



# Lunar **ROADSTER**

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

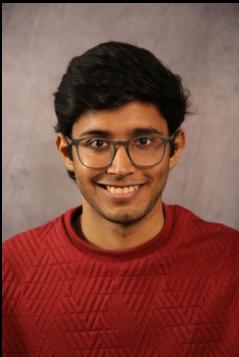
*“Starting with a foothold on the Moon, we pave the way to the cosmos”*



# The Team



Ankit Aggarwal



Deepam Ameria



Bhaswanth Ayapilla



Simson D'Souza



Boxiang (William) Fu



Dr. William "Red" Whittaker

# Goal: Resolve Arduino connection & reset issue

## In Progress:

- Implementing a software-based auto-reset feature
- Using GPIO Pins on Jetson and Digital Pins on Arduino

## Software Procedure:

- Arduino (micro-ROS): Publishes /arduino\_heartbeat
- Jetson (ROS 2 C++ node):
  - Subscribes to /arduino\_heartbeat
  - On timeout (> 5 s): toggle GPIO via **libgpiod**
  - Pulse ~200 ms → reset Arduino

# Goal: Implement new total station resection method

**Resection:** Uses three or more known reference points to solve both position and orientation of the total station (more accurate)

**Key Motivation:** Swapping batteries will not lead to total station offsets anymore

**Validation:** A fixed target measured from different total station setups yields consistent coordinates



# Goal: Finalize validation stack code methodology

**Step 1:** Obtain point cloud from ZED camera FOV

**Step 2:** Use RANSAC to fit a plane

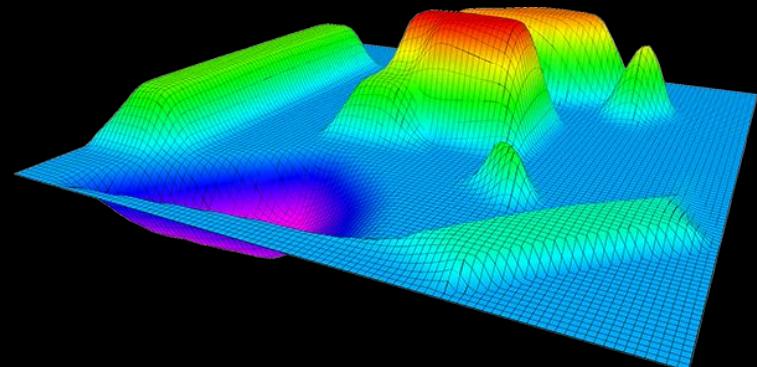
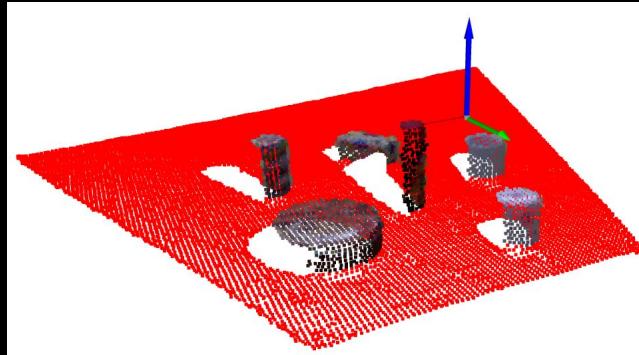
**Step 3:** Discretize map into grids and bin points in each cell

**Step 4:** Compute an elevation map relative to the RANSAC plane

**Step 5:** Compute gradients using finite differences

**Step 6:** Remove excessive gradients (walls)

**Step 7:** Compute maximum gradient





# **Goal: Finalize validation stack code methodology (PERCEPTION)**

- **Device Used:** ZED 2i RGBD Stereo Camera
- **Methodology:** Train a Deep Learning Model (YOLO) on a custom Crater dataset to determine geometric data: centroid, diameter, distance to crater, etc.
- **Data collection** using ZED 2i and annotation to be done manually
- **Output:** Crater dimensions and position in robot frame which will be used by the planning stack to get robot poses

# Goal: Finalize global path planner methodology

## Inputs:

- Occupancy grid with crater classification
- Reference circular path (ring)
- Obstacles are large sized craters ( $>0.5m$ )
- Costmap with Ring Bias: adds extra cost for leaving the ring corridor

## Planner:

- Hybrid-A\* (lattice) search with Ackermann primitives
- Forward-only Dubins arcs (for minimum turn radius)
- g-cost: arc length + curvature penalty + obstacle cost + deviation from ring
- Heuristic: Dubins distance to goal + small penalty for ring misalignment

## Directionality:

- Only Counter Clockwise motion
- Reverse moves allowed but heavily penalized

## Outputs:

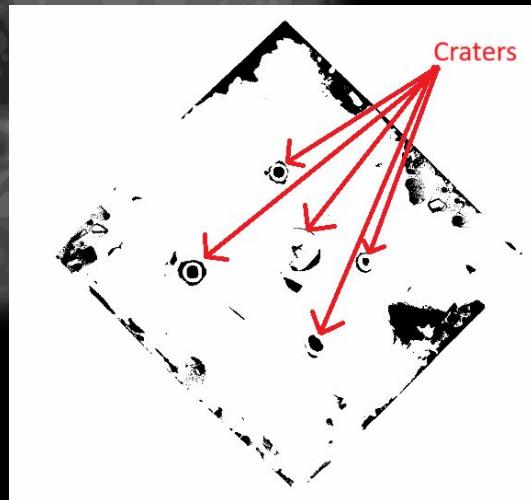
- Collision-free, curvature-feasible path hugging the latitude
- Path naturally passes through crater centers on the ring
- Smooth detours around large craters, rejoining the ring

# Goal: Finalize global navigation controller methodology

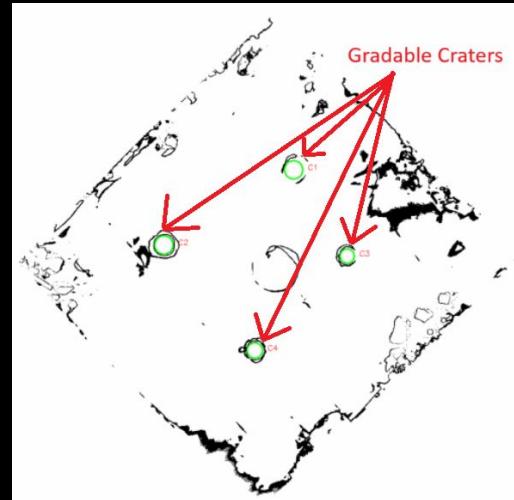
## Pure Pursuit Controller Workflow

1. Receive **global path** (list of waypoints)
2. Get **current robot pose** from localization
3. Select **lookahead point** at distance  $L$  along path
4. **Transform** lookahead point to robot frame
5. Compute **curvature and steering angle**
6. Send **steering & velocity commands** to actuators
7. Repeat steps 2 - 6 until goal is reached

# Goal: Finish implementing selection of gradable craters from global map generated from FARO scanner



Global Cost Map



Identified gradable craters based on diameter and depth,  
extracting their coordinates which will be used for navigation

Centroids of Gradable Craters (World Coordinates):  
Crater 1: X = 10.298 m, Y = 3.360 m  
Crater 2: X = 7.726 m, Y = 4.857 m  
Crater 3: X = 11.359 m, Y = 5.070 m  
Crater 4: X = 9.526 m, Y = 6.944 m

# Risk Management

Risk ID	Risk Title	Risk Owner	Risk Type:	Logistics
R30	No spares available	Team	5	
Description	Date Added	4		
Discontinued model, spare parts unavailable	3/4/2025	3		
Consequence	Date Updated	2		
The whole project falling through, or redo almost all subsystems on a different rover.	8/30/2025	1		
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	Likelihood	
Securing Moonyard access for testing/demos will be restricted and challenging		8/29/2025	5	
Consequence		Date Updated	4	
		8/29/2025	3	
			2	✗
			1	⊕
			1	2
			3	4
			4	5
			Consequence	
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		

# Risk Management

Risk ID	Risk Title	Risk Owner	Risk Type:	Technical
R34	Arduino requires reset before operation	Bhaswanth		
<b>Description</b>		<b>Date Added</b>		
Arduino needs to be manually reset each time before starting autonomy or switching between autonomy and teleoperation modes.		3/4/2025		
		<b>Date Updated</b>		
		4/10/2025		
<b>Consequence</b>			Likelihood	
Slows down setup time and impacts operational readiness, delaying mission start and mode transitions.			5	
			4	
			3	
			2	⊕
			1	
				Consequence
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		

# Issues Log

I12	08/29/2025	09/12/2025	<b>Bhaswanth Ayapilla</b>	Localization frame shift after total station battery swap	1. Implement resection method using three known prism locations instead of orientate-to-line 2. Explore and test alternative localization methods (using SkyCam)	Learned and implemented resection method for localization using total station	Quick solution to implement first before exploring alternatives
I13	09/08/2025	09/14/2025	<b>Team</b>	Compute unit (Orin & Jetson) unable to communicate with Arduino	1. Replace old Arduino with new one 2. Find code workarounds to force communication 3. Retrace wiring to make sure everything is wired correctly	MicroROS versions between Orin and Arduino were incompatible. Reinstalled MicroROS resolved the problem	This allows Orin to communicate with the Arduino, which is used to send motor commands
I14	09/14/2025		<b>Team</b>	Steer pinion tooth chipped and worn-out due to wear-and-tear. Unable to find exact replacement for the pinion	1. Replace with similar pinion that has different tooth count 2. Switch to using another chassis		
I15	09/14/2025		<b>Ankit Aggarwal Deepam Ameria Simson D'Souza</b>	Wires keep on coming loose during operations due to bad soldering	1. Re-solder every wire 2. Switch to plug connectors and buy adaptors for the RoboClaws and motors		

# Future Work

- Tuned validation stack - perception and verification separately
- Local navigation controller ready
- Navigation stack tested and tuned
- Methodology for integration
- Hardware finalized



# THANKS!

*Team Lunar ROADSTER*

