering assembly might be worn out, leading to suboptimal vehicle dynamics, and potentially mechanical failure			**4**	**4**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Rover drive system fails and may require a lot of repair and maintenance			**2**	**2**
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Thoroughly check the Crater Grader's assembly and carry out maintenance of any worn-out parts		Successfully understand and carry out maintenance of existing parts and assemblies		
Completely replace the assembly parts with the same/similar new parts for better performance and reliability		Order and stock spares		
Add limit switches to avoid steering gears to operate beyond their limits		Limit switches added		
Comments				

Figure 35: Drive System Wear-and-Tear Causes Malfunction

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R11	Dust ingress	William	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Due to significant sand manipulation, the flying sand/dust can enter and accumulate over sensitive electronics (PDB, drivers, Arduino) and sensors (cameras, IMU), leading to component failure or incorrect sensing			**5**	**3**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Component failure during testing or demonstrations. Highly inhibits all future scheduled tasks			**3**	**3**
Risk Reduction Plan Summary			**Risk Type:**	Technical, Cost
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Design proper sand enclosures and mounts for sensitive components		Successfully design and manufacture enclosures		
Review placement of components		Components are placed aptly, away from dust		
Review scale and speed of sand manipulation to eliminate root-cause of flying sand/dust		Select the sweet spot for apt tool speed with least flying dust/sand		
Allocate contingency budget and order spares of the sensitive components in case of component failure		Order and stock spares		
Comments				

Figure 36: Dust Ingress

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R12	Code version control	Simson	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Code modifications or config parameter changes during testing might not be saved, affecting the final demo. Reverting to a stable version is difficult if changes do not work as expected			**3**	**4**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Delay in code integration and implementation			**1**	**4**
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Implement GitHub version control to store and retrieve the best versions of code and configuration		Successful tracking of code changes		
Use Google Drive to back up important documentation explaining setup processes		Reduces delays during testing due to quick access to setup processes		
Comments				

Figure 37: Code Version Control

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R13	Items missing	Ankit	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Critical project items may go missing if not stored properly or tracked. Items may be misplaced or borrowed without proper logging			**4**	**3**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Delay in hardware implementation			**2**	**2**
Risk Reduction Plan Summary			**Risk Type:**	Logistics
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Maintain an inventory tracking spreadsheet		Ensures availability of required tools and materials		
Include spare inventory		Reduces downtime caused by missing or damaged items		
Comments				

Figure 38: Items Missing

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R14	Sensor ROS packages not available	William	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Finalized sensors might lack compatible ROS packages, leading to delays or significant changes in the software architecture			**3**	**3**
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Delay in software implementation			**1**	**3**
Risk Reduction Plan Summary			**Risk Type:**	Technical, Scheduling
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Perform trade studies to pick sensors that are compatible with ROS versions before finalizing		Sucessful sensor-ROS compatibility		
Select sensors and ROS versions that minimize potential conflicts		Streamlined integration with minimal issues		
Comments				

Figure 39: Sensor ROS Packages not Available

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R15	Lunar-accurate cut/fill regions are not possible to groom	Simson	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
The rims of the craters may not be enough to fill the whole crater. Going to a different region to carry the sand to the crater may prove to be inefficient			3	3
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
The basic assumption of sand availability fails. We may need to rethink the basic concept of tool planner to fit the new parameters of the environment.			2	2
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Accurately create the environment and assess if the rims are enough to fill		Assessment gives us adequate information		
If not, modify PRs accordingly		Achievable Performance Requirements		
Comments				

Figure 40: Lunar-Accurate Cut/Fill Regions are not Possible to Groom

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R16	Sensor data is too noisy to fulfill performance requirements	William	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
Performance requirements are tough and ambitious, sensor noise may prevent us from achieving it			4	4
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Failure to demonstrate performance requirements may cause us to lose marks in the demonstrations			2	4
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Relax the performance requirements enough to ensure that they are achievable		Achievable Performance Requirements		
Ensure enough testing time to tune parameters		Fully planned testing cycle	11/28/2024	
Comments				

Figure 41: Sensor Data is too Noisy to Fulfill Performance Requirements

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R17	Off-the-shelf wheels don't interface with the rover	Ankit	11/27/2024	11/27/2024
Description			**Original Likelihood**	**Original Consequence**
No off-the-shelf wheels fit the rover, We'll have to redesign wheel hubs and mountings as per the new wheels.			3	3
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Continue with sub-optimal wheels that the rover currently has, thus, not meeting one of the non-functional requirements			2	3
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Shift requirements to FVD		Updated SVD and FVD requirements for wheels		
Good enough market research to see find the best fit, with least amount of changes		Finding and replacing current wheels with new wheels, with least modifications		
Comments				

Figure 42: Off-the-Shelf Wheels don't Interface with the Rover

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R27	TX2 Integration	William	1/2/2025	2/24/2025
Description			**Original Likelihood**	**Original Consequence**
Unable to login to TX2 and interface with a LAN network for transmitting data over WiFi ro Jetson			3	3
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Delay in finalizing localization stack			2	2
Risk Reduction Plan Summary			**Risk Type:**	Technical
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Set up a new TX2 (Re-flash the TX2).		Successfully set up a new TX2 or re-flash the existing TX2 with updated credentials and settings	01/02/2025	
Reach out to previous teams to understand their methodology and retrieve credentials		Successfully retrieve credentials and interface with LAN	01/02/2025	02/19/2025
Comments				

Figure 43: TX2 Integration

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R28	Electrical Hardware Finalization	Ankit	2/14/2025	2/14/2025
Description			**Original Likelihood**	**Original Consequence**
E-box Design dependence on to-be manufactured PDB.			3	4
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Not meeting the hardware deadline			2	4
Risk Reduction Plan Summary			**Risk Type:**	Technical, Logistics
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Use previous knowledge and account for a placeholder in the design.		Successfully design and manufacture E-box compatible with the new PCB using placeholder PCB design	02/14/2025	04/06/2025
Order PCB and components (and spares) outside of MRSD schedule		Successfully order and assemble the PCB	03/26/2025	04/04/2025
Comments				

Figure 44: Electrical Hardware Finalization

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R29	Access to FRC Workshop	Deepam	2/7/2025	2/7/2025
Description			**Original Likelihood**	**Original Consequence**
Without access, no hardware fabrication/repairs can be carried out in the absence of Tim			4	4
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
			2	2
Risk Reduction Plan Summary			**Risk Type:**	Logistics
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Try other fab-labs on campus.		Successfully access other fab-labs and manufacture components	2/9/2025	
Request Tim, John or Red for getting temporary access, if not permanent		Successfully get temporary/permanent access to FRC Workshop	2/12/2025	
Comments				

Figure 45: Access to FRC Workshop

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R30	No spares available	Team	3/4/2025	3/4/2025
Description			**Original Likelihood**	**Original Consequence**
Discontinued model, spare parts unavailable			4	5
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
The whole project falling through, or redo almost all subsystems on a different rover.			3	5
Risk Reduction Plan Summary			**Risk Type:**	Logistics
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Check out eBay and other similar platforms for spares		Successfully find exact spares on these platforms	3/6/2025	
Check out and stock similar parts if not same		Successfully find and stock similar parts	3/6/2025	
Find a twin rover that was used by a previous team on campus		Successfully find the twin rover and scavenge parts	3/6/2025	3/7/2025
Maintain all parts, especially mechanical parts		Successfully avoid future breakdowns and part failures	3/7/2025	
Comments				

Figure 46: No spares available

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R31	Localization Accuracy	Team	4/2/2025	4/2/2025
Description			**Original Likelihood**	**Original Consequence**
The robot experiences delays in localization and minor positional offsets during movement			3	4
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
May prevent the robot from reaching the goal location, blocking the transition to the tool planner and affecting grading			2	2
Risk Reduction Plan Summary			**Risk Type:**	Logistics
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Tune EKF parameters and ensure synchronized motor speeds for accurate odometry		Successfully localize the robot in the moon yard during movement	4/2/2025	
Comments				

Figure 47: Localization Accuracy

Risk ID	Risk Title	Risk Owner	Date Added	Date Updated
R32	Navigation and Tool Planner Stack Integration	Team	4/4/2025	4/4/2025
Description			**Original Likelihood**	**Original Consequence**
Navigation and tool planners use differently processed costmaps generated from the point cloud. As the navigation package requires waypoints in a specific format and in a sequence, the tool planner will have to modified and tuned accordingly			2	4
Consequence			**Mitigated Likelihood**	**Mitigated Consequence**
Errors in manipulation of sand, leading to sub-par grading			2	2
Risk Reduction Plan Summary			**Risk Type:**	Logistics
Action/Milestone		**Success Criteria**	**Date Planned**	**Date Implemented**
Extensive testing and tuning		Successfully tune parameters and integrate navigation and tool planner stack	4/5/2025	
Minimize differences between the maps used by tool planner and navigation stack by setting the correct map frame in the point cloud		One ground truth map frame	4/1/2025	04/01/2025
Comments				

Figure 48: Navigation and Tool Planner Stack Integration

Risk ID	Risk Title		Risk Owner	Date Added	Date Updated
R33	Localization frame shift after total station battery swap		Bhaswanth	3/4/2025	4/10/2025
Description				Original Likelihood	Original Consequence
Battery replacement in the total station causes small frame offsets, leading to localization inaccuracies.				5	5
Consequence				Mitigated Likelihood	Mitigated Consequence
Leads to poor navigation performance and risk of missing the crater during grading operations.				3	4
Risk Reduction Plan Summary				Risk Type:	Logistics
Action/Milestone			Success Criteria	Date Planned	Date Implemented
Implement resection method using three known prism locations instead of orientate-to-line			Successfully fix the frame consistently after battery swaps	4/26/2025	
Explore and test alternative localization methods (using SkyCam)			Successfully maintain localization accuracy	4/26/2025	
Comments					

Figure 49: Localization frame shift after total station battery swap

Risk ID	Risk Title		Risk Owner	Date Added	Date Updated
R34	Arduino requires reset before operation		Bhaswanth	3/4/2025	4/10/2025
Description				Original Likelihood	Original Consequence
Arduino needs to be manually reset each time before starting autonomy or switching between autonomy and teleoperation modes.				5	5
Consequence				Mitigated Likelihood	Mitigated Consequence
Slows down setup time and impacts operational readiness, delaying mission start and mode transitions.				3	4
Risk Reduction Plan Summary				Risk Type:	Logistics
Action/Milestone			Success Criteria	Date Planned	Date Implemented
Check USB port permissions and drivers issues on Jetson			Successfully establish consistent serial connection without reset	4/26/2025	
Verify that Arduino is connected via USB 3.0 instead of USB 2.0 port			Ensure stable high-speed communication	4/26/2025	
Check for ROS node frequency mismatches causing packet loss to Arduino			Match ROS publish/subscribe rates	4/26/2025	
Comments					

Figure 50: Arduino requires reset before operation