Final Report

Lunar Autonomous Regolith Excavator

Dhruv Tyagi
Anishram Senathi
Hariharan Ravichandran
Vibhakar Mohta
Vivek Chervi

Academic Sponsor

Field Robotics Center

Dr. William "Red" Whittaker

Industry Sponsor

CATERPILLAR®

Caterpillar Inc.

13th December 2023

Carnegie Mellon University

The Robotics Institute

Master of Science Robotic Systems Development

Abstract

In order to establish a sustainable human presence on the Moon's surface, a significant amount of raw material is required to construct the necessary infrastructure. Lunar regolith is one of the primary materials that can be employed for many purposes in pursuance of that goal, while also being abundant on the Moon. Vast amounts of regolith must be acquired to be used for the construction of lunar landing sites and other facilities. This report outlines the design, development, and performance of a robotic system called the Lunar Autonomous Regolith Excavator (LunAR-X), aimed at excavating regolith to be used for building elemental shapes starting with a berm. The LunAR-X system is built to allow complete autonomous excavation and construction while adhering to many constraints imposed by the lunar conditions. The operation starts with the operator specifying the desired berm configuration, along with the areas in the worksite designated for acquiring and depositing the regolith. Following this, the robotic system executes the required excavation and deposition cycles autonomously, resulting in the constructed berm. This report also discusses the project management methodologies and tools that were followed to ensure this project's success.

Table of Contents

1. Project Description

The Moon is the key to enabling humanity's quest to explore the solar system. A lunar presence will be a gateway to other planets and a research site to develop deep-space technologies and study more about our celestial neighbor. This endeavor requires considerable construction and volumetric site work on the Moon. For this objective, it would be highly impractical to transport raw material in bulk from Earth. Therefore, we will rely on the prevailing materials on the Moon's surface, specifically the lunar regolith, adopting the concept of In-Situ Resource Utilization (ISRU).

Lunar regolith can be utilized to serve numerous other purposes as well. Studies conducted on the chemical makeup of the regolith suggest that it has elements that can help synthesize propellants and run life support systems [1, 2, 3, 4, 5]. Radiation protection, In-Situ manufacturing, habitat construction, and metal production are also some valuable applications of the material [6].

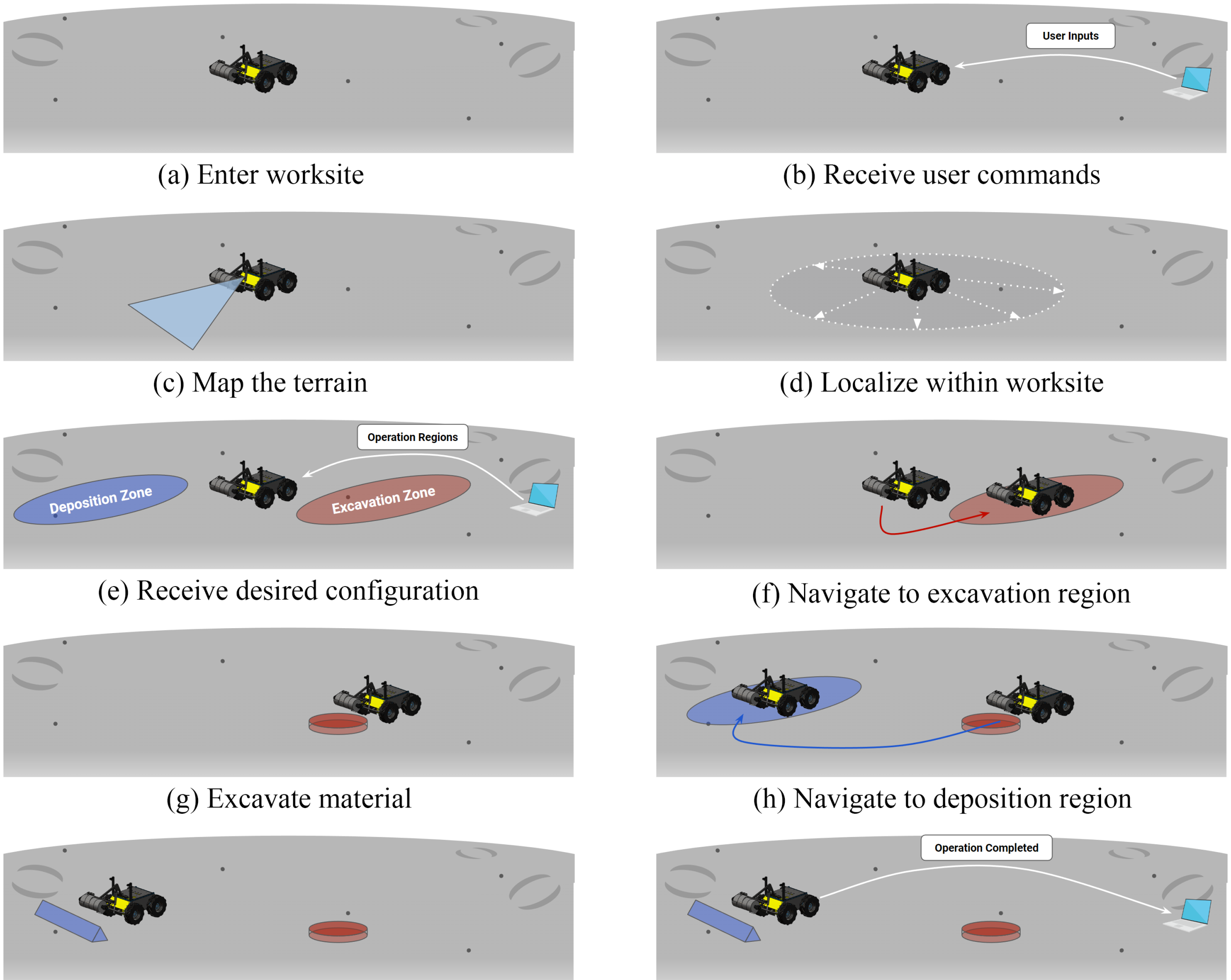
Regolith would be continually required in vast supplies to actualize the abovementioned applications. Having humans excavate and transport these amounts would be perilous, inefficient, and uneconomical. An autonomous solution is ideal, considering the strenuous, mundane, and repetitive nature of the job. A robotic system can be designed to conduct these operations for extended durations efficiently.

Before we build complex structures and habitats on the moon, we would need to shield these structures from the dust kicked-up due to the exhaust/plume of the rockets during takeoff and landing at the lunar launchpads. The Lunar Autonomous Regolith Excavator (LunAR-X) is a robotic system developed to autonomously excavate and build protective raised barriers, or berms, around launchpads. These berms would protect human infrastructure from the damages that can be caused during rocket operation, as mentioned above. The system demonstrates construction capabilities that are representative of the unprecedented class of volumetric site work required for future lunar establishments. Furthermore, these berms can also serve as versatile groundwork for further construction projects by acting as borders and layouts for expansive lunar settlements.

2. Use Case

The Con-Ops for the LunAR-X system is visualized in Figure 1. The process begins when the system arrives at a relatively flat worksite (Fig. 1a) and receives a user command to start autonomous operations (Fig. 1b). The system then constructs a map of the worksite terrain and topography (Fig. 1c), using this map to localize itself within the worksite (Fig. 1d). Once the mapping is complete, the system reports the map to the user and receive the desired worksite configuration. This configuration describes the shape of the berm that the user wants to build, as well as the operating regions for excavation and dumping (Fig. 1e). With this information, the system can plan the required operations and begin executing them. The first step in the excavation process is for the system to navigate to the source region (Fig. 1f), where it begins excavating material (Fig. 1g). This material is transported to the dump/build region (Fig. 1h), where it is used to construct the berm (Fig. 1i). This cycle of excavation, transportation, and dumping is repeated until the desired berm configuration is achieved.

The system is equipped to assess the progress and re-plan actions if necessary. This is done



(a) Enter worksite
(b) Receive user commands
(c) Map the terrain
(d) Localize within worksite
(e) Receive desired configuration
(f) Navigate to excavation region
(g) Excavate material
(h) Navigate to deposition region
(i) Build a berm
(j) Generate evaluation metrics

Figure 1: Concept of Operations

using sensor feedback and information about the status of the operation, allowing the system to adapt to changing conditions and continue working efficiently.

The terrain/worksite map is also updated frequently to assist with navigation and to provide the user with a visualization of the system's progress. This ensures that the system always has accurate and up-to-date information about the worksite, allowing it to operate effectively.

Once the berm has been built, the system generates evaluation metrics and reports them back to the user (Fig. 1j). These metrics provide important information about the performance of the system and the quality of the berm that was constructed.

3. System Requirements

The requirements of the system are first categorized into mandatory and desirable, with each further classified as either performance or non-functional. These requirements are tabulated in Table 1, 2, 3, and 4. There have been no modifications to the requirements since the conceptual design review.

3.1. Mandatory Requirements

Table 1: Mandatory Performance Requirements

ID	Requirement	Performance Metric
M.P.1	Receive commands from the user	The system will receive **2** types of commands: start/stop & desired configuration
M.P.2	Map the terrain	The system will update the worksite map at a frequency of **≥ 0.1 Hz**
M.P.3		The system will output **≥ 2** data products of worksite map pre and post operation
M.P.4	Localize within the site	The system will localize itself within a positional accuracy of **≤ 30 cm**
M.P.5	Operate autonomously	The system will operate autonomously with **0** human intervention after the operation starts
M.P.6	Traverse site terrain	The system will be capable of traversing a site terrain with **≤ 10 deg** incline
M.P.7	Excavate material	The system will be capable to excavate **≥ 3 kg** material per cycle
M.P.8	Build berm	The system will build a berm that has minimum dimensions of **15 cm** in height and **30 cm** in length
M.P.9		The system will build a berm with an error tolerance within **±3 cm** in height and **±5 cm** in length

Table 2: Mandatory Non-Functional Requirements

ID	Requirement	Description
M.N.1	Appropriate Size	The system will have a size appropriate to the test site
M.N.2	Maintain Traction	The system will be capable of maintaining traction during operation
M.N.3	Environmental Robustness	The system will be able to operate in a dusty environment
M.N.4	Safety Features	The system will operate safely (minimize human hazard)
M.N.5	Evaluate Performance	The system will be capable of evaluating the similarity between the desired and the constructed berm

3.2. Desirable Requirements

Table 3: Desirable Performance Requirements

ID	Requirement	Performance Metric
D.P.1	Receive commands from the user	The system will receive **2** types of commands: start/stop & desired configuration
D.P.2	Map the terrain	The system will update the worksite map at a frequency of ≥ **1.0 Hz**
D.P.3		The system will output ≥ **10** data products of worksite map pre, post and throughout operation
D.P.4	Localize within the site	The system will localize itself within a positional accuracy of ≤ **10 cm**
D.P.5	Operate autonomously	The system will operate autonomously with **0** human intervention after the operation starts
D.P.6	Traverse site terrain	The system will be capable of traversing a site terrain with ≤ **20 deg** incline
D.P.7	Excavate material	The system will be capable to excavate ≥ **5 kg** material per cycle
D.P.8	Build berm	The system will build a berm that has minimum dimensions of **20 cm** in height and **100 cm** in length
D.P.9		The system will build a berm with an error tolerance within ±**1.5 cm** in height and ±**2.5** cm in length

Table 4: Desirable Non-Functional Requirements

ID	Requirement	Description
D.N.1	Weight	The system will have mass < **100 kg**
D.N.2	Traverse Worksite Terrain	The system will maintain traversability of terrain throughout operation
D.N.3	Technological Extensibility	The system will be well documented and designed so that future teams can easily access and build on the work

4. Functional Architecture

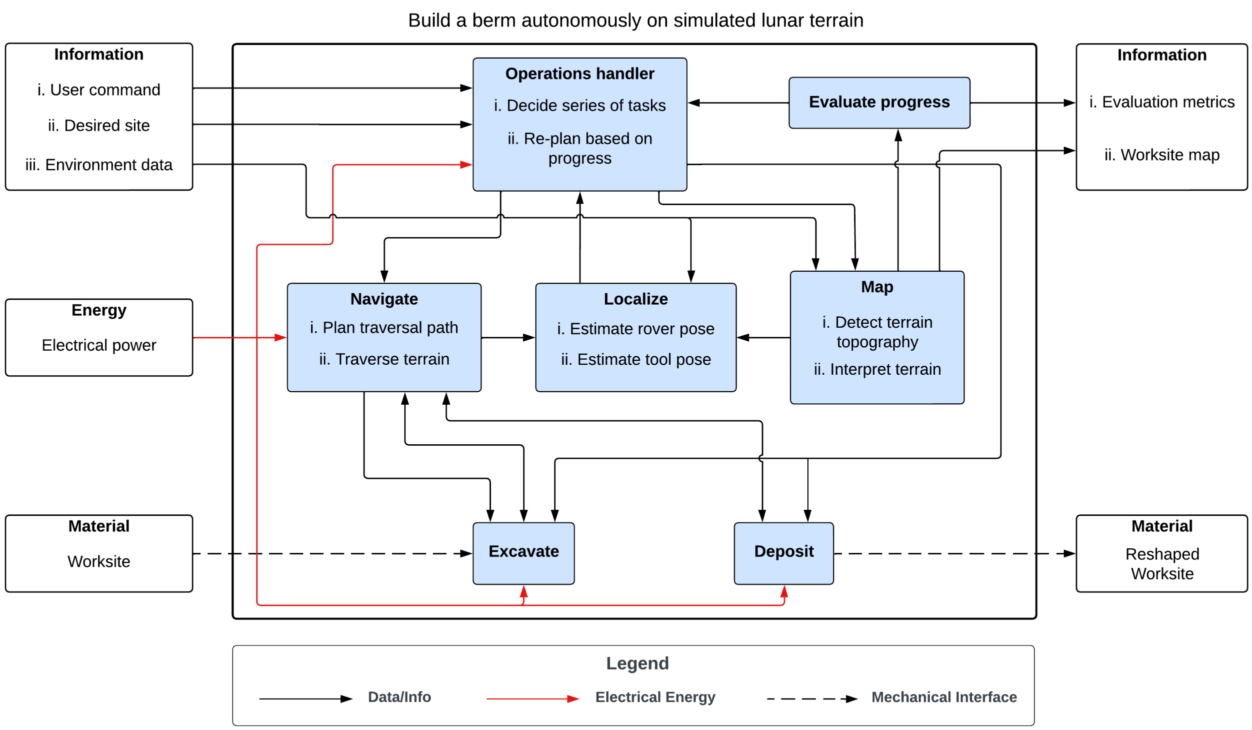


Figure 2: Functional Architecture

Figure 2 shows the functional architecture of the system. The system takes information input in the form of user commands, desired site configuration, and environment data. These inputs specify the source region, the build region, and the shape of the berm to be built. It also has electrical power as the energy input and the worksite as the material input. The worksite constraints for navigation and traversal are specified in the performance requirements (M.P.6).

The system outputs are the reshaped worksite, evaluation metrics, and the worksite map. The reshaped worksite will be evaluated against the desired site configuration to obtain the evaluation metrics. Tolerances for various parameters for the constructed berm have been specified in the performance requirements (M.P.8) and (M.P.9). Per-cycle constraints have been specified as well (M.P.7).

Table 5: Mapped Performance Requirements for Functional Blocks

ID	Block Description	Mapped Requirements
F.1	Operations Handler	M.P.1, M.P.5
F.2	Evaluate Progress	M.P.5, M.P.8, M.P.9
F.3	Navigate	M.P.6
F.4	Localize	M.P.4
F.5	Map	M.P.2, M.P.3
F.6	Excavate	M.P.7
F.7	Dump	M.P.8, M.P.9

Table 5 shows the description of each functional block and the corresponding mapped requirement. Based on user inputs, the operations handler plans the high-level tasks of the system necessary to achieve the objective autonomously. The progress evaluation element is responsible for evaluating the system's percentage completion of the task and whether the berm shape lies within the tolerances specified in performance metrics (M.P.8) and (M.P.9). The operations handler is responsible for transitioning the system between navigation, excavation, and dump modes. Each of these functional blocks has corresponding mapped requirements for terrain traversal, amount of material excavated, and shape of the constructed berm. The mapping element is responsible for satisfying performance requirements on the frequency and number of generated worksite maps reported to the user (M.P.2) and (M.P.3).

5. Cyberphysical Architecture

5.1. Mechanical subsystem

The essential components of the mechanical subsystem are the excavation tool and the lifting mechanism. These, along with the sensors and the reflector prism, have mechanical interfaces with the chassis of the robot. 3D CAD applications were utilized to design, analyze and document the various elements and linkages of the subsystem.

5.2. Actuation and Electronics

Motors, actuators, and their respective controllers are required to drive the mobile platform and operate the excavation tool. These constitute the actuation and electronics subsystem. The actuators include feedback capabilities to measure the state of the wheels and the tool to allow for adequate control.

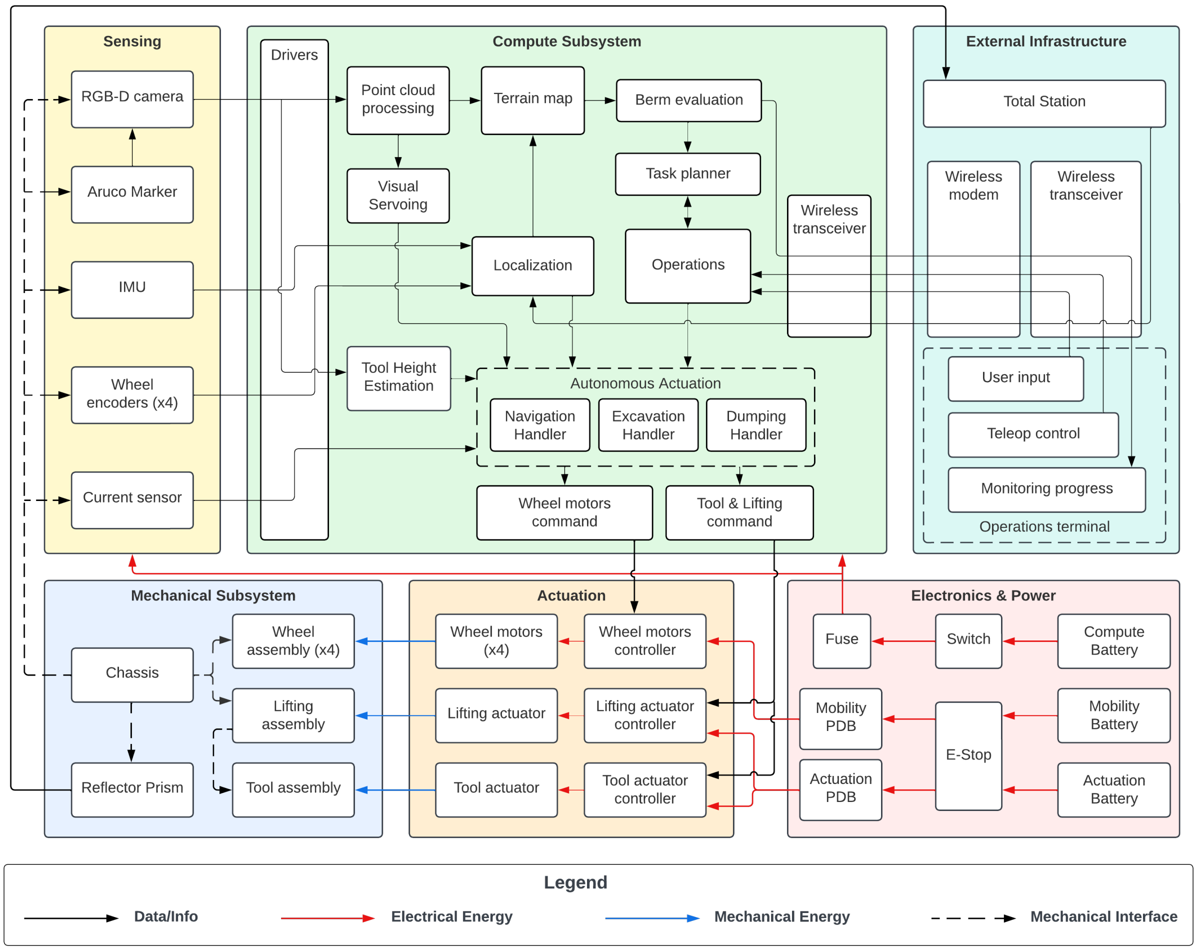


Figure 3: Cyberphysical Architecture

5.3. Electrical Power

The system is powered using a three power-source setup. The electrical power input is fed into the emergency stop buttons and the primary fuse as the initial steps toward safe operation. The following circuits appropriately distribute the power to all the components which might require it. Wireless/software emergency stops are additional means to ensure the safety of the users and the system. The power for the compute has been set up in the way that the batteries can be replaced without having to restart it.

5.4. Sensing

The sensor suite comprises various sensors required for the autonomous operation of the system. All of the sensors will be fixed on the robot chassis. The system uses an RGBD camera to get a close-up view of the berm under construction and some of the ground area surrounding it. Due to the unavailability of GNSS infrastructure on the Moon, a total station is deployed to aid the system in estimating its current position. The system is also equipped with wheel encoders for the velocity and an IMU for the absolute heading, to be utilized by the localization module. Finally, a current sensor is installed to measure the tool motor current for aid in estimating the amount of material excavated by the tool.

5.5. External Infrastructure

The external infrastructure encompasses all components outside the system required to achieve the operational objectives. This includes the total station deployed for localization, the wireless network for communication, and an operations terminal. The user monitors progress and communicates with and controls the system through this terminal.

5.6. Compute

The compute subsystem constitutes the algorithms and software required for autonomous operation. After appropriate pre-processing, sensor data, along with user commands, are the inputs to this subsystem. To calculate the subsequent dump pose, the planner requires an estimate of the present status of the berm and the system's position relative to the berm. Pointcloud processing is followed by registering the terrain map and then estimating the berm geometry using that map. The localization module fuses data from various sensors to calculate the position and orientation of the system. The operations handler fetches a plan from the task planner and assigns individual tasks to the respective action handlers. It can transition the operation of the system between three major modes: navigation, excavation, and dumping. These three modes plan and communicate directly with low-level actuation controllers to execute the desired behavior.

6. System Description and Evaluation

6.1. System Depiction

The entire system has been depicted in Figure 4. The figure shows the rover at the worksite, with the excavation tool, the lifting mechanism, the electronics enclosure, and the sensing apparatus. All the subsystems of the LunAR-X system are detailed below.

6.2. Rover

The rover subsystem is the mobility platform base on which all the sensors, tooling, and electronics necessary for the autonomous operation of the system are mounted. After a trade study that included critical factors like traversability, traction, and payload capacity, the Clearpath Husky A200 was selected as the mobility platform, shown in Figure 5. The Husky A200 is built for mobility in outdoor & field environments, making it suitable for this application. It is driven by a skid steer drive and has sufficient ground clearance to traverse over terrains well within the constraints set by the system requirements. The range of motion of the platform is restricted to turns of high radius to maintain traversability of the worksite, power efficiency, and avoid sensor noise due to jittery motion. The platform is equipped with high-torque motors and wheel encoders.

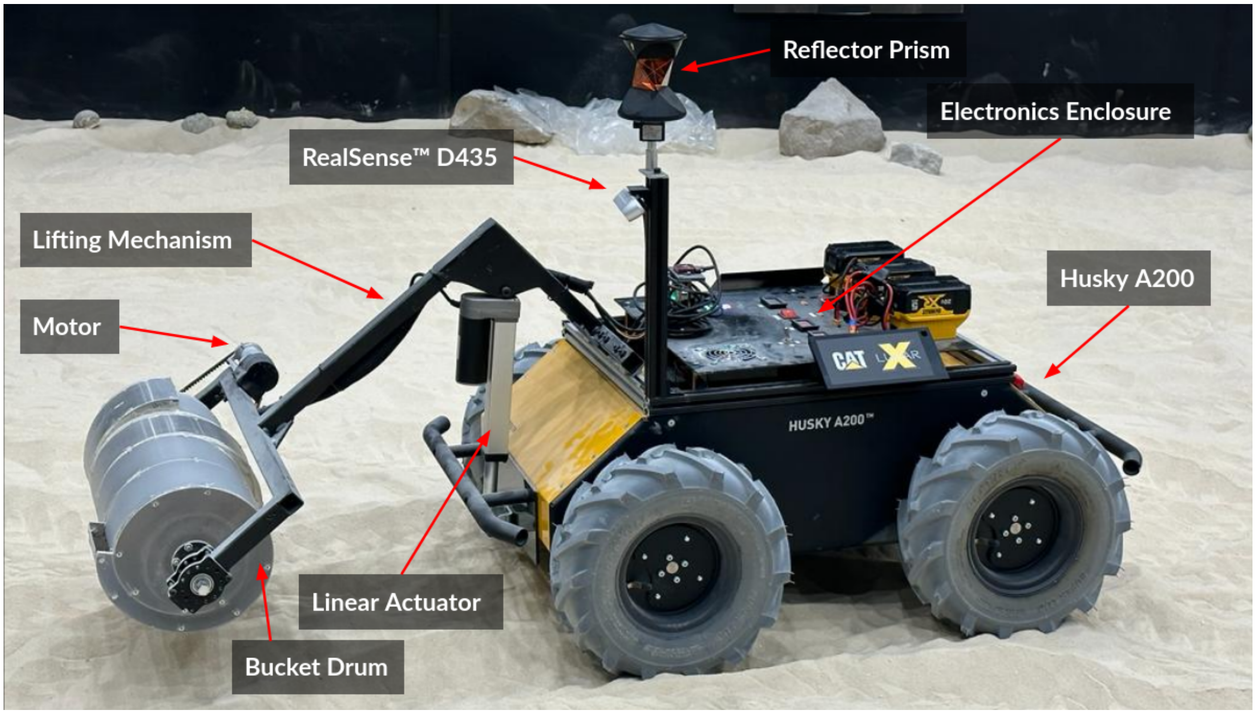


Figure 4: The LunAR-X System

Figure 5: Clearpath Husky A200

6.3. Tool

The tool is crucial for the system, consisting of two main components: the bucket drum and the lifting arm. The bucket drum is responsible for regolith-related operations such as excavation and deposition. Its unique design excavates material when rotated in one direction and deposits material when rotated in the opposite direction. The bucket drum is made up of four separate sections, each with a cutting edge phased out at 90 degrees to reduce the cutting force acting on the tool at any given moment. The bucket drum is driven by a belt and a 135 kg-cm, 60 RPM motor. The effective gear ratio of 2.25:1 increases the operational torque, providing an additional factor of safety. The labeled CAD design of the bucket drum is shown in Figure 6. The assembly has the capability to have an additional motor and belt on the other face of the drum as a potential for increasing the excavation power.

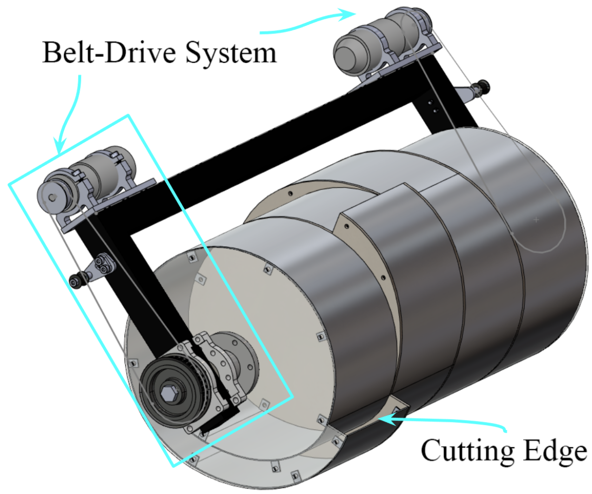


Figure 6: Bucket Drum Assembly

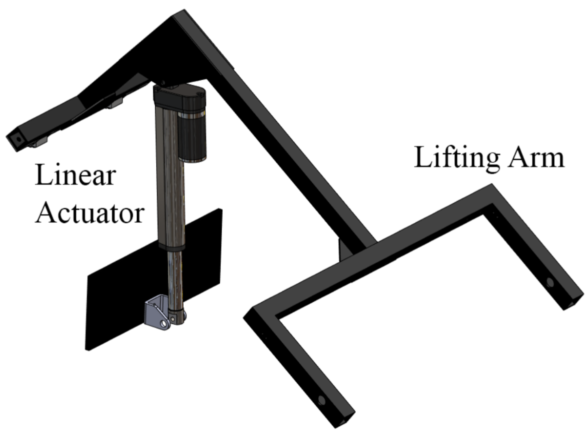


Figure 7: Lifting Arm Mechanism Assembly

The lifting arm is responsible for raising and lowering the bucket drum. It has been built to lift the bucket drum to 45 cm above the ground level and lower it to as low as 5 cm below ground level. This range of motion ensures the system can build a berm bigger than the minimum requirements while also being able to excavate in low-lying areas. The lifting mechanism is actuated using a linear actuator. The labeled CAD design of the bucket drum is shown in Figure 7.

6.4. Electronics

The electronics architecture that powers the system is visualized in Figure 8. There are three sources for three different power circuits. These circuits are for powering the computer, the actuation, and the mobility separately. This separation ensures that the emergency button only cuts off the power to the mobility and actuation, leaving the compute and sensing subsystem unaffected.

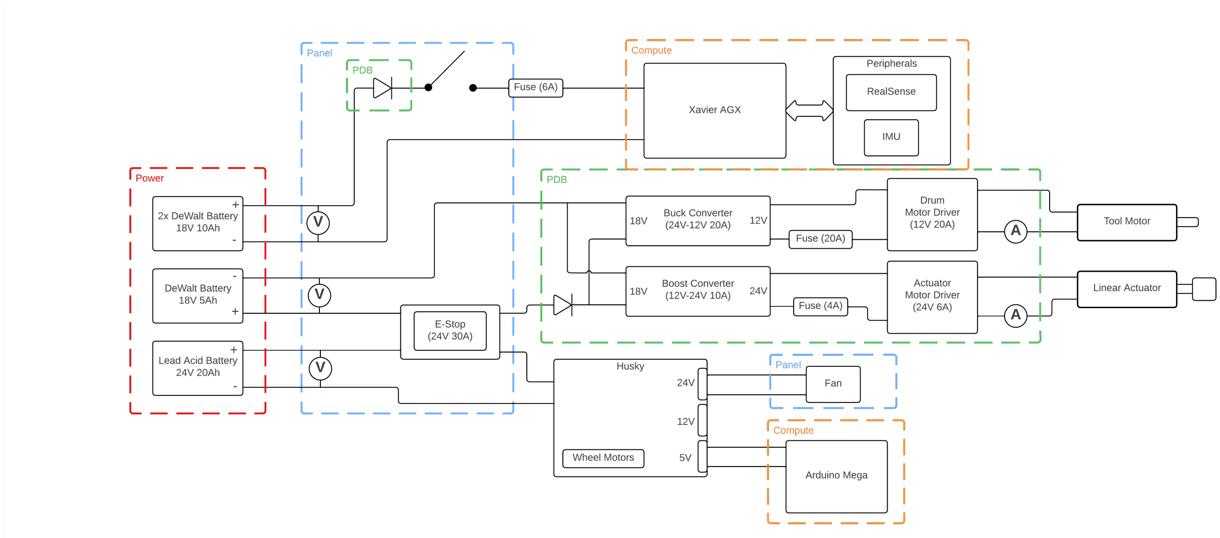


Figure 8: Electronics Architecture

To maintain the safety of the system, there have been fuses installed wherever required. Voltmeters help in keeping the battery voltages in check during operation. There are two voltage converters to provide the appropriate voltage levels to the actuators. These actuators are controlled using the respective motor drivers.

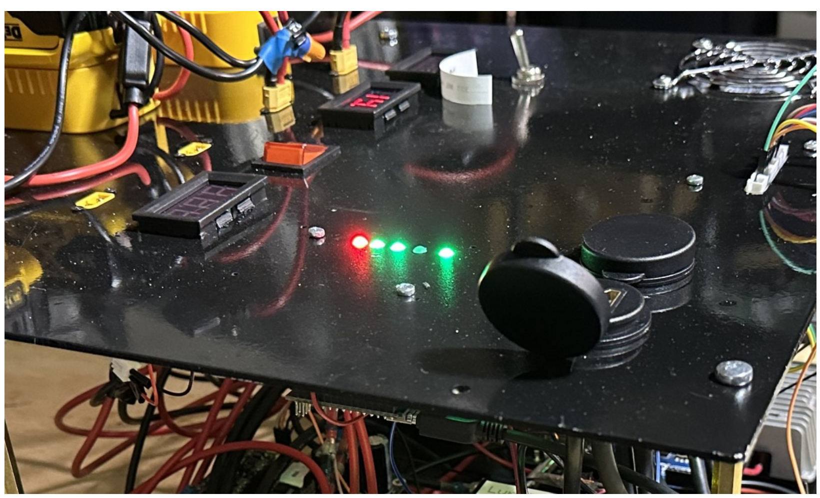


Figure 9: Status LEDs

To help monitor the system conveniently, there are five status LEDs installed on the electronics panel of the system, as seen in Figure 9. The function of each of the LEDs is mentioned in Table 6.

6.5. Software

The software architecture onboard the system is divided into multiple modules. The majority of the components are implemented using the ROS 2 framework, except the lower-level control, which currently has better support in ROS 1. Docker containers are also used to maintain the

Table 6: Status LEDs

ID	Name	Function
PWR	Power	Display if the system is powered on
HBT	Heartbeat	Display if the software pipeline is online
OPR	Operation	Display the current active operation mode
TSK	Task	Display the current active task mode
LCK	Lock	Display the system's lock status

modularity of the software while ensuring it can be developed on different machine architectures easily. All sensor data is subject to pre-processing. This reduces noise in the data.

The Nvidia Xavier AGX, which is the main compute onboard the system, executes two docker containers. The first is the autonomy container, which consists of all software packages responsible for the autonomous operation of the system. The autonomy software elements along with their interfaces are visualized in Figure 10.

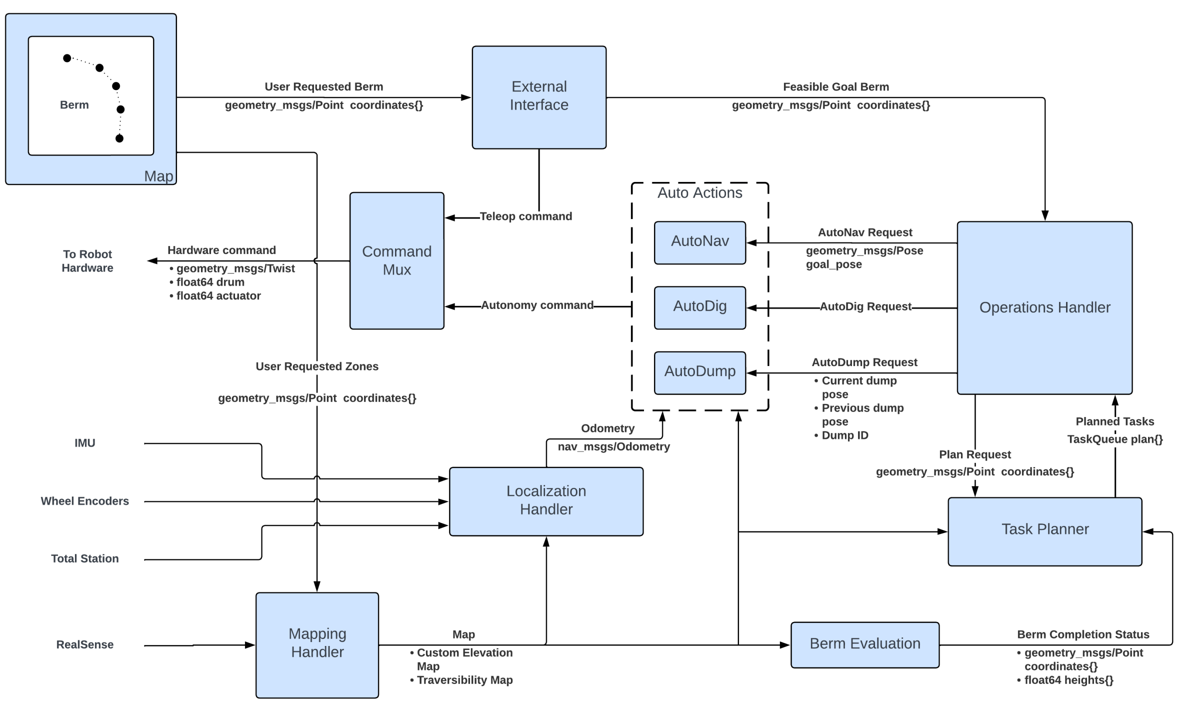


Figure 10: Software & Interfaces

The pipeline begins with the user marking the desired berm configuration, restricted zones and excavation zones on the control station. These point coordinates are processed and then passed on to the rover. The 'External Interface' node, which is responsible for all communication with the operator, checks these desired berm coordinates for feasibility. This node also enforces security locks in case of communication loss. The feasible berm coordinates are then provided to the 'Operations Handler' node to start autonomous construction of the berm. A plan is fetched from the 'Task Planner' node, which is used by the operations handler to respectively call the AutoDig, AutoNav and AutoDump actions that are detailed in subsection 6.8, subsection 6.9, and subsection 6.10.

The 'Command Mux' node selects which command will be used to drive the system at any given moment. This is provided for efficient switching between teleoperation commands from the user and the autonomous commands from the system itself. The 'Mapping Handler' and the 'Localization Handler' nodes keep the terrain map updated while also keeping track of the system's pose within that map. The 'Berm Evaluation' node gives crucial feedback on the progress of the construction to the task planner to dynamically re-plan if required.

The second container is the hardware-interface container, which contains all software packages required to control and communicate with the lower-level hardware. These include the package for controlling the Husky mobility base and processing data from the IMU, RGBD Camera, and tool encoders. This container communicates with the Arduino Mega microcontroller that is employed to control the tool subsystem.

6.6. State Estimation

An accurate estimate of the rover's pose with respect to the berm it is building is required for autonomous operation. The subsystem makes use of the total station, wheel encoders, IMU, and aruco markers to estimate the system's pose within the work site.

6.6.1. Localization

The following data is fused from each of the sensors:

- **Wheel Encoders:** $[\dot{x}, \dot{y}, \dot{\psi}]$
- **IMU:** $[\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]$
- **Total Station:** $[x, y, z]$

As the position feedback from the total station is discontinuous, two different odometry streams are generated. Local odometry is generated by fusing the data from the wheel encoder and the IMU using an Extended Kalman Filter. This fused data is guaranteed to be continuous for a short period of time in the local frame. Odometry for the global frame is generated by using another Extended Kalman Filter that fuses the data from all the sensors to produce a highly accurate pose estimate in the global map frame. As the total station provides extremely accurate but discontinuous measurements, the covariance and the system process matrices are tuned such that the position estimate converges rapidly to the estimate from the total station. This ensures the system achieves reliable localization. We conducted loop closure tests and measured the loop closure accuracy of our localization subsystem to be under 2 cm, hence demonstrating the capability of this subsystem.

6.6.2. Tool Height Estimation

For the purpose of facilitating autonomous digging operations, a critical requirement is the precise determination of the tool's height at any given moment. This was accomplished using the

Intel Realsense D435 camera to calculate the tool's depth using an ArUco marker on the tool arm that would serve as a distinctive visual reference point for the system.

Data were collected at various heights in steps of 5cm and and a 3D plane fitting was performed on the data to estimate the height. The data fitting method was preferred to calculating the camera transform which would require extrinsic calibration. The method provided reasonably good estimates of the height for autonomous operations to be performed.

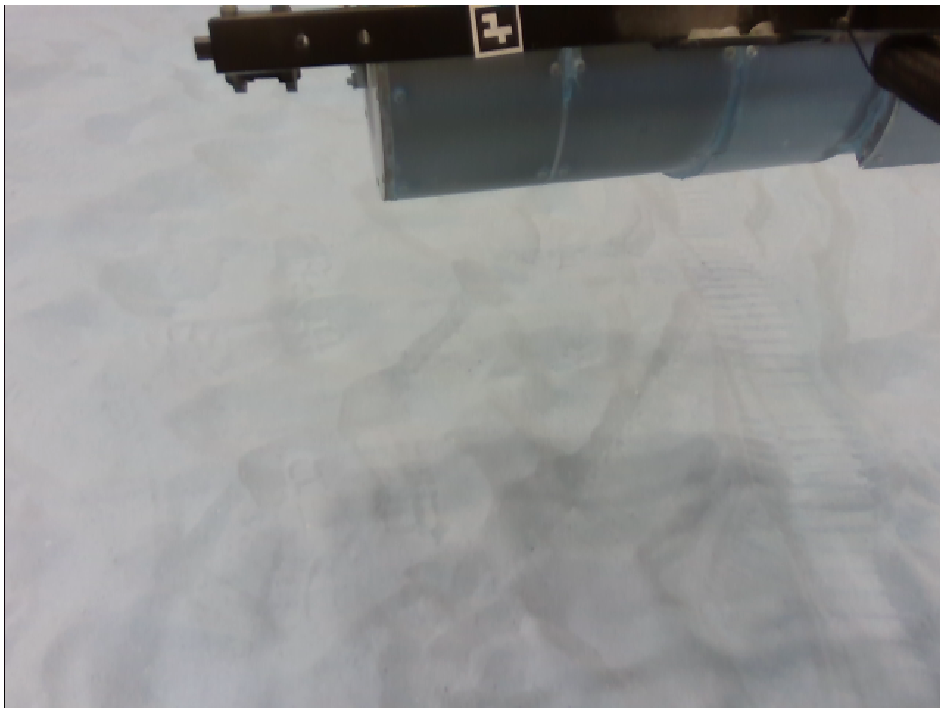


Figure 11: Tool view from the camera with the marker

6.7. Task Planning

The task planning subsystem plays a vital role in optimizing the construction of the berm for efficiency by devising a feasible high-level task plan. Maneuvering through a 3D environment while manipulating terrain offers numerous possibilities for constructing a berm. For instance, in the scenario of a straight berm, the initial layer might follow a straight line, and subsequent layers could be deposited at the same locations, in between sections, or randomly distributed. Additionally, exploring multiple dumping positions within a single excavation cycle adds to the complexity. Planning an optimal sequence of dump operations in this expansive and continuous space poses a computationally challenging task. To simplify the problem, the task planner operates under the following assumptions:

Assumption 1: The desired berm configuration can be segmented in the X-Y dimension, completing the entire berm by building each section.

Assumption 2: The angular change between any two consecutive berm sections will always be less than or equal to 10 degrees.

Assumption 3: Each operation cycle comprises a sequence of excavation, transportation, and deposition, all occurring only once.

In the process of experimentation, the length of each berm section, accommodating angular changes from 0 to 10 degrees, was determined to be approximately 45 cms. Furthermore, the number of dumps required in each section remains constant since the desired height for the entire berm is uniform. This number is calculated by the formula:

$$n = \left(\frac{h_d}{h_0}\right)^2$$

where h_d represents the berm height, and h_0 is the height of the berm section in one material deposition (approximately 8 kgs of material), experimentally measured at 9 cms.

Streamlining the problem to one dimension, the task planner processes feasible excavation sections and computes an optimal sequence for visiting and depositing material in the berm sections. At a high level, this problem mirrors a traveling salesman problem with state-dependent movement costs. Each construction of a berm segment introduces an obstacle to the costmap, altering the cost of visiting all remaining berm segments. Since our transition costs are solely a function of the state, we frame the problem as a bi-level optimization scenario. The top-level planner engages in a graph-based search to uncover a feasible min-cost solution. Simultaneously, at the lower level, a hybrid A* planner calculates the edge cost. Our approach also uses a TSP-based heuristic, which solves a relaxed problem to accelerate the top-level search.

The low-level graph search problem involves the navigation of the robot from the excavation zone to the desired berm section. This is discussed in detail in Section 6.9.

For computing the TSP-based heuristic, our approach leverages the Google OR-Tools TSP Solver, employing an octile distance matrix between each state. The octile distance, not factoring in collisions, provides an underestimation of the material transport cost between the excavation and berm sections. Additionally, as we don't need to return to the start node, in contrast to the general Traveling Salesman Problem, we set the cost to reach the start state as zero from all other states. This effectively neutralizes the impact of reaching the starting node at the end. In the distance matrix, we assign a 0 cost to travel to the same node or go to any excavation zone. When traveling from the start excavation zone to a segment, we choose the minimum octal distance to reach any dump pose for that segment. Finally, for traveling between segments through an excavation zone, we again select the minimum across all potential transitions to consistently underestimate the cost.

6.8. AutoDig

To dig material from the ground while maintaining low ground reaction forces, the tool needs to be lowered to its desired depth of cut and rotated while the robot is moved forward. The depth of cut is visualized in Figure 12. The height of the tool needs to be adjusted dynamically throughout this maneuver to account for uneven terrain and maintain excavation efficiency.

We use two intuitive observations to design a controller for this purpose. The first observation is that if there is more material present inside the drum, it consumes more power and, hence, more current to rotate it. Secondly, if the tool excavates too deep, it will also consume more power (and

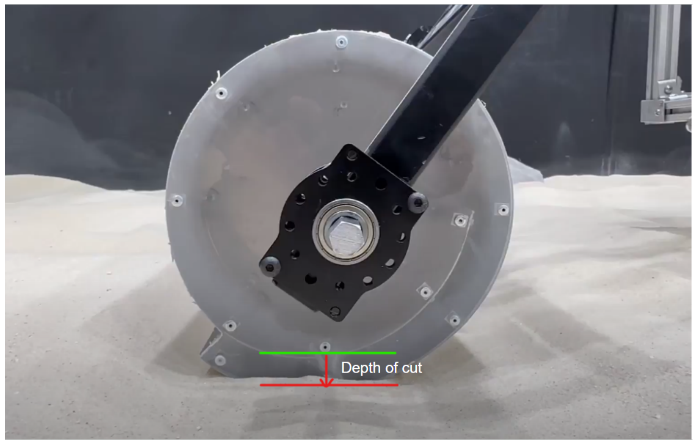


Figure 12: Tool Control for Excavation Depth

therefore current) to excavate. Therefore, tool current feedback is incorporated and used with a cascaded PID controller to command the tool's depth of cut. This helps track a desired tool current profile, as shown in Fig 13.

The outer-loop of this controller predicts the delta dynamics of the system: given an error in the currents, it predicts the change in height required in order to stabilize the current errors. This differential is then added to the current tool height estimate, which is used by the inner-loop PID to command the linear actuator. To reduce the initial overshoot, which could occur if the start tool height is very far from the desired tool height for excavation, absolute ground height measurements from the point cloud were used. This absolute height helped initialize the tool's excavation height at a suitable position. Data from manual excavation operations was used to calibrate the profile of the target current trajectory and tune the outer-loop proportional gain. This methodology helped achieve robust operation across a variety of uneven terrains.

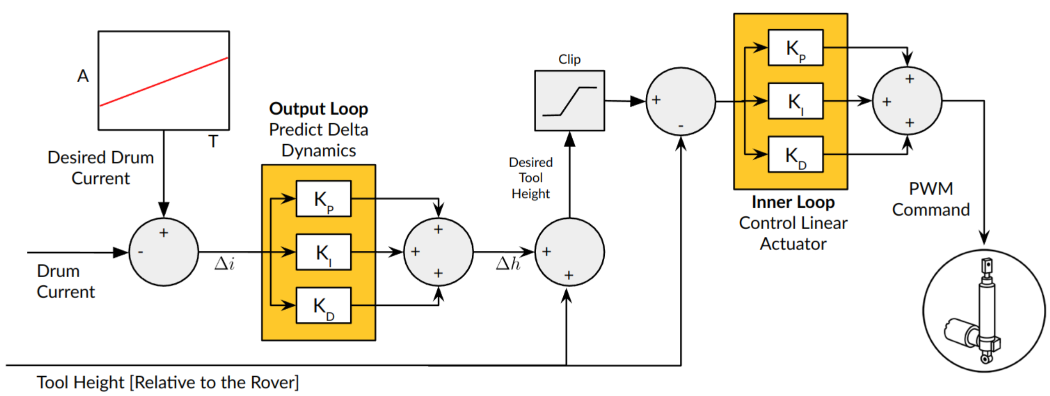


Figure 13: Block Diagram for Autodig Control

6.9. AutoNav

The Autonomous Navigation (AutoNav) subsystem facilitates the robot's movement to efficiently transport excavated material from the excavation region to the designated berm section. The primary challenge involves navigating through a high-resistance and deformable sandy terrain. Although the Clearpath Husky is a differential drive robot that can take point turns, i.e., rotate in place, we avoid this to prevent creating grooves in the terrain, rendering it untraversable. Therefore, the AutoNav subsystem generates a smooth path, incorporating a minimum turning radius constraint of 2.5 meters.

The AutoNav leverages the ROS 2 Nav2 Smac Hybrid A* path planner to generate a smooth trajectory, utilizing Reeds-Shepp motion primitives. This choice not only ensures a smooth trajectory but also introduces the valuable capability of reverse movement, a feature absent in Dubin's path. To track the generated path with precision, AutoNav employs the Regulated Pure Pursuit controller from the Nav2 package.

However, the controllers in the Nav2 package were initially designed for flat terrains and may not be optimized for sandy landscapes. Given the deformable nature of the terrain and potential localization errors, achieving perfect tracking, especially during turns, becomes challenging. In response to these real-world complexities, AutoNav takes a proactive approach by dynamically generating a new path plan every 5 seconds. This adaptive strategy allows the controller to automatically commence tracking the freshly generated path, mitigating issues associated with deformable terrains.

6.10. AutoDump

The primary objective of the AutoDump subsystem is to correct the navigation and localization errors and deposit material in the right position in order to minimize construction errors.

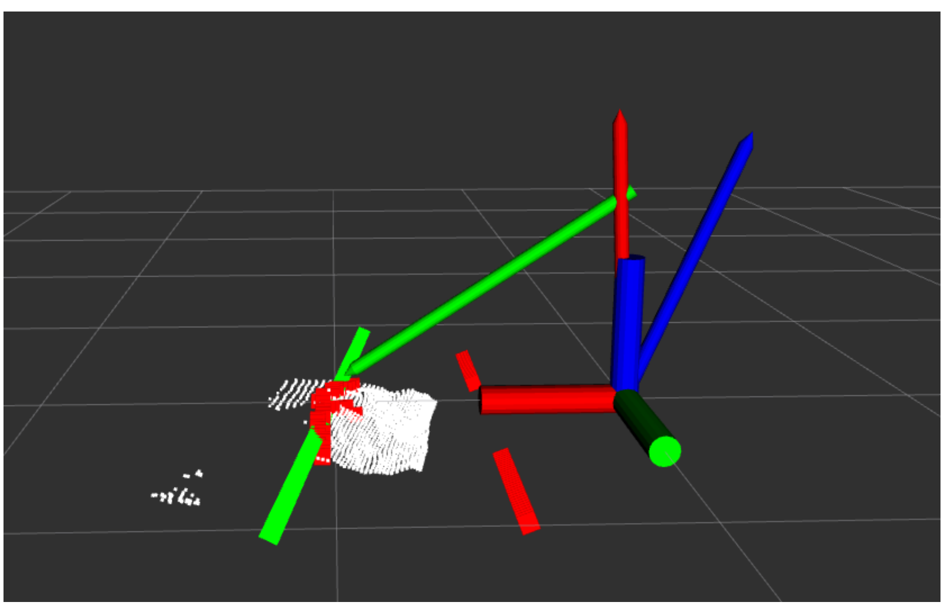


Figure 14: Visualization of Visual Servoing to deposit material

The first step in this process is to detect the previously built berm in the region near the present dump location. A median-based clustering method was implemented to segment the berm in the point cloud, followed by RANSAC to fit the peak line of the visible berm. After detecting the peak line, the segment ID of the detection in the point cloud was identified by finding the berm segment that has the centroid closest to the detected peakline. Using this information, the target deposition point is recalculated using the input berm geometry, as shown in Fig 14. This point is then used to calculate the errors in the distance X, yaw angle ϕ, and tool height. A complementary filter is added to the error estimates to ensure robustness single-frame detection failures.

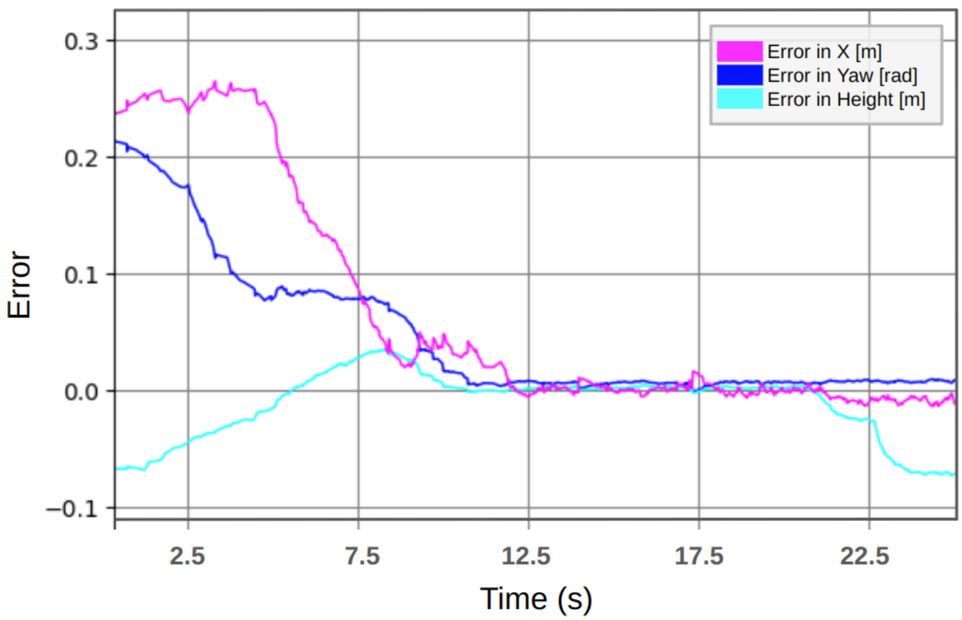


Figure 15: Visual Servoing Errors during AutoDump. Three parallel PID controllers are used in the first phase to mitigate the errors; the material is then dumped before the tool is taken to the highest position for clearance

Three parallel PIDs are used to mitigate the errors generated by visual servoing. The rover is moved to the correct position and tool height for dumping material, as seen in Fig 15. Finally, the rover deposits material at this position by rotating the drum until it is empty and raises the tool back to its highest point, marking the end of one AutoDump cycle.

6.11. Perception

The goal of the perception subsystem is to derive an understanding of the environment and provide the required information to the other subsystems. These include tool height estimation, local ground elevation estimation, visual servoing for AutoDump, berm evaluation, and mapping for traversal costmap & visualization. The perception functions can be divided on the basis of whether they were processed in the robot frame or the global map frame. The perception subsystem takes as inputs the pointcloud stream from Intel RealSense and the pose data generated by the localization subsystem.

Functions performed in the robot frame include tool height estimation, ground elevation estimation and visual servoing for autodump. Ground elevation estimation is performed by cropping the part of the pointcloud which is right under the tool and reporting its average elevation with respect to the rover. This is used by the AutoDig subsystem while initially actuating the tool to touch the ground and start excavation, and the visual servoing algorithm to estimate the target height as described in subsection 6.10

Berm evaluation and traversal costmap generation are described in subsubsection 6.11.2.

6.11.1. Pointcloud Processing

The decimation filter provided by RealSense ROS libraries was used to first sub-sample the point cloud. Given the pointcloud stream in the camera frame, the pointcloud must be processed to the robot base link frame and filtered to be useful for downstream perception algorithms.

To achieve this, we first get the static transform from the camera frame to the robot base link frame and apply eigen transformations in the fixed frame to the pointcloud data. Next, we subscribe to the tool height and appropriately crop out the points corresponding to the tool. This is important since the visible tool would hamper performance in the downstream perception tasks, none of which require the tool points. Finally, we perform noise removal by outlier detection and publish the resultant pointcloud in the robot base link frame.

6.11.2. Mapping

The mapping subsystem uses various inputs like the transformed pointcloud in the robot base link frame, the robot's pose from localization, user-defined berm locations, and predefined restricted zone coordinates. Its goal is to generate a global elevation map, track berm progress, and create a traversability costmap.

The map frame is centered at the moonyard's center, aligned with Total Station axes, and has a resolution of 1.5cm, resulting in an 800x800 grid where each cell holds the average elevation in that area. To build the elevation map, the processed pointcloud is transformed to fit the map frame, and then divided into map resolution bins. Each cell in the map is treated as a Bayes filter unit to accommodate Total Station prism and camera sensor mast vibrations.

For the AutoNav subsystem's traversability costmap, we create another grid within the map frame, initializing all cells as 0 (denoting traversable terrain). The process involves setting the moonyard walls to a cost of 100 to indicate them as obstacles. Then, areas defined by the restricted zones receive a cost of 100 by identifying intersecting cells within the polygon specified by the restricted zone coordinates. Finally, the desired berm regions are marked as cost 100 to ensure the rover avoids disrupting its own construction.

6.11.3. Berm Evaluation

To plan and re-plan autonomous construction operations, accurate measurement of the progress is required. This will allow for the planning of the subsequent excavation and build-poses. The berm evaluation unit subscribes to the map and interfaces with user inputs for desired berm configuration and pre-defined restricted zones.

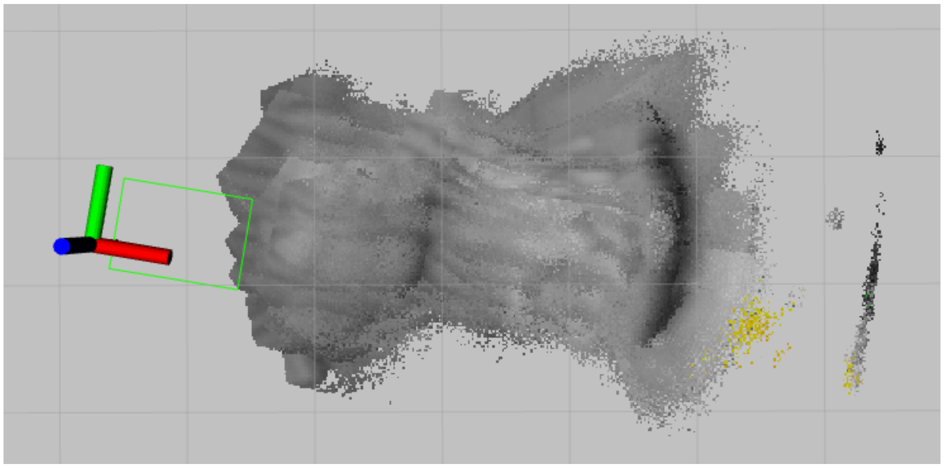


Figure 16: Map during the construction of curved berm. The lighter pixels are lower elevation and the darker pixels are higher elevation.

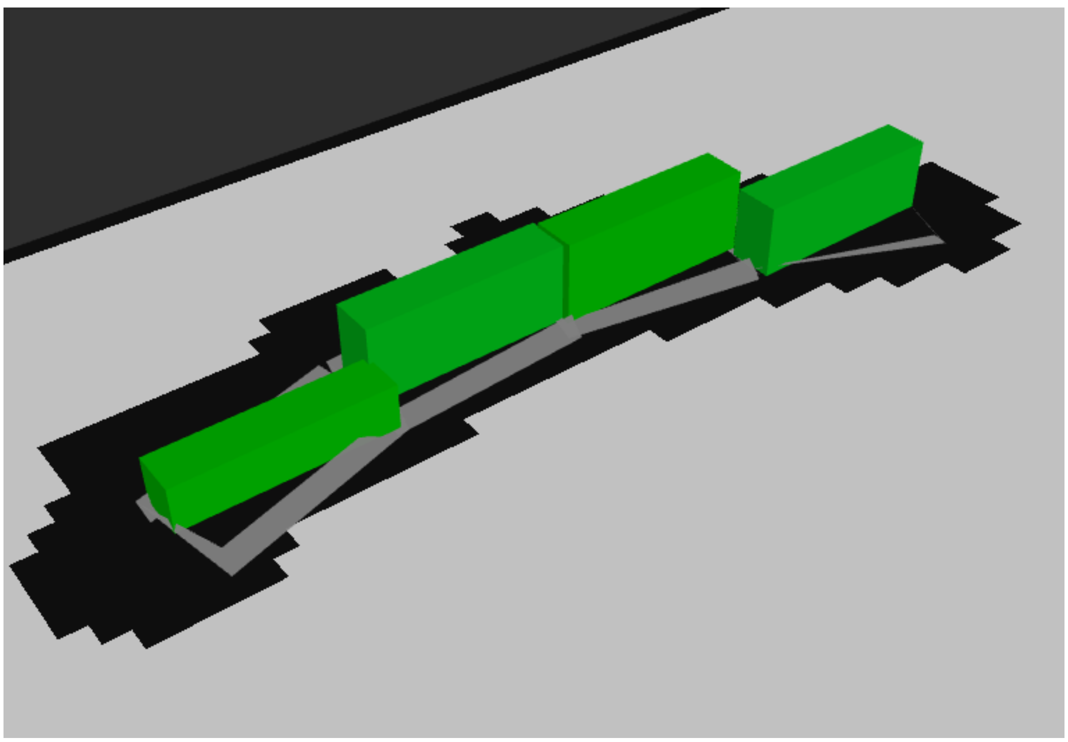


Figure 17: Berm Evaluation. The height of each green box represents the evaluated height of the respective berm section.

To evaluate the berm, the height, length, volume and positional error are calculated. Given the desired berm points, the berm is segmented into sections of length equal to the drum length. For each such section in the global elevation map, the 95th percentile height is reported as the height in each section, and the average of berm section heights is reported as the height of the berm. To calculate the length of the berm, the berm region is binned and peakpoints are selected as maxima in each column perpendicular to the berm section. The ground elevation is estimated by the average elevation map value in the region between 40 to 50cms of the neighborhood of the berm region. The number of columns with minimum 70% of the berm section height are counted as peakpoints and its product with the hypotenuse of patch size is reported as the berm height.

Height of each berm section = 95th percentile height of the section

Height of berm = $\frac{1}{N}\sum_{i=1}^{N} h_i$

Length of berm = Number of peakpoints $\times\sqrt{2}$ patch size

The volume of the berm is estimated by summing up the elevations in 30cm neighborhood of the desired berm peakline and marking it with the timestamp. The error in construction is computed by taking root mean square (RMSE) error of the current berm peakline obtained through binning to the desired peakline.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{current\_peakline}_i - \text{desired\_peakline}_i)^2}$$

6.12. External Infrastructure

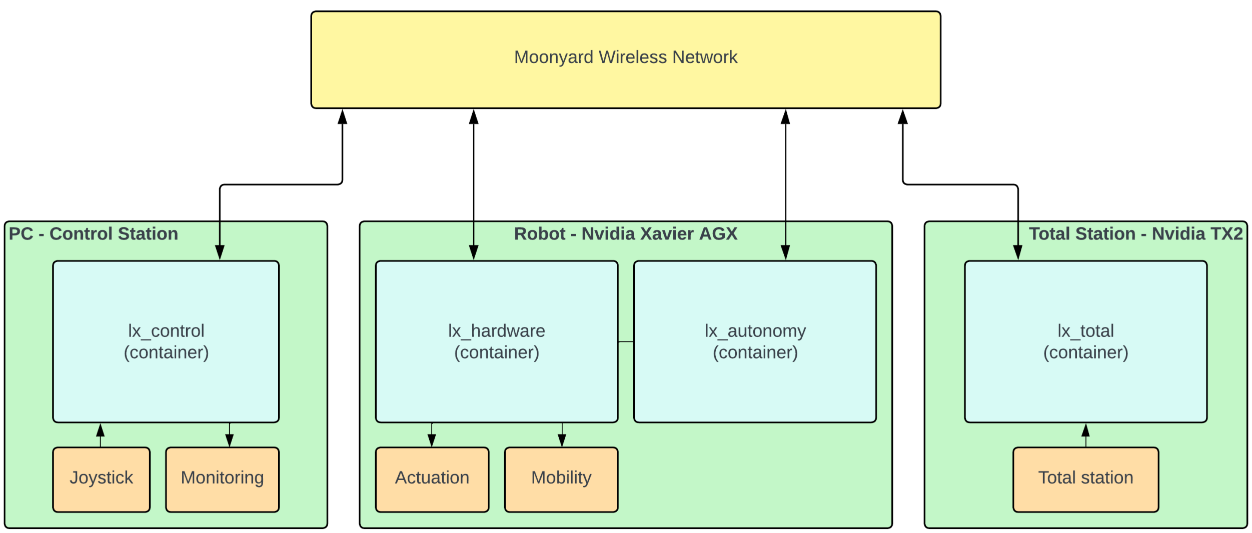


Figure 18: Wireless Network Layout

The external infrastructure is the combination of elements that are not onboard the system but are necessary for operation. The infrastructure elements have been shown in Figure 19. It includes the control station, joystick for teleoperation, interfaces for wireless communication, and total station for localization. The wireless network over which the robot, control station, and the total station communicate has been visualized in Figure 18.

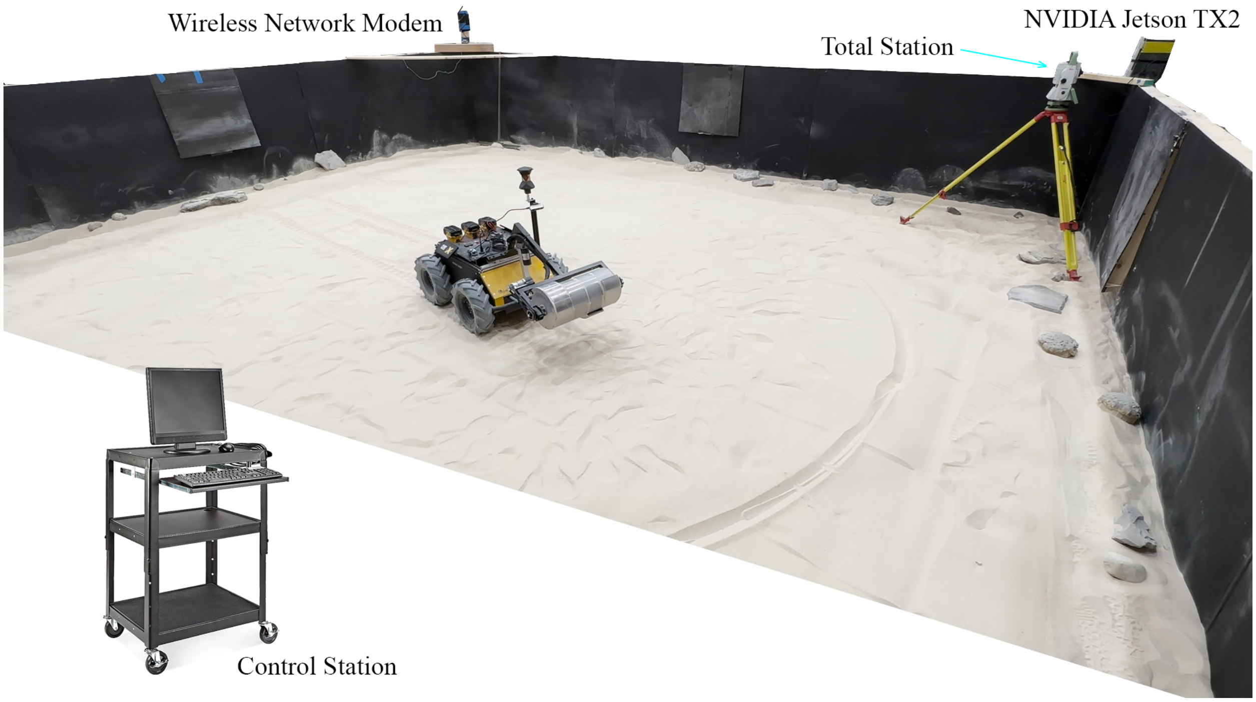


Figure 19: External Infrastructure Layout

6.13. System Performance

The Fall Validation Demonstration (FVD) and the FVD Encore were held on the 20th & 29th of November at the Moonyard (Figure 20), a sandbox commonly used to test space robotic systems, situated in the Planetary Robotics Lab at CMU. In the demonstration, the system was tested to evaluate the performance. Table 7 details the achieved performance of the system as compared to the system requirements. The system successfully met all of the mandatory performance and non-functional requirements.

To evaluate the system, it was given a desired berm configuration that was 15 cm high and 75 cm long. The rover was then left to operate autonomously in-order to construct the desired berm. There was no human intervention required throughout the autonomous construction. The system even met some of the desirable performance requirements, which are mentioned in Table 8.

6.14. System Evaluation

Throughout the development cycle, the various design decisions taken and issues faced led to multiple characteristics and traits possessed by the system. These can be classified into strengths and weaknesses depending on how they affect the system and its operation.

6.14.1. Strengths

A significant strength of the system is the excavation mechanism. It is built to be durable and robust. The entirety of the mechanical subsystem has survived multiple unplanned collisions without damaging itself. This allowed for testing of the autonomy of the system with lesser constraints

Table 7: FVD Performance - Mandatory Requirements

ID	Promised Metric	Achieved Performance
M.P.1	Operate with **2** types of user command	Operated with **2** types of user command
M.P.2	Update worksite map at ≥ **0.1 Hz**	Updated worksite map at **1.0 Hz**
M.P.3	Output ≥ **2** data products of worksite map	Output **2** data products of worksite map
M.P.4	Localize with a positional accuracy of ≤ **30 cm**	Localized with a positional accuracy of ≤ **2 cm**
M.P.5	Operate autonomously with **0** manual intervention	Operated autonomously with **0** manual intervention
M.P.6	Traverse a site terrain with ≤ **10 deg** incline	Traversed a site terrain with an incline up to **15 deg**
M.P.7	Excavate ≥ **3 kg** material per cycle	Excavated **8 kg** material per cycle
M.P.8	Build a berm that has minimum dimensions of **15 cm** in height and **30 cm** in length	Built a berm that has dimensions of **15 cm** in height and **76 cm** in length
M.P.9	Build a berm with error tolerance of ± **3 cm** in height and ± **5 cm** in length	Built a berm with an error of **2.1 cm** in height and **1.3 cm** in length

Table 8: FVD Performance - Desirable Requirements

ID	Promised Metric	Achieved Performance
D.P.2	Update worksite map at ≥ **1.0 Hz**	Updated worksite map at **1.0 Hz**
D.P.4	Localize with a positional accuracy of ≤ **10 cm**	Localized with a positional accuracy of ≤ **2 cm**
D.P.7	Excavate ≥ **5 kg** material per cycle	Excavated **8 kg** material per cycle

and risks.

Secondly, due to the electronic architecture of the system, the batteries for the onboard computer can be swapped without switching it off or hindering the operation of the software. This adds a system capability to continue constructing bigger berms without having to reset and restart the software.

Another strength of the system is that hardware and software architectures are highly modular. This results in all the components of both subsystems being easier to debug, maintain and replace if necessary.

Lastly, the husky mobility platform is powerful and robust, which helped us avoid multiple potential issues while testing the system.

6.14.2. Weaknesses

Due to budget and time constraints, the excavation subsystem was built with limited degrees of freedom. The system lacks the ability to align with the uneven ground while excavating. This results in slightly longer excavation times and some sections of the drum having more excavated material than others.

The system is also currently missing direct weight feedback for the amount of material excavated. A weight feedback could have added another layer of intelligence to the autonomous decision-making of the robot.

The weight of the system, along with the wheel design, limits the maneuverability of the system on the worksite. Single-point turns with the skid-steer mechanism result in the system losing traction and displacing large amounts of material from underneath the wheels.

The last weak point of the system is that, currently, the computational power of the hardware is operating at the limits. This limitation also restricted the addition of a second RGBD camera, which in-turn limits the view of more area of the worksite from the perspective of the robot. For any future development or more complex additions to the autonomy, the system would either require optimization or replacement with a more powerful platform.

7. Project Management

7.1. Schedule

To understand the schedule of the development of the robot, the timeline can be broken into 2 phases. The first phase involved electro-mechanical development, while the autonomy development took place in the second phase. Although both the development phases mostly stayed on schedule, they each had a slight delay due certain dependencies.

In the first phase, the integration of the hardware subsystems largely had a dependency on the delivery of the Husky A200 mobile platform. The expected lead time was 1.5 months whereas the actual time for the platform to be delivered took nearly 3 months. This external dependency caused some unexpected delays. The effect of the delay was controlled due to assistance from the Field Robotics Center at CMU, where a similar mobile platform could be arranged for some days of integration tests while awaiting delivery.

In the second phase, the development of navigation and mapping subsystems depended on the localization subsystem. Localization faced multiple issues throughout the phase, with challenges ranging from miscalculated transforms to outdated device drivers.

Despite the above-mentioned delays, the project and the robot capabilities were successfully able to meet the final milestone of the Fall Validation Demonstration.

7.2. Budget

Over the course of the development, \$4,218 have been utilized, which accounts for approximately 84% of the total \$5000 project budget. The final list of the purchases is mentioned in Table 9, which also included necessary spare parts for redundancy. Some items were sourced from the program inventory and AirLab at CMU, thus having no impact on the project budget.

Table 9: Budget

ID	Description	Subsystem	Price (USD)	Quantity	CoA (USD)
B.1	IMU	Mechanism	1020	1	1020
B.2	Mechanical Hardware	Mechanism	586	1	586
B.3	Electronic Accessories	Infrastructure	414	1	414
B.4	Electronic Hardware	Electronics	264	1	264
B.5	Motors	Mechanism	188	1	188
B.6	Arduino Mega	Electronics	48	1	48
B.7	Step Down Converters	Electronics	21	2	42
B.8	Current Sensors	Sensing	10	2	20
B.9	PCB	Electronics	12	1	12
B.10	Load Cell Amplifiers	Electronics	12	1	12
B.11	Strain gauges	Electronics	8	1	8
B.12	Clearpath Husky	Rover	18000	1	0
B.13	Intel RealSense D435	Sensing	320	2	0
B.14	Wi-Fi Router	Infrastructure	280	1	280
B.15	Monitor/Display	Infrastructure	117	1	117
B.16	Custom Project Merchandise	Misc.	400	1	400
B.17	Spare Mechanical & Electronic parts	Electronics & Mechanism	807	-	807
Total					**4218**

*CoA - Cost of Acquisition

7.3. Risk Management

Project risks were identified early and mitigated swiftly by either coordination, redundancies or design choices. Risk owners were assigned to facilitate accountability and efficient management. The major risks that were tracked throughout the development are summarized in Table 10. All risks were successfully managed and mitigated before the Fall Validation Demonstration.

Table 10: Project Risks

ID	Name	Type	Description	Status
R.1	Mobility platform unavailability	Technical	Mobility platform not available or unsuitable	Mitigated, Husky A200 acquired
R.2	Tool Incapable	Technical	Tool not capable of collecting and transporting enough material	Mitigated, Robust tool developed
R.3	Hardware Failure	Technical, Schedule, Cost	Electro-mechanical component failure	Mitigated, Spare parts acquired
R.4	Test site unavailability	Scheduling	Test site not available due to other project priority	Mitigated, coordinated with PRL
R.5	Data loss	Technical	Loss of code	Mitigated, Maintaining online repositories with backups
R.6	Dust damage	Technical	Dust ingress into actuators, electronics, and sensors	Mitigated, Electronics enclosure manufactured
R.7	Excavation depth estimation	Technical	Accuracy of the estimation of tool depth to start excavation	Mitigated, Controller tuned using extensive experiments
R.8	Weight estimation	Technical	Accuracy of the estimation of weight of sand in the drum	No longer applicable
R.9	Mapping accuracy	Technical	Accuracy of the mapping subsystem affects berm evaluation metrics	Mitigated, Camera calibrated and algorithm tested for robustness

8. Conclusions

8.1. Lessons Learned

Throughout the project's design and development phases, there were multiple hurdles, issues and risks that the team faced and addressed. This led to the team members learning various valuable lessons, some of which are discussed below.

The most valuable lesson is that all assumptions, especially technical, should be validated as early as possible in the development cycle. This ensures that the foundation for later development is strong and no unforeseen issues arise closer to the demonstrations.

It is recommended that the localization subsystem be tested rigorously with multiple test cases to ensure robustness. The transform values should also be confirmed carefully.

Another crucial lesson is making sure the compute subsystem is compatible and has sufficient hardware specifications to be able to execute the required algorithms and process raw camera data from the desired number of cameras.

All the functional and elemental blocks of the systems architectures should be broken down and given extensive thought to make sure the information and data flow is logical.

Lastly, in the hardware development cycles, it is important to not delay production just for the sake of optimizing the designs. The designs can be developed iteratively, which also ensures weaknesses and flaws are discovered in the initial few iterations itself.

8.2. Future Work

The LunAR-X system was built as a proof-of-concept for autonomous lunar excavation operations and it successfully met all the expected system requirements. It can autonomously build a berm that is atleast 15 cm high, restricted in length only by the worksite constraints and the battery. Even though the system is quite capable, the challenging problem of continuous autonomous excavation and the unforgiving environment of space require far more capabilities and robustness.

Future work, building towards a flight mission for lunar excavation, includes both, extensive hardware and software development.

The excavation tool will have to be modified for efficient digging, increased capacity, versatility and robustness against unexpected obstacles in the sand. The camera will have to be replaced with a model that satisfies standards set for operation in space. The camera placements will also have to be recalculated for better coverage. The base platform will have to be redesigned from scratch to have better mobility in regolith-like terrain. An accurate weight feedback for the excavation drum will also prove to be extremely useful for advanced autonomous operations.

The tool control software should be made capable of detecting and addressing excavation obstacles like bedrock, pebbles and debris. The task planner will have to be tested with more complex scenarios and developed to address continuous construction capabilities rather than just the simple application of discrete segments of dumps. The visual servo and mapping pipelines will have to be made more robust with the ability to handle even more edge and failure cases. Due to scope of the project, the localization subsystem used the Total Station as an external aid, although the dependence on a separate device should be removed in the future. The localization subsystem should be built to make use of the image data from the camera(s).

With these efforts as stepping-stones, it is our hope that these autonomous technologies one day aid in ensuring the future of humanity on the Moon, Mars, and beyond.

9. References

[1] M. Anand, I. A. Crawford, M. Balat-Pichelin, S. Abanades, W. Van Westrenen, G. Péraudeau, R. Jaumann, and W. Seboldt, "A brief review of chemical and mineralogical resources on the moon and likely initial in situ resource utilization (isru) applications," *Planetary and Space Science*, vol. 74, no. 1, pp. 42–48, 2012.

[2] V. Badescu, *Moon: Prospective energy and material resources*. Springer Science & Business Media, 2012.

[3] I. A. Crawford, "Lunar resources: A review," *Progress in Physical Geography*, vol. 39, no. 2, pp. 137–167, 2015.

[4] P. Eckart, *The lunar base handbook: an introduction to lunar base design, development, and operations*. McGraw-Hill New York, 1999.

[5] J. S. Lewis, M. S. Matthews, and M. L. Guerrieri, "Resources of near-earth space," *Resources of near-earth space*, 1993.

[6] G. Just, K. Smith, K. Joy, and M. Roy, "Parametric review of existing regolith excavation techniques for lunar in situ resource utilisation (isru) and recommendations for future excavation experiments," *Planetary and Space Science*, vol. 180, p. 104746, 2020.

A. Appendix

A.1. System Test Site

Figure 20: Planetary Robotics Lab - Moonyard