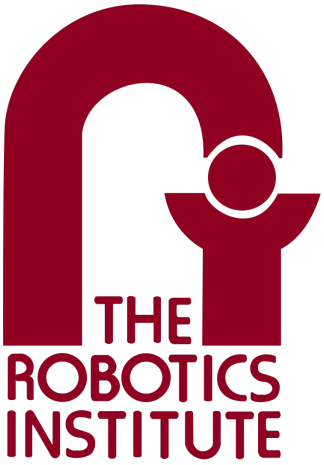
Individual Lab Report - ILR07



Lunar ROADSTER

Team I

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

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

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

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

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

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

September 25, 2025

Carnegie Mellon University
Robotics Institute

Contents

1 Individual Progress

In this phase of the project, I have worked on the following tasks:

1.1 Methodology for Perception

This phase, I worked with Ankit to finalize the methodlogy for the Perception stack of our system. We decided on the following methodology:

Step 1: Camera Connection: Connect the ZED 2i stereo camera to the NVIDIA Jetson Orin through a USB 3.0 interface.

Step 2: Activation of ROS Topics: Launch the ZED ROS/ROS2 wrapper to publish all necessary topics, including:

- Rectified left and right images
- Depth map
- Point cloud
- Camera information and transforms

Step 3: YOLO Inference: Deploy a custom-trained YOLO model to perform real-time inference on the live ZED 2i camera feed.

Step 4: Crater Detection: Use the inference results to identify bounding boxes corresponding to craters in the captured scene.

Step 5: Depth Cropping: Crop the depth points from the ZED 2i point cloud to match the bounding box of each detected crater.

Step 6: Geometric Estimation: Using the `PCL` and `OpenCV` libraries:

- Estimate the diameter of the crater by fitting a circle to the cropped point cloud.
- Compute the center position of the crater relative to the camera frame.

Step 7: Frame Transformation: Transform the crater measurements (diameter and center coordinates) from the camera frame into the world frame using the available TF tree.

Step 8: Distance Computation: Calculate the Euclidean distance between the robot base frame and the crater center position.

Step 9: Navigation Integration: Provide this distance information as an input to the navigation stack, enabling the robot to plan and execute movements relative to the detected crater.

1.2 Data collection and preparation for training model

As mentioned above, we decided to use a YOLO for detecting craters using the ZED 2i Camera. For training the model, I worked on data collection. I used a phone camera to collect pictures of different sized craters from different angles and under varying light conditions. I then used an annotation tool, Roboflow, to annotate the image with labels for efficient training with YOLO. (Fig. 1 and Fig. 2)

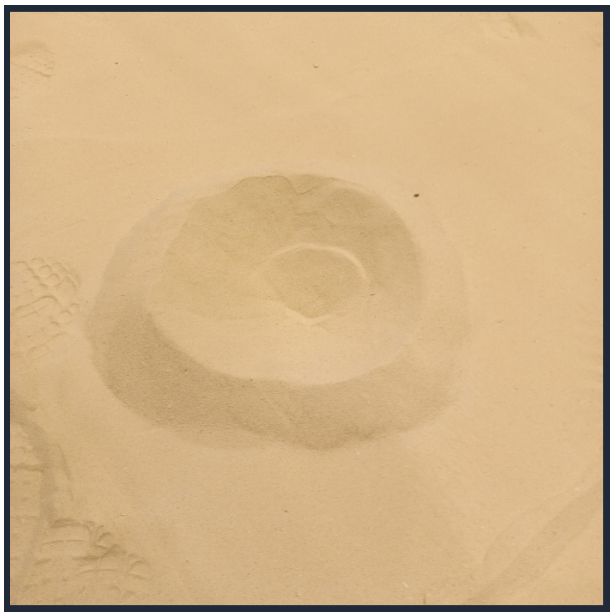


Figure 1: Crater Image example

Figure 2: Annotated Crater in the image

1.3 Dozer Refinement

I also worked on some Dozer refinement tasks. I worked with Ankit on this one, to eliminate the jittery motion of the linear actuator at intermediate positions. We switch the pin from a Digital Pin on the Due, to a DAC Pin and noticed less to no jitters. We aim to test it in sync with the whole system once the rover is up and ready for testing.

1.4 Selecting and ordering new spare parts

As we carried out hardware maintenance and some initial testing, we figured that some parts required complete replacement and makeover. I decided to look for the appropriate motors and matching pinions and was able to find a pinion with 15 teeth (as opposed to the other 16 teeth pinion). I will be working with the team to get the new motors fixed to the rover in the upcoming days.

2 Challenges

- One of the key challenges was the biggest risk we track every time: Limited Spares availability. Just like last sem, some of the teeth of the steering pinion got damaged and we were down to our last spare. That's when we decided to get a pinion of the same module but one less tooth, leading to just 0.6mm of difference in pitch diameter. We purchased the new pinions and are working on integrating it to the rover.
- Another major issue is the Arduino Reset issue. WE have to manually reset Arduino every time we lose communication between the Roboclaws and the Arduino Due. To tackle this problem, we decided to implement a software reset that will be triggered using the Jetson Orin.

3 Teamwork

The team completed many individual and collaborative tasks in this phase, which are listed here:

- **Ankit**: Ankit and I worked on debugging th dozer jitter issue. We tried implemententing different fixing, and are on the verge of solving the hitter issue. We also collaborated to ideate methodologies for the Perception stack. He also designed and 3D Printed new mounts for the new Orin and ZED Camera in this phase.
- **Bhaswanth**: Bhaswanth worked on implementing the resection method for the total station. He collaborated with William to carry out unit testing for fixing the connection issues between Arduino, Roboclaws, and the motors. He also worked with Simson to finalize the methodology for the global controller, and with me and Ankit for fixing the Arduino Reset problem.

- **Simson**: Simson focussed this phase to work on crater detection using the initial map that we get from the FARO Scanner. He explored different methods for the same, but finally convereged to the similar method he used in the previous semester for this task. In addition, he collaborated with Bhaswanth on the Navigation stack, especially the Global Navigation Controller.
- **William**: William's work since the last progress review focused on debugging software connections between the new Orin compute with the Arduino (in collarboration with Bhaswanth) and joystick. The joystick issue was resolved by installing updated Logitech joystick dependencies and the Arduino problem was resolved by reinstalling MicroROS. William also collaborated with Bhaswanth in resolving the sudden jolts caused by steering commands. Finally, William worked on researching on how to implement the rover's validation unit using point cloud sensor data and elevation mapping.

4 Plans

Until next Progress Review, I'll be working on training the YOLO model with the dataset, tuning the parameters for detection, running inferences on test videos and live videos, and setting up a methodology to extract geometric features from the video feed.