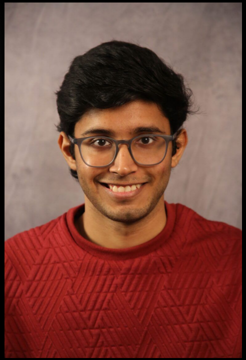
# Lunar ROADSTER

## (Robotic Operator for Autonomous Development of Surface Trails and Exploration Routes)

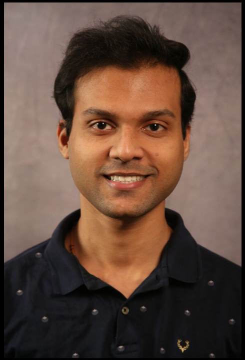
*"Starting with a foothold on the Moon, we pave the way to the cosmos"*

# The Team


Ankit Aggarwal


Deepam Ameria

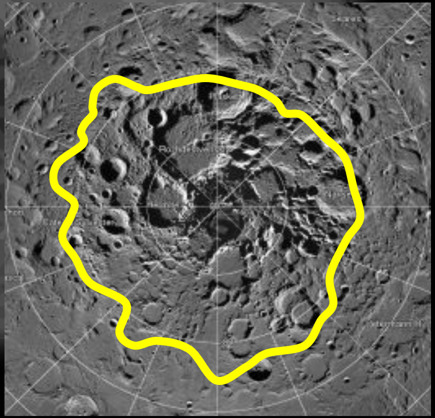

Bhaswanth Ayapilla

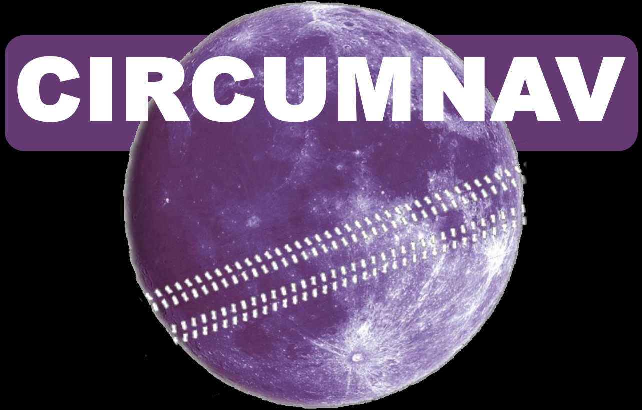

Simson D’Souza

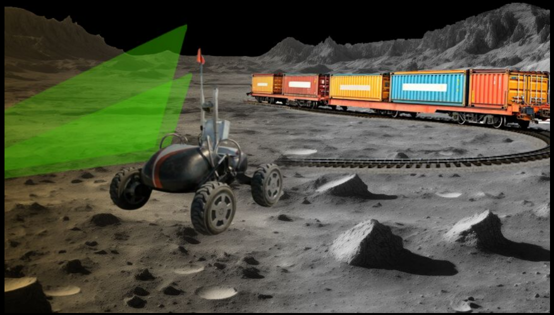

Boxiang (William) Fu


Dr. William “Red” Whittaker

# Motivation: The Lunar Polar Highway





Is it possible for a solar-powered rover to repeatedly drive around the Moon and never encounter a sunset?

# **Motivation:** The Lunar Polar Highway

Sun-synchronous circumnavigation around Moon at
28 days x 24 hr = 672 hour sun rotation

| | | |
|---|---|---|
| At equator | 11,000 km | 16 kph |
| At 50 deg | 7,040 km | 10 kph |
| At 60 deg | 5,500 km | 8 kph |
| At 70 deg | 3,700 km | 6 kph |
| At 75 deg | 2,800 km | 4 kph |
| At 80 deg | 1,870 km | 3 kph |
| At 81 deg | 1,529 km | 2.5 kph |

Jogging speed if the route was flat, circular and traversable

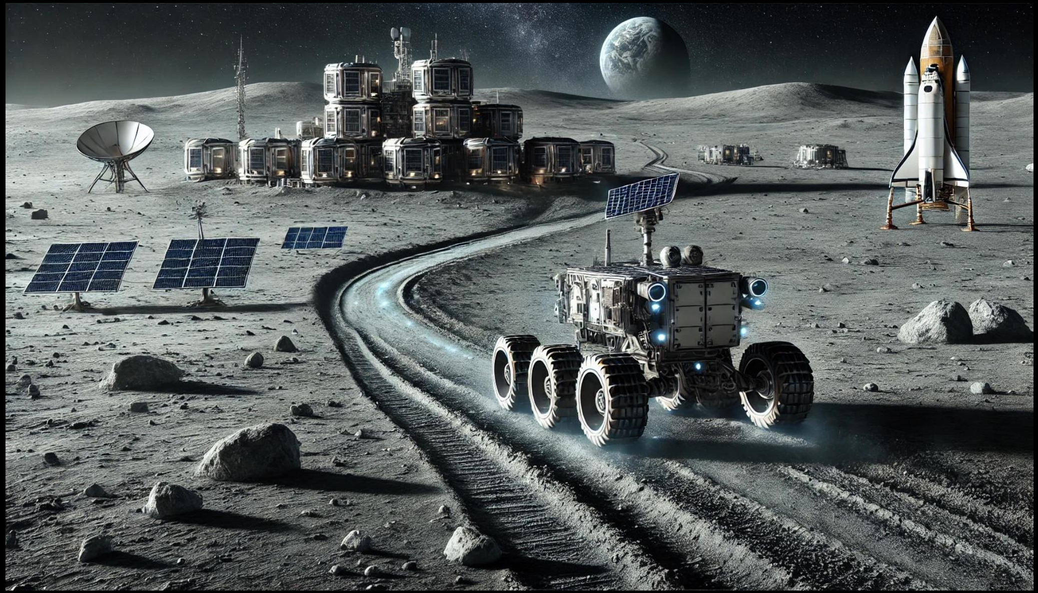
# The Project: Lunar ROADSTER

An autonomous moon-working rover capable of finding ideal exploration routes and creating traversable surface trails.

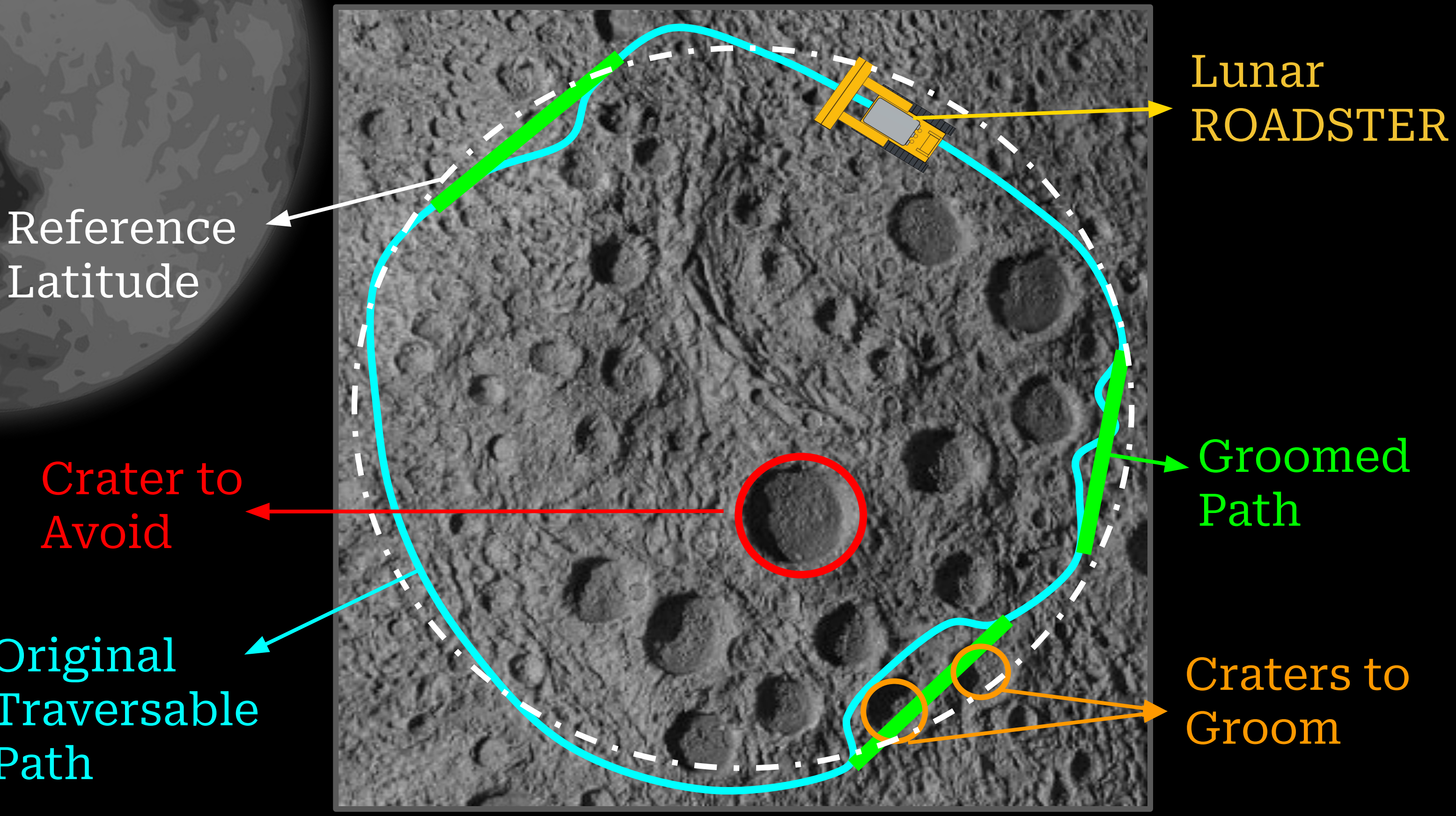
By grooming trail paths, rovers with less traversing capabilities will be able to travel at higher speeds and higher power efficiencies.

A traversable and circuitous trail path will allow rovers to maintain sun-synchronicity, thereby allowing machines to run for much longer.

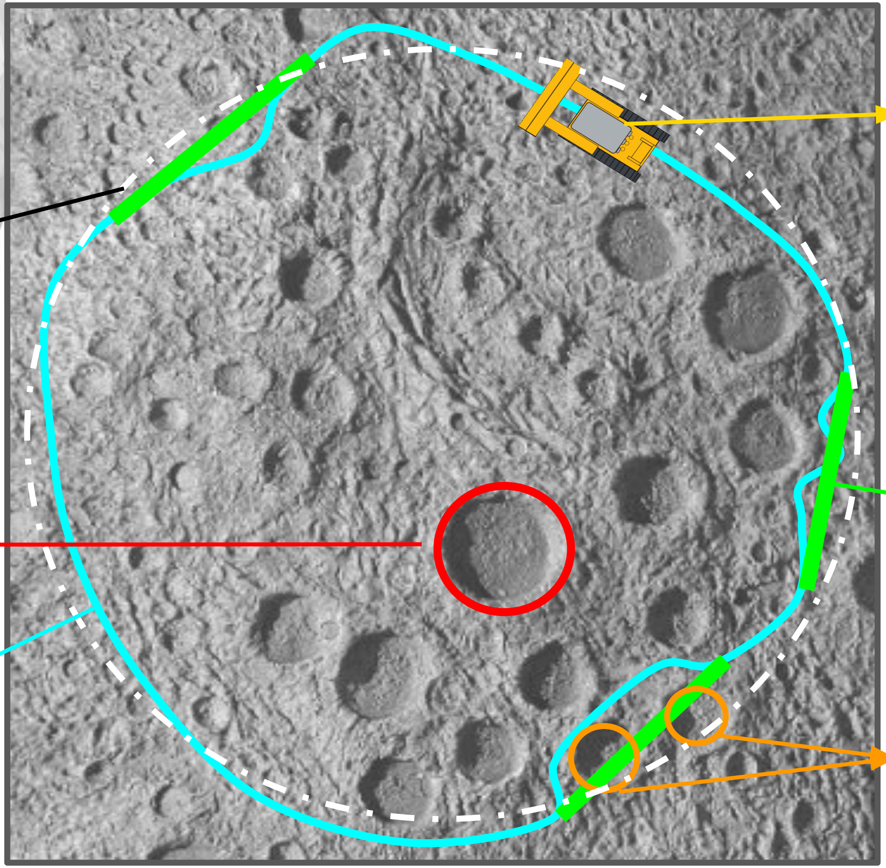
The groomed trails will become the backbone for colonization of the Moon by enabling transportation, logistics and enterprise development.



# Use Case: Circular Path Grooming



# Use Case: Circular Path Grooming



Lunar ROADSTER

Reference Latitude

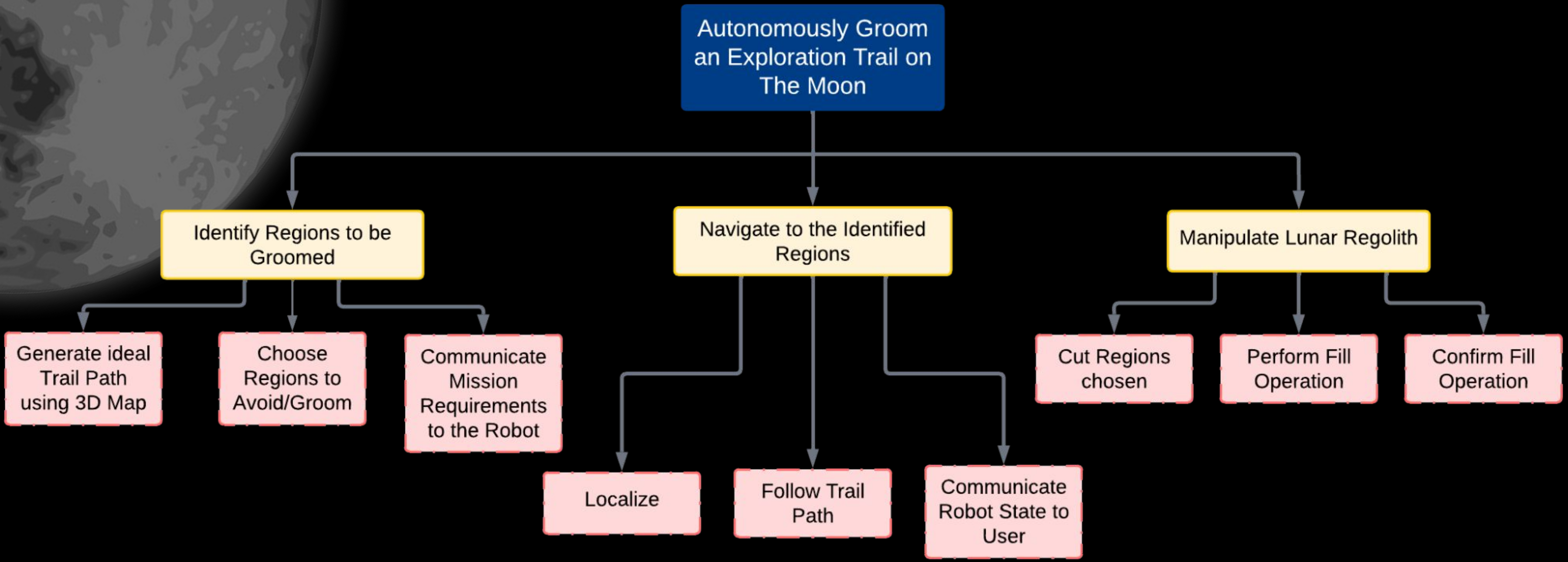
Groomed Path

Crater to Avoid

Original Traversable Path

Craters to Groom

# Objectives Tree



# Functional Requirements (Mandatory)

| Sr. No. | Mandatory Functional Requirement |
|---|---|
| M.F.1 | Shall perform trail path planning |
| M.F.2 | Shall operate autonomously |
| M.F.3 | Shall localize itself in a GPS denied environment |
| M.F.4 | Shall navigate the planned path |
| M.F.5 | Shall traverse uneven terrain |
| M.F.6 | Shall choose craters to groom and avoid |
| M.F.7 | Shall grade craters and level dunes |
| M.F.8 | Shall validate grading and trail path |
| M.F.9 | Shall communicate with the user |

# Non-Functional Requirements (Mandatory)

## Modified

| Sr. No. | Parameter | Description |
|---|---|---|
| M.N.1 | Weight | The rover must weigh under 50kg |
| 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 1m in all dimensions~~ |

# Non-Functional Requirements (Desirable)

| 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 |

# Performance Requirements (Mandatory)

## Modified

| Sr. No. | Performance Metrics |
|---|---|
| M.P.1 | Will plan a path with 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° |

# Current System Status

# Subsystem Completion Status

| Subsystem | Completion % | Future Work |
|---|---|---|
| **Sensors** | **100%** | None |
| **Computations** | **60%** | |
| 1. Jetson and Docker | **100%** | None |
| 2. Localization Stack | **85%** | Tune and improve Skycam based localization |
| 3. Navigation Stack | **60%** | Test and tune navigation stack (global + local) |
| 4. Perception Stack | **60%** | Fine tune, integrate with planning and navigation |
| 5. Validation Stack | **100%** | None |
| 6. Planning Stack | **40%** | Obtain Robot poses |
| 7. Behavior Executive Node | **40%** | Integrate all computation units into FSM |
| **External Infrastructure** | **100%** | None |
| **Mechanical** | **100%** | |
| 1. Dozer Assembly | **100%** | None |
| 2. Wheel Assembly | **100%** | None |
| **Actuation** | **100%** | None |
| **Electrical Power** | **100%** | None |

# **Description:** Sensors Subsystem

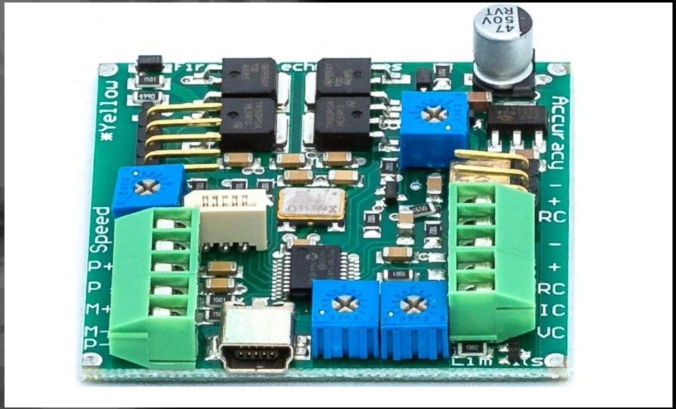

**Description:** All sensors used on the rover for computations.

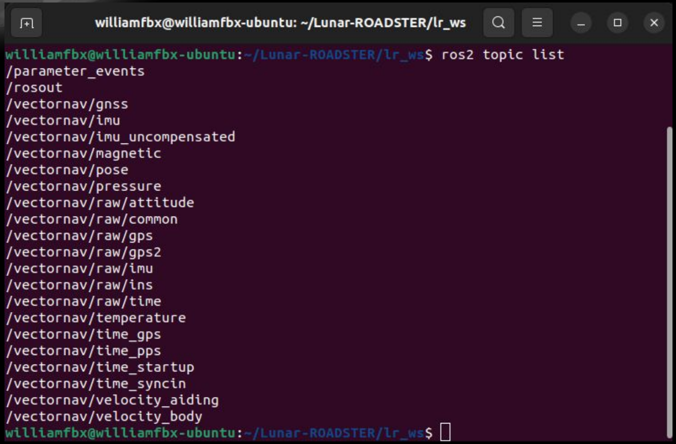
**Requirements:**

- Wheel Encoders (x4)
- Mast Depth Camera (ZED 2i)– New!
- IMU (VectorNav)
- Actuator Feedback Sensor

**Expected Functionality:** The sensor data is published to various ROS topics and can be used inside the Docker container to perform computations.

# **Status:** Sensors Subsystem






**Implementation:**

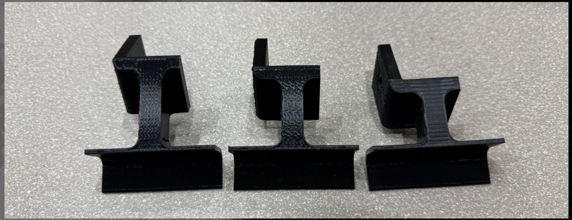
- IMU implemented using VectorNav ROS package
- Wheel Encoder parsed using micro-ROS
- Purchased actuator has a feedback potentiometer
- Obtained point cloud feed using ZED 2i wrappers

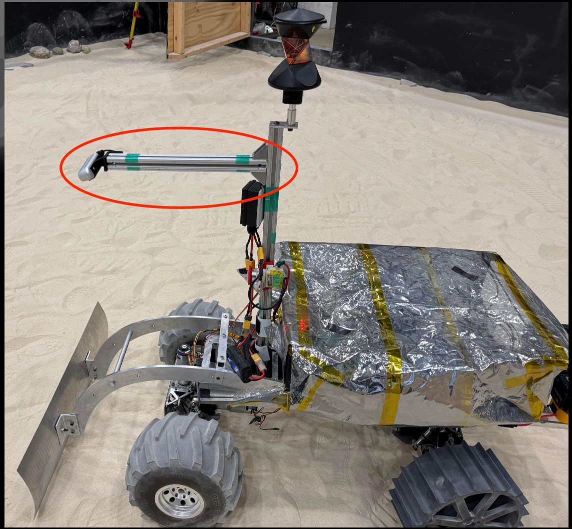
**Challenges:**

- CUDA compatibility issues with ZED SDK on Jetson Xavier – tackled by upgrading to Orin
- IMU issues – tackled by purchasing new IMU

**Status:** 100% Complete
IMU, wheel encoder, and linear actuator finalized.
Upgraded RealSense to ZED 2i during summer.

# Evaluation: Sensors Subsystem





**Modelling:**

- New CAD designed and 3D printed mast depth camera mounts of 30, 40, 45, and 50 degree angles relative to mast

**Analysis:**

- 50 degree angled camera mount gives the best view of craters
- However, if tool height is above 15%, occlusion occurs
- Point cloud obtained from ZED 2i is much denser than Realsense

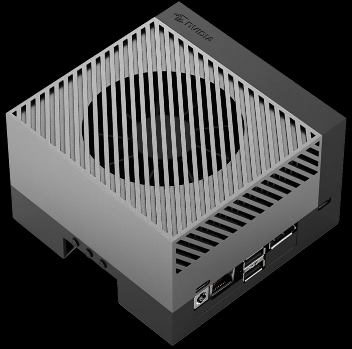
**Testing:**

- Fall T09: CPU/GPU Usage of Autonomous Stack is Below Orin Compute Limits
- Fall T10: Maintenance, Reliability and Quality Assurance Test

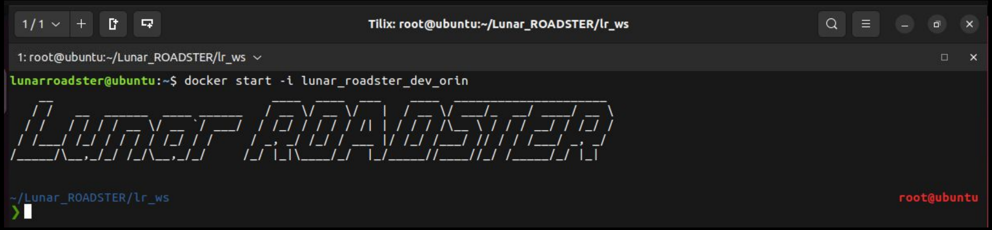
# Description: Computations Subsystem Jetson and Docker

Jetson AGX Xavier

~4x CUDA cores
~1.7x faster CPU performance
~2x memory
~1.5x higher bandwidth



Jetson Orin




**Description:** Set up the Jetson Orin with Docker to host and run all critical system packages

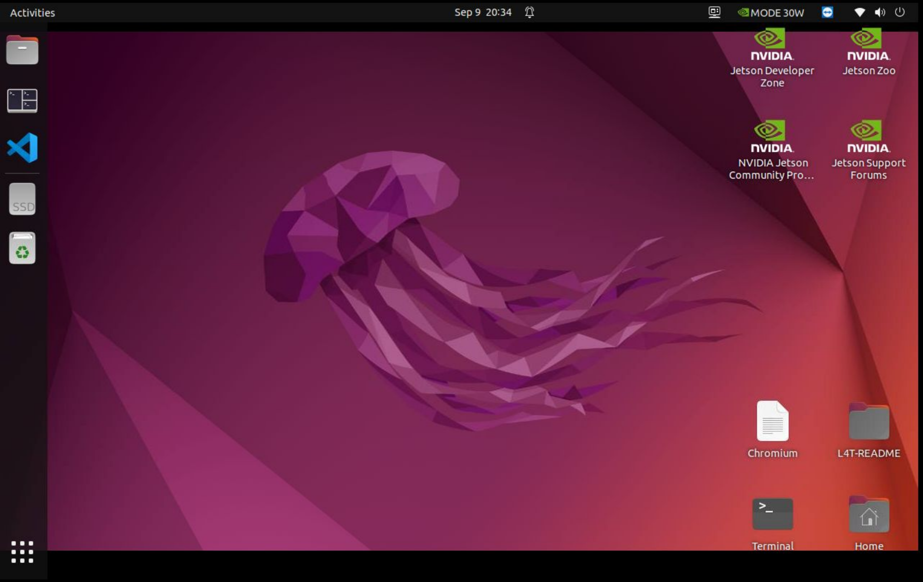
**Requirements:**

- NVIDIA Jetson Orin – New!
- LAN Router
- Team laptop (operations terminal)

**Expected Functionality:**

- Acts as primary on-board compute
- Runs ROS2 Humble
- Runs micro-ROS
- Hosts and manages all necessary packages and device drivers inside Docker containers

# **Status:** Computations Subsystem Jetson and Docker Unit




**Implementation:**

- Connect Jetson to rover's power system
- Assigned static IP on the LAN to enable SSH-based remote access from operations terminal
- Start docker container and initialize all core services and packages
- Tmux used for master launcher

**Challenges:**

- Setup of a VNC server for remote GUI access

**Status:** 100% Complete
Integrated ZED SDK drivers

# **Evaluation:** Computations Subsystem Jetson and Docker Unit

Lunar-ROADSTER / lr_ws / docker / **lrdev_jetson_zed.dockerfile**

**LunarROADSTER** updated dockerfiles

Code | Blame — Executable File · 251 lines (208 loc) · 9.85 KB

```
# ------- ZED Camera SDK --------
FROM deepamameria/zed_ros2_sdk:36.4.0

# ------- ROS 2 Humble image --------
# FROM dustynv/ros:humble-desktop-l4t-r36.2.0

# Set environment variables
ENV DEBIAN_FRONTEND=noninteractive \
  ROS_DISTRO=humble \
  WORKSPACE=/root/Lunar_ROADSTER/lr_ws

# --------------- NVIDIA Libraries Configuration ------------------------
# Ensure NVIDIA drivers and CUDA libraries are accessible inside the container
ENV NVIDIA_VISIBLE_DEVICES=all
ENV NVIDIA_DRIVER_CAPABILITIES=all
ENV LD_LIBRARY_PATH=/usr/local/cuda/lib64:/usr/lib/aarch64-linux-gnu:/usr/lib/aarch64-linux-gnu/tegra:$LD_LIBRARY_PATH


COPY ./ $WORKSPACE

# --------------- Base environment configuration ------------------------
# Setup shell
COPY docker/.p10k.zsh /root/.p10k.zsh
ENV TERM=xterm-256color
RUN apt-get update && apt-get install -y zsh bash wget \
  && PATH="$PATH:/usr/bin/zsh" \
  # Install Oh-My-Zsh with default theme
  && sh -c "$(wget -O- https://github.com/deluan/zsh-in-docker/releases/download/v1.1.3/zsh-in-docker.sh)" \
  # Zsh Autocomplete
  && git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-/root/.oh-my-zsh/custom}/plugins/zsh-autosuggestions \
  && echo "ZSH_AUTOSUGGEST_HIGHLIGHT_STYLE='fg=#777777'" >> ~/.zshrc \
  # Initialize custom zsh theme
  && echo "[[ ! -f ~/.p10k.zsh ]] || source ~/.p10k.zsh" >> ~/.zshrc \
  # Source ROS
  && echo '# ROS' >> /root/.zshrc \
  && echo '# ROS' >> /root/.bashrc \
  && echo 'source /opt/ros/$ROS_DISTRO/setup.zsh' >> /root/.zshrc \
```

## Modelling:

- Created custom Dockerfile built on ZED SDK image that installs all required system packages

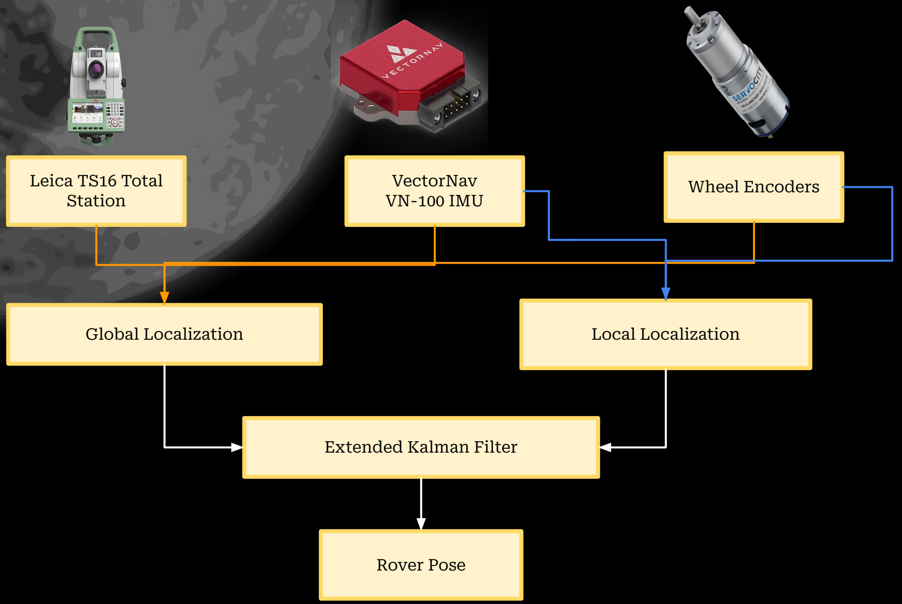
## Analysis:

- Verified that all required nodes and drivers start successfully inside the container

## Testing:

- Fall T01: Rerun Spring Validation Demonstration
- Fall T09: CPU/GPU Usage of Autonomous Stack is Below Orin Compute Limits

# Description: Computations Subsystem Total Station Localization



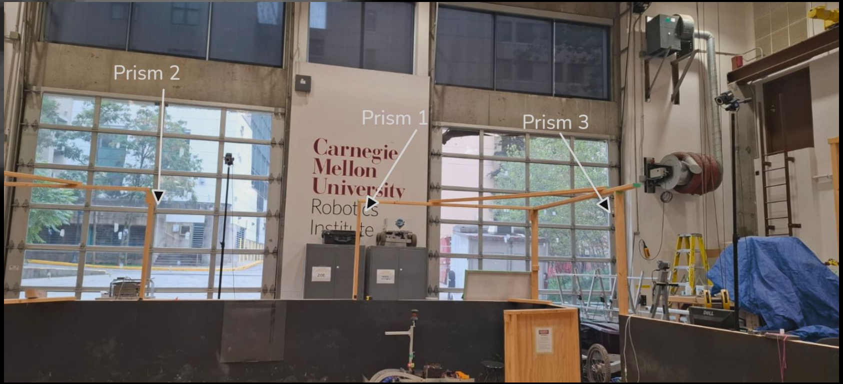

**Description:** Localize the rover in the Moon Yard

**Requirements:**

- Leica TS16 Total Station
- NVIDIA TX2 Relay Chip + LAN
- VectorNav IMU
- Wheel Encoders

**Expected Functionality:** Accurately localize rover pose inside the Moon Yard, to be used further for navigation

# Status: Computations Subsystem Total Station Localization




**Implementation:**

- On-board IMU and encoders used for local localization
- Total station data fused with IMU and encoders for global localization
- EKF running on Jetson using robot_localization package, now tuned to prevent odometry drift
- Yaw calibration to ensure IMU data is w.r.t map frame
- Fixed frame shifts issue caused by total station battery replacement by using resection method (uses 3 total station prisms instead of 2)

**Challenges:**

- Incorrect Jetson Docker network permissions blocked two-way communication
- Minor offset introduced when total station battery is replaced, causing frame inconsistencies

**Status:** 100% Complete

# Evaluation: Computations Subsystem Total Station Localization

**Modelling:**

- Configured and tuned EKF to fuse sensor inputs
- Set up frame transforms to ensure all sensor data aligns properly at base_link frame
- Performed yaw calibration to ensure consistent orientation data from IMU

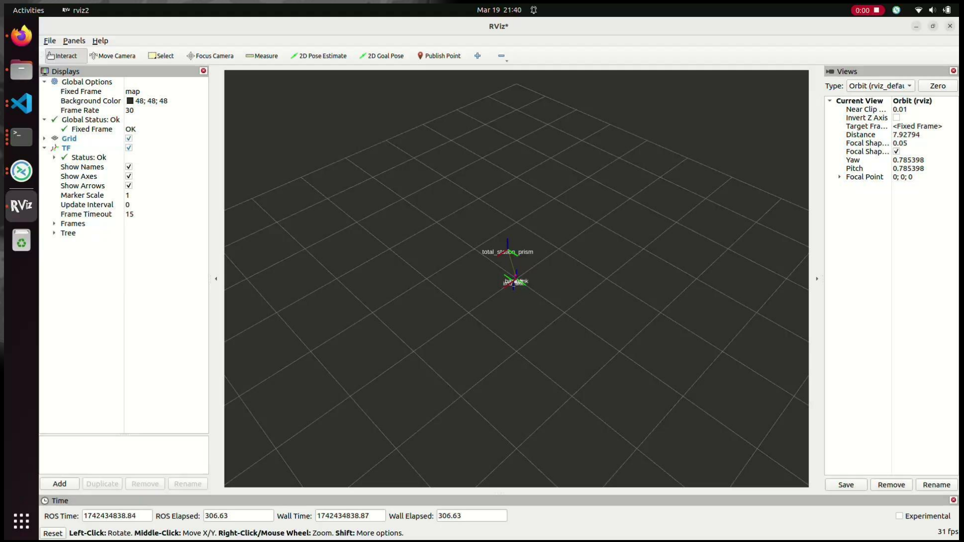
**Analysis:**

- Analyzed pose stability and drift over time during testing
- Analyzed sensor noise and measurement delays, tuned EKF parameters to minimize odometry drift
- Assessed effect of total station resections and battery swaps on accuracy

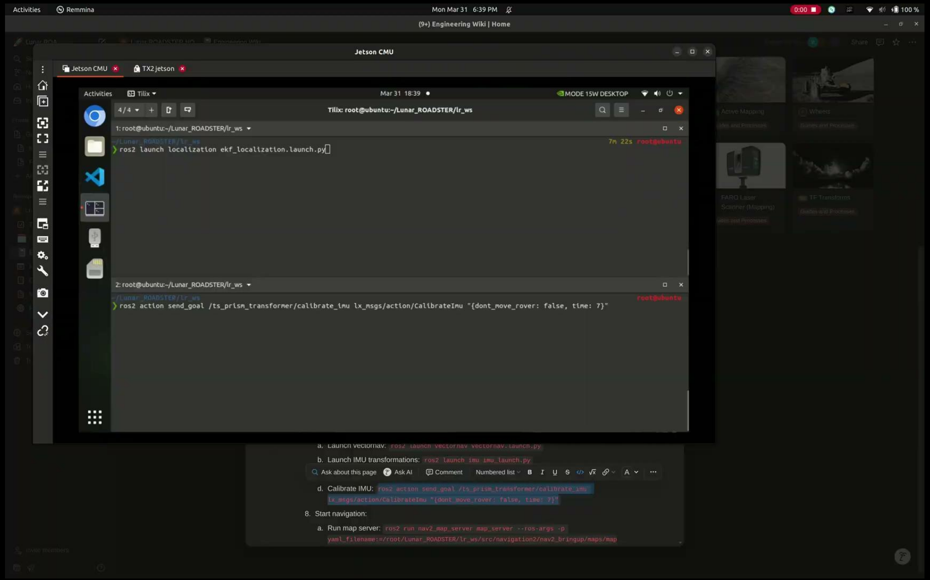
**Testing:**

- Spring T09: Rover can localize itself accurately
- Spring T15: Spring Validation Demo Test

# Evaluation: Computations Subsystem
# Total Station Localization

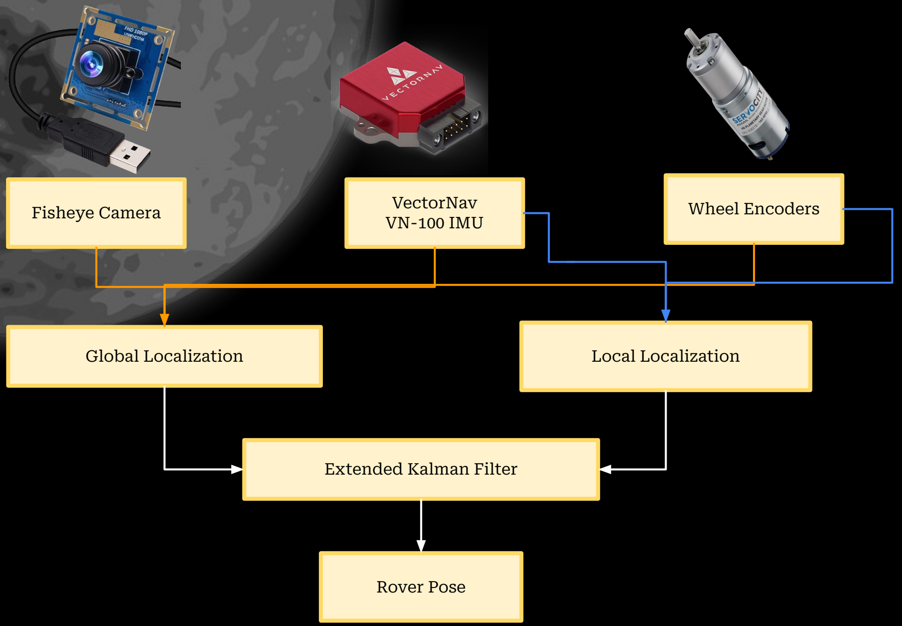


Localization drift issue




Drift corrected + Yaw calibrated

# Description: Computations Subsystem Skycam Localization




**Description:** Localize the rover in the Moon Yard using rover mounted Skycam – New!

**Requirements:**

- Fisheye camera (Skycam)
- VectorNav IMU
- Wheel Encoders

**Expected Functionality:** Accurately localize rover pose inside the Moon Yard, to be used further for navigation

# Status: Computations Subsystem Skycam Localization

**Implementation:**
- **Data Collection Pipeline:** Saves and indexes raw Skycam camera images along with ground truth localization provided by total station
- **Training Pipeline:** Train a neural network using PyTorch by regressing camera image against position
- **Deployment Pipeline:** Deploys network to a ROS node at runtime

**Challenges:**
- Trained model is inaccurate (MSE localization error is ~1 meter)
- Corners of Moon Yard is recently covered, need to retake training data to account for this

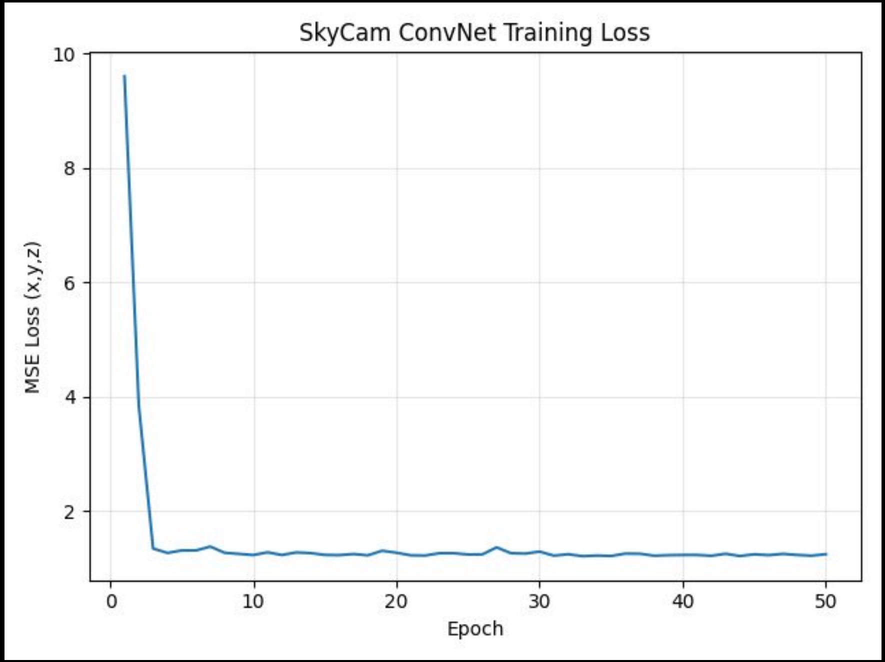
**Status:** 85% Complete
- Tune MSE localization error to be below 10cm

# Evaluation: Computations Subsystem Skycam Localization

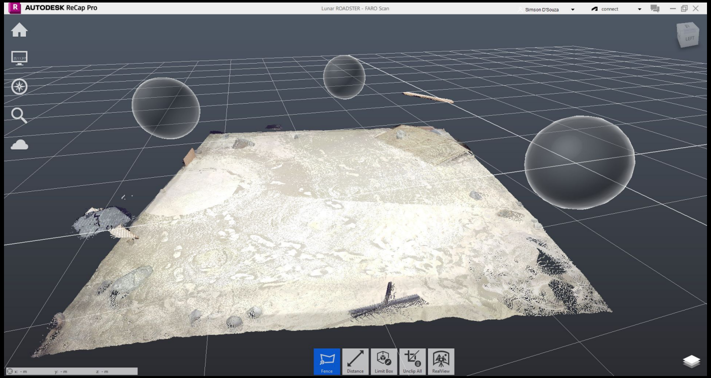
**Modelling:**
- Tried different neural network architectures (vanilla, ConvNet, etc)

**Analysis:**
- Split the training data into training and validation datasets
- Analyzed the localization MSE loss on validation dataset
- Compared localization error against total station method

**Testing:**
- Fall T08: SkyCam Localization Validation



# Description: Computations Subsystem Navigation Stack


**Description:** Obtain a global plan between craters based on target poses and navigate the rover to their locations

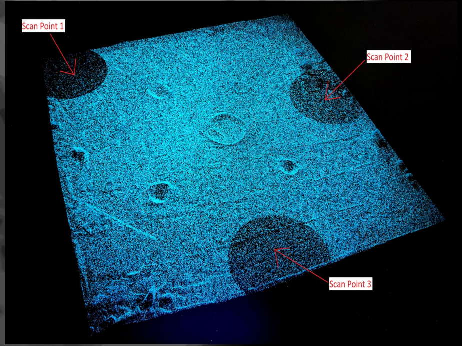
**Requirements:**

- FARO Laser Scanner
- NVIDIA Jetson Orin
- ZED 2i Depth Camera
- Rover
- Team laptop (operations terminal)

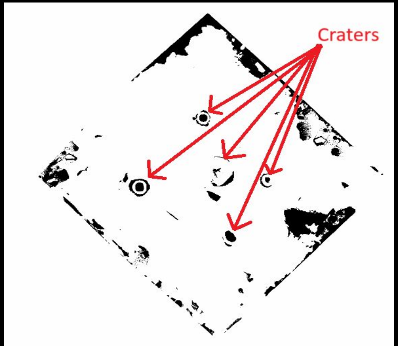
**Expected Functionality:**

- Accurately detect and classify gradable craters
- Ensure no gradable crater is overlooked
- Compute an optimal navigation path while avoiding obstacles (non-gradable craters)
- Navigate and reach the goal location correctly

# Status: Computations Subsystem Navigation Stack




**Moon Yard Scan Visualization**




**Global Costmap**

**Implementation:**

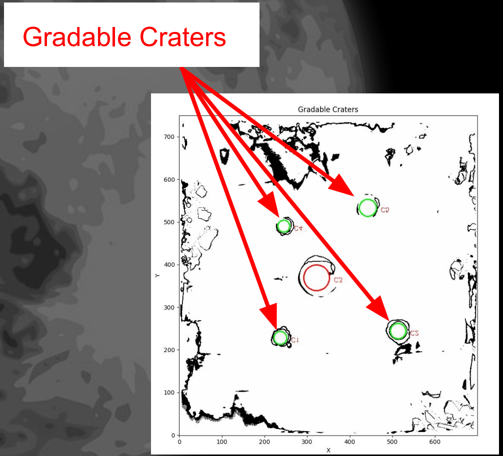
- FARO Laser Scanner used to map the Moon Yard and obtain a high density point cloud
- The point cloud is converted into an occupancy grid using RANSAC and thresholding – contains craters centroid and diameter information
- Global Path Planner uses the occupancy grid map to compute a smooth trajectory between craters, taking into account Ackermann primitives and ring bias
- Pure Pursuit Global Controller executes the planned path and reaches the desired goal (close to the crater)
- MPC Local Controller uses information from perception and planning stacks to reach desired target poses
- Navigation stack is integrated with tool planner to perform transport assignments

**Challenges:**

- FARO output file (.fls) was incompatible with the PCL library and ROS.
- Fine-tune global planner and global/local controller parameters for optimal performance on our robot

**Status:** 60% Complete (Tune Navigation Stack) - Fall Goal

# **Evaluation:** Computations Subsystem Navigation Stack




**Identified Gradable Craters**

```
Gradable Craters Location
Crater C1: Diameter = 0.300 meters
  Centroid of Crater C1: X = 2.380 m, Y = 2.289 m

Crater C2: Diameter = 0.360 meters
  Centroid of Crater C2: X = 5.131 m, Y = 2.443 m

Crater C3: Diameter = 0.600 meters

Crater C4: Diameter = 0.280 meters
  Centroid of Crater C4: X = 2.453 m, Y = 4.909 m

Crater C5: Diameter = 0.400 meters
  Centroid of Crater C5: X = 4.421 m, Y = 5.335 m
```

**Gradable Craters Location**

**Modeling:**

- Generated dense point cloud maps by taking multiple FARO scans and updated the map origin for consistent localization and navigation frames
- Custom Global Path Planner that takes into account various costs and heuristics for optimal planning
- Custom Global and Local Controllers to allow more control over robot movement
- Used RViz extensively to visualize and debug transformations and paths

**Analysis:**

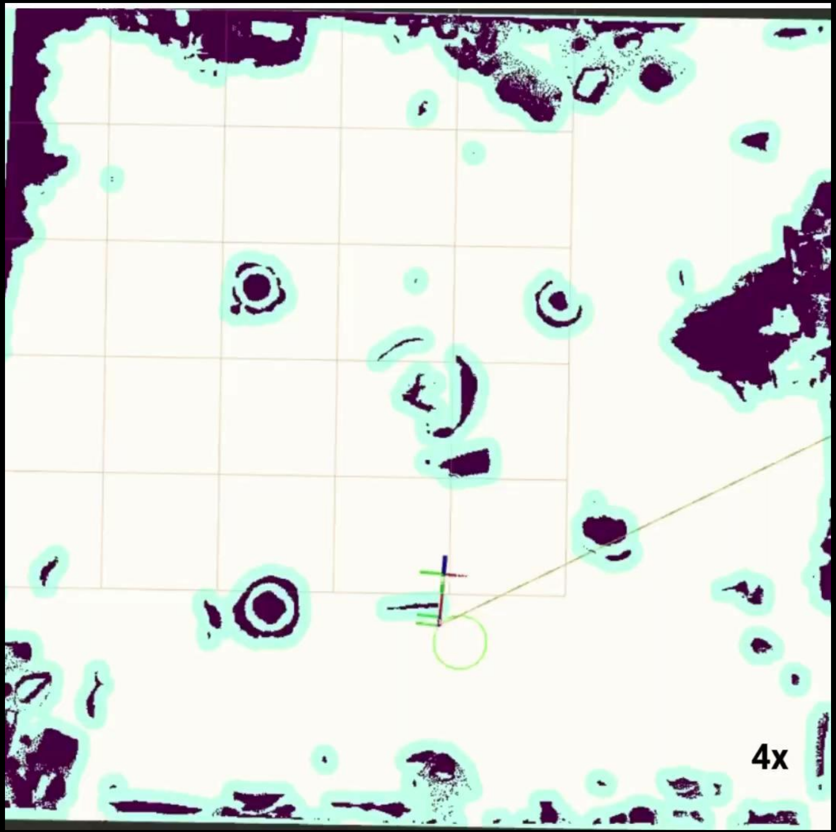
- Evaluated planned paths for smoothness, obstacle avoidance, and feasibility
- Analyzed factors affecting navigation: localization noise, data delay, computation load, and parameter tuning

**Testing:**

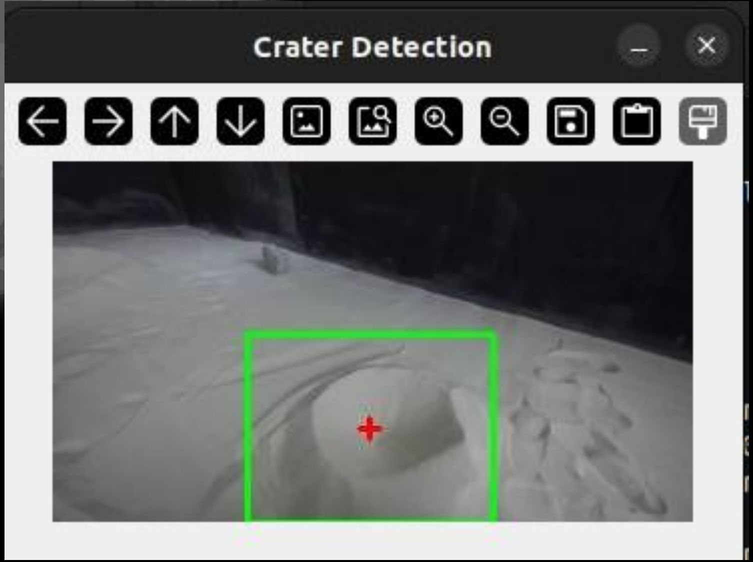
- Fall T02: Global Path Planner Accuracy Test
- Fall T03: Filtering and Selection of Gradable Craters
- Fall T04: Navigation planner maximum deviation test
- Fall T06: Repeatability Test of Local Navigation Controller
- Fall T11: Fall Validation Demo Preparation Test

# Evaluation: Computations Subsystem Navigation Stack

# Evaluation: Computations Subsystem Navigation Stack



# Description: Computations Subsystem Perception Stack




**Description:** Uses RGB-D sensor to detect craters and extract coordinates for planning and navigation subsystems.

**Requirements:**

- ZED 2i stereo camera
- Camera driver packages/ROS2 Wrappers
- Jetson Xavier AGX

**Expected Functionality:**

- Output coordinates of the detected crater(s) in the world frame.

# Status: Computations Subsystem Perception Stack

**Implementation:**

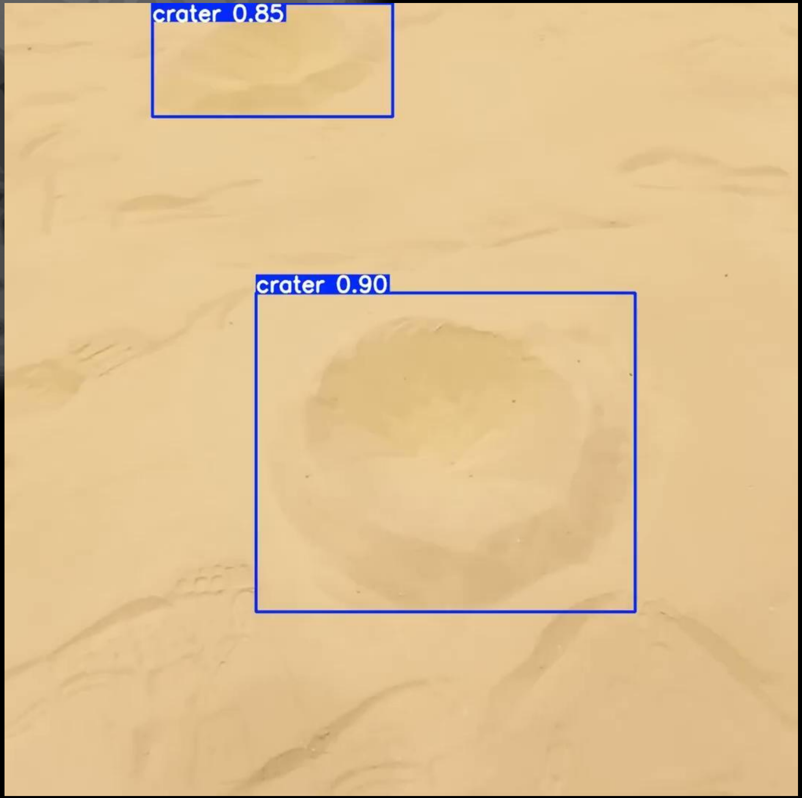
- Collect data for training and validating a CNN Object Detection Model
- Load the trained model on edge compute
- Obtain RGB raw images from ZED Camera over ROS topics
- Detect craters in real-time
- Obtain centroid of crater and publish on ROS topic for planning and navigation stacks

**Challenges:**

- Rover mast moves due to rover suspension. May cause jitter in coordinates.
- Compute or  inference is not optimized for real-time detection

**Status:** 60% Complete

# Evaluation: Computations Subsystem Perception Stack




**Modelling:**

- Collected dataset and annotated using Roboflow Annotation Tool
- Trained YOLO v8 and validated on test videos
- Transferred to Orin for online real-time perception and inference
- Used bounding box analysis to determine crater centre coordinates

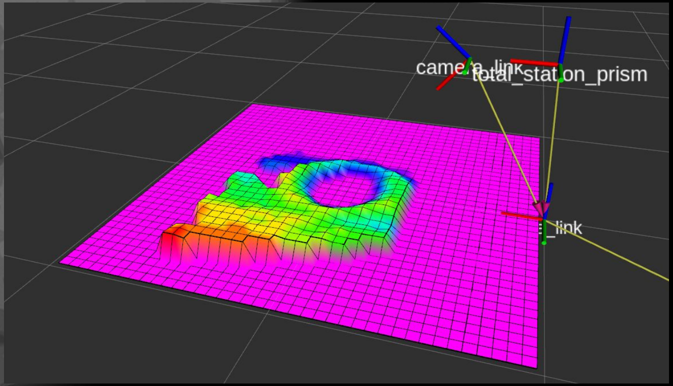
**Analysis:**

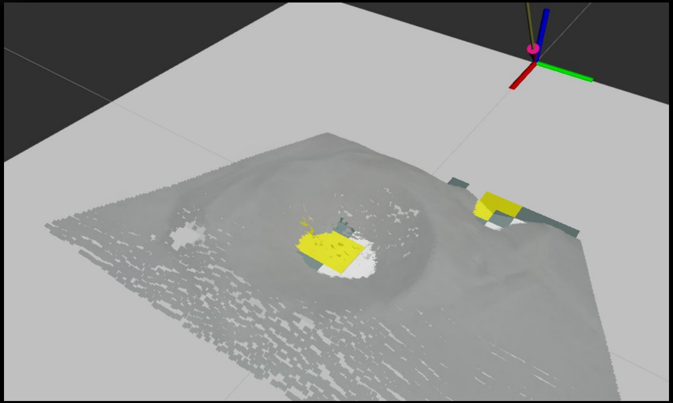
- Inference is laggy and requires optimization and smoothening for persistent detection

**Testing:**

- Fall T05: Perception Stack Crater Geometry Extraction Test

# Description: Computations Subsystem Validation Stack





**Description:** Validates if groomed crater satisfies maximum traversability requirement (M.P.6)

**Requirements:**
- ZED 2i stereo camera
- Camera driver packages
- Jetson Orin

**Expected Functionality:**
- Outputs boolean of grading success or failure plus additional gradient information in FOV region

# **Status:** Computations Subsystem Validation Stack

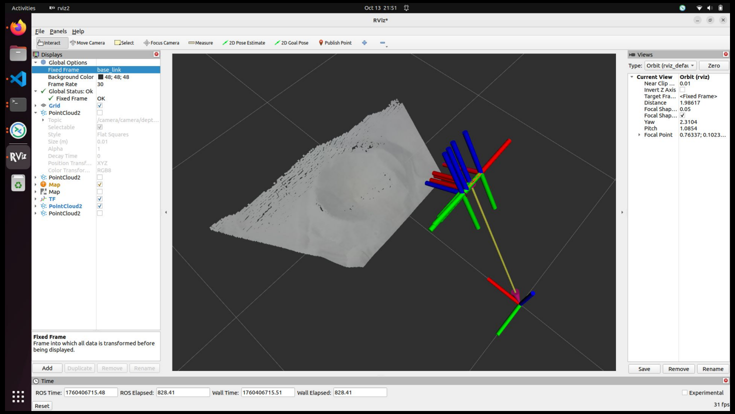
## Implementation:

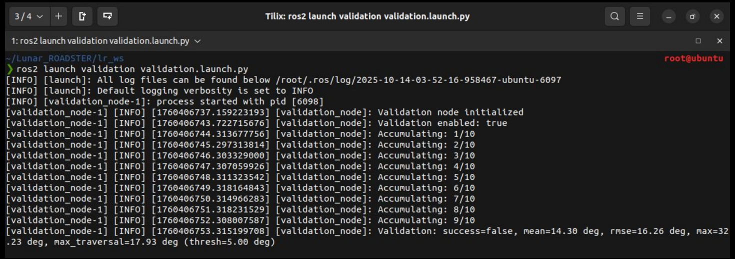
- Obtains most recent point cloud feed from the ZED camera
- Voxel downsampling, and builds a KdTree
- Surface normal estimation and calculate per-point slope from the vertical z-axis
- Nearest neighbor smoothing and masking to filter out phantom points and walls/edges
- Compute aggregate slope statistics over several frames

## Challenges:

- Rover mast moves due to rover suspension
- Phantom points in point cloud data sometimes causes erroneous gradient estimates

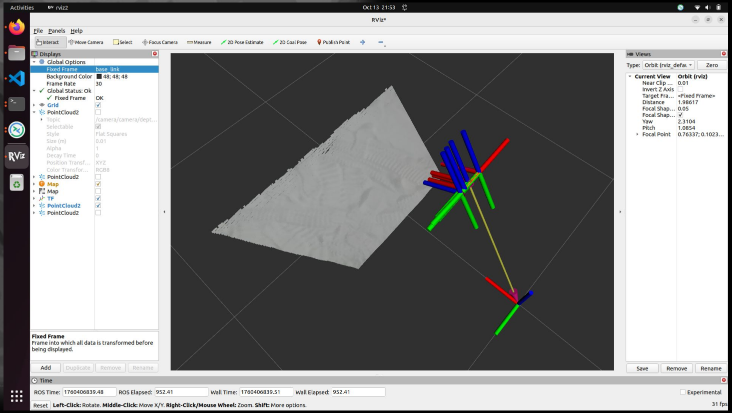
**Status:** 100% Complete






Validation before grooming

# Evaluation: Computations Subsystem Validation Stack




```
Tilix: ros2 launch validation validation.launch.py
1: ros2 launch validation validation.launch.py
~/Lunar_ROADSTER/lr_ws                                                root@ubuntu
❯ ros2 launch validation validation.launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2025-10-14-03-54-10-850300-ubuntu-6261
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [validation_node-1]: process started with pid [6263]
[validation_node-1] [INFO] [1760406851.046740258] [validation_node]: Validation node initialized
[validation_node-1] [INFO] [1760406857.613517088] [validation_node]: Validation enabled: true
[validation_node-1] [INFO] [1760406858.315540620] [validation_node]: Accumulating: 1/10
[validation_node-1] [INFO] [1760406859.316497540] [validation_node]: Accumulating: 2/10
[validation_node-1] [INFO] [1760406860.319700535] [validation_node]: Accumulating: 3/10
[validation_node-1] [INFO] [1760406861.320069391] [validation_node]: Accumulating: 4/10
[validation_node-1] [INFO] [1760406862.314155168] [validation_node]: Accumulating: 5/10
[validation_node-1] [INFO] [1760406863.314263795] [validation_node]: Accumulating: 6/10
[validation_node-1] [INFO] [1760406864.321429016] [validation_node]: Accumulating: 7/10
[validation_node-1] [INFO] [1760406865.324008797] [validation_node]: Accumulating: 8/10
[validation_node-1] [INFO] [1760406866.322223838] [validation_node]: Accumulating: 9/10
[validation_node-1] [INFO] [1760406867.316454935] [validation_node]: Validation: success=true, mean=2.82 deg, rmse=2.92 deg, max=4.55
deg, max_traversal=1.73 deg (thresh=5.00 deg)
```

Validation after grooming

## Modelling:

- Calibrated camera {X, Y, Z, R, P, Y} so that flat ground has zero elevation and no rotation

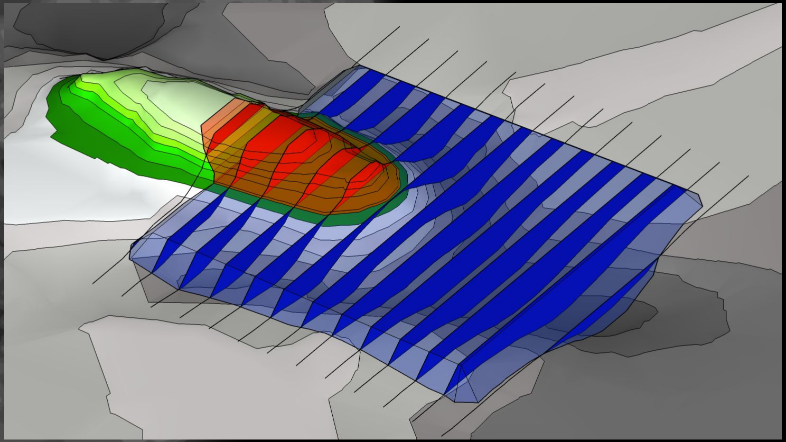
## Analysis:

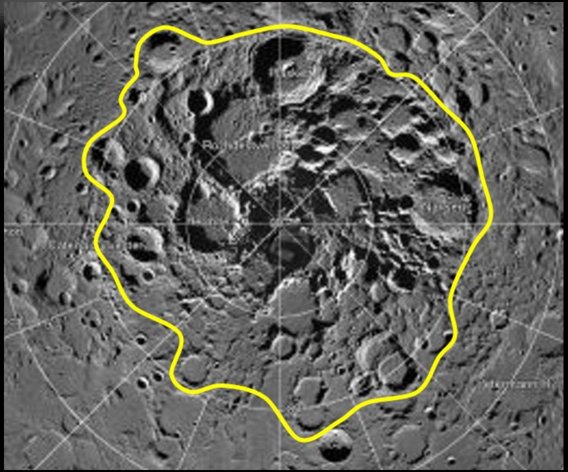
- Analyzed accuracy and robustness of output on different terrains (crater, flat ground, smoothed sand, facing wall, facing rock)

## Testing:

- Fall T07: Trail Grooming Slope Validation

# **Description:** Planning Stack





**Description:** Subsystem that plans sand manipulation by using the tool and drive train.

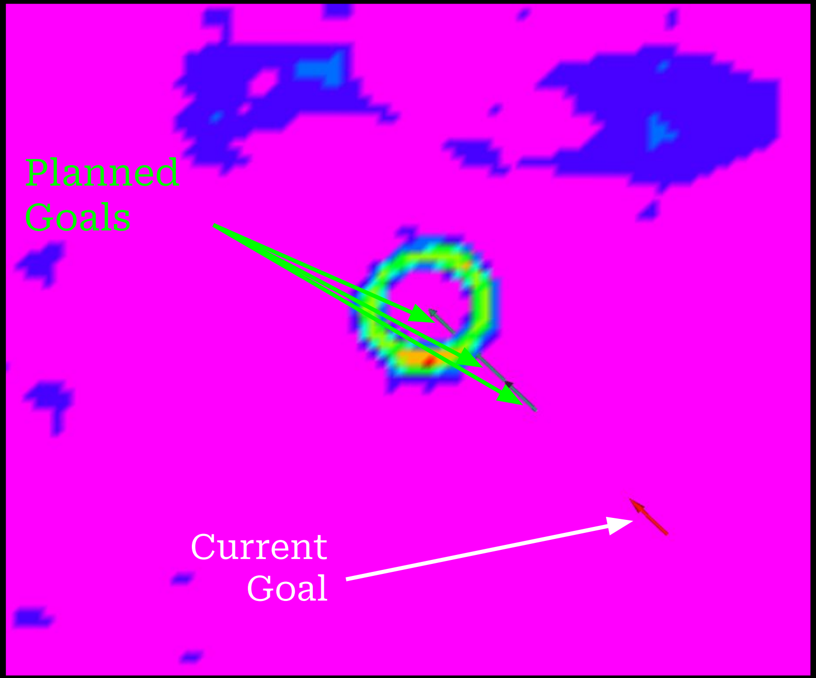
**Requirements**

- Jetson Orin
- Dozer Assembly
- Drivetrain (with Wheel Assembly)
- Global Map

**Expected Functionality**

- Plans a control input for the tool
- Plans a trajectory for the rover for manipulation
- Outputs waypoints and tool trajectories to the navigation planner

# **Status:** Planning Stack




**Implementation:**

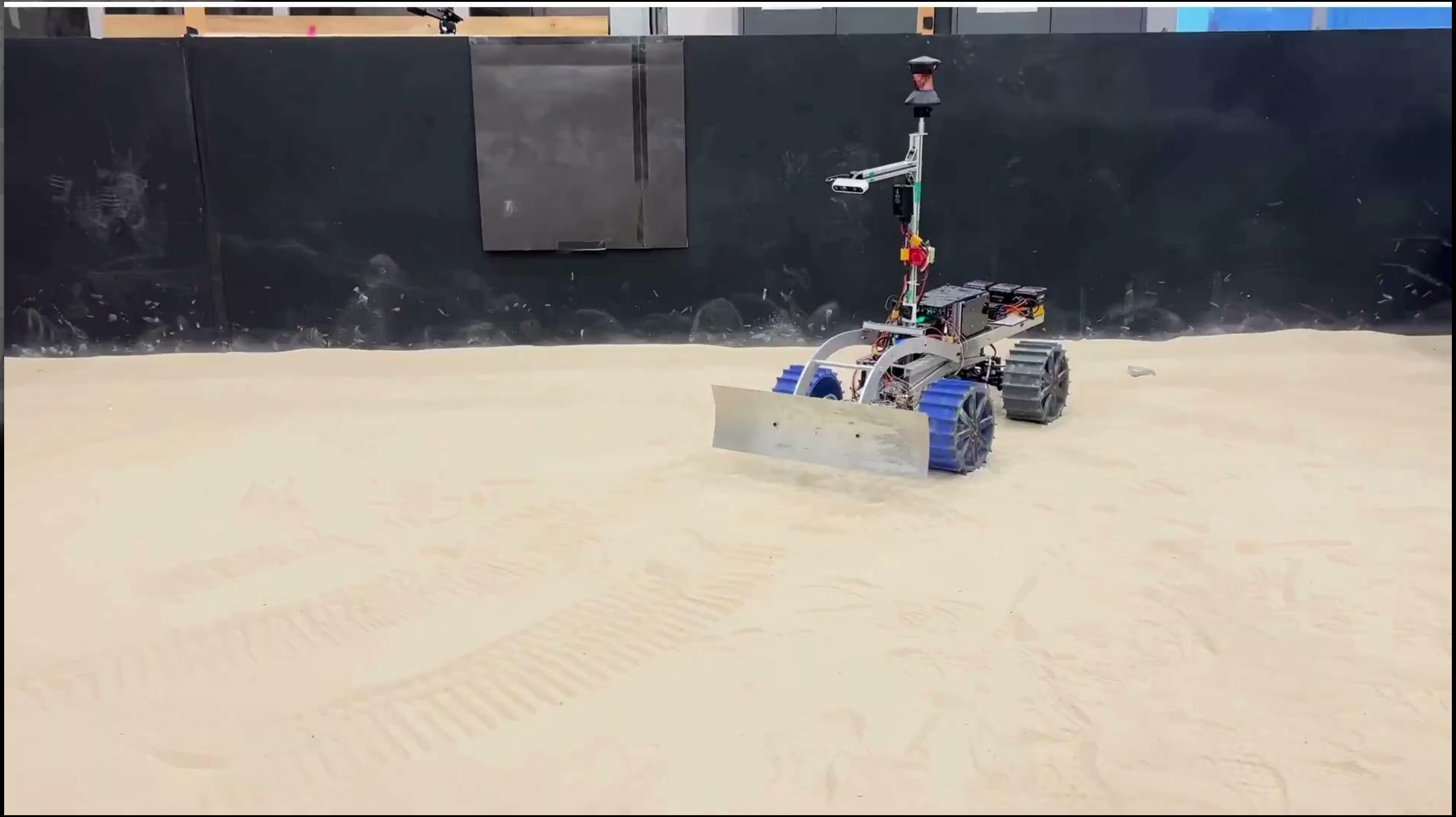
- The new planning stack will be using the geometry of the crater as outputted by the perception stack.
- The diameter and coordinates of the crater's centroid will be used to create robot poses for manipulation
- The tool height will be determined by the depth data obtained from the ZED

**Challenges:**

- Accurate crater geometry will be needed
- Robot poses are an estimate of an estimate, meaning they will need to be tuned

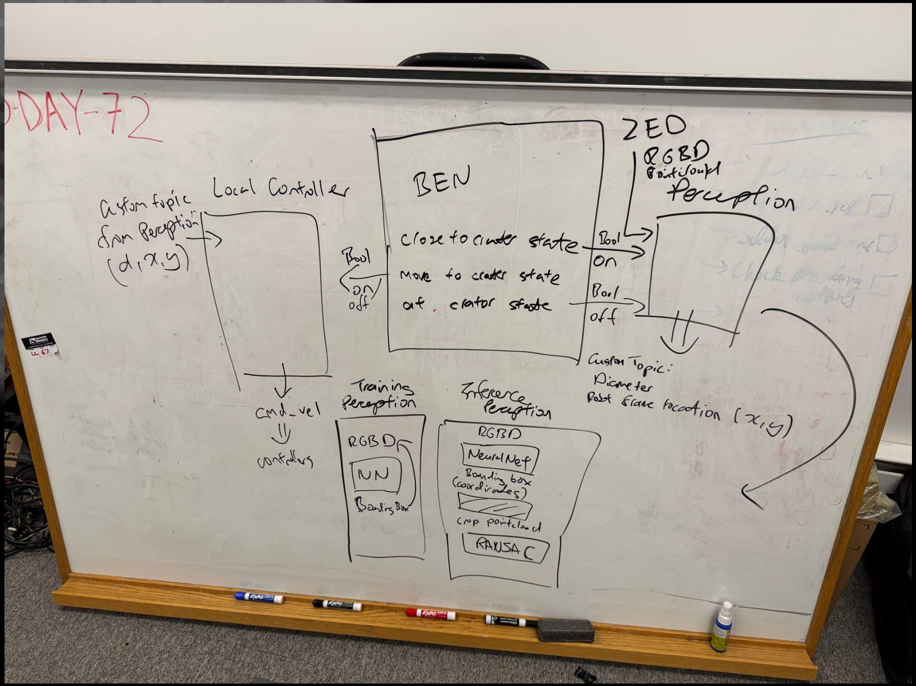
**Status:** 40% complete

# Evaluation: Planning Stack



4x

# Description: Computations Subsystem Behavior Executive Node




Initial Methodology for Integration

**Description:** High level behaviour tree for entire autonomy stack

**Requirements:**

- Sensors subsystem
- Computations subsystem
- External infrastructure
- Mechanical subsystem
- Actuation subsystem
- Electrical power subsystem

**Expected Functionality:**

- Entire system is able to grade craters autonomously

# Status: Computations Subsystem Behavior Executive Node

**Implementation:**

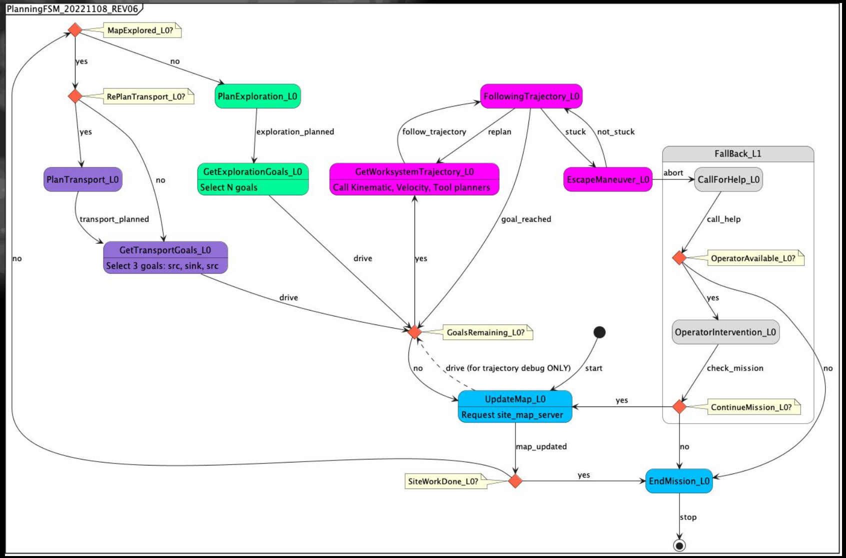
- FSM callbacks implemented at 2 Hz
- Heavy computations parallelized and detached from main thread to not block FSM node from iterating

**Challenges:**

- Compute may not be sufficient with new modules

**Status:** 40% Complete
(Integrate new navigation, perception and validation unit into FSM)

# Evaluation: Computations Subsystem Behavior Executive Node




## Modelling:

- Initial FSM design sourced from Crater Grader. Adapted and modified design based on our needs to include all new packages

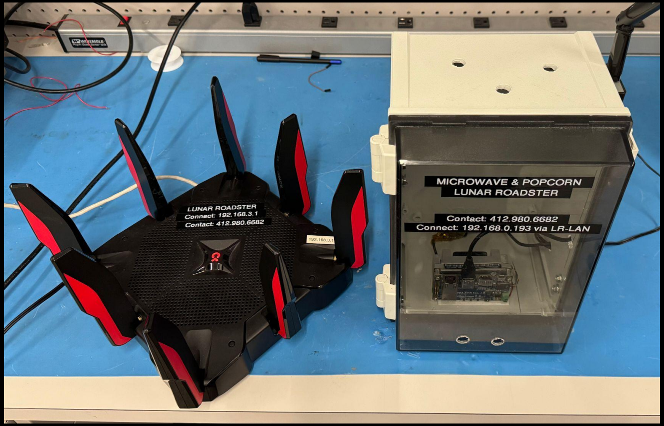
## Analysis:

- Computations take too long. Rover often completely stops for calculations
- Sometimes topic messages gets missed due to high network layer traffic

## Testing:

- Fall T09: CPU/GPU Usage of Autonomous Stack is Below Orin Compute Limits
- Fall T11: Fall Validation Demo Preparation Test

# **Description:** External Infrastructure Subsystem

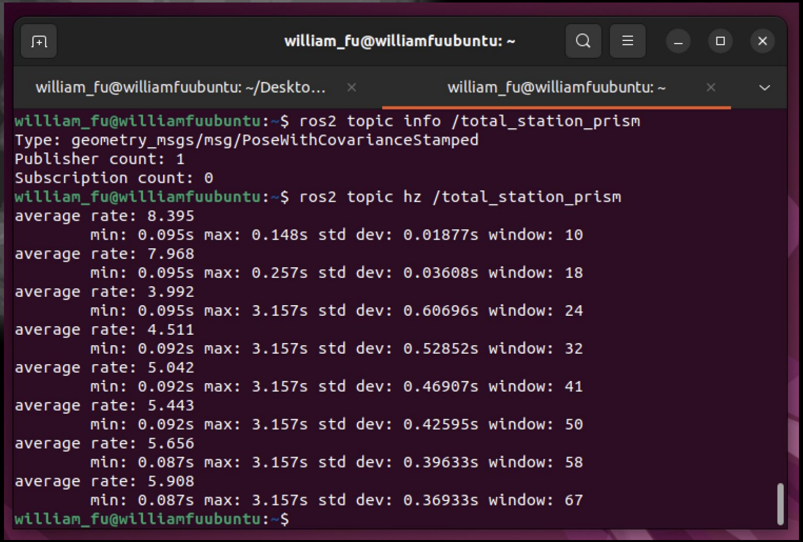




**Description:** Mission components deployed offboard the rover to support localization and communication

**Requirements:**
- Leica TS16 Total Station
- NVIDIA TX2 Relay Chip
- LAN Router
- Team laptop (operations terminal)

**Expected Functionality:**
- Accurately localize {X, Y, Z}-coordinates of the rover inside Moon Yard
- Establish two-way communication

# Status & Evaluation: External Infrastructure Subsystem




**Implementation:**

- Total Station sends rover coordinates to TX2 relay chip, forwards data packet via LAN network to rover
- Established static IP on LAN network so operations terminal can communicate via SSH

**Challenges:**

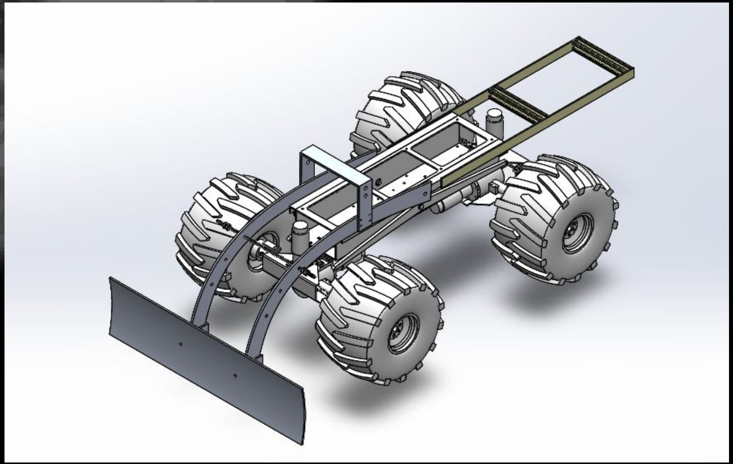
- Unable to obtain access to the TX2 relay chip login details
- Incorrect Jetson Docker network permissions blocked two-way communication

**Status:** 100% Completed

**Analysis:** Investigated network permission issues inside Jetson Docker

**Testing:** Spring T05: External Infrastructure Test

# Description: Mechanical Subsystem Dozer Assembly

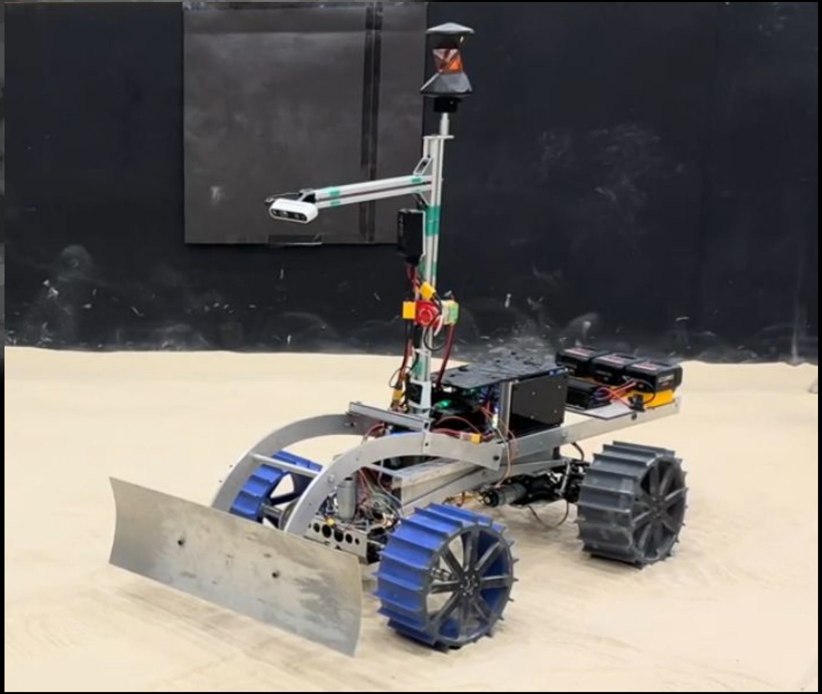


**Description:** Lunar terrain manipulation tool

**Requirements:**
- Dozer blade
- Dozer arms assembly
- Linear actuator
- Arduino Due

**Expected Functionality:**
- Perform dozing of sand
- Perform backblading
- Actuate automatically based on commands from tool planner

# Status: Mechanical Subsystem Dozer Assembly Unit



**Implementation:**

- Research: Dozer shapes and sizes, actuation methodologies.
- Designed the dozer assembly (blade, arms, yoke, mounts) using SolidWorks.
- Shortlisted linear actuators of different gear ratios.
- Manufactured and assembled all parts on the rover.
- Testing with different actuators

**Challenges:**

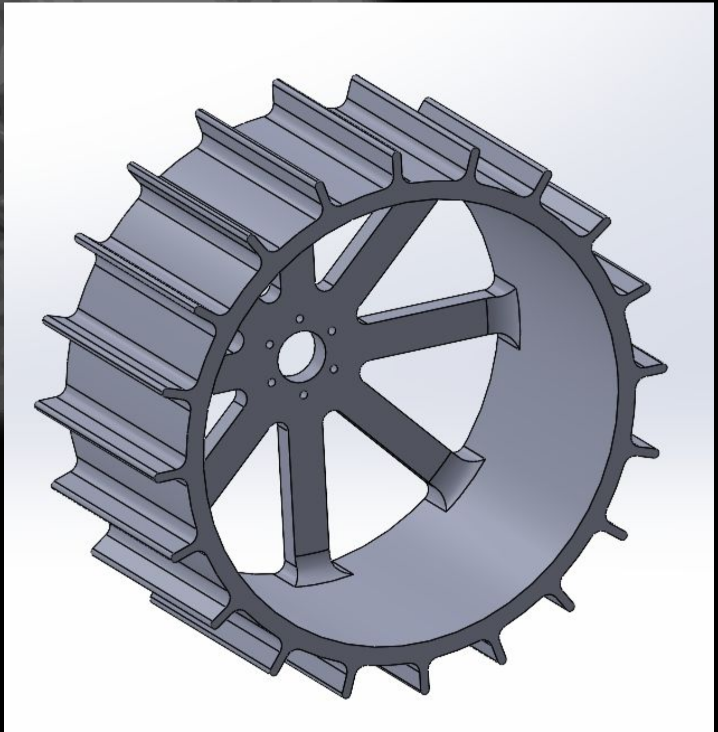
- Fabrication problems
- Limited access to FRC Workshop

**Status:** 100% Completed

# Evaluation: Mechanical Subsystem Dozer Assembly Unit

Excellent pushing and grading capability, owing to the shape and size of the blade, and the robust dozer arms

# Description: Wheel Assembly Unit



**Description:** Assembly that enables movement of the rover by acting as an interface between the drivetrain and ground.

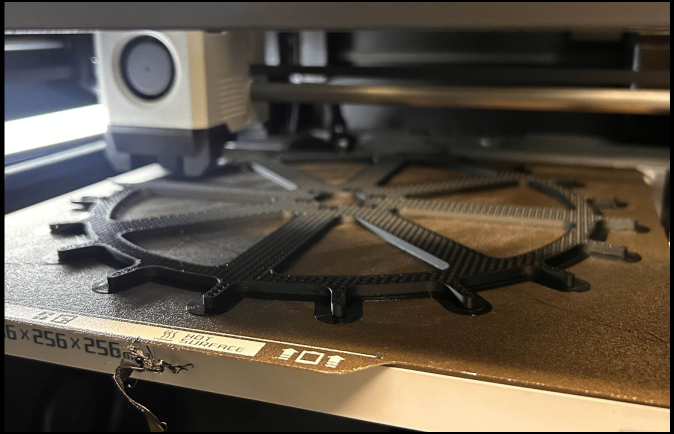
**Requirements:**

- Wheel (3D Printed - PLA)
- Mounting Assembly to the Suspension
- Drive Train (Differential and Steering)
- Motors

**Expected Functionality:**

- Provide required traction for movement and grading of sand
- Minimize wheel slip
- Allow for steering in sand
- Desirable - Use materials that can function on the Lunar Surface

# **Status:** Wheel Assembly Unit



**Implementation:**

- Design single-part iterations in SolidWorks
- 3D-print designs and test in the MoonYard
- Observe and re-design until the design is satisfactory

**Challenges:**

- 3D Printing the wheel is a long process with frequent failures
- Steering system of the rover is fragile and frequently disengages - solved.

**Status:** Completed

- Demonstrated mobility and pushing power with 4 printed wheels in SVD

# Evaluation: Wheel Assembly Unit

Great performance in traction and generating pushing power in all tests

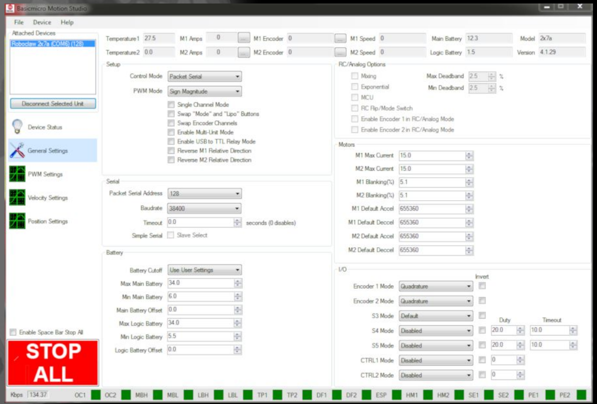
# Description: Actuation Subsystem

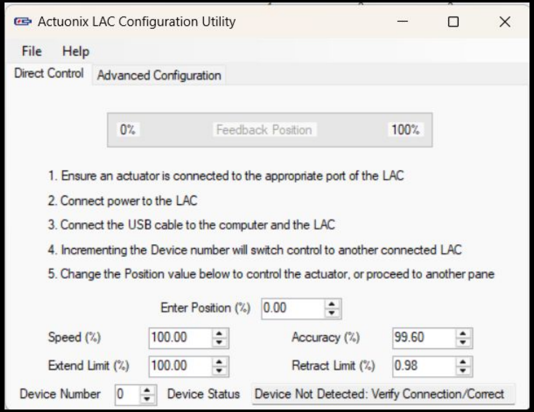

**Description:** Power transfer methodologies for rover mechanisms.

**Requirements:**
- DC Motors with Encoders (x4)
- Linear Actuator (with feedback)

**Expected Functionality:**
- Deliver power to wheels for mobility
- Steer the front and rear wheels
- Actuate the dozer assembly to facilitate mobility and dozing

# **Status:** Actuation Subsystem




## Implementation:

- Selected drive motors with higher torque for better traction and mobility
- Selected linear actuator for dozer assembly
- Interfaced drive motors and steering motors with Roboclaw motor controller
- Interfaced Actuator with Linear Actuator Controller Board and Arduino Due

## Challenges:

- Worn-out pinion gears
- Hard-to find spares (resolved)

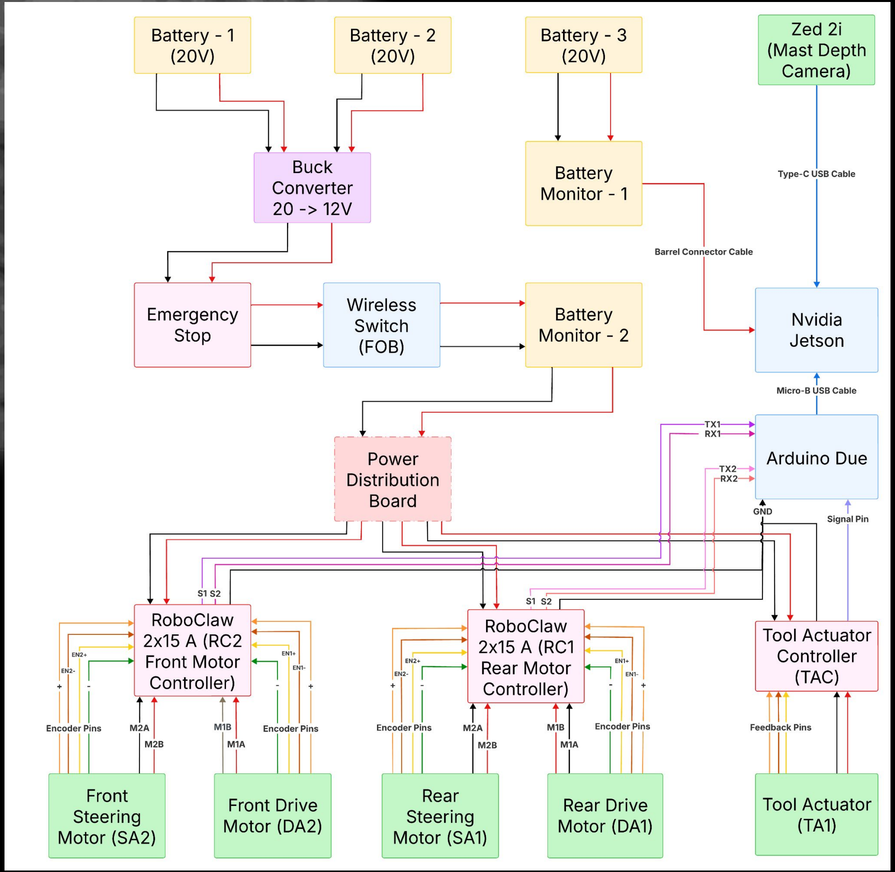
**Status:** 100% Complete

- Demonstrated enhanced steering and driving in SVD
- Linear actuator has been tuned by switching to analog control, eliminating oscillations

# Evaluation: Actuation Subsystem

New drive motors provide adequate motion in the Moonyard. Actuator has the capability to lift the rover, as demonstrated in SVD, oscillations have been resolved.

# Description: Electrical Subsystem




**Description:** Rover’s power and logic circuitry.

**Requirements:**

- Buck Converter (20 V -> 12 V)
- Power Distribution Board (PDB)
- RoboClaw Motor Controller - 2
- Linear Actuator Controller
- Linear Actuator
- Zed 2i Depth Camera
- E-Stop
- Wireless Switch
- IMU - VectorNav VN100
- Arduino Due
- Nvidia Jetson Xavier AGX
- Wireless Receiver for Joystick
- DC motors with Encoders

**Expected Functionality:**

- Distribute power efficiently to all rover components
- Ensure stable voltage levels for uninterrupted operation

# Status: Electrical Subsystem


**Implementation:**

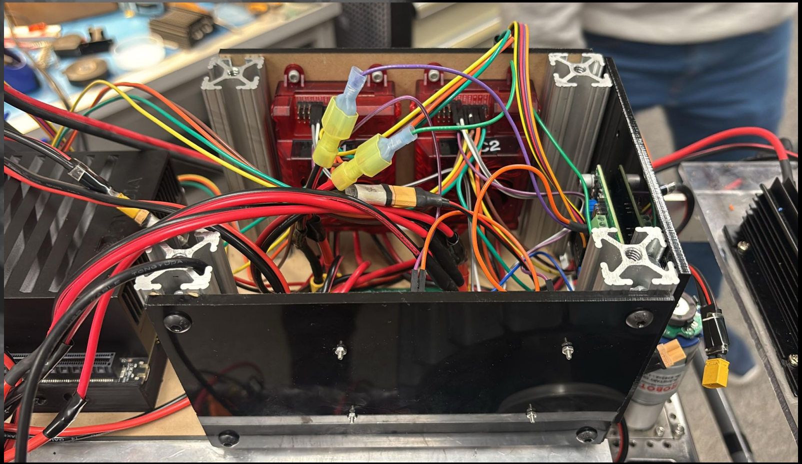
- Integrated new electrical components and designed a custom Power Distribution Board (PDB) based on updated power requirements
- Designed a compact and accessible electronic box to streamline the electrical subsystem setup
- Successfully integrated and tested the PDB within the rover’s electrical system
- Created robust connections by replacing jumper wires with screw terminals and lever wire connectors

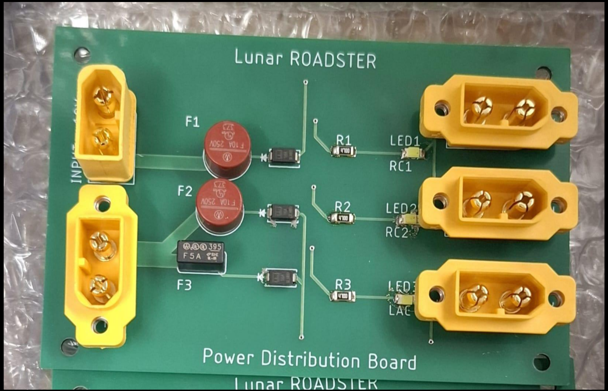
**Challenges:**

- Understanding and adapting to the existing circuit design and limitations.
- Ensuring the compact design of the electronic box while maintaining accessibility and proper cooling.

**Status:** 100% Completed

# **Evaluation:** Electrical Subsystem



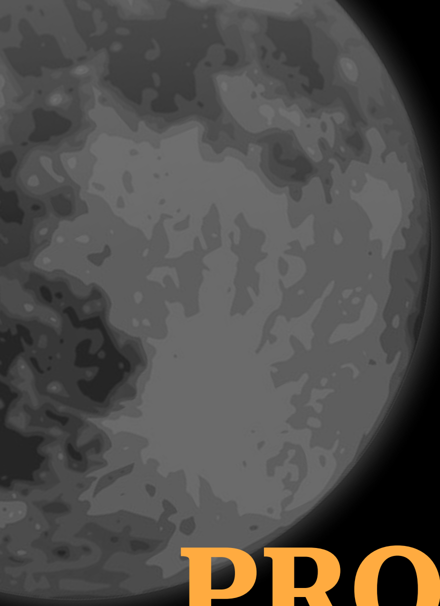



**Modeling:**
- Developed detailed circuit diagram integrating existing and new electrical components
- Modeled custom PDB with over-current, reverse-voltage, and power indication features
- Designed electronics box for minimal footprint, cooling, and accessibility

**Analysis:**
- Verified voltage and current demands for each subsystem component to ensure PDB output stability
- Assessed cable routing and hardware quality assurance

**Testing:**
- Fall T10: Maintenance, Reliability and Quality Assurance Test

# Video Excerpt



# PROJECT MANAGEMENT

# Schedule

**Major Milestones:**

- Implemented Objectives and Key Results (OKR) to track Progress Review Goals
- Achieved promised goals during PRs

**Current Status:**

- Achieved all promised goals for SVD
- On track with planned schedule
- Delays in planned Continuous Integration Continuous Testing approach due to hardware issues – made up for this in Fall break

**Plan to Improve:**

- Extreme testing until FVD to ensure hardware robustness
- Testing and tuning navigation, perception and planning stacks
- CICT

# Fall Test Plan

| Date | Event | Capability Milestones | Tests | Requirements |
|---|---|---|---|---|
| 09/10 | PR7 | • Hardware and software refinement | Validate hardware upgrades, software fixes, and system stability improvements | M.F.2<br>M.F.3<br>M.F.4<br>M.F.5<br>M.F.7<br>M.F.9 |
| 09/24 | PR8 | • Validation stack setup<br>• Navigation stack setup<br>• Obtain gradable craters location | Detect craters that meet grading criteria based on diameter and depth, and determine their poses | M.F.6 |
| 10/08 | PR9 | • Perception stack detects craters accurately and provides waypoints<br>• Validate grading<br>• Rover navigates to goal location accurately | Test navigation accuracy and perception stack geometry extraction | M.F.1<br>M.F.2<br>M.F.3<br>M.F.4<br>M.F.6<br>M.F.8<br>M.F.9 |

# Fall Test Plan

| Date | Event | Capability Milestones | Tests | Requirements |
|---|---|---|---|---|
| 10/29 | PR10 | • SkyCam-based localization for improved global positioning<br>• Ensure compute usage is below orin limits<br>• Tool Planner stack completed and integrated with necessary subsystems | Test SkyCam-based localization by checking rover's ability to self-localize accurately with / without external infrastructure and compute usage | M.F.3<br>M.F.7 |
| 11/12 | PR11 | • Full system integration<br>• Conduct quality assurance testing | Check all subsystems and units are functioning correctly | M.F.1 M.F.6<br>M.F.2 M.F.7<br>M.F.3 M.F.8<br>M.F.4 M.F.9<br>M.F.5 |
| 11/17<br>11/24 | PR12 (FVD and FVD 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.6<br>M.F.2 M.F.7<br>M.F.3 M.F.8<br>M.F.4 M.F.9<br>M.F.5 |

# Fall Validation Demonstration

| Test Location |
| --- |
| Planetary Robotics Lab Moon Yard |

| Sequence of Events |
| --- |

**Prior Setup:**

1. Prepare the Moon Yard with several craters and dunes in a circular path.
2. Perform a FARO scan of the environment and preprocess the scan to generate a map used for identifying gradable crater poses and for navigation planning.
3. Set up the external infrastructure by positioning the Leica total station at the corner of the Moon Yard, configuring the LAN router, and connecting the Jetson TX2 relay.
4. Position the rover in the Moon Yard and perform localization calibration.

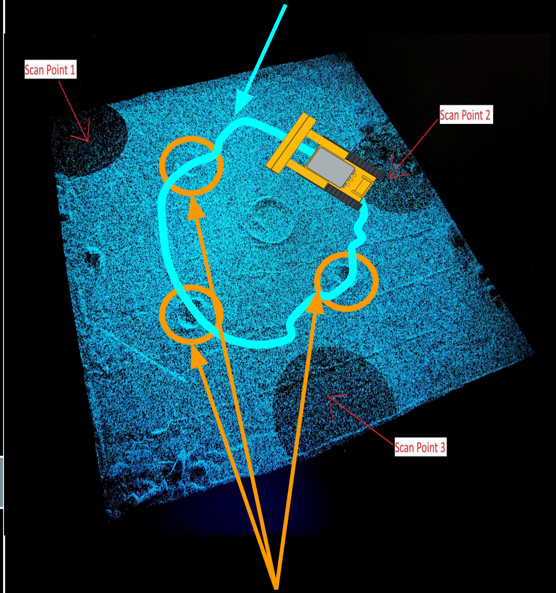
**During Demonstration:**

1. Switch the rover to autonomous mode and run the start-up procedure.
2. Observe the rover autonomous 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. Monitor the job status through the GUI, and use the emergency stop button if any unexpected behavior occurs.

| Quantitative Performance Metrics |
| --- |

**M.P.1:** Will plan a path with 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 have a contact pressure of less than 1.5 kPa
**M.P.4:** Will avoid craters >= 0.5 metres
**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°

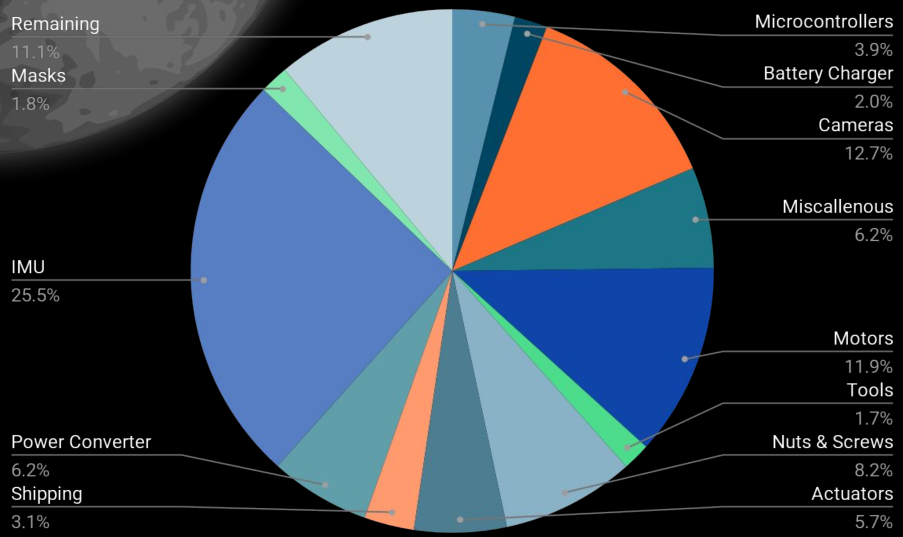
Follow a circular path




Groom several craters in a circular path

# Budget

| MRSD Budget | MRSD Budget Spent ($) | MRSD Budget Spent (%) | Total Budget Spent* | Remaining Balance |
|---|---|---|---|---|
| $5,000 | $4,437.09 | 88.7% | $7507.09 | **$562.91** |



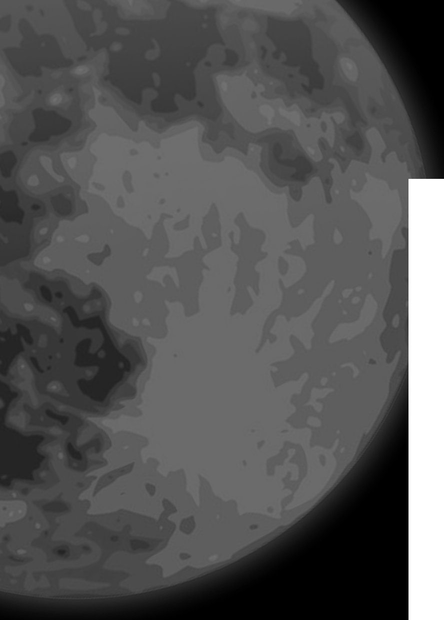

| Critical Parts | Extra Stock |
|---|---|
| Steer motor | 2 Units |
| Drive motor | 2 Units |
| Pinions | 8 Units |
| IMU | 2 Units |
| Chassis | 1 Disassembled |

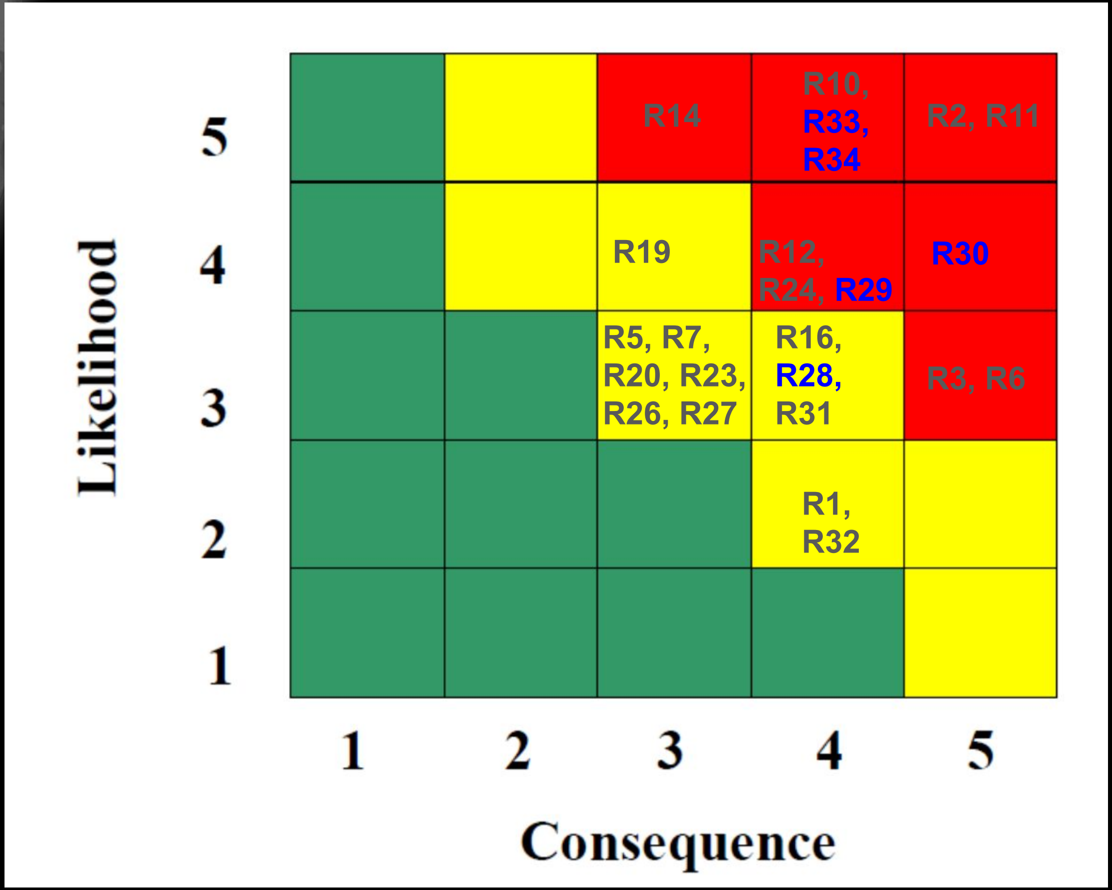
**Emergency Budget: Red**

* Includes $3,070 worth of items inherited from Crater Grader and Supervisor

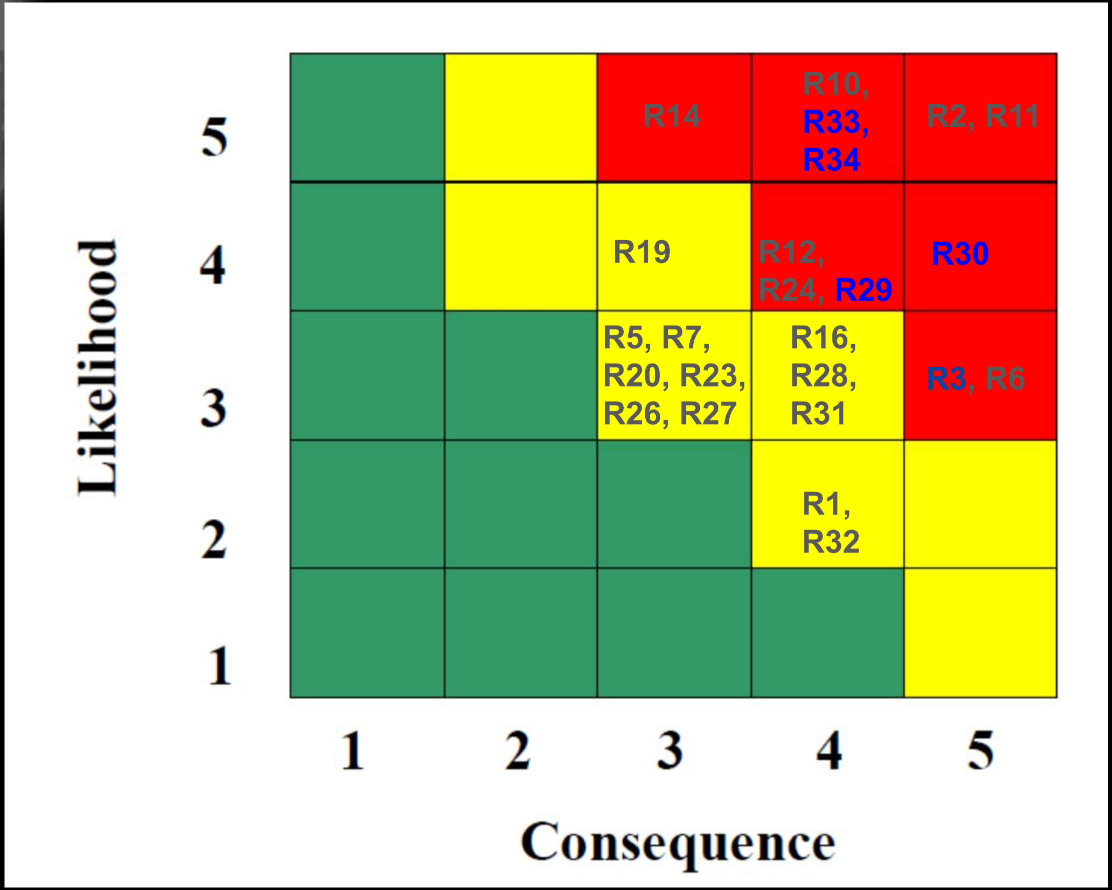
# Risk Management (Updated)

## Risk Summary



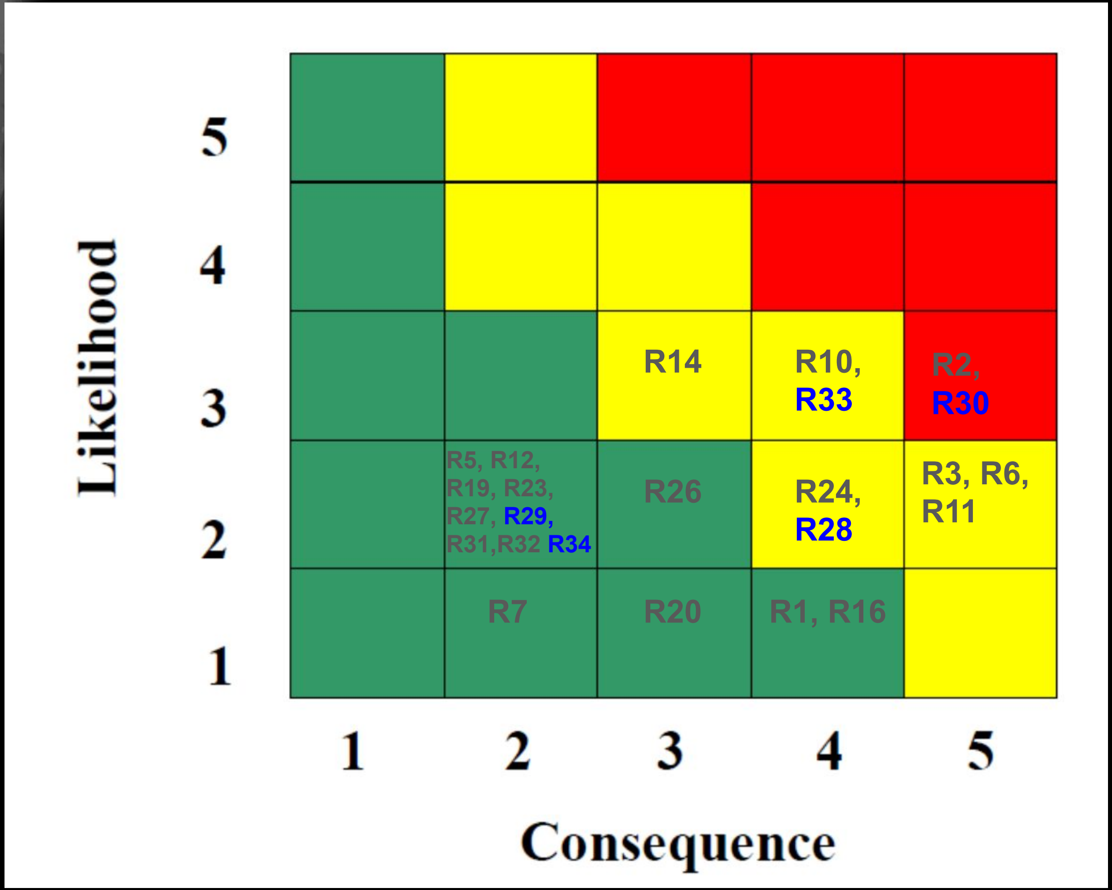


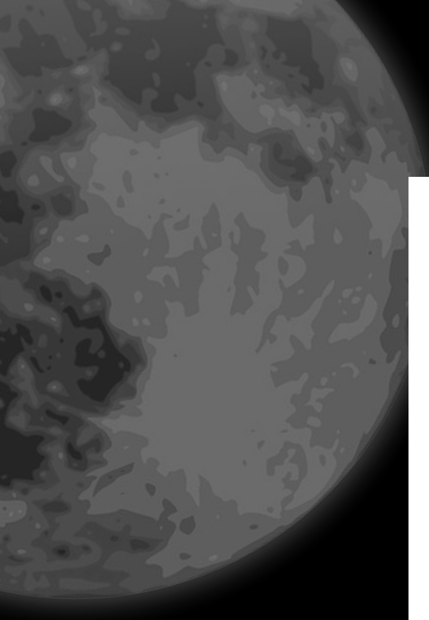
# Risk Management (Updated)
## Risk Summary



# Risk Management (Updated)

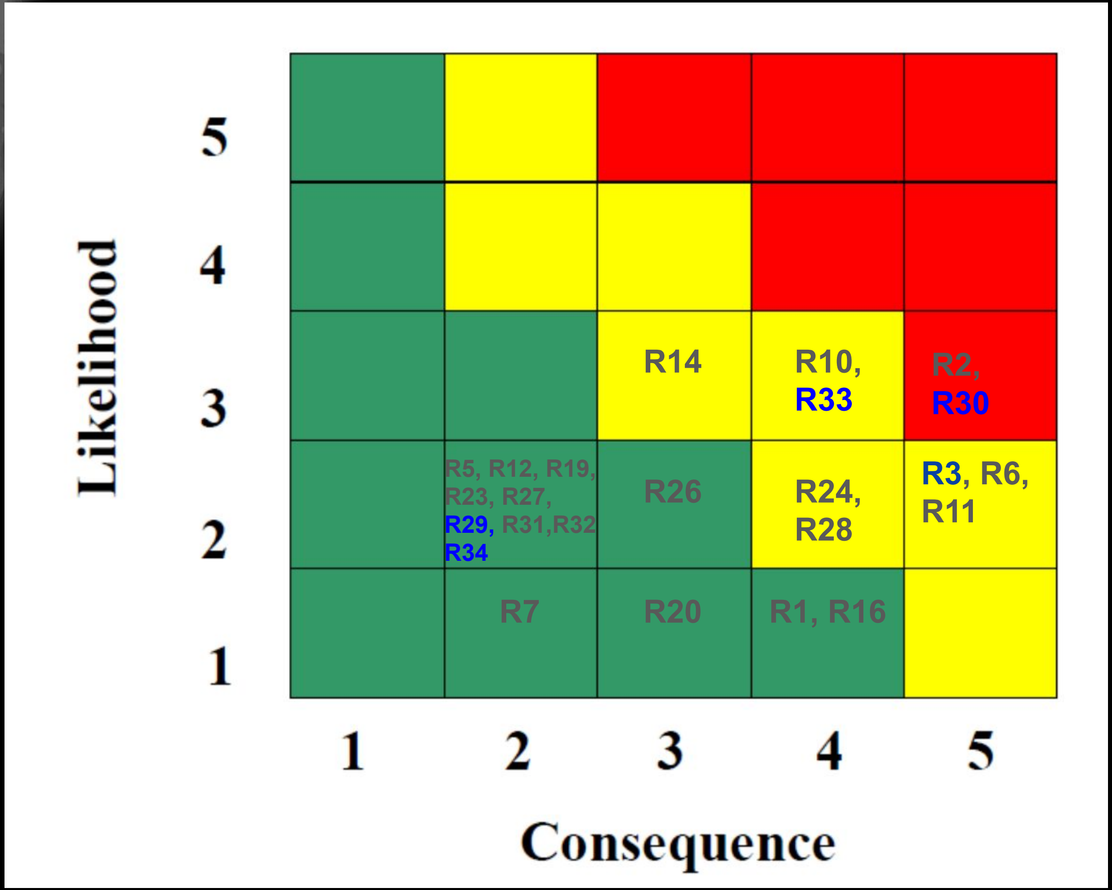
## Reduced Risk Summary





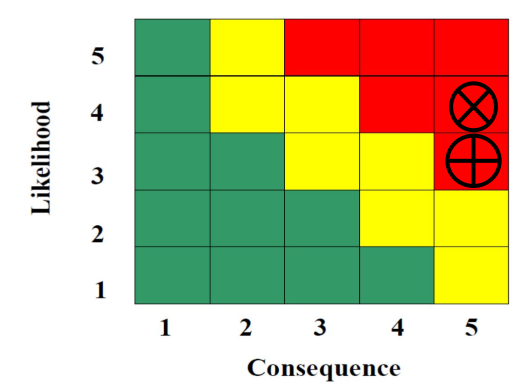
# Risk Management (Updated)

## Reduced Risk Summary



# Risk Management

| Risk ID | Risk Title | Risk Owner | Risk Type: | Logistics |
|---|---|---|---|---|
| R30 | No spares available | Team | | |
| **Description** | | **Date Added** | | |
| Discontinued model, spare parts unavailable | | 3/4/2025 | | |
| | | **Date Updated** | | |
| | | 9/22/2025 | | |
| **Consequence** | | | | |
| The whole project falling through, or redo almost all subsystems on a different rover. | | | | |



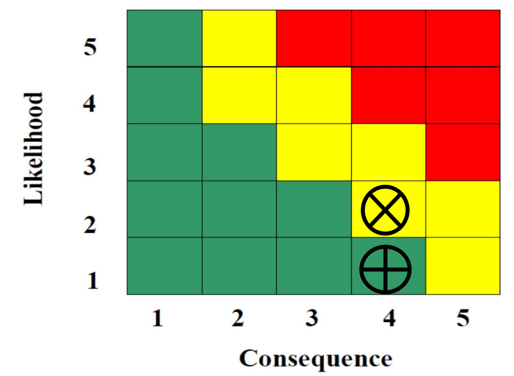

| Action/Milestone | Success Criteria | Date Planned | Date Implemented |
|---|---|---|---|
| Check out eBay and other similar platforms for spares | Successfully find exact spares on these platforms | 3/6/2025 | 9/22/2025 |
| Check out and stock similar parts if not same | Successfully find and stock similar parts | 3/6/2025 | 9/22/2025 |
| 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 |
| Find similar parts - a slightly smaller pinion and motor set | Spares problem will be solved | 9/10/2025 | 9/22/2025 |

# Risk Management

| Risk ID | Risk Title | Risk Owner | Risk Type: | Logistics |
|---|---|---|---|---|
| R36 | PRL Moonyard Access | William | | |

| Description | Date Added |
|---|---|
| Securing Moonyard access for testing/demos will be restricted and challenging | 8/29/2025 |
| | **Date Updated** |
| | 9/30/2025 |

| Consequence |
|---|
| No testbed available for testing and/or FVD |




| Action/Milestone | Success Criteria | Date Planned | Date Implemented |
|---|---|---|---|
| Devise and discuss a testing and demo plan with Prof. Red and Prof. David Wettergreen beforehand and reserve slots | Successfully meet and discuss the schedule of high priority projects | 9/11/2025 | 9/11/2025 |
| Complete Medical Evaluation to get unrestricted but controlled access | Successfully complete the Medical Evaluation and get unrestricted access to the Moonyard | 9/5/2025 | 9/11/2025 |
| Respirator Training | Complete training and get custom masks | 9/30/2025 | 9/30/2025 |

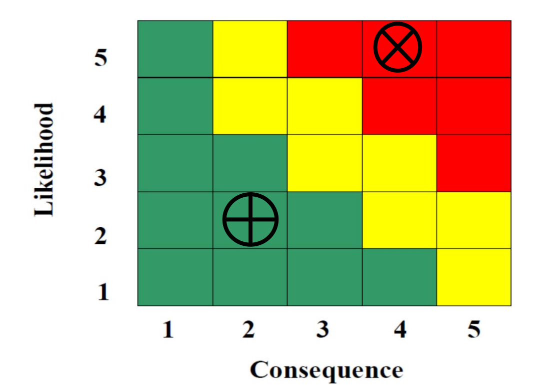
# Risk Management

| Risk ID | Risk Title | Risk Owner | Risk Type: | Technical |
|---|---|---|---|---|
| R34 | Arduino requires reset before operation | Bhaswanth | | |

| Description | Date Added |
|---|---|
| Arduino needs to be manually reset each time before starting autonomy or switching between autonomy and teleoperation modes. | 3/4/2025 |
| | **Date Updated** |
| | 9/7/2025 |

**Consequence**

Slows down setup time and impacts operational readiness, delaying mission start and mode transitions.



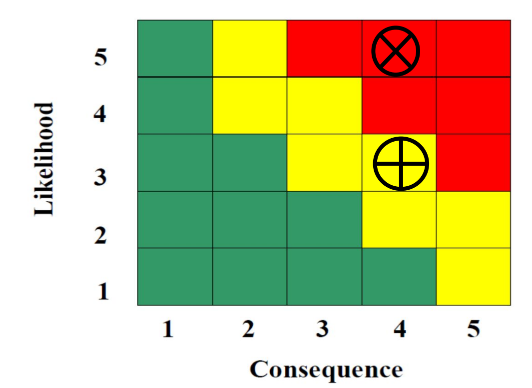

| Action/Milestone | Success Criteria | Date Planned | Date Implemented |
|---|---|---|---|
| Check USB port permissions and drivers issues on Jetson | Successfully establish consistent serial connection without reset | 4/26/2025 | 9/5/2025 |
| Verify that Arduino is connected via USB 3.0 instead of USB 2.0 port | Ensure stable high-speed communication | 4/26/2025 | 9/5/2025 |
| Check for ROS node frequency mismatches causing packet loss to Arduino | Match ROS publish/subscribe rates | 4/26/2025 | 9/5/2025 |
| Implement a software reset trigger | Reset can be called from the operations terminal | 9/7/2025 | |

# Top Risks

| Risk ID | Risk Title | Risk Owner | Risk Type: | Technical |
|---|---|---|---|---|
| R33 | Localization frame shift after total station battery swap | Bhaswanth | | |

| Description | Date Added |
|---|---|
| Battery replacement in the total station causes small frame offsets, leading to localization inaccuracies. | 3/4/2025 |
| | **Date Updated** |
| | 9/12/2025 |

| Consequence |
|---|
| Leads to poor navigation performance and risk of missing the crater during grading operations. |



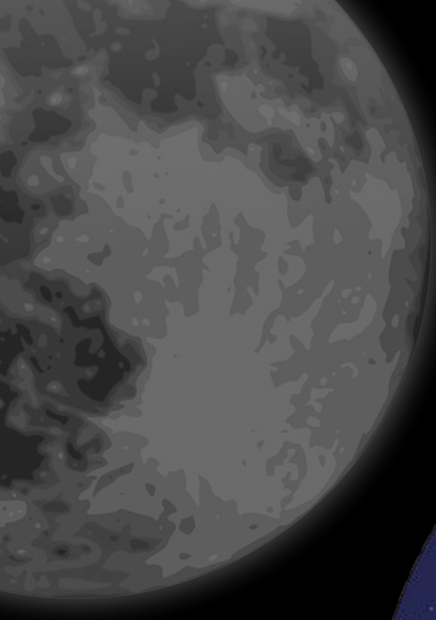

| Action/Milestone | Success Criteria | Date Planned | Date Implemented |
|---|---|---|---|
| Implement resection method using three known prism locations instead of orientate-to-line | Successfully fix the frame consistently after battery swaps | 4/26/2025 | 9/12/2025 |
| Explore and test alternative localization methods (using SkyCam) | Successfully maintain localization accuracy | 4/26/2025 | |

# Lessons Learned

- Check for common issues online while choosing hardware (Jetson and Zed issue)
- Schedule based on resource availability such as lab access and lead times
- Plan for Progress Review goals at the beginning of the semester
- Have a proper plan for demo split between SVD and SVD Encore
- Do not choose a project where you have to build both software and hardware from scratch :)

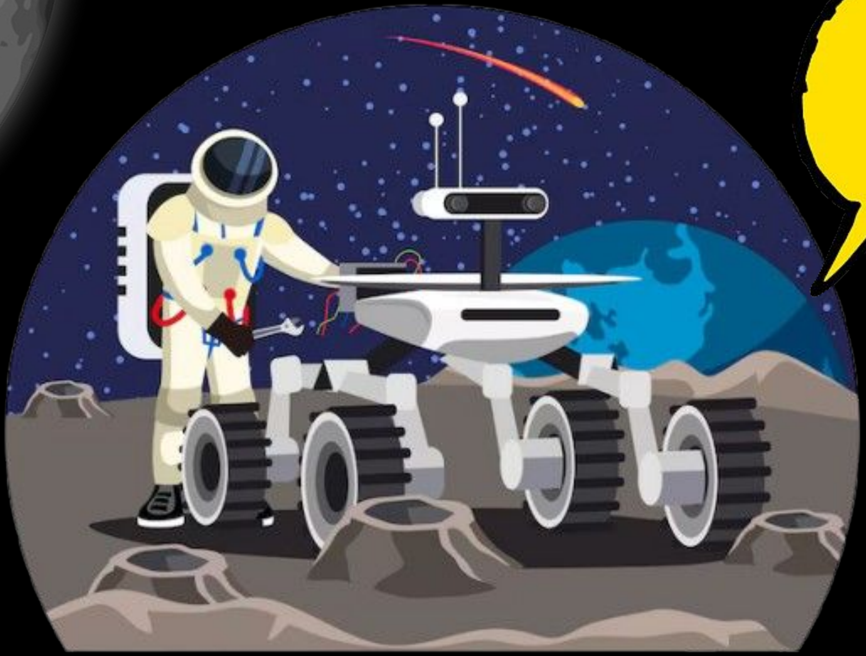
# Fall Activities

- Integrate and test continuously
- Improve tool actuator
- Wheel torque feedback
- Tune overall software stack
- Solution to compute problem (new Jetson/code optimization)
- New localization method - SkyCam
- Planning for grooming multiple craters



# Colonize the Moon!

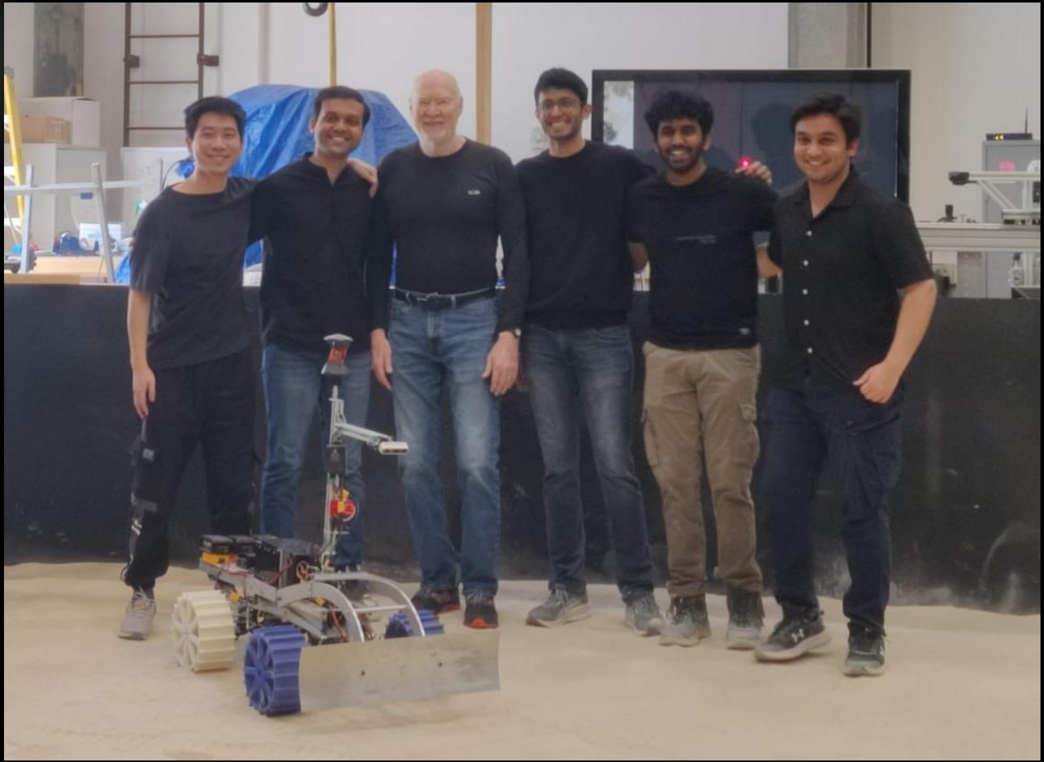
## - *Team Lunar ROADSTER*



**Any Questions?** 🤖🦾🚀

*"Starting with a foothold on the Moon, we pave the way to the cosmos"*

# Thank You!



**https://mrsdprojects.ri.cmu.edu/2025teami/**


Pose 3 - Sink
2x Speed

# Appendices

A.1. Derivation for M.P.5:

- Chang'e-4's landing site was surveyed and found that 97.5% of nearby craters were below 15.5 meters in diameter.
- Our rover is approximately 1/30th the size of a commercial grader, so it shall be able to grade $15.5/30 \approx 0.5$ meter craters at least.
- Source: DOI 10.3390/rs14153608

A.2. Derivation for M.P.3:

- Average depth-to-diameter (DtoD) ratio of 0.07 near the North pole
- Assuming worst-case scenario of a crater with twice DtoD ratio of 0.14, the gradient is $\theta = \arctan(0.14*2) \approx 15$ degrees
- Contact pressure requirement follows recommendation from NASA
- Source: DOI 10.1029/2022GL100886, NASA/TP—2006–214605

A.3. Derivation for M.P.1:

- Recommendation from Nature paper on extraterrestrial path-planning metrics
- Source: DOI 10.1038/s41598-023-49144-8

Credits for images:

- Generative AI
- Google Images
- Dr. William Red Whittaker's slides

# Risk Management (Extra)

| Risk ID | Risk Title | Risk Owner | Type | Description | Consequence | Risk Reduction Plan |
|---|---|---|---|---|---|---|
| R1 | PRL Testbed Scheduling | Ankit | Scheduling | PRL Testbed unavailable due to scheduling conflicts with other high priority projects | No testbed available for testing and/or SVD | Devise and discuss a testing and demo plan with Red and other stakeholders of the PRL testbed beforehand and reserve slots<br>Reach out to external testing facilities like Astrobotic or CAT for a backup testing facility<br>Schedule tests at night |
| R2 | Excavator and grader tool planner takes longer than expected to deliver | Simson | Technical | Integration of the excavator and grader software with hardware takes longer than expected | Unable to meet SVD deadline and potential requirements change | Shift requirements for SVD<br>Integrate the grader during Fall semester<br>Potentially use off-the-shelf code if available, preferably from CraterGrader |
| R3 | Integration issues between subsystems | Deepam | Technical | Subsystems work individually, but integration and communication between the subsystems are flawed | Delay in integration causing scheduling overruns, requirements change and failure of the demo | Perform unit testing and subsystem validation continuously<br>Integrate one subsystem at a time<br>Use a common framework (e.g. ROS2 interfaces) for communication between subsystems to reduce bugs<br>Keep to planned schedule and have at least 5 weeks for testing and integration |
| R4 | Belly depth sensor is not suitable for validation | Bhaswanth | Technical | 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 | Will result in major revision and changes to the validation architecture and functional requirement, causing delays in scheduling | Mount the depth camera at another location on the rover (e.g. on a mast)<br>Use another sensor to determine depth variations (e.g. LIDAR, visual odometry, IR sensor)<br>If all else fails, use the total station for validation |

# Risk Management (Extra)

| Risk ID | Risk Title | Risk Owner | Type | Description | Consequence | Risk Reduction Plan |
| --- | --- | --- | --- | --- | --- | --- |
| R5 | Unable to get Crater Grader to perform autonomous crater filling | Bhaswanth | Technical | Our rover builds on top of the work accomplished by Crater Grader. If we cannot get Crater Grader to perform autonomous crater filling, we may need to spend more time working on the navigation stack and designing the entire pipeline | Extra time commitment to start from scratch or obtaining a suitable replacement | Thoroughly go through Crater Grader's code and the mechanical schematics provided<br>Test each component and wiring to see if they are working<br>If it is still not working, inherit only the software component from Crater Grader and build hardware ourselves |
| R6 | Delay in arrival and manufacture of hardware components | William | Schedule | Shipping delays of components ordered and/or manufacturing delays on custom made components | Delays in hardware integration, causing pushbacks in scheduling and software development | Use off-the-shelf components that are available on hand (e.g. from CMU labs or Red's workshop)<br>Start ordering and designing components during Winter break so there is adequate leeway for delivery and manufacturing before Spring semester starts<br>Use simulations to work on software components while we wait for the components to be delivered and/or manufactured<br>Implement other subsystems that are independent from the subsystem that is missing parts<br>In case of delay in wheels, work with the existing wheels and proceed with the timeline while waiting for the new ones to arrive. |
| R7 | Lack of proper simulation environment | Simson | Technical | Inability to accurately simulate the rover in a Lunar-like environment can lead to suboptimal performance | The rover's performance in the Moon Pit may be compromised, leading to inefficiencies, mission delays, or potential failure in achieving key objectives | Ask CraterGrader how they ran all their simulations and gather resources<br>Explore LunarSim - https://github.com/PUTvision/LunarSim and check how useful this will be, during the winter break<br>Develop Gazebo environment |

# Risk Management (Extra)

| Risk ID | Risk Title | Risk Owner | Type | Description | Consequence | Risk Reduction Plan |
|---|---|---|---|---|---|---|
| R10 | Mast depth camera FOV is blocked | William | Technical | 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. | Hinders the rover's ability to perceive its surroundings accurately, resulting in navigation errors and inefficiencies in excavation tasks | 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 |
| R11 | Too many performance requirements | Ankit | Technical, Schedule | We have a lot of performance requirements and we may not be able to meet all of them by April for SVD | Delays in testing and validation, impacting project timelines and April SVD Demo results | Have revised performance requirements separately for SVD and FVD (focus more on SVD)<br>Talk to CraterGrader and discuss what is feasible and what is not in the given time<br>PM should track schedule properly and team members have to push to meet the timeline |
| R12 | Drive system wear-and-tear causes malfunction | Deepam | Technical | The transmission and steering assembly might be worn out, leading to suboptimal vehicle dynamics, and potentially mechanical failure | Rover drive system fails and may require a lot of repair and maintenance | Thoroughly check the Crater Grader's assembly and carry out maintenance of any worn-out parts<br>Completely replace the assembly parts with the same/similar new parts for better performance and reliability<br>Added limit switches to avoid steering gears to operate beyond their limits |
| R14 | Dust ingress | William | Technical, Cost | 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 | Component failure during testing or demonstrations. Highly inhibits all future scheduled tasks | Design proper sand enclosures and mounts for sensitive components<br>Review placement of components<br>Review scale and speed of sand manipulation to eliminate root-cause of flying sand/dust<br>Allocate contingency budget and order spares of the sensitive components in case of component failure |

# Risk Management (Extra)

| Risk ID | Risk Title | Risk Owner | Type | Description | Consequence | Risk Reduction Plan |
|---|---|---|---|---|---|---|
| R16 | Code version control | Simson | Technical | 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. | Delay in code integration and implementation | Implement GitHub version control to store and retrieve the best versions of code and configuration<br>Use Google Drive to backup important documentation explaining setup processes |
| R19 | Items missing | Ankit | Logistics | Critical project items may go missing if not stored properly or tracked. Items may be misplaced or borrowed without proper logging | Delay in hardware implementation | Maintain an inventory tracking spreadsheet<br>Include spare inventory |
| R20 | Sensor ROS packages not available | William | Technical, Schedule | Finalized sensors might lack compatible ROS packages, leading to delays or significant changes in the software architecture | Delay in software implementation | Perform trade studies to pick sensors that are compatible with ROS versions before finalizing<br>Select sensors and ROS versions that minimize potential conflicts |
| R23 | Lunar-accurate cut/fill regions are not possible to groom | Simson | Technical | 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 | 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 | Accurately create the environment and assess if the rims are enough to fill<br>If not, modify PRs accordingly |

# Risk Management (Extra)

| Risk ID | Risk Title | Risk Owner | Type | Description | Consequence | Risk Reduction Plan |
|---|---|---|---|---|---|---|
| R24 | Sensor data is too noisy to fulfill performance requirements | William | Technical | Performance requirements are tough and ambitious, sensor noise may prevent us from achieving it | Failure to demonstrate performance requirements may cause us to lose marks in the demonstrations | Relax the performance requirements enough to ensure that they are achievable |
| | | | | | | Ensure enough testing time to tune parameters |
| R26 | Off-the-shelf wheels don't interface with the rover | Ankit | Technical | No off-the-shelf wheels fit the rover, We'll have to redesign wheel hubs and mountings as per the new wheels. | Continue with sub-optimal wheels that the rover currently has, thus, not meeting one of the non-functional requirements | Shift requirements to FVD |
| | | | | | | Good enough market research to see find the best fit, with least amount of changes |

# Risk Management (Extra)

| | | | | | | |
|---|---|---|---|---|---|---|
| R27 | TX2 Integration | William | Technical | Unable to login to TX2 and interface with a LAN network for transmitting data over WiFi ro Jetson | Delay in finalizing localization stack | Set up a new TX2 (Re-flash the TX2).<br>Reach out to previous teams to understand their methodology and retrieve credentials |
| R28 | Electrical hardware finalization | Ankit | Technical | E-box Design dependence on to-be manufactured PDB. | Not meeting the hardware deadline | Use previous knowledge and account for a placeholder in the design |
| R29 | Access to FRC Workshop | Deepam | Logistics | Without access, no hardware fabrication/repairs can be carried out in the absence of Tim | Not meeting the hardware deadline | Try other fab-labs on campus.<br>Request Tim, John or Red for getting temporary access, if not permanent |
| R30 | No spares available | Team | Logistics | Discontinued model, spare parts unavailable | The whole project falling through, or redo almost all subsystems on a different rover. | Check out eBay and other similar platforms for spares<br>Check out and stock similar parts if not same<br>Find a twin rover that was used by a previous team on campus<br>Maintain all parts, especially mechanical parts |