University of Arkansas – CSCE Department
Capstone I – Final Proposal – Fall 2020

NASA Robotic Mining Competition

William Burroughs, Calvin Franz, Z. Gunner Lawless, Jett McCullough, Carson Molder

Abstract

The goal of this project is to design, implement, and evaluate the computer systems of a robotic lunar mining rover developed by the Razorbotz team at the University of Arkansas. In this project, the old robot code will be refactored and documented with good programming practices on a newer platform as well as expanded with new autonomy features. Over several months, the team will first refactor the old code, document the updated code, and then expand upon it. The goal is to win the NASA Robotics Mining Competition (RMC) in May 2021 with the updated robot and control system. In the future, lunar mining robots can be used to extract minerals on the moon, reducing the costs of such materials on Earth while eliminating the environmental impacts from their extraction.

1.0 Problem

NASA’s Artemis program has goals including putting the first woman and next man on the moon by 2024, exploring more of the lunar surface than ever before, and later sending astronauts to Mars. This mission will require an incredible quantity of resources and infrastructure to be present in the lunar environment to house astronauts and provide a base for lunar operations. A 2019 estimate of the commercial cost of transporting payloads to the moon is about $1.2 Million per kilogram [1]. Given the many resource needs of humans and the extreme cost of sending resources into space, improving the technologies that allow us to harvest resources for the production of fuel, water, and/or oxygen from local materials while on space missions will be an important step towards enabling sustainable lunar surface operations while decreasing supply needs from Earth. Without the ability to gather resources while in space, extended space exploration will be infeasible due to the incredible costs of supplying resources from Earth.

Rovers will be a valuable tool to future astronauts, especially if they can autonomously gather useful local materials. However, autonomy presents its problem. Rovers are complex machines that need to operate in extreme environments continuously and reliably. The Artemis Student Challenge (this project) challenges college-level teams to develop a lunar excavator prototype capable of excavating lunar regolith to extract local resources so that those resources need not be transported from Earth. The challenge requires our rover to be capable of teleoperation as well as various levels of autonomy, and it strongly rewards the development of lighter rovers. The rover will be faced with specific challenges from the abrasive nature of the regolith and icy-regolith simulants it must mine, weight and size limitations for the rover itself, and the ability to navigate and operate remotely and autonomously.

Our Capstone team is part of the Computer Systems sub-team of the U of A’s “Razorbotz” team dedicated to this challenge. Our specific tasks will be to implement all the software systems required by the rover, including controls, operator interfaces, sensor reading, and computer vision, and autonomy.

2.0 Objective

The objective of this project is to design a strong, sturdy, fully modular robot that can maneuver on the arena terrain, collect the maximum amount of icy regolith within the time limit, and make two collection trips autonomously using ROS (Robot Operating System) and several types of sensors. Each sub-team has its separate objective which combined, gave the overall objective of the project. Our team is part of the Computer Systems sub-team and will be responsible for executing the objectives of autonomy and control. Other objectives include gaining experience with efficient hardware design and software-architecture planning, learning the processes involved in working with a multidisciplinary team, and obtaining valuable engineering skills for future careers. If our project is competitive enough, it may win the competition and benefit NASA by providing innovative robotic and excavations concepts that may inspire future ideas and solutions.

3.0 Background

3.1 Key Concepts

A wide variety of technologies and key concepts will be used to build the computer systems for the robot. This year, the Computer Systems sub-team’s main goals are to refactor old code and add further autonomy to the robot’s functionality with the help of computer vision.

The robot will use computer vision, specifically an object recognition neural network, to seek out objective locations, identify key targets, and perform path generation within the competition arena. Python, ZED SDK, and PyTorch will all be used to develop the computer vision model needed for the robot [2]. One potential candidate for an object detection neural network is YOLO [3]. The computer vision software will be run on an Nvidia Jetson Nano board, a small microcomputer that is designed for artificial intelligence models in low-power environments.

The robot will have a ZED camera mounted to it which will allow for visual sensor input to be passed to the computer vision model via the ZED SDK. Sensor input will be passed to the computer vision model built using PyTorch. The model will be application-specific but based on a pre-trained network initially trained on a large, diverse image dataset. A new dataset built using ZED image senor data in the robot testing environment will be used to retrain the initial model into the model that will be used by the robot. The model will decide which functions the robot will need to perform automatically.

All robot functionalities will be built using ROS (Robot Operating System). ROS supports functionality wrappers programmed in multiple languages, but background knowledge of embedded systems programming will be helpful for interfacing with the robot’s hardware and sensors. This year’s Computer Systems sub-team has experience with a wide variety of programming languages including Python, C, C++, and Java, so many of these languages will be used to program ROS is a real-time operating system that seeks to accomplish desired goals within time limits. While there is an existing codebase for manually controlling the robot's functionalities, this year's Computer Systems sub-team will take on the goal of completely rebuilding the interface for operating the robot manually. The current code for interfacing with the robot was built primarily by mechanical engineering students, so the usability and functionality were limited to their programming knowledge. This year's sub-team contains more students in computer-related fields, so modern computer interface concepts will be used in the manual interface redesign to produce more intuitive controls. This includes using Git for the software's version control. Many new functionalities and standards will be implemented by this year's Computer Systems sub-team, but this year's team will primarily focus on learning and using computer vision, the ROS real-time operating system, embedded systems technologies, and modern computer interface design.

3.2 Related Work

Teams from previous years have built computer systems focusing on navigation, internal systems, controls, and manual operation with feedback [4, 5, 6]. The existing codebase was primarily built by mechanical engineering students with limited software development experience. This year's computer systems team plans to rebuild a large portion of the existing codebase while implementing good coding practices. This entails creating well-documented code, developing unit testing, and using Git as a version control manager. Additionally, our code will improve upon readability so future teams can continue using the codebase we design. Another key goal this year's computer systems team has is to implement further autonomy of the robot's functions with the help of computer vision. In prior years, the competition did not have as much of an emphasis on the robot's autonomy with its tasks [7, 8]. For this year's competition, many points are allocated to the autonomous functionality of the robot [1], so autonomy will be a primary focus when developing the new computer systems codebase. Since this year's computer systems team has an abundance of students from computer-related fields, it is expected that the robot's computer systems will see drastic improvements compared to previous years concerning programming standards, software stability, system usability, and robot autonomy.

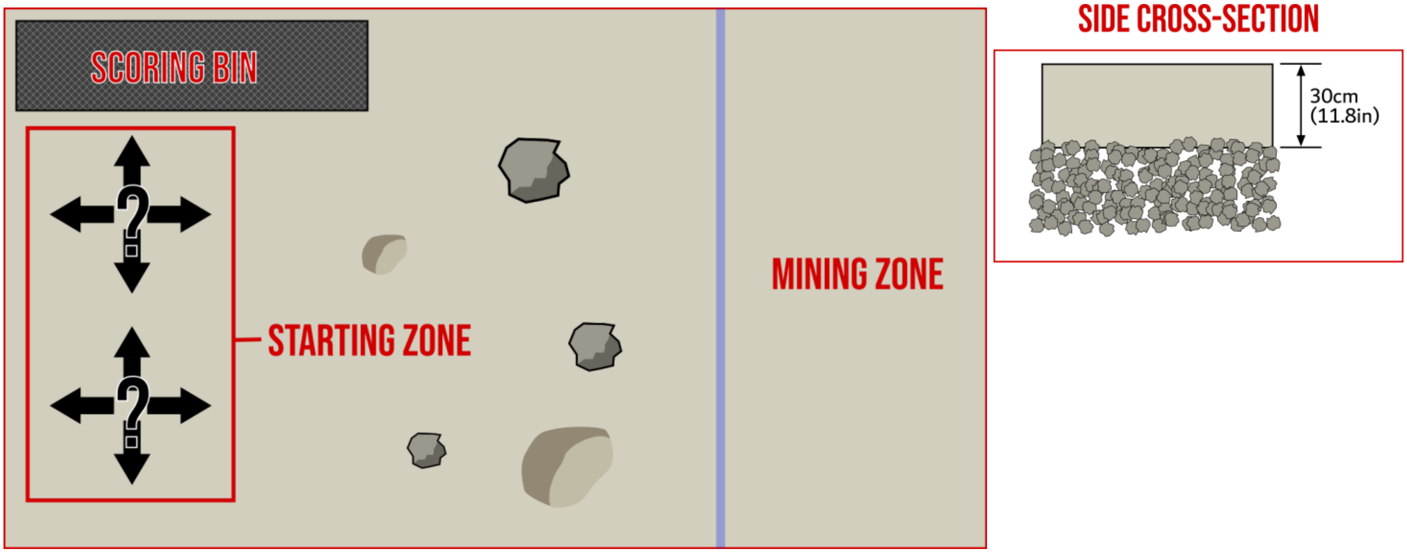
4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

- Design must satisfy volume specifications set by the NASA rubric
- Design must satisfy volume specifications of .5m x .5m x 1m, as set by the NASA rubric
- Prove team can communicate wirelessly with the robot by March
- Autonomous navigation of the arena
- Autonomous excavation of the regolith

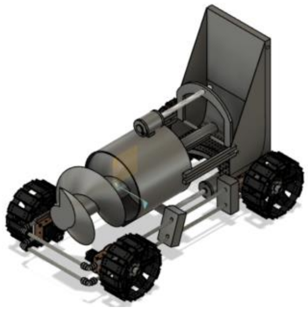
4.2 High-Level Architecture

The design of the robot for this year was heavily modified from the previous years. In years past, the robot relied on a spinning drum to remove the soil simulant called BP-1 from the gravel beneath and used a shovel to scoop the gravel into a collection bin. Both the soil composition and the arena layout are shown in the figures below. This year, due to changes in the size requirements, the robot will use an auger to drill into the soil and collect the gravel. The significant changes in the design have brought equally significant changes in the hardware and software needed to run the robot. The robot will use four motors to run the drivetrain and four ODrive motors to run the excavation assembly.



The software needed to run the robot will be broken into four main systems: Navigation, Movement, Excavation, and Autonomy. The navigation system will be responsible for using the camera inputs to plot a path around obstacles and keep track of the position of the robot. The movement module is made of the drivetrain motors and the various sensors attached to keep track of the movement of the wheels and the robot. The excavation system consists of the ODrive motors, the attached encoders, and the other sensors attached to ensure the operation of the excavation assembly is working smoothly.

The final and most important system is the autonomy module. This will be responsible for all aspects of autonomous operation, including using the other three systems to read in the information necessary and to take the actions required to achieve the goals laid out by the system. It will use a combination of sensor inputs, graphics processing, and logic to run the course fully autonomously. Attached below is a picture of an old model of the robot and an overview of the current ROS node setup. The robot will be programmed using the Robotic Operating System 2 (ROS2). This operating system allows the users to write individual programs, known as nodes in ROS, in many different languages and have them all interact without errors if the message passing is compatible. ROS2 will be used in place of ROS1 because of Windows support.



4.3 Risks

Risk	Risk Reduction
Autonomy system fails during competition	Train on a diverse set of rock images, attempt to mimic the competition rock pit and rock samples as accurately as possible, use a neural network architecture that is robust to changes
Glitches from porting ROS1 code to ROS2	Document all changes, keep code compartmentalized to make debugging easier, write and use unit tests to evaluate changes
ME students may not understand our changes because of limited computer science knowledge	Write detailed, comprehensive documentation that explains concepts and functions well enough that someone who has only taken Programming Foundations I (or equivalent) can understand them
Setting the ME team back by losing old code	Back up the old code before making our changes, using GitHub for effective version control
Loss of key functionality from porting ROS1 to ROS2	Keep a checklist of functionalities to implement in the refactored code, checking items off as they are ported and updated, and only removing functionalities that are deemed no longer applicable

4.4 Tasks

1. Refactor and Upgrade Old Code
    - Update ROS1 code to ROS2
        - Learn ROS2 – This is essential to move forward as all code will be put through ROS2
    - Trim current Modules
        - Study current code to understand functionality
        - Remove code that is no longer applicable to the current project.
            - Communicate these specifics with design teams.
    - Create/Update Documentation
        - Ensure documentation is long-lasting
        - Must be extremely detailed.
2. Improve Manual Control and UI
    - Create a more User-Friendly UI Design
        - Consult with Project Lead on Recommendations.
3. Implement full Autonomy
    - Excavation Autonomy: Digging rocks from the surface
        - Explore potential datasets for a computer vision-based object recognition model
            - May be implemented using YOLO [3] or another real-time object recognition neural network

         - Take photographs/video of camera to predict where rocks, obstacles, and the mining base are in the given frame
         - 36 FPS on ResNet-50 using Jetson Nano [9], allows for real-time image analysis
      - Design, implement, and train the model
      - Simulate the model on the rover
      - If computational power permits, run the trained model directly on the rover
   - Dump Autonomy: Depositing payload at mining base
      - Use object recognition to recognize mining base
      - Create an analysis algorithm that determines the correct path to the dumpsite
   - Travel Autonomy: Movement around the test environment
      - Use object recognition to recognize rocks and hazards
      - Create an analysis algorithm that parses recognized objects and hazards to calculate the optimal path to the next rock
   - Failure Management: Handling of software errors and mechanical failures
      - Detect any component failures such as camera or accelerometer loss
      - Implement backup features for detectable losses

4. Win the NASA RMC Competition

4.5 Schedule

Tasks	Dates
GUI/CLI Design Started	9/1/2020
GUI Design Review	9/26/2020
GUI Programming/Code Rewrite Started	10/1/2020
Start Manual Controls	10/1/2020
Preliminary Design Review	10/10/2020
Finish GUI/CLI Design Overhaul	12/18/2020
Revisit Scope of Project	1/16/2020
Start Excavation Macro	1/23/2021
Start Camera Implementation	1/23/2021
Start Sensor Implementation	1/23/2021
Finish Manual Controls Overhaul	2/6/2021
Begin Full Scale Manual Mining Testing	2/6/2021
Start Training AI and Dataset Creation	2/20/2021
Complete Target Localization with ZED camera	3/6/2021
Begin Testing Camera and Sensors While Driving	3/20/2021

Finish Excavation Macro	3/27/2021
Begin Full Scale Autonomous Testing	4/10/2021
Finish Final Testing of Autonomy	4/24/2021
Graduate	5/7/2021
Compete at NASA	5/18-23/2021

4.6 Deliverables

- Design Document: For the design document, we will submit a report of the software systems to the Project Manager. The design document will include a diagram of how the ROS2 nodes are connected, a description of the computer vision system used for navigation autonomy, and additional details about the software that controls the robot.
- ROS2 Nodes: ROS2 nodes for autonomy, excavation, navigation, and movement will be designed and implemented. These nodes will be made up of many subfiles and will comprise most of the software created during this project.
- Documentation: The previous code had very little documentation explaining functions, subroutines, and variables. As this code is refactored and written with good programming practices, the code will also be well documented through comments in the code and a document explaining the utility of each function and class. Documentation will also be provided for all new code written that adds functionality to the rover.
- Robot Testing Data: A dataset of images containing rocks and obstacles will be built. After the neural network that analyzes these images is completed, the robot will be tested in the campus testing pit. Results of the robot’s performance will be documented as a deliverable.
- Final Report: At the end of the project, a final report will be written that explains the steps taken to complete the project, challenges and setbacks faced, tasks that were altered from this report, and the results. This report will be supplemented by all the code written in the project, for both the ROS2 nodes and computer vision system.
- Project Website: Along with the final report, the project website will be updated to include additional information on the project. It will host the results of the project, a link to the code repository, the final report, and additional relevant material. This website will be hosted on the capstone.csce.uark.edu website.
- RMC Competition Prize: At the RMC Competition in May 2021, the first prize in the competition will be sought. Winning this prize is the primary goal of this project. The scores from the competition will be kept and used for future competitions to note weaknesses and strengths that could be addressed in future robot designs.

5.0 Key Personnel

William Burroughs – Burroughs is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has been on the team for 1.5 years and was the Controls sub-team lead in 2020. He is the current Computer Science sub-team lead for the RMC project. For this project, he will oversee time management and project deliverables for the subteam.

Calvin Franz – Franz is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has experience in writing software for vehicle systems in the F-16 fighter plane and has worked on projects featuring Robotics, AI, and Software Development. This is his first year on the team, and he plans to use his software engineering experience in vehicle systems to help with the implementation of ROS2 nodes and write descriptive multi-level documentation for the project.

Zachary Lawless – Lawless is a senior Computer Science and Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. This is his first year on the team; however, his previous research experiences attacking vulnerabilities in deep neural networks and working with embedded systems should help contribute to building the robot’s computer systems. Lawless will be responsible for contributions in developing the robot’s new computer vision model and performing software upgrades from ROS to ROS2.

Jett McCullough – McCullough is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. This is his first year on the team. He has experience with working on diverse project teams and foundational knowledge in AI, Big Data, and Software Development. He will be responsible for developing ROS nodes and writing documentation.

Carson Molder – Molder is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has experience in artificial intelligence, machine learning, and computer systems. He is currently performing computer vision research with Professor Justin Zhan. For this project, Molder will develop the robot’s computer vision system.

Alex Westbrook – Westbrook is a senior Mechanical Engineering major in the Mechanical Engineering Department at the University of Arkansas. He is a fourth-year team member and is the project manager this year. He oversees all sub-teams and oversees the overall schedule for the project.

Dr. Uche Wejinya– Dr. Wejinya is an Associate Professor in the Department of Mechanical Engineering at the University of Arkansas. He performs research in robotics and mechatronics and is the faculty advisor for Arkansas Razorbotz.

6.0 Facilities and Equipment

6.1 Facilities

- Mechanical Engineering Robotics Lab: Laboratory in the Mechanical Engineering building where the robot is assembled, and the software is tested
- Campus Test Site: Shipping container with a simulated lunar environment to test designs
- Data Science and Artificial Intelligence Lab: Contains computers with GPUs to train computer vision models (pending permission from Professor Justin Zhan)

6.2 Equipment

- Talon SRX Motor Controller: Uses CAN protocols to relay position information of motors to the Jetson
- Victor SPX Motor Controller: Similar functionality to Talon
- NVIDIA Jetson Nano Developer Kit: Run ROS nodes, includes neural network hardware
- Vex BAG Motors: Motors used to power each wheel
- ODrive Boards: Motor controller boards used to interface between Jetson and ODrive motors
- ODrive Motors: Used to control excavation assembly
- ODrive Encoders: Tracks positions of excavation motors, allowing for more precise control
- Zed Stereo Camera: Used for depth perception and navigation; images are passed through to the neural network to predict where rocks and obstacles are
- TI Tiva TM4C123GXL: TI board used to interface with the IMU
- Sparkfun LSM6DS3: Accelerometer/Gyroscope IMU
- TI IWR1443 Evaluation Module
- Power Distribution Panel: Used to distribute and track power consumption of various components
- Logitech Extreme 3D Pro Joystick: Used to manually control the robot
- Arduino Modules
- NVIDIA RTX 2080 Ti: Used to train and simulate computer vision models for the rover

7.0 References

[1] "NASA Robotic Mining Competition (RMC) Lunabotics 2021, Registration, Rules and Rubrics," NASA, 2020, url: https://www.nasa.gov/sites/default/files/atoms/files/000_rmc_lunabotics_rules_rubrics_2021.pdf

[2] "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Adam Paszke, Sam Gross, Franciso Massa, et.al., 2019, url: https://papers.nips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[3] "You Only Look Once: Unified, Real-Time Object Detection," Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, arXiv, 2016, url: https://arxiv.org/abs/1506.02640

[4] "Razorbotz CS Intro," William Burroughs, 2020, url: https://www.youtube.com/watch?v=BSq90vnPwJ4&feature=youtu.be

[5] "2020 Systems Engineering Report," Razorbots Team 2019-2020, University of Arkansas, 2020.

[6] “Razorbotz/RMC-Code-19-20,” Razorbotz Team 2019-2020. 2019-20, url: https://github.com/Razorbotz/RMC-Code-19-20

[7] “RMC 2019 Registration, Rules and Rubrics,” NASA, 2018, url: https://www.nasa.gov/sites/default/files/atoms/files/rmc2019_registrationrulesrubrics_09062018.pdf

[8] “2020 Systems Engineering Report,” Razorbotz Team 2019-2020, University of Arkansas, 2020.

[9] “Jetson Nano: Deep Learning Inference Benchmarks,” Nvidia Developer, n.d, url: https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks