

# NASA Lunabotics 2022-2023

A Major Qualifying Project  
Submitted to the Faculty of  
WORCESTER POLYTECHNIC INSTITUTE  
In partial fulfillment of the requirements for the  
degree of Bachelor of Science

**Submitted April 27th, 2023**

Alexander Brattstrom (RBE)

Carter Bullock (RBE)

Roopsa Ghosh (RBE/CS)

Patrick Hagearty (RBE/CS)

Helen Le (RBE/CS)

Joshua Moy (RBE/PH)

Report Submitted to:  
Professor Joshua Cuneo, Advisor, WPI  
Professor Hektor Kashuri, Advisor, WPI  
Professor Kenneth Stafford, Advisor, WPI

*This report represents the work of six WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review.*

## Abstract

The Lunabotics MQP represents an effort to compete in the NASA RMC Lunabotics competition. This requires the project objectives to follow NASA's established rules and regulations for the competition. As the Lunabotics MQP is a legacy project, this year's team worked on an inherited robot from last year's competition. The main focus for this year's team was to incorporate full autonomy into the competition cycle to earn more points on the scoring rubric and increase overall competition efficiency. Additionally, the team took on the task of redesigning several mechanical systems from last year to increase subsystem efficiency.

In order to achieve the objective of full autonomy, the team went about designing a sensor integration process in order to measure every state of the robot with accuracy. This included the incorporation of encoder monitoring and hall effect sensors to measure the position of each subsystem relative to starting configuration. In addition, the task of full autonomy meant that the robot would need to perceive and interpret the competition field around it. This was done by integrating a D455i camera for both visual and stereo-depth processing. Visual processing was included to assist in the localization of the robot using April Tags, and Stereo depth processing was included to assist in obstacle avoidance capabilities.

The problem presented from last year's mechanical design was an issue of ground clearance that made it difficult for the robot to drive. The main mechanical objective of the team included redesigning the storage and deposit subsystem used by last year's team to overcome the clearance issue. An additional redesign of the excavator subsystem was conducted so that it would require less time to reconfigure from the storage position used to meet size requirements and the fully extended digging configuration.



## Acknowledgements

The team wants to thank our project advisors (Professor Kenneth Stafford, Professor Joshua Cuneo, Professor Hektor Kashuri), the partnering Business Management MQP team (Jaden Meng, Corey Logan, and Professor Walter Towner) FRC Team 190, Kate Lindsay, Daniel Ali Tribaldos, Neil Rosenberg, Washburn Lab staff, Worcester Sand and Stone, and the African Community Education for their contributions and support throughout this project.

Table of Authorship

<b>Report Section</b>	<b>Author(s)</b>	<b>Editor(s)</b>
<b>Abstract</b>	Carter	Helen
<b>Acknowledgements</b>	Carter	Helen, Alex
<b>Executive Summary</b>	Carter, Josh	Helen
<b>Introduction</b>	Joshua	Patrick, Helen
<b>Background</b>	Joshua, Roopsa, Carter, Helen, Alex, Patrick	Carter, Alex, Helen
<b>Design and Development: Excavator</b>	Carter	Joshua
<b>Design and Development: Storage and Deposit</b>	Joshua, Roopsa	Carter
<b>Design and Development: Vision Processing/Navigation</b>	Patrick	Patrick, Roopsa, Alex
<b>Design and Development: Autonomy Integration</b>	All	Patrick, Joshua, Alex
<b>System Testing and Validation: Excavator</b>	Carter	Patrick
<b>System Testing and Validation: Storage and Deposit</b>	Joshua, Roopsa	Carter

<b>Report Section</b>	<b>Author(s)</b>	<b>Editor(s)</b>
<b>System Testing and Validation: Vision/Navigation</b>	Patrick	Roopsa, Alex
<b>System Testing and Validation: Autonomy Integration</b>	Helen, Roopsa	Carter
<b>Analysis and Discussion: Excavator</b>	Carter	Joshua
<b>Analysis and Discussion: Storage and Deposit</b>	Joshua	Joshua
<b>Analysis and Discussion: Vision/Navigation</b>	Roopsa, Patrick	Joshua, Alex
<b>Analysis and Discussion: Autonomy Integration</b>	Helen	Joshua, Alex
<b>Conclusions and Recommendations</b>	Joshua, Roopsa, Alex, Helen	Patrick
<b>References</b>	Joshua, Alex	Alex
<b>Appendices</b>	Alex	Alex

## Table of Contents

<b>Abstract</b> .....	<b>1</b>
<b>Acknowledgements</b> .....	<b>2</b>
<b>Table of Authorship</b> .....	<b>3</b>
<b>Table of Contents</b> .....	<b>5</b>
<b>List of Figures</b> .....	<b>7</b>
<b>List of Tables</b> .....	<b>10</b>
<b>Executive Summary</b> .....	<b>11</b>
<b>1 Introduction</b> .....	<b>1</b>
<b>2 Background</b> .....	<b>2</b>
2.1 The Competition.....	2
2.2 Previous Lunabotics Robots.....	2
2.3 Statement of Work.....	8
2.3.1 Level of Autonomy.....	8
2.3.2 Dust Protection.....	9
2.3.3 Amount of Regolith Collected.....	9
2.3.3 Competition Cycle Speed.....	9
2.3.4 Maximum Mass.....	10
2.3.5 Competition Defined Cycles.....	10
2.3.6 Safety Rating.....	10
2.4 Timeline.....	11
<b>3 Design and Development</b> .....	<b>12</b>
3.1 System Overview.....	12
3.1.1 Excavator.....	12
3.1.2 Storage and Deposit.....	13
3.1.3 Vision/Navigation.....	13
3.1.4 Autonomy Integration.....	15
3.2 Conceptual Designs and Prototyping/Modeling Feasibility Studies.....	16
3.2.1 Excavator.....	16
3.2.2 Storage and Deposit.....	19
3.2.3 Vision/Navigation.....	22
3.2.4 Autonomy Integration.....	24
3.3 Design of Core Functions.....	26
3.3.1 Excavator.....	26
3.3.2 Storage and Deposit.....	33
3.3.3 Vision/Navigation.....	40

3.3.4	Autonomy Integration.....	49
3.4	System Integration.....	50
3.4.1	Excavator.....	50
3.4.2	Storage and Deposit.....	51
3.4.3	Vision/Navigation.....	52
3.4.3	Autonomy Integration.....	52
<b>4</b>	<b>Systems Testing and Validation.....</b>	<b>56</b>
4.1	Excavator.....	56
4.1.1	Lead Screw System.....	56
4.1.2	Belt System.....	57
4.2	Storage & Deposit.....	59
4.2.1	Bucket Frame.....	59
4.2.2	Depositing.....	59
4.3	Vision/Navigation.....	61
4.4	Autonomy Integration.....	62
<b>5</b>	<b>Discussion.....</b>	<b>64</b>
5.1	Excavator.....	65
5.2	Storage and Deposit.....	65
5.3	Vision/Navigation.....	66
5.4	Autonomy Integration.....	66
<b>6</b>	<b>Conclusions and Recommendations.....</b>	<b>67</b>
	<b>References.....</b>	<b>68</b>
	<b>Appendix A.....</b>	<b>70</b>
	<b>Appendix B.....</b>	<b>71</b>
	<b>Appendix C.....</b>	<b>72</b>
	<b>Appendix D.....</b>	<b>73</b>

## List of Figures

The following is a comprehensive list of all figures referenced in the paper.

<b><u>Figure No.</u></b>	<b><u>Description</u></b>	<b><u>Page No.</u></b>
1	The CAD of WPI's 2020 Lunabotics robot	2
2	The CAD of WPI's 2021 Lunabotics robot	3
3	The University of Alabama's 2022 robot	3
4	Astrobotics team's 2015 NASA Luabotics robot	4
5	Montana Technological University's entry	5
6	Purdue's robot with mesh as filter	5
7	MSOE's entry from last year	6
8	SDSU's entry from last year	6
9	Astrobotics' 2015 NASA Luabotics robot	7
10	The system hierarchy for the robot and its surroundings	12
11	The field the robot had to navigate	14
12	Conops Diagram	16
13	Belt tensioner failure	17
14	Designed belt tensioner placement	17
15	Misaligned gears due to belt tensioner	18
16	WPI 2017 Lunabotics Robot, Markhor	20
17	Initial design for the bucket	21
18	Prototype of the bucket system	22
19	2-Channel CAN hat for Raspberry Pi	25
20	roboRIO 2.0	26
21	Free body diagram of the robot when it is digging	27

<b><u>Figure No.</u></b>	<b><u>Description</u></b>	<b><u>Page No.</u></b>
22	BAG motor efficiency graph	30
23	Free body diagram of torque applied to the linear screw	31
24	Free body diagram of linear screw	32
25	Stress analysis of the bucket when loaded	33
26	The calculations done to determine the angle at which the backplate would go flat	34
27	The complete CAD of the bucket and transmission system design before manufacturing	35
28	Free Body Diagram used to calculate the minimum torque required to raise a fully loaded bucket	37
29	Curve Sheet for PG188 Motor (AM-3656 Motor curve sheet)	38
30	The bucket with the front wedge attached	39
31	Visual representation of a projection transformation	41
32	Visual representation of the two-tag localization method	41
33	17HS4023 Stepper Motor and FEETECH continuous rotation servo	42
34	Depiction of the point cloud processing pipeline	44
35	Visualizing the initial mapping	45
36	Visualization of edge detection	47
37	Visualization of path to the target frontier	48
38	Toolpaths for CAM program to make the new mountings	50
39	The new mountings machined on the VM-2	51

<b><u>Figure No.</u></b>	<b><u>Description</u></b>	<b><u>Page No.</u></b>
40	Hall effects two channel encoder in the PG188	53
41	MD30-60 Pressure Sensor	54
42	SmartDashboard	55
43	Testing the excavator digging down to depth	57
44	Testing various materials for their dust protection effectiveness	58
45	Testing the frame of the bucket	59
46	The bucket depositing gravel on sieve	60



## List of Tables

<b><u>Table No.</u></b>	<b><u>Description</u></b>	<b><u>Page No.</u></b>
1	The team's goals for the robot this year	8
2	NASA Deadlines for the 2023 Competition	11
3	Initial Storage & Deposit Decision Matrix	21
4	Critical design requirements for the computer vision and navigation subsection	23
5	Preliminary assessment of localization methods	23
6	Preliminary assessment of different mapping technologies	24
7	The variables for analysis of the excavator	27
8	BAG Motor calculation sheet	29
9	Localization system requirements	40
10	Map generation system requirements	43
11	Path-planning system requirements	45
12	Dust protection results	58
13	Bucket deposit testing results	60
14	Localization accuracy testing	61
15	A* calculation times and node expansion in relation to the implemented heuristic	62
16	The team's goals for the robot this year with descriptions on what was achieved	64

## Executive Summary

The NASA Lunabotics Challenge is a competition in which teams all over the nation participate in building robots that are designed to dig on a simulated lunar surface for gravel, a simulant for ice water. The Lunabotics MQP is a project designed around the framework of the competition. The goals of this project are meant to improve the design of the inherited robot from last year's team in a way to achieve faster and more reliable performance in a competition environment. This included implementation of full obstacle avoidance and navigation autonomy, full subsystem autonomy, a redesign of the storage and deposit system, and improving the excavator's reconfiguration speed. The team wanted to combine the above listed systems into a fully autonomous run.

All of these goals required their own separate design process. The storage conveyor on last year's robot was replaced by a motor-actuated bucket. This bucket was created using 1" x 1" aluminum square tubing and reinforced with steel wire to filter out non-gravel materials. We performed finite element analysis to confirm the design was strong enough to hold a full bucket of gravel. This full load of gravel also gave us the amount of torque and power that would be required to actuate the bucket to its final position. The torque and power requirements allowed us to, first, select our gear ratio and gears which were bought from VEX Pro's lineup and second, actuate the system with the PG188 motor from the Andymark PG Series Gearbox Motors.

The storage and deposit system was a success overall as it was able to easily deposit three kilograms of gravel and it could deposit this load in four seconds. The bucket system was also 5.5 kilograms lighter than the old storage conveyor and it used over 50% less moving parts than last year's conveyor.

The linear screw on the excavator linear slider was re-designed to provide a much shorter total reconfiguration time -down to 57 seconds from nearly a minute and a half- for excavation purposes. Its siding was also replaced with better-fitting Lexan plates padded with black rubber foam for dust protection.

The obstacle avoidance and navigation goal required the team to start researching methods of obstacle detection. We came to the conclusion that a stereo depth camera would be necessary to detect obstacles.

The team was overall able to increase the efficiency of the robot by achieving subsystem autonomy through the implementation of sensors.

The Moon has long been an essential target in astronomy for NASA, from the Apollo program over fifty years ago to the Artemis program today. In its recent decadal survey, NASA identified the Moon as a critical mission target, with sample return and human exploration missions being a priority to address important questions in astronomy about the origins of the solar system and planetary formation. Specifically, the decadal survey stated that a medium-class lunar rover mission to collect samples from the Moon's south pole is the highest priority for addressing lunar science goals in the next decade (National Academies of Sciences, Engineering, and Medicine, 2022). In order to collect samples, the robot has to dig through lunar regolith to reach the icy regolith below the surface to collect valuable lunar samples (Artemis Plan, 2020). This is where the NASA Lunabotics competition came in. The WPI Lunabotics team, in addition to the other competitors in this year's competition, aimed to build a robot that can dig through a regolith simulant known as BP-1 (the layer of this is 30 centimeters deep) and collect gravel, which is the simulant for the icy regolith (which exists 15 centimeters deep beyond BP-1). The BP-1 regolith is very fine and somewhat conductive, so dust protection for the many components on the robot components is critical to its success. The robot has to conform to several constraints, including size, energy consumption, and mass. The terrain is also challenging, with craters and boulders having a diameter of up to 40 centimeters, creating challenges for navigation.

The WPI Lunabotics team has created a robot to meet these challenges, which has met the Lunabotics guidelines and the requirements of the MQP. This team had access to some resources to work with, including last year's robot, code from previous years, and documentation from previous teams. We used these resources to determine the direction of this year's work with many design iterations- including changing from a backhoe excavator to a more conventional conveyor system that can be seen on many successful robots from past Lunabotics competitions.

The ultimate goal of this project was to improve on last year's robot and add autonomous features as intended by last year's team. Last year's chassis was excellent at navigating the rugged terrain but needed improvements in its digging and delivery systems. As a result, it did not score as well as it could have at the Lunabotics competition. This year's team considered past WPI designs and designs from other teams to create improvements for this year's robot.

### 2.1 The Competition

The NASA Lunabotics competition provides college students across the country an opportunity to interact with the NASA Systems Engineering process by designing and building robotics Lunar excavators capable of mining regolith and icy regolith simulants. Through conducting this challenge for more than a decade, NASA has gathered data about necessary excavation hardware and surface locomotion processes that it can utilize to return to the Moon through the Artemis program.

The 2023 challenge was a virtual competition in which we were required to design, fabricate, assemble, and operate a Lunabotics robot. The Lunabotics robot needed to complete several tasks. This included driving on a simulated arena, excavating simulated icy regolith buried under an overburden of granular material, then returning to the starting site and delivering the collected material to a simulated receiving hopper (NASA Lunabotics Guidebook, 2023). During the competition, the team was required to achieve critical deliverables that had to be submitted to NASA, including the Project Management Plan, Public Outreach Report, Systems Engineering Paper, and Proof of Life video, all of which were successfully submitted.

### 2.2 Previous Lunabotics Robots

While we inherited a complete robot from last year's team and generally stuck to that design, this year's team still found it valuable to research other successful Lunabotics robots to see how last year's robot could be improved. This meant researching both previous WPI robots as well as other teams' that have been successful at achieving the challenge in the past.

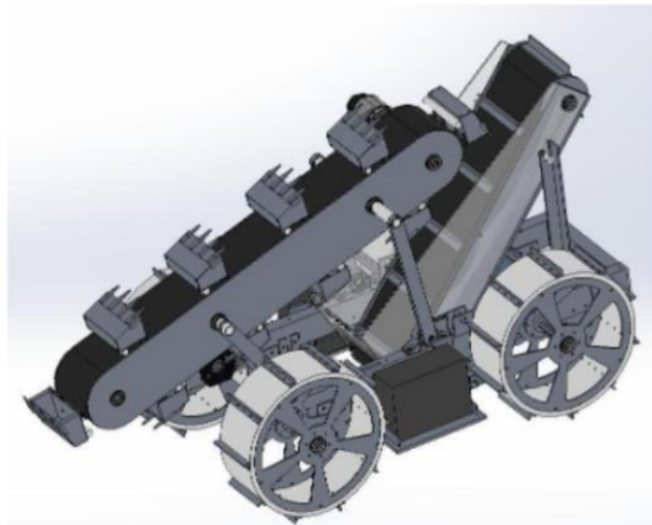


Figure 1: The CAD of WPI's 2020 Lunabotics robot

WPI's most successful designs came from before 2019, when the BP-1 regolith simulant could be mined for points, but that is not to say WPI's past designs did not provide some insight for our designs this year. Notably, the WPI Lunabotics 2020 robot (Figure 1) bears many

similarities to last year's robot with a double conveyor belt system for mining and storage/depositing of the gravel collected. However, unlike last year's robot, the 2020 robot had buckets for mining instead of the modified L-Channels on last year's robot (Bimonte et al., 2021). This design was never built in 2020 due to the COVID-19 pandemic. The 2021 robot (Figure 2) was never completed in full, but the excavator scoop was made and showed some potential when tested (Houbre et al., 2021).

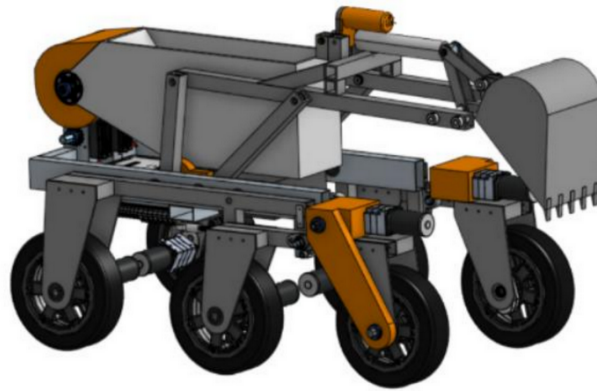


Figure 2: The CAD of WPI's 2021 Lunabotics robot

The first team we looked at outside of WPI was the University of Alabama Astrobotics team, one of the most successful teams in the history of the Lunabotics competition. Their robot for last year's competition can be seen in Figure 3, with a chassis and excavator actuated by a set of lead screw mechanisms. While the team did not switch to a chassis driven by lead screws, there were still some valuable takeaways from their robot. Instead of implementing a filter before the mined material reaches the bottom conveyor, their robot had a mesh-like conveyor belt which allowed it to filter some of the BP-1 regolith out when the mined material landed on it (Alabama Astrobotics, 2022).



Figure 3: The University of Alabama's robot from last year

The next team we looked at was the Montana Technological University team which won the competition last year over the University of Alabama. Unlike many other teams, they used a scoop as an excavator, instead of the more common belt-driven excavator. This design is simpler and potentially lighter/more reliable if done well. The team also used a bucket for depositing gravel instead of a belt-driven depositing system. Also, in contrast to many other teams, they used treads for mobility instead of wheels (Montana Tech NASA Robotics, 2022). We could not find any footage of this robot during the competition, but the fact that it was a successful design informed some of our design changes.

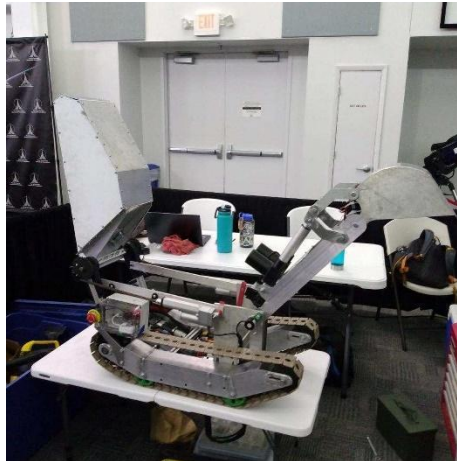


Figure 4: Montana Technological University's entry (winner of Lunabotics last year).

While the following teams in Figures 5, 6, 7, and 8 were less successful than the University of Alabama or Montana Technological University, a more brief look at their robots informed us of general design trends in the competition. Two that stuck out to us compared to our robot from last year was the lack of side shielding on most belt-driven excavators (barring where the pulleys are) and how most teams used a bucket of some sort for depositing gravel instead of a conveyor belt. A selection of robots from Purdue, UIC, MSOE, and SDSU show these design trends in the Figures below.

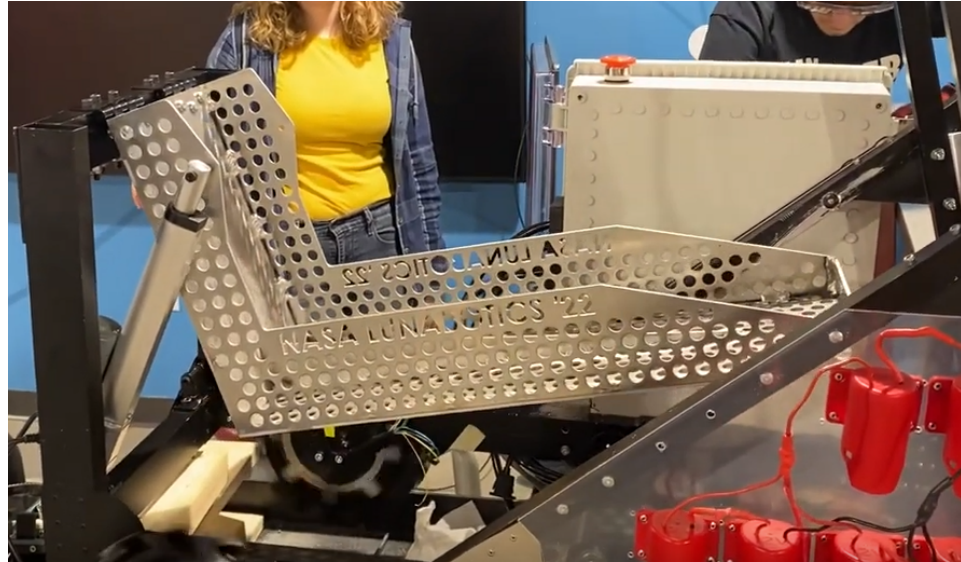




(Purdue Lunabotics Proof of Life Submission 2022, 2022)  
Figure 5: Purdue's entry from last year which uses a mesh attached to some frame members as a filter

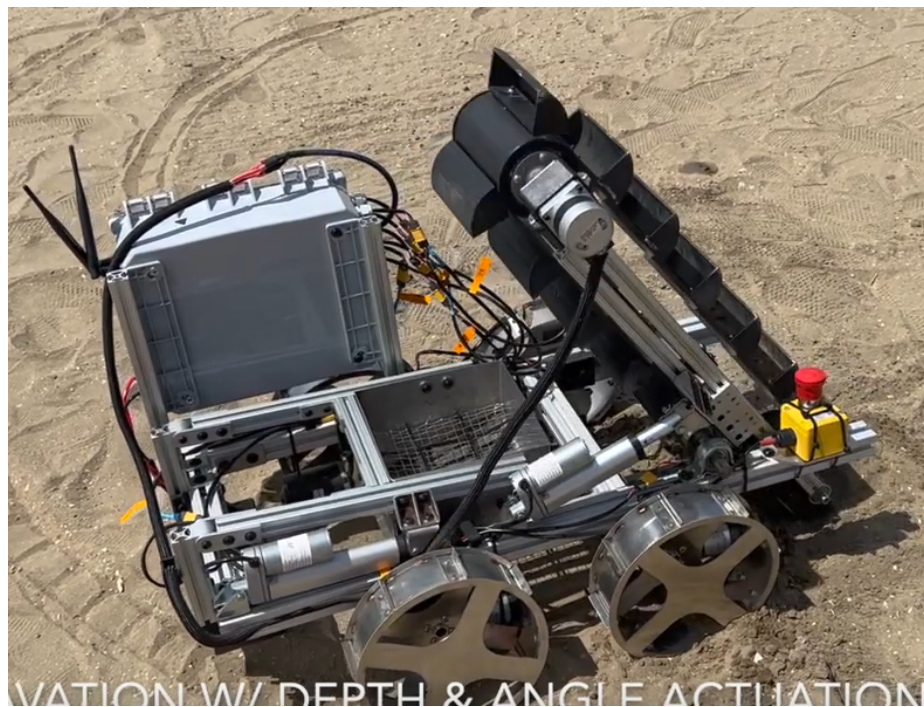


(NASA Lunabotics 2022 Pol | Atlas | University of Illinois at Chicago, 2022)  
Figure 6: UIC's entry from last year whose conveyor is angled at 90 degrees to bring regolith into their bucket. Their filter was not the best though.



(2022 Space Raiders Proof of Life, 2022)

Figure 7: MSOE's entry from last year. There was no footage available of the robot digging but their bucket can be seen in this figure.



(Proof of Life Team Hades San Diego State University NASA RMC 2022)

Figure 8: SDSU's entry was one of the few entries to demonstrate autonomous navigation during its proof of life video. Its filter is similar to the one on the Purdue robot but the mesh is only on the bottom.



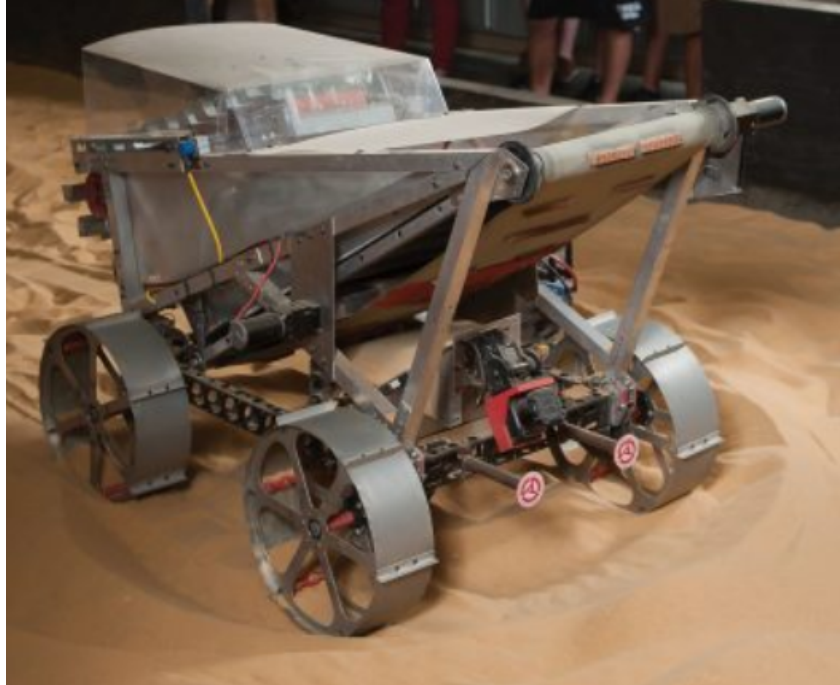


Figure 9: Astrobotics' 2015 NASA Luabotics robot, Astobot, with LiDAR for sensing

We looked at Astrobotics' 2015 robot and noticed they used LiDAR for sensing. Astrobotics was the only team with a robot that used LiDAR to detect a physical target placed in the arena for localization (Martin, 2015).

## 2.3 Statement of Work

The table shown below summarizes our goals for the project. We aligned our goals with the NASA Lunabotics competition.

Parameters	Minimum Goals	Reach Goals
<u>Level of Autonomy</u>	Obstacle Avoidance	Autonomous Navigation
<u>Dust Protection</u>	IP50	IP60
<u>Amount of Gravel Collected</u>	2 kg in 15 min	3 kg in 15 min
<u>Competition Cycle Speed</u>	12 minutes	10 minutes
<u>Maximum Mass</u>	55 kg	50 kg
<u>Competition Defined Cycles</u>	Complete 1 full cycle	Complete 2 full cycles
<u>Safety Rating</u>	Lunabotics requirements (e.g.: switches, handles, e-stop)  Smart Safety Telemetry System (e.g.: temp., current draw, battery voltage)	Visual indicator system or GUI to convey issues back to the ground station.

Table 1: The team's goals for the robot this year

This section covers the design goals of the team in greater detail.

### 2.3.1 Level of Autonomy

#### **Minimum Goal:**

Through computer vision algorithms and stereo depth imaging, we hope to provide the robot with rudimentary object detection by developing a point cloud of its surroundings. With object detection, we will then provide autonomous programming to enable collision/object avoidance so that it will be able to move autonomously through its surroundings when given a path to follow.

#### **Reach Goal:**

Beyond object avoidance, we also wanted to implement localization, map generation, and path planning. Ideally, given a specified target destination, the robot should be able to navigate towards it using its sensors and obstacle avoidance capabilities to reach the selected destination. With integrated localization, the robot will be aware of its position in relation to surrounding objects, whether it has identified them or not. The end goal of autonomous navigation would be to have the robot navigate to and from the excavation zone without human input.

### **2.3.2 Dust Protection**

#### **Minimum Goal:**

The current protection level of the robot is at IP40, with some minor dust collection on the inside of the main excavator. A major goal for us was to fully reach IP50, as the presence of dust inside the excavator decreases its performance. To test this, we will run the robot for 5 minutes, then measure how much dust is in the system. If IP50 is met, there should be less than a spoonful of dust in important systems like the electronics box.

#### **Reach Goal:**

To reach an IEC IP60 rating, specified as being “dust-tight, no ingress of dust”, would be a desirable goal. The primary concern is to eliminate the regolith causing any electrical or mechanical problems. Examples include an electrical short between wires or increased resistance inside a conveyor.

### **2.3.3 Amount of Regolith Collected**

#### **Minimum Goal:**

We did not have a formal record of the amount of gravel the team could obtain during the competition last year. From last year’s competition footage, the collection seemed minimal, though we have been told by our advisor that 1 kg was doable, "if we had done all things correctly". In the competition rules, we would have to obtain 1 kg of gravel in 15 minutes to qualify, so we would like to go beyond that and shoot for 2 kg of gravel.

#### **Reach Goal:**

Our reach goal was to obtain 3 kg of gravel in the allotted time. With an efficient excavation and collection system along with autonomous systems in place, we would be able to collect more in a faster and more accurate manner.

### **2.3.3 Competition Cycle Speed**

#### **Minimum Goal:**

When we received the robot, one of its notable issues was that from the point where it was completely level (at driving configuration) to the point where the excavator reached the ground (at digging configuration), a total of approximately two and a half minutes (reconfiguration cycle) passed. This is a large amount of wasted time that could be better used digging, driving to/from the excavation zone, depositing gravel, or in running another cycle. Our goal was to improve this system and others using autonomy and mechanical redesigns in order to reduce the time it takes the robot to complete one full competition cycle to 12 minutes.

#### **Reach Goal:**

If it is possible to decrease cycle time even further to 10 minutes without causing stability or functional issues to the robot, that would be ideal.

### **2.3.4 Maximum Mass**

#### **Minimum Goal:**

Last year's team was able to keep the robot's mass below 60 kg. This is already significantly below the 80 kg limit by Lunabotics Rules, which is a very strong benefit for the robot, so we aim to keep the mass below 55 kg even with our additions and modifications.

#### **Reach Goal:**

Ideally, our redesign of the robot should be able to shave off some mass as well so being able to take off 10 kg would be a great accomplishment and help the robot drive better without excessive torque on the motors. We will be adding a bucket to the robot and with the use of better materials, we want to reduce the original mass to 50kg.

### **2.3.5 Competition Defined Cycles**

#### **Minimum Goal:**

Last year's team suffered some technical difficulties during the competition. This prevented them from scoring points that would have been easily attainable in other runs. Our minimum goal this year is to make sure the robot can consistently run one competition cycle without error or technical difficulty.

#### **Reach Goal:**

If we can perform one cycle consistently, then we plan to run two cycles uninterrupted without error. This would put us in a position that makes it hard to run into critical issues at events.

### **2.3.6 Safety Rating**

#### **Minimum Goal:**

The current form of the robot is missing some required safety features for the Lunabotics competition. To compete this year, we have added an indicator light and an emergency stop button for all teleoperation and systems.

When initially handling the robot this year, we ran into many potentially hazardous situations. The current robot has no form of self-limitation and will proceed with whatever the given command is, no matter what. This means if the robot was told to continue folding the excavator when it is already at its lowest physically-limited state, it would continue pushing the excavator down, folding in on itself. The team would like to combat this by providing what we call "Smart Safety". To define it, the robot would monitor its internal systems and prevent itself from bringing those systems past a certain parameter.

#### **Reach Goal:**

Our reach goal was to implement a feedback system that allows the ground station to understand what systems function nominally and which systems are having issues. This could be in the form of visual indicators on the robot itself or a GUI dashboard that present the information from the smart safety system. Our team, if time permits, would like to try to integrate this system into the robot's autonomy to allow for the robot to make rudimentary decisions based on its status.

## 2.4 Timeline

<b>NASA Deadlines</b>		
<b>Description</b>	<b>Opens</b>	<b>Closes</b>
NASA Gateway Application	09/14/2022	10/19/2022
Statement From Advisors	09/14/2022	10/19/2022
Statement of Rights of Use	09/14/2022	10/19/2022
Project Management Plan	09/14/2022	10/19/2022
Public Outreach Report	02/08/2023	02/15/2023
Proof of Life Video	03/22/2023	03/29/2023
Systems Engineering Paper	03/22/2023	03/29/2023

Table 2: NASA Deadlines for the 2023 Competition

## 3 Design and Development

### 3.1 System Overview

In this section we will discuss the core functionality of each subsystem and how that functionality is implemented by the team. Figure 10 shows the system hierarchy for the robot and indicates the extent to which changes were made in each system. We highlight the major redesigns and large design changes in this section giving a high level overview of what we intended to achieve.

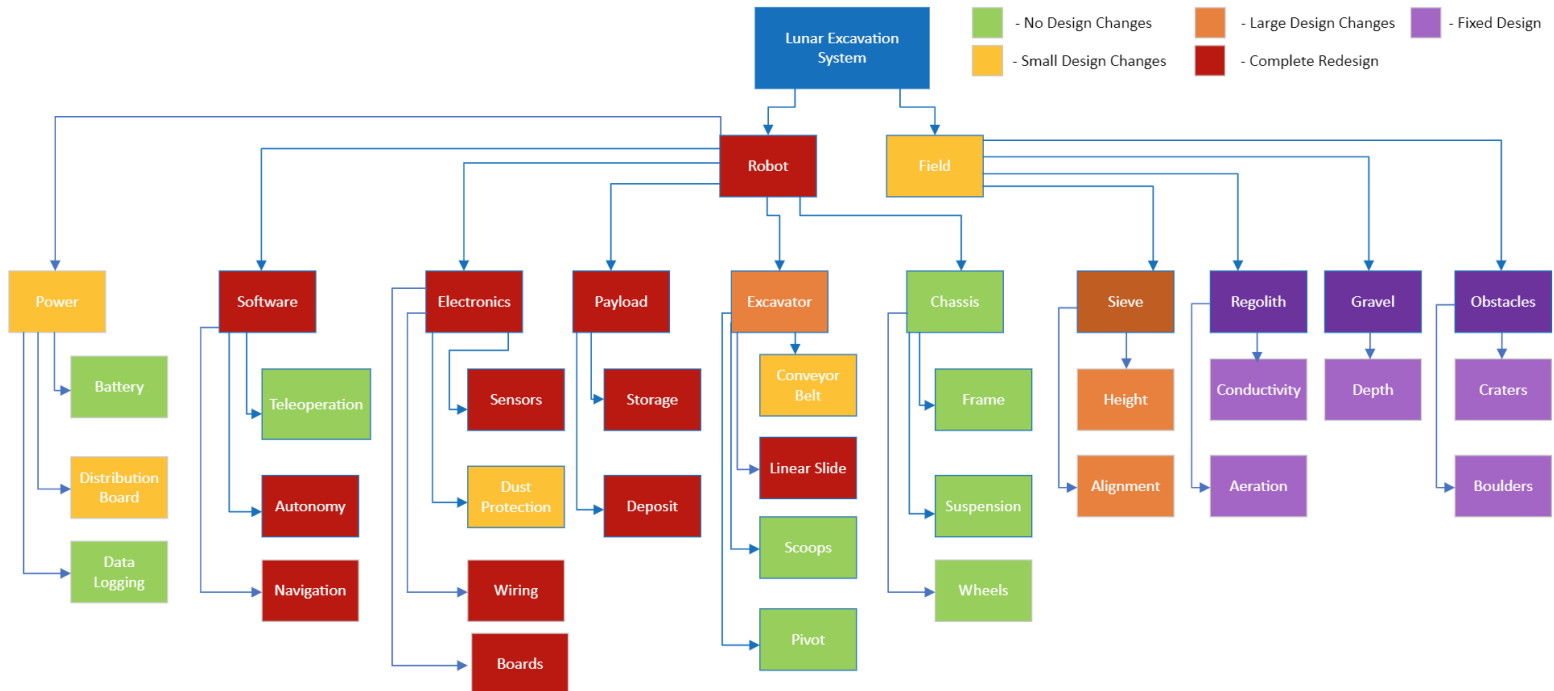


Figure 10: The system hierarchy for the robot and its surroundings

Per the System Overview, the subsystems and robot functionality were modeled to follow the Operations Criteria from the Lunabotics 2023 Guidebook via the Systems Engineering process by NASA.

#### 3.1.1 Excavator

The excavator design was inherited from last year's team and worked relatively well as an overall system. The collection scoops worked well in digging sand and gravel. While no redesign was needed, some repair was necessary for some of the scoops that were broken. The sprocket and chain transmission that pivoted the excavator had some small tolerance issues in chain length, but no redesign of the system was necessary as it was able to provide enough torque to move the excavator between configurations.

The main criteria for the redesign of the excavator was to increase the speed at which it could move linearly. The team decided that shortening the previous speed of 2 minutes and 30 seconds by 1 minute would be considered an achievement of this goal. The primary constraint on this redesign was being able to increase speed while still providing an adequate amount of linear force to dig into the sand and gravel of the test environment.

The secondary criteria for the redesign of the excavator subsystem was to fix the critical failure in the belt tensioner that caused damage to the transmission and its housing. A successful tensioning redesign would keep the belt in tension for the duration of the digging period while protecting critical components like the transmission from coming into contact with the belt. The main constraint for this design was the number of viable locations on the linear slide system to mount a tensioner without interfering with linear movement.

The tertiary objective of the excavator redesign was to incorporate dust and material protection to prevent obstruction in the belt movement. An ideal protection system would keep all forms of foreign material out of the excavator, however because the excavator will be deep into sand while digging, the team considers the protection against any significant amounts of material that could obstruct belt movement a success.

### **3.1.2 Storage and Deposit**

The storage and deposit system design was a new design created from scratch to replace the conveyor system from last year. The conveyor had several issues, with the biggest being that it significantly reduced ground clearance when in the storage configuration and could not get gravel onto the sieve during a competition run. The ground clearance issue negated the chassis's main strength, which was its ability to traverse obstacles, and could even lead to the robot getting stuck while driving back to the sieve.

The primary objectives for the storage and deposit system were to solve the ground clearance issue and to make a system that could deposit gravel onto the sieve consistently. A successful design would not significantly reduce the ground clearance of the robot when depositing. Ideally, there would be no changes to ground clearance when depositing. The design would also be able to consistently deposit gravel on the sieve without having gravel fall out or get stuck. The chassis of this robot was the most significant constraint with the design as it limited the space available for a new system. System integration in general with a robot not designed for a different storage and deposit system was challenging.

Other objectives for the new storage and deposit system included reducing the overall mass of the system and reducing the amount of time it takes to deposit a load of gravel. Reducing the mass of the system allowed the robot to score better in competition, and reducing the deposit time gave more time for the robot to do other tasks, such as digging up more gravel. These were similarly constrained by the chassis of last year's robot having to be used to house a system it was not originally designed to.

### **3.1.3 Vision/Navigation**

The previous WPI Lunabotics had not attempted autonomous navigation, creating a novel challenge for this team. Given the complexity of autonomous navigation, the system was divided into three subcategories: localization and odometry correction, map generation, and path planning. To accomplish autonomous navigation, the team developed a dual camera system utilizing a traditional webcam (Logitech c920) and a depth-sensing camera (Intel RealSense D455i). The information from these cameras would be managed and processed by a Raspberry Pi 4B. Each of these subsystem components was critical to the function of the robot allowing it to traverse from the starting zone to the excavation zone and back. Each subsystem required

rigorous testing to ascertain whether the system met the design specifications that are detailed below.

Computer vision-based localization and odometry were necessary due to the sandy environment in which lunar robots operate. Initially, a fixed camera was considered for localization; however, wheelbase odometry proved unreliable during early testing. Therefore, a computer vision approach was selected to provide localization and odometry correction, using a fiducial marker called an AprilTag. This marker is a fixed size known to the robot and contains a series of black and white squares to represent information visually to the camera, similar to the techniques used in a barcode. These characteristics allow for six degree of freedom localization of the robot with respect to the AprilTag. Therefore, an actuated camera system was designed to track an AprilTag and provide localization and odometry correction. To give context to the fiducial marker in the scope of this project, an AprilTag is used to mark the final delivery point for the regolith that has been mined (See Figure 11). The success of this functionality depends on the consistency and accuracy of the localized data. To determine success, the subsystem must find and provide localization information consistently and accurately. This includes re-identifying the AprilTag if the tag is lost during the process of the robot traversing the field, and ensuring the AprilTag stays within the frame as the robot navigates. In addition, the robot must be capable of producing this information quickly to avoid hindering navigation. To be considered accurate, the localization data derived from the computer vision approach must be within  $\pm 2$  cm at a distance of 2.5m or closer and  $\pm 3.5$  cm if further than 2.5m.

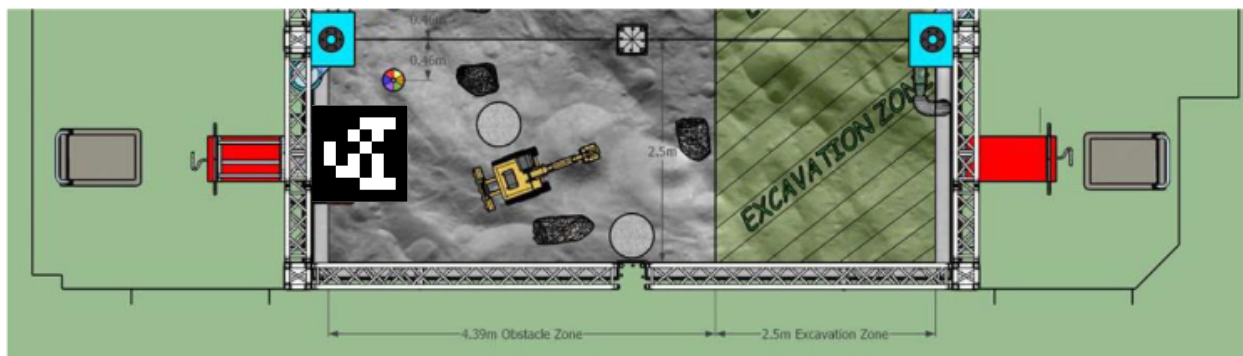


Figure 11: The field the robot had to navigate with an AprilTag placed at the location of the sieve. Adapted from the NASA Lunabotics 2023 Guidebook

After the robot has been localized within the frame, it is necessary to identify objects that could interfere with the robot's navigation to the excavation zone. In order to accomplish this, a point cloud was generated from the RGB-D (Red Green Blue-Depth) data provided by the Intel RealSense D455i camera mounted on the front of the robot. The D455i camera accomplished this task by aligning information from its two stereo depth sensors and its regular RGB camera. The two stereo sensors collected infrared noise and then compared their images. Given the distance between the two cameras was known and fixed, it was possible to generate depth information from the corresponding image. This depth information was then converted to a point cloud where extraneous and unnecessary data was filtered out. After filtering, the data was projected onto a two-dimensional plane and represented by an occupancy grid that was used to navigate. The success of this core functionality was determined by quick and accurate map data



for the robot's navigation. To determine success in terms of accuracy, the map generated must represent objects within three percent of both object size and location. Success in terms of speed was displayed by map generation occurring at a minimum of three times per second with a preferred goal of three times per second.

Once the mapping of the robot's current environment was complete, the robot can be actuated toward its end goal: the excavation zone in the playing field. To navigate toward the excavation zone, the map is utilized to perform path planning for the robot. This was accomplished by using several techniques that will be covered in-depth later. However, the basic system will require C-Space generation, frontier detection, and a path-planning algorithm. The success of this core functionality depends on generating a path in a usable form for the roboRIO (robot controller), developing an efficient path in terms of time and energy consumption, implementing a margin of safety during navigation, and using an algorithm that was computationally quick. To be considered successful in terms of the core requirements, the robot must communicate the path to the roboRIO in a usable manner, identify the shortest safe path from the starting location to the end location, avoid the implemented C-Space (margin of safety), and generate a path in less than 0.5 seconds from any pair of locations within the playing field.

### **3.1.4 Autonomy Integration**

The incorporation of autonomy integration was a significant upgrade to the previous year's robot. Our team aimed to achieve a certain level of autonomy by enabling the robot to perform tasks without human intervention. To this end, we set a project goal to develop autonomous operation by localizing the robot, detecting and avoiding obstacles through path planning, and enabling each subsystem to operate autonomously. We planned to achieve this by incorporating sensors for each subsystem, which would help identify the current state of a subsystem and signal transfer over to the next state of the robot. Successful integration of autonomy ensured that the robot could respond to mechanical stimuli or completion of a certain state.

The primary objective of the project was to enhance the efficiency and reliability of the robot by leveraging sensor technologies to achieve autonomous operation. For instance, we used cameras to enable the robot to localize and path plan autonomously, while sensors in the excavator facilitated accurate movement and digging. Hall effect sensors helped determine the precise position at which the excavator should start and stop digging, while pressure sensors determined when the bucket was fully loaded, signaling the end of the digging process and the need to move to the dumping zone. These goals were for these sensors to work in harmony to facilitate autonomous operation of the robot. Encoders were also used for all the motors to provide readings, enabling the robot to drive to specific points, make turns, reconfigure the excavator, and raise the bucket to dump the gravel. The Conops chart in Figure 12 depicted the different states that the robot had to iterate through to complete a competition cycle, and our aim was to achieve autonomous operation through successful sensor integration.

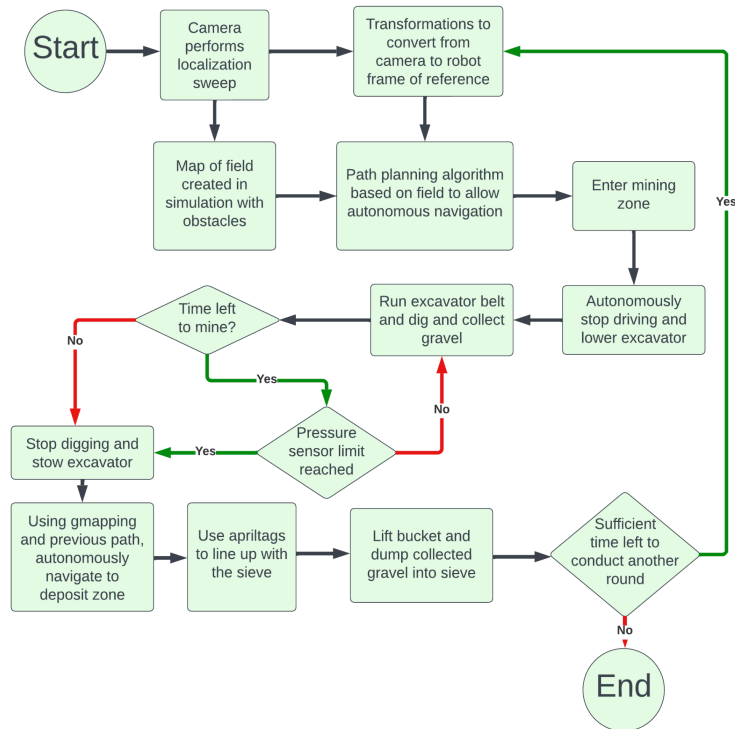


Figure 12: Conops Diagram representing the overall functioning of the robot

### 3.2 Conceptual Designs and Prototyping/Modeling Feasibility Studies

This section will cover the early design process and concepts used for the major subsystems of the robot.

#### 3.2.1 Excavator

After an initial assessment of the robot, it was observed that the linear slide and top conveyor, designed for excavation, were still partially functional. The linear slider still had its full range of motion, while the sliding mechanism was in good condition. However, the conveyor belt itself was not running smoothly, as it would stop after a few seconds of operation, and the motor would overheat. Upon visual inspection, it was noted that there was a slight tension in the belt, which could be the root cause of the problem.

The team focused on disassembling the excavator to identify any issues that may have occurred after the competition. During the disassembly process, the team discovered many problems that may have been caused by negligence from the previous team. The primary issue that was identified was that the belt tensioner from the previous design had been under tension in the wrong direction for the entire summer following the competition (Figure 13 and Figure 14). Over time, this caused the spring system that applied tension to the belt to move out of its set position, eventually leading to the springs' failure and the tensioner's collapse onto the linear slider's transmission system.

This discovery not only explained the lack of tension in the belt system observed earlier, but it also shed light on the overheating of the linear slider's driving motor. The gears were rubbing against the belt tensioner, resulting in large amounts of friction which stopped the gears at some point. As a result, the driving motor was drawing more current than necessary to turn the transmission system. This extra current draw caused the motor to exceed the designed electrical draw specifications, leading to its overheating.

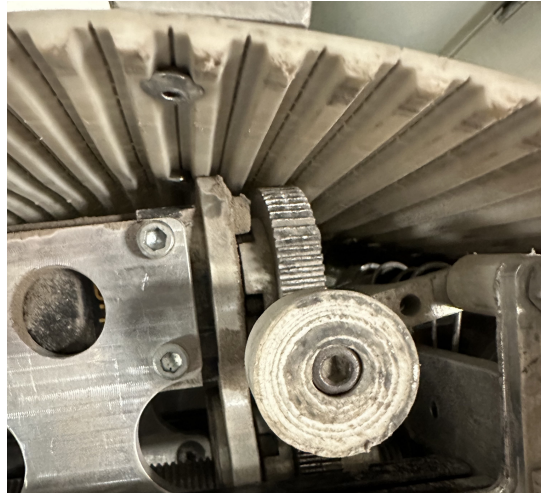


Figure 13: Belt tensioner failure

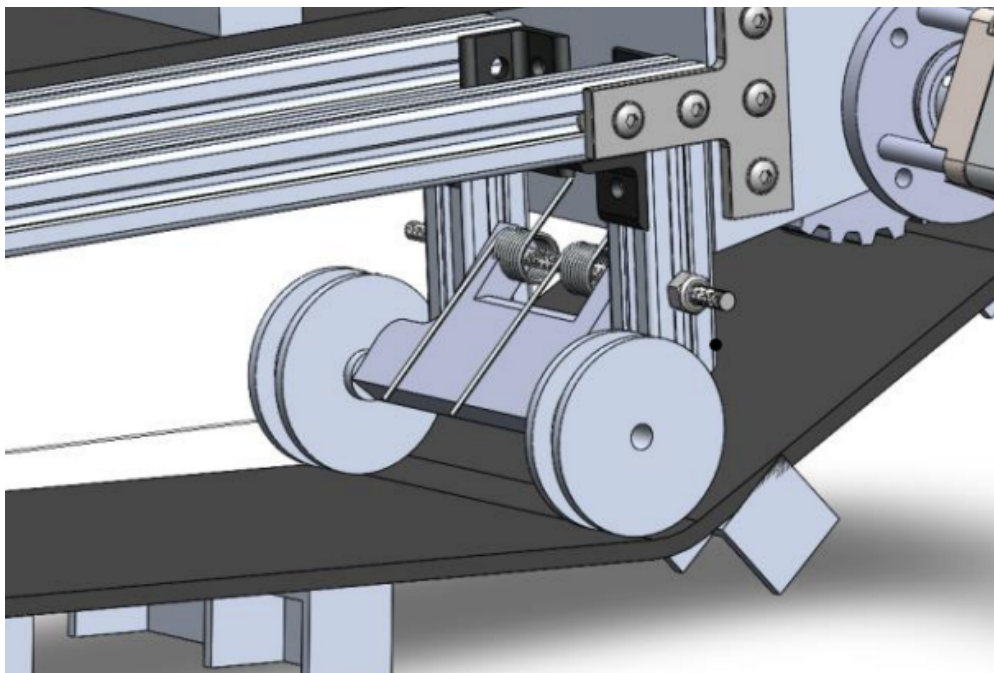


Figure 14: Designed belt tensioner placement

In addition to the aforementioned issues with the conveyor belt, the team also found that the failure of the belt tensioner had caused a misalignment in the gears of the linear slider's transmission system (Figure 15). Although the gear ratio was a direct 1:1 ratio, the misalignment still presented problems. Upon observation, the team noted that the linear slide system would stop moving approximately every 5 seconds. While some of this failure could be attributed to the belt slip previously mentioned, the team concluded that most of the failure was caused by the slipping of gears in the transmission system due to misalignment.

The misalignment in the transmission system led to a decrease in the system's efficiency, which affected the overall performance of the robot's excavation system. This reduction in performance was a critical issue that needed to be addressed to improve the robot's functionality. To address this problem, the team needed to realign the gears in the transmission system, which required a complete disassembly of the subsystem.

To prevent such problems from recurring, the team considered using a more robust and durable tensioning system that can withstand prolonged use without the opportunity to damage other systems. This redesign involved moving the tensioner to the opposite end of the linear slider where more slack was present and adding an additional roller above the contact point for the gears to ensure the belt would not interfere with the transmission.

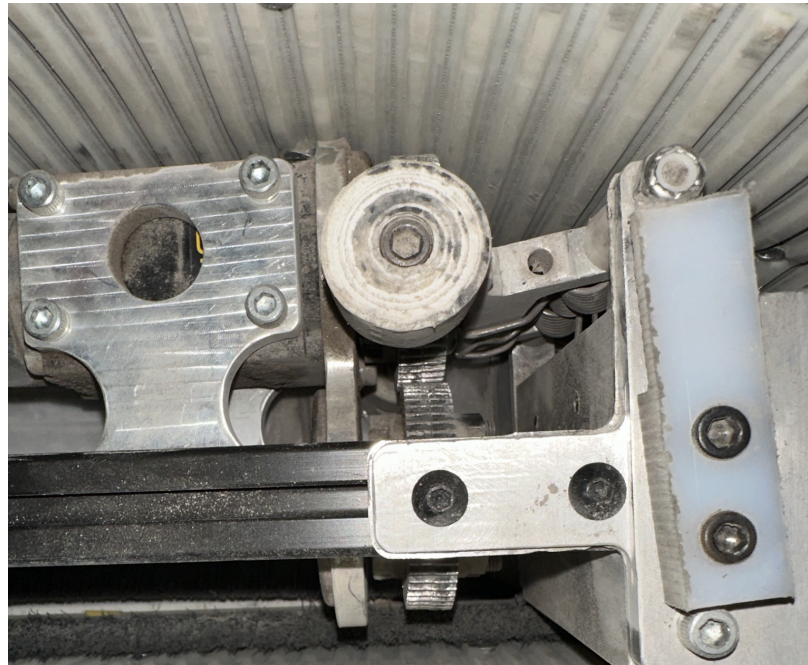


Figure 15: Misaligned gears due to belt tensioner

In addition to the critical failures previously mentioned, there were other areas of the robot's design that required improvement. One of these areas was the time it took for the robot to move from the top of the linear slider position to the bottom digging configuration. After a thorough analysis, this delay was attributed to the high thread per inch count in the lead screw used for movement. The high thread count resulted in a disproportionate relationship between the force applied and the speed of the system, causing the system to move slower than desired.

To further investigate this issue, the team reviewed last year's force analysis of the entire robot and excavation system. The team discovered that the torque required to dig successfully

was significantly lower than the torque provided by last year's slider. As a result, there was an excess of applied force that could have been more useful to the digging mechanism as a whole. This inefficiency led to additional delays in the sliding system's movement.

Furthermore, the lead screw used for the sliding system was made of dated scrap steel found in Washburn shops. As a result, the screw was rusted, which had a high coefficient of friction and further contributed to the delays in the movement of the sliding system. This delay was a critical issue that needed to be addressed to improve the robot's performance.

To address this issue, the team considered using a lead screw with a lower thread per inch count, which provided a better balance between force and speed. Additionally, the team used a more durable and rust-resistant material for the lead screw to reduce friction and improve the sliding system's efficiency. These measures helped to decrease the time it takes to move from the top of the linear slider position to the bottom digging configuration, improving the robot's overall performance during excavation.

### **3.2.2 Storage and Deposit**

This year's storage and deposit goals were developed based on issues last year's team had with ground clearance and getting gravel from the storage system into the sieve. Last year's robot would have trouble driving when the storage conveyor was in the depositing configuration (see Figure ), as the bottom of the storage conveyor was low enough to get caught on the ground and make it difficult for the robot to line itself up with the sieve. It also had issues getting gravel from the storage area to the sieve once it was lined up. Last year's team added an extender plate in an attempt to remedy this issue, but gravel would get stuck on the extender plate instead of falling onto the sieve. Attempted tests of the storage conveyor found that its angle could be adjusted, but the conveyor would not run. During disassembly, the team discovered that the PLA pulleys used by the conveyor had warped due to excessive thermal loads and likely stopped the storage conveyor belt from moving.

After identifying these issues, the team researched other designs, as seen in section 2.2, and found that most teams did not use a conveyor belt to deposit the gravel into the sieve. Instead, most teams used a bucket, including last year's winner, Montana Technological University (Montana Tech NASA Robotics, 2022). Going to a bucket for our storage and deposit system would solve the ground clearance issue as the bucket would never have to go lower than its starting point when depositing, and it could easily actuate to a dumping position where gravel would not get stuck when depositing. It was also a simpler system than a conveyor belt, with many teams using some form of a single axle pivot at the back of the robot that was powered by an electric motor. Previous WPI teams had also used a bucket such as Markhor in 2017, as seen in Figure 16. This motion is very similar to that of a dump truck unloading.





Figure 16: WPI 2017 Lunabotics Robot, Markhor. (Wamback, 2017)

To determine whether to try and fix the issues with the current storage conveyor or to make a new bucket system for the storage and depositing of gravel, the team made a decision matrix which can be seen in Figure . This is how we ultimately decided to design a new storage system from scratch. The two options we decided between were evaluated on various criteria and given a grade from 1 to 5, with a 1 indicating less than desirable performance in that criteria to a 5, representing close to ideal performance. Ultimately, despite the conveyor already being on the robot, the team was unsure of how to solve the ground clearance and dumping issues while still being within the design constraints (especially dimensional) set by the Lunabotics rules. It was easier to start developing a concept for a bucket that could solve the issues faced by last year's storage system.

Criteria	Priority	Motor Bucket	Conveyor
Low Cost	2	3	2
Easy to Manufacture	4	4	2
Low Mass	4	3	3
Small Size	1	5	3
Good Dust Protection	3	4	2
High Storage Volume	4	3	4
Fast Dumping Speed	5	4	3
Low Mechanical Complexity	4	5	4
<b>Total</b>		<b>31</b>	<b>23</b>

Table 3: Initial Storage & Deposit Decision Matrix.

After determining that switching to a bucket for the storage and deposit system was the best way forward, the team made an initial design in CAD which can be seen in Figure 17. This design used  $\frac{3}{4}$ " by  $\frac{3}{4}$ " aluminum square tubing with a  $\frac{1}{8}$ " wall thickness. The gusset plates and backplate were made from  $\frac{1}{16}$ " thick aluminum. A servo actuated extender plate was added to hold gravel in the bucket before it reached its final position to deposit. This extender plate was also made from  $\frac{1}{16}$ " thick aluminum.

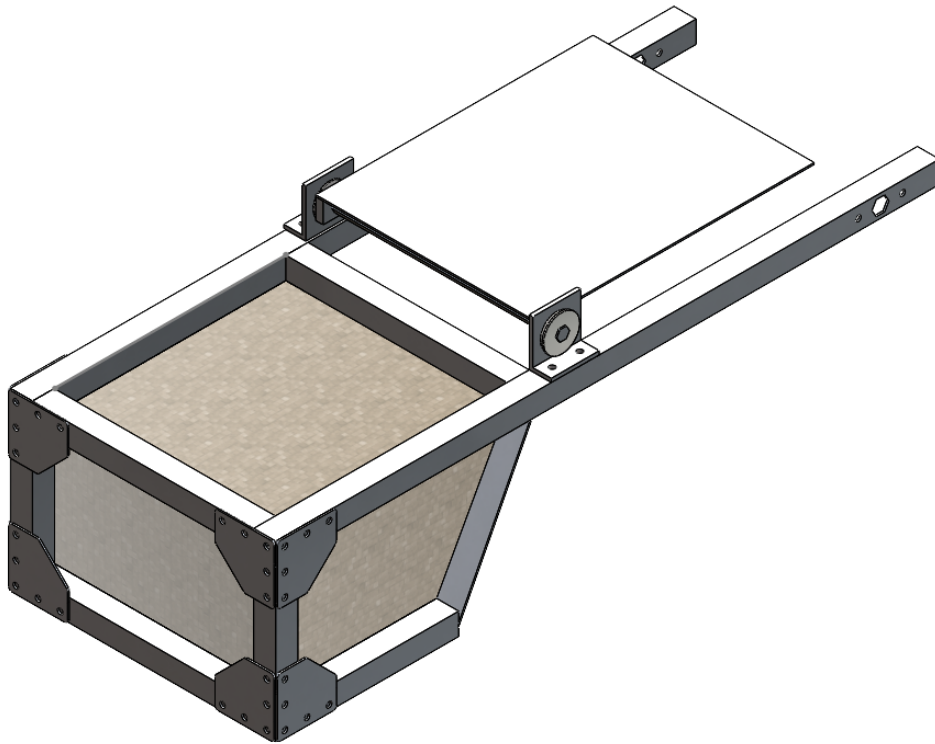


Figure 17: Initial design for the bucket

Once we created a design for the bucket in CAD, the team built a prototype to test its viability. The prototype for the bucket system was made from two pieces of lumber for the arms and sections of hot-glued cardboard, as seen in Figure 18. The axle was made from a cut section of PVC pipe and laser-cut wood scraps from last year, representing how we could attach it to the chassis. An extender plate made of cardboard was actuated by a servo which held on with duct tape and powered by a power supply for testing. With last year's team and our background research demonstrating the effectiveness of a metal mesh to filter lunar regolith, we decided to make the filters out of the remaining cardboard to save some money and time. On the final version of the bucket, the mesh would be placed on the inside of the bucket to stop gravel from getting stuck in the corners created by the frame rails. These cardboard pieces were placed at intervals small enough not to let gravel fall through, with gravel size being determined by the bag of gravel we bought for testing.



Figure 18: Prototype of the bucket system

The prototype bucket had to be moved manually, but the servo was tested by operating it with a power supply similar to how the robot would provide power to a servo used on the actual robot. The team used a tall bucket to represent the height of the sieve and placed the bucket at a height representative of where it would be compared to the sieve on the actual robot for testing. The testing yielded promising results, with all of the gravel falling out of the bucket and making it into the tall bucket, and the servo could hold the gravel inside the bucket until it was time to deposit the gravel. This confirmed to our team that this design was viable and that we could move forward with building it.

### 3.2.3 Vision/Navigation

Several components and concepts were evaluated in the early development stage of the computer vision and navigation subsystem. Using preliminary testing, our team studied and compared design choices to make decisions regarding the system's development. Building upon the section analysis discussed in Section 3.1.3, two critical design choices needed to be evaluated within the system before the development continued. These critical design choices are indicated in Table 4.



Needed Functionality:	Primary Design Options:
Localization method	<ol style="list-style-type: none"> <li>1. Fiduciary markers specifically AprilTags</li> <li>2. Infrared beacons</li> <li>3. Adaptive Monte Carlo Localization based on point cloud mapping</li> </ol>
Mapping technology	<ol style="list-style-type: none"> <li>1. LiDAR</li> <li>2. Depth sensing camera</li> </ol>

Table 4: Critical design requirements for the computer vision and navigation subsection

The first major design choice was selecting the method of localization within the navigation zone that precedes the excavation zone. The primary choices were fiduciary markers, infrared beacons, or an implementation of Adaptive Monte Carlo Localization (AMCL). Each of these options were investigated; the fiduciary markers and infrared beacons were selected for preliminary testing. The team performed research on AMCL to determine its usefulness after an initial mapping was done; however, the algorithm did not fulfill the needed functionality. According to the Lunabotics rule book, the walls of the playing field are not permitted for navigational use which would make AMCL localization challenging. Additionally, AMCL is costly in terms of computation and time; therefore, it was removed from consideration. Preliminary testing was performed with the two other methodologies, evaluating their performance in the following categories: location accuracy, orientation accuracy, ease of implementation, effective range, and the number of sensors/markers needed.

	Minimum Fucntionality:	Fiduciary Marker	Infared Beacon
<b>Location accuracy</b>	±0.08m at a range of 4m	Pass (up to 5m)	Fail (data becomes unreliable at 3-3.5m)
<b>Orientation accuracy</b>	±3-5 degrees at a range 4m	Pass (up to 5m)	Due to more complex implemented skipped
<b>Ease of implementation</b>	Scored from 1-5	2	3
<b>Effective range</b>	Minimum of 4m	Pass (up to 5m)	Fail (data becomes unreliable at 3-3.5m)
<b>Number of sensors required</b>	Less is preffered	Single camera	Single infared camera
<b>Number of markers required</b>	Less is preffered	1	4

Table 5: Preliminary assessment of localization methods

Table 5 demonstrated the feasibility and preliminary assessment results for both the AprilTags and the infrared beacons. Based on these results, the AprilTag was selected for the ongoing development in the localization system.

Another critical design decision was selecting between different mapping technologies. LiDAR and depth-sensing cameras were the two technologies investigated. Their preliminary assessment considered the following factors: accuracy of mapping, size and mass of the system, ease of integration into the current robot (scored one to five with one representing easiest integration), power consumption, and the ability to detect craters/obstacles beneath the ground. The results of this preliminary assessment are documented in Table 6. After analyzing the results, the depth-sensing camera was selected as the primary mapping sensor. This was because it had similar functionality to the LiDAR (within a 5m range) while possessing a more usable form

factor and consuming less power. Although the LiDAR was more accurate, the depth-sensing camera provided reasonable measurements that would be acceptable for the mapping and navigation processes.

	Minimum Fucntionality:	LiDAR System (2D LiDAR Sick TiM)	Depth Sensing Camera (Intel Realsense D455i)
<b>Mapping accuracy</b>	±0.01m at a range of 1m	Pass (up to 5m)	Pass (up to 5m)
<b>Size/Mass</b>	Small form factor and mass (less 1kg)	Pass, 0.25 Mass 60x60x86mm	Passed, 0.388kg 124x26x29mm
<b>Ease of implementation</b>	Scored from 1-5	5 (Failed) no position where laser scan is unimpeded, unique connectors used for data communication	1 (Passed) Small form factor, and standard usb communication allows for easy intergration
<b>Power consumption</b>	Compared based on information provided from datasheets	~ 4-16w	~2 watts
<b>Identifaction of low elevation obstacles</b>	Pass/Fail	Failed - horizontal laser scans were unable to detect features	Inconclusive - mixed succes with certain camera angles

Table 6: Preliminary assessment of different mapping technologies

### 3.2.4 Autonomy Integration

The software design of the robot was broken down into three major parts. The roboRIO with the radio is used to interface the physical aspects of the robot, such as the wheel motors, the bucket, and the linear slider. The goal was to control all these autonomously with feedback from our sensors - a LiDAR and a RealSense camera that a Raspberry Pi 4B will control.

In order to achieve the team’s goal of an autonomous system, encoder values had to be extracted from the motors to implement driving with feedback. The existing software infrastructure utilized a Raspberry Pi, which communicated with the Spark Max motor controllers over PWM. Unfortunately, PWM cannot receive encoder values, so the team implemented a CAN bus instead.

On the CAN path, through research, it was learned that there is not a great way to incorporate CAN with a Raspberry Pi, so a CAN shield extension board (Figure 19) was ordered to mount onto the Raspberry Pi. It was still not able to fix the issue even after this and enabling SPI on the Pi (the CAN data had to be converted to SPI protocol to be recognized by the Pi). After more research and consultations with WPI’s resident FRC190 team, it was established that if the team continued using CAN with the Pi, a custom CAN network overlay would have to be written to communicate with the Pi effectively.

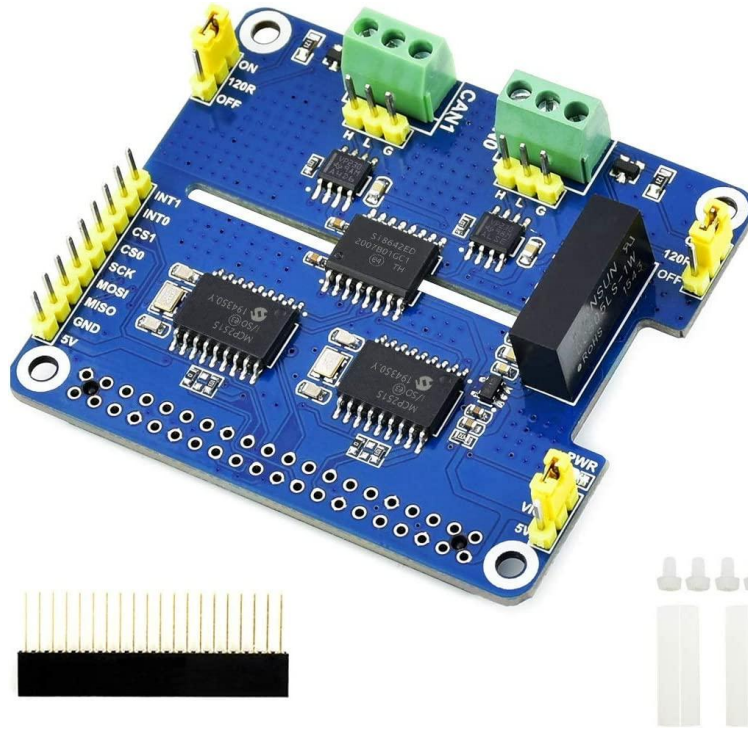


Figure 19: 2-Channel CAN hat for Raspberry Pi (Amazon)

The next course of action was to investigate other microcontrollers that could be used. The highest contenders were the VMX Robot Controller and the CTRE Hero board. Both had their pros and cons, which we discussed and evaluated. The VMX Robot Controller seemed to accommodate all needs and wants. Its major drawback was its cost, with a far-off shipping date, which was decided would not be worth the time and money given the project constraints. The Hero board was far cheaper and highly recommended, though no clear documentation indicated that it could be used with the Spark Max Controllers as they were meant to be used with the VEX Talons.

Through FRC team 190, the team acquired an FRC roboRIO 2.0 (Figure 20) and radio. Both devices were flashed, imaged, and configured as required by FRC specifications. The radio is used as an intermediary connection between the roboRIO and the network through ethernet. Through these devices, the team was able to communicate and transmit data to and from the Spark Max motor controllers wirelessly over a Wifi connection.



Figure 20: roboRIO 2.0 (Studica)

### 3.3 Design of Core Functions

This section describes the design of the final solutions used for the core functions on the robot and the design process to arrive at these solutions.

#### 3.3.1 Excavator

In order to determine the amount of linear force the excavator should output, we started by revisiting last year's free-body diagram calculations (Figure 21). The figure below is last year's diagram for the robot in the digging configuration. The team concluded that using last year's free body diagram was applicable as the excavator and chassis systems have remained approximately the same in size and mass. Note that some of the values of these variables were translated from last year's paper (Section 5.2.2). Table 7 represents the variables used in a conservative analysis as in actuality the mass of the robot was decreased over the course of the project.

Variable	Definition
$W_e$	Weight of excavator = 19kg * G
$W_r$	Weight of robot = 43kg * G

$F_f$	Reaction force on front wheel
$F_r$	Reaction force on rear wheel
$F''$	Tractive force on wheels
$F_x$	Force of ground on excavator (x-axis) = 400N
$F_y$	Force of ground on excavator (y-axis)

Table 7: The variables for analysis of the excavator

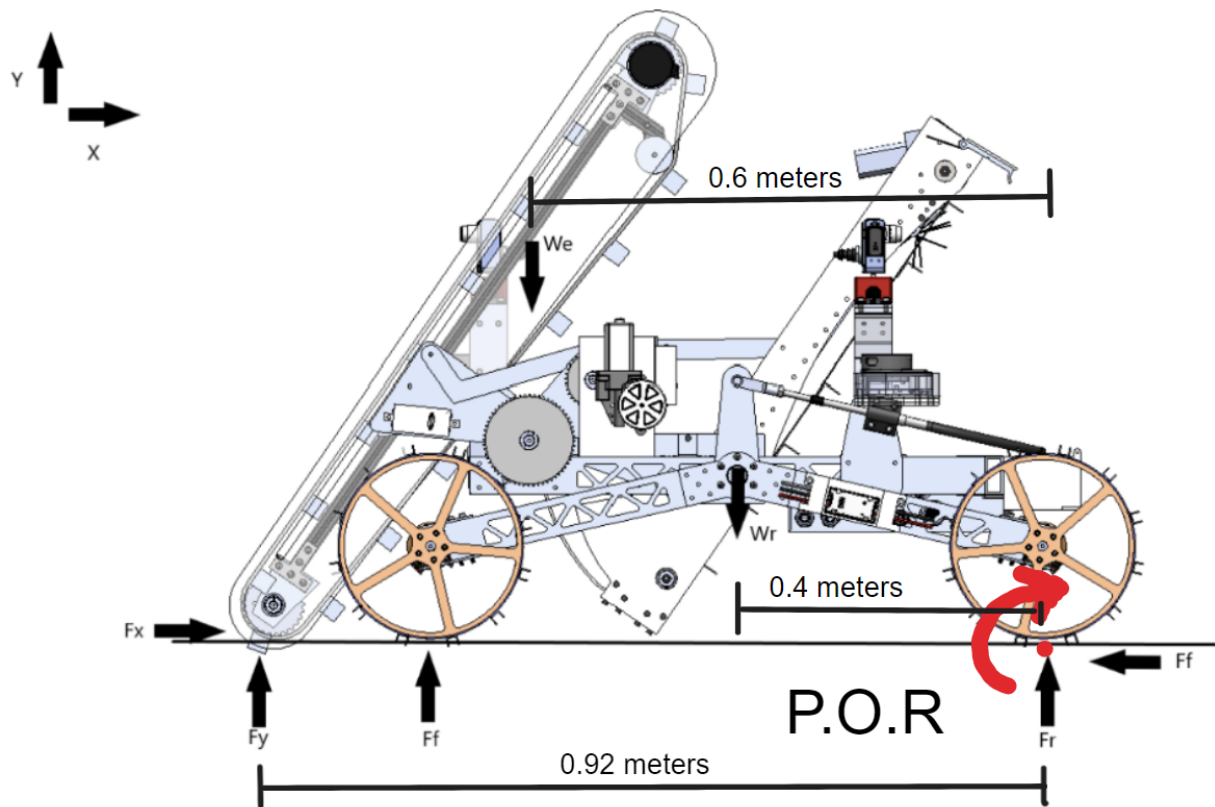


Figure 21: Free body diagram of the robot when it is digging. Adapted from Hartman (2022)

Using Figure 21, we were able to derive a set of equations. These calculations solve for the maximum linear force that can be applied by the excavator before the robot tips about its rear wheels. This number represents the upper bound of the amount of force that the linear slide system can provide, so we know moving forward that our new system should not output more force than this.

*Before Digging:  $F_f \equiv 0$*

$$EQ1: \sum F_x = F_x - F'' = 0$$

$$EQ2: \sum F_y = F_y - W_e - W_r + F_r = 0$$

$$EQ3: \sum \tau = -(0.92(m)*F_y) + (0.4(m)*W_r) + (0.6(m)*W_e) = 0$$

$$\text{From EQ3: } (0.92(m)*F_y) = 280N$$

$$F_y = 304N$$

$$F_x = 400N$$

$$F = \sqrt{(F_x)^2 + (F_y)^2} = 502N$$

Before making any mechanical changes to the robot, it was essential to verify that the driving motor for the linear system was operating at the best parameters for providing both desired force output and speed requirements. To accomplish this, the team thoroughly analyzed the data supplied by the VEX Robotics spec sheet. In addition, the team created a separate data table to ensure the accuracy of the motor's torque output. This was done using a template (Stafford, 2023) that takes in the base specifications of the motor and projects its output and efficiency for each speed. Table 8 is for the VEX Bag Motor assuming a planetary gearbox with an e of ~0.01. The gearbox coefficient is approximate because the team could not find updated documentation from last year's team following the competition, and the planetary gearbox is in a very difficult position to remove. Instead, the team observed the physical relationship between a set motor speed and the output speed of the gear to come to the current conclusion.

Speed (rpm)	Torque (N m)	Torque (in lbs)	Current (A)	Power (wt)	Efficiency	Heat (wt)
0	33.91	300.00	41.0	0	0%	492
9	31.65	280.00	38.4	29	6%	432
17	29.39	260.00	35.9	53	12%	377
26	27.13	240.00	33.3	74	18%	326
35	24.87	220.00	30.7	90	24%	279
43	22.61	200.00	28.2	103	30%	235
52	20.35	180.00	25.6	111	36%	196
61	18.08	160.00	23.0	115	42%	162
69	15.82	140.00	20.5	115	47%	131
78	13.56	120.00	17.9	111	52%	104
87	11.30	100.00	15.3	103	56%	81
95	9.04	80.00	12.8	90	59%	63
104	6.78	60.00	10.2	74	60%	49
113	4.52	40.00	7.6	53	58%	38
121	2.26	20.00	5.1	29	47%	32
130	0.00	0.00	2.5	0	0%	30
<b>Motor Type</b>	<b>Bag Motor/Planetary</b>					
Desired Volt	12	V				
Ref Volt	12	V				
Ref Free Spd	130	RPM				
Ref Stall Torq	300	in-lbs				
Ref Stall Cur	41	A				
Ref Free Cur	2.5	A				
Weight	0.71	Lbs				

Table 8: BAG Motor calculation sheet (Stafford, 2023)

After analyzing the data and comparing it with the system's requirements, the team concluded that the motor was operating at its best for the amount of RPM and torque required by the system. Specifically, the team observed the operating speed of the motor while driving the linear system by using a marker that could be used to determine rotation and a slow motion camera to ensure the count was accurate. After filming for 1 minute we determined that the motor was operating at approximately 60% efficiency, with an output of roughly 60 in\*lbs of torque. This level of efficiency was considered satisfactory for the system's requirements, and the team concluded that the motor should be kept under a 20 amp current draw for future use in order to keep it on the higher end of the efficiency curve shown in Figure 22. Given that the motor was not operating at a low rpm, the team decided to shift focus from improving the motor to improving the linear screw that it drives.



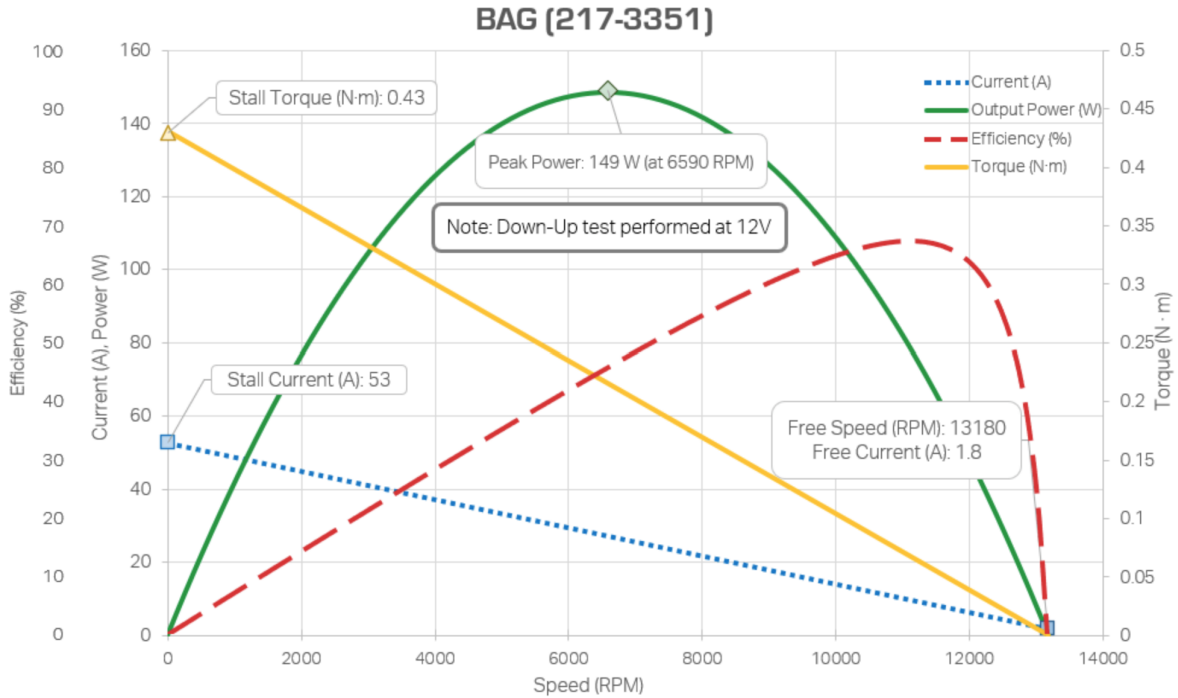


Figure 22: BAG motor efficiency graph (*BAG Motor*)

The team recognized that other areas of the robot's design could be improved to enhance the system's overall performance. As previously mentioned, the team planned to address the sliding system's movement delays by implementing a lead screw with a lower thread per inch count and using a more durable and rust-resistant material for the lead screw. In order to determine the specifications for the linear screw we had to start by analyzing the forces that would be placed on the screw(Figure 23).



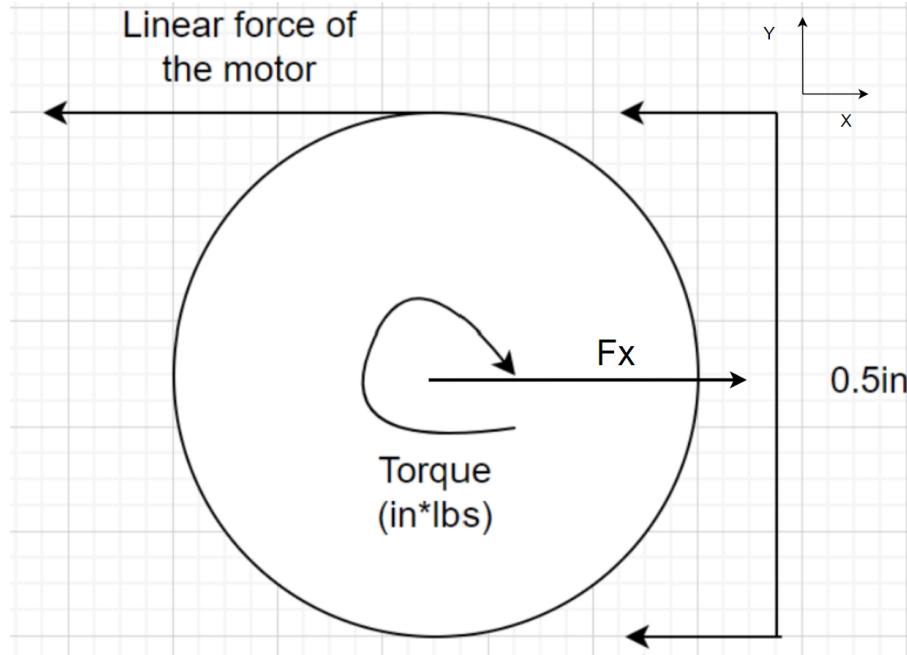


Figure 23: Free body diagram of torque applied to the linear screw

The team discovered that decreasing the thread per inch count to fulfill the desired speed requirements led to an increase in  $\alpha$ . Using this relationship we were able to find the maximum alpha angle that would satisfy the motor requirements of remaining under 20A of current draw. Using the maximum torque load of 15 (N\*m) and a free body diagram developed from a white paper on linear screws (Appendix D) we calculated the maximum helix angle for the system of approximately 10 degrees.

Upon conducting a free body analysis (Figure 24) on the lead screw system that uses the previously calculated torque requirements, a system of equations was established to determine the alpha angle of each screw, including the one used last year. We were able to compare the linear force-to-speed ratio change of a new linear screw system to the previous one. Using this process, we iterated through multiple screw options and determined how much of a difference there was in speed. In each screw we would determine its helix angle and compare it to the angle of last year's screw and the maximum angle available to meet the motor requirements. We decided upon a new angle of 7.26 degrees which increased the speed of the system while keeping in well within the bounds of the determined maximum angle

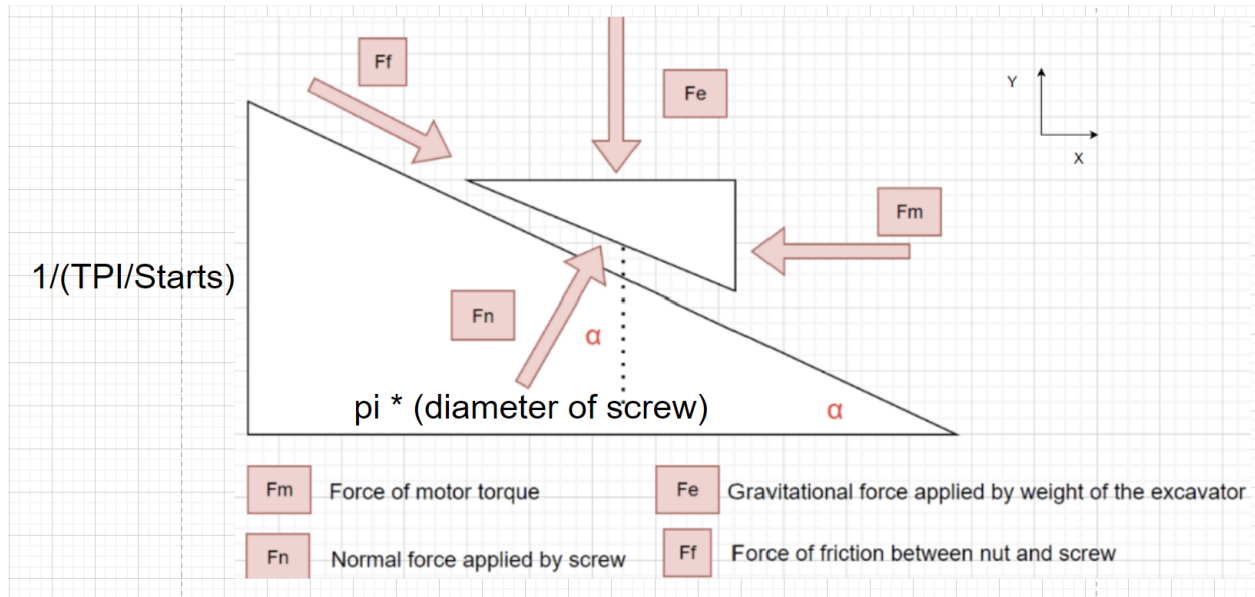


Figure 24: Free body diagram of linear screw and nut system

$$\alpha = \arctan\left(\frac{\left(\frac{1}{TPI}\right)}{\pi \cdot \text{Diameter of screw}}\right)$$

$$\alpha = \arctan\left(\frac{\left(\frac{1}{14(TPI)}\right)}{\pi \cdot 0.5(in)}\right) = 2.60 \text{ degrees}$$

$$\alpha = \arctan\left(\frac{\left(\frac{10(TPI)}{2(Starts)}\right)^{-1}}{\pi \cdot 0.5(in)}\right) = 7.26 \text{ degrees}$$

### 3.3.2 Storage and Deposit

With a simple motor-powered bucket, the storage system and deposit system became the same system with the bucket both holding the gravel that the robot has excavated so far and moving backwards in an arc to deliver the gravel to the sieve. With the storage and deposit system being the most significant mechanical redesign of a subsystem on the robot, the team took it as an opportunity to reduce the mass of the robot which would allow it to score better at the competition. As a result of this, the team designed the bucket frame to be made from 6061-T6 aluminum square tubing with a wall thickness of 1/8" and an overall outer cross section of 1" by 1". To hold the square tubing together, a set of 1/16" aluminum gusset plates were riveted to the square tubing using aluminum rivets. To ensure that the material would be strong enough to handle the mass of a fully loaded bucket, the team performed a FEA simulation (Figure 25) and found that the stress did not exceed yield strength of aluminum which meant that the minimal deformation would not be permanent.

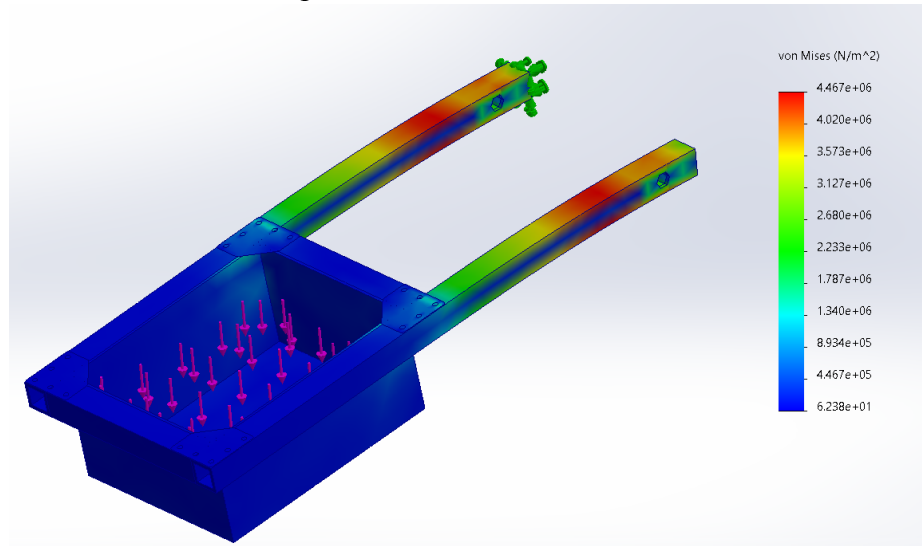


Figure 25: Stress of the bucket when load was applied which is less than the yield strength of 6061-T6 aluminum (240 MPa)

Originally, the team was just planning on making a bucket frame with no extension but there was a problem. If the bucket was mounted on the back of the chassis (which was about nine inches short of the maximum length the robot could extend back), gravel would fall out of the bucket too early as a result of the angled backplate in the initial design (see Figure 26). The calculations for this can be seen in Figure , which show that the backplate of the bucket will be flat once the bucket has reached an angle of 63 degrees relative to the chassis of the robot. Any further actuation would result in gravel falling out, but this gravel would fall onto the robot instead of the sieve.

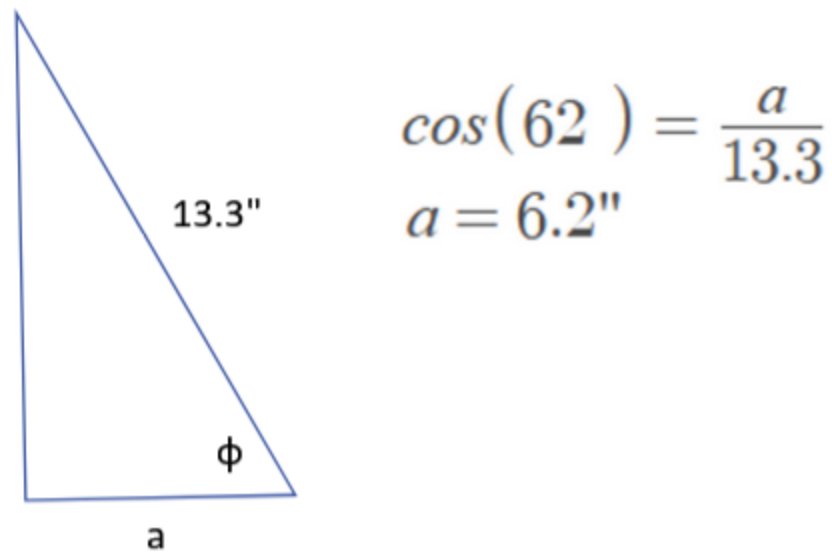
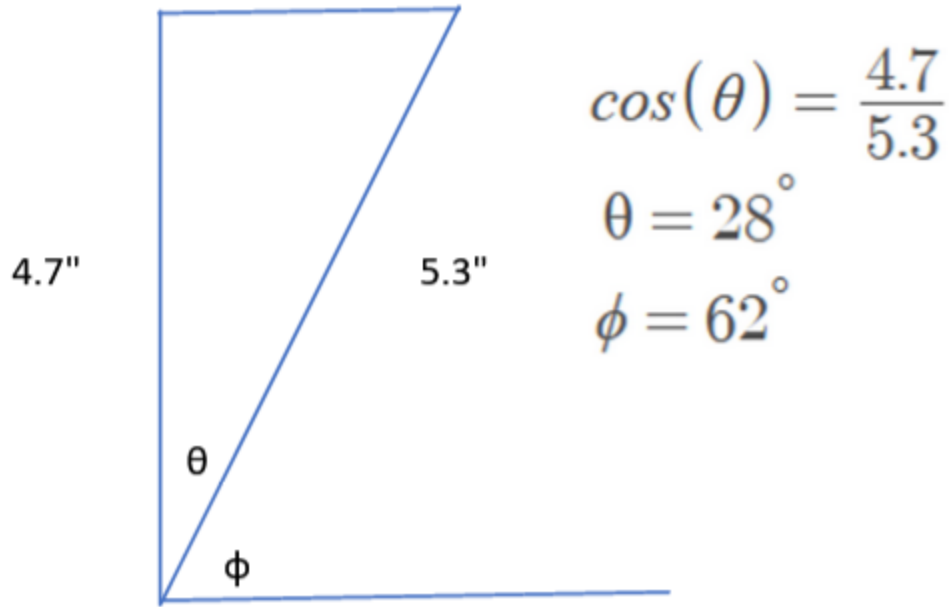


Figure 26: The calculations done to determine the angle at which the backplate would go flat and where gravel would start falling out of the initial bucket design. This led to design changes later.

To remedy this issue, the team changed the design to a square bucket and added a Lexan backplate to stop the gravel from falling out too early. This would also make it easier to dump gravel on the middle of the sieve instead of the edge of the sieve. The team chose this solution over the extender plate in the initial design because the Lexan plate setup was lighter and required no additional actuators. The final version of the bucket can be seen in Figure 27.

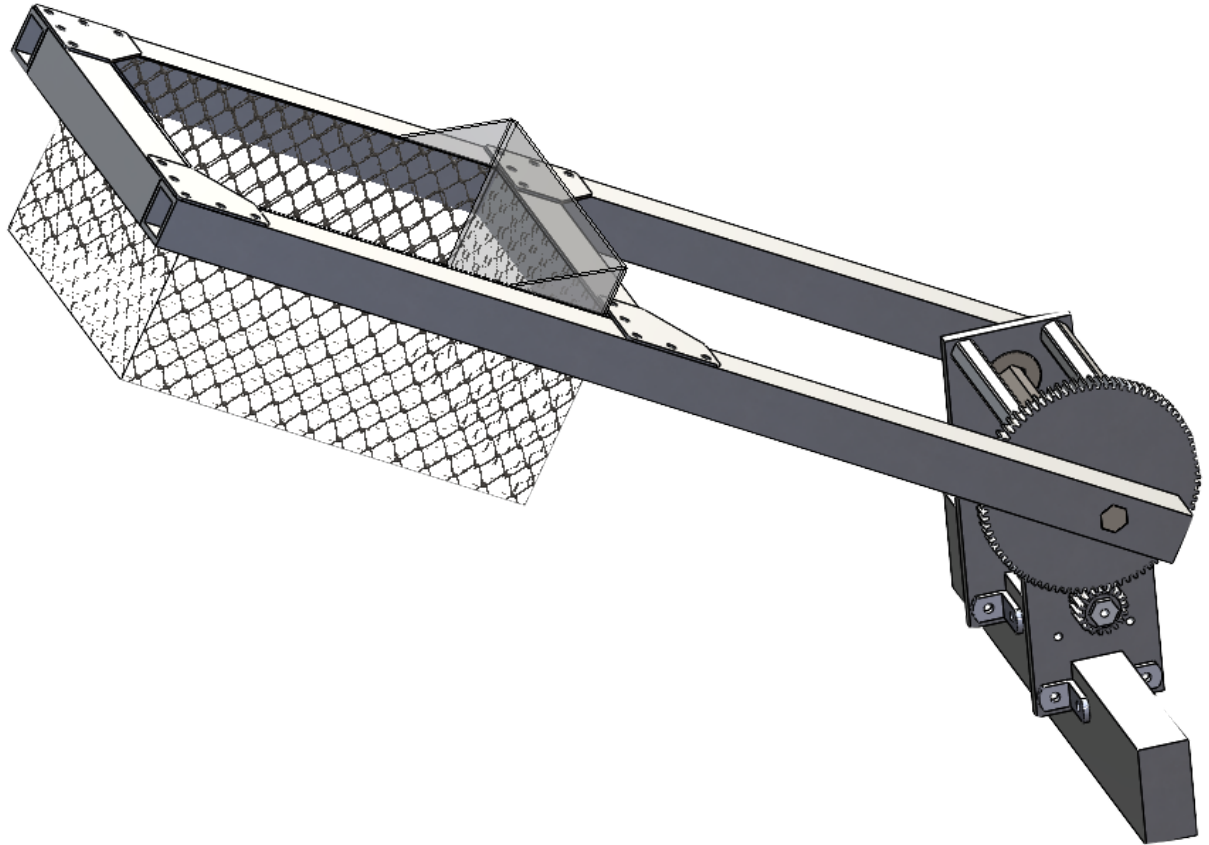


Figure 27: The complete CAD of the bucket and transmission system design before manufacturing

The bucket's actuation was initially designed to use gas springs, but the team soon discovered a problem with this method. The stroke length of the gas springs required to actuate the bucket to a 90-degree position relative to the chassis would have to be 24 inches long, and there was not enough space to fit two feet of gas springs without raising the bucket beyond what was feasible or introducing ground issues (one of the main issues we were trying to solve). The actuation method was switched to an electric motor with a transmission because the system could be placed horizontally above the chassis.

We first determined the transmission system's requirements through the calculations below. We calculated the total torque required to raise a fully loaded bucket about the point the bucket was mounted as seen in Figure . The required torque was around 30 Nm, and the change in potential energy when the bucket is raised and moved to an angle of 120 degrees is around 24 J. The amount of work done helps us determine the average power needed to raise the bucket, which was 6 W for raising the bucket in 4 secs.

Initially, we looked into the AndyMark Snowblower motor (am-2235a) and the AndyMark PG series PG188 motor to determine which would match our requirements. Both these motors are powerful enough to provide us with 6W. We further looked into these motors to determine which would help us achieve our torque and transmission speed requirements.

We calculated the work done and the torque required to raise a fully loaded bucket as shown in Figure. Although both motors met the power requirements, we ran some preliminary tests with the Snowblower motor and a 5:1 gear ratio, which indicated that the transmission

speed was faster than required. The Snowblower motor has a no-load RPM of 100 RPM, and our gear reduction made the bucket move at 20 RPM, which was fast and difficult for us to control. This small test helped us finalize the Andymark PG series motor, PG188, with a ½ inch hex shaft with no load RPM of 28 RPM. Initially, as the torque required is higher to raise the bucket, the speed would be around 21 RPM, and then with our reduction of 4.65:1 it produced on average 5 RPM transmission speed. We achieved the 4.65:1 gear ratio with 18 teeth and 84 teeth VEX Pro Gears.

We also chose this motor because it has encoders, which can be used to stop the bucket when it is upright instead of having a mechanical hard stop. The motor and a speed controller like the Talon SRX attached to it can be used to start and stop the bucket at desired positions and drive back the bucket after the gravel is dumped. We primarily used the Talon SRX to drive the bucket back so that the robot could run multiple competition cycles. We tested the motor and gear ratio by plugging the motor directly into the 12V battery. This was important to ensure that the transmission was not geared too high and that it achieved the transmission speed we required.

We calculated our requirements for the motor to power the bucket by first finding the coordinates of the center of mass of the loaded bucket in Solidworks and then calculated the work done to raise the bucket. See Figure 28 for the FBD of the bucket for reference to labels. Figure 29 contains the motor curve data we used for our calculations.

$$\begin{aligned}
 & x, y, z \text{ coordinates of Center of Mass of Loaded Bucket in Initial Position} \\
 & = (4.5, 14.18, -1.21) \text{ inches} = (0.11, 0.4, -0.03) \text{ meters}
 \end{aligned}$$

$$\begin{aligned}
 & x, y, z \text{ coordinates Center of Mass of Loaded Bucket in Final Position} \\
 & = (x, y, z) = (4.5, -7.09, 12.28) \text{ inches} = (0.11, 0.18, 0.31) \text{ meters}
 \end{aligned}$$

$$\begin{aligned}
 & \text{Maximum mass of gravel that the bucket can hold} = 5.4 \text{ kg} \\
 & \text{Mass of a fully loaded bucket} = 7 \text{ kg}
 \end{aligned}$$

$$\begin{aligned}
 \text{Work done to raise the bucket} &= \Delta PE = mg(h_2 - h_1) = 6.5 \text{ kg} * 9.8 \text{ N} * (0.31 - ( \\
 & \qquad \qquad \qquad 0.03))\text{m} \\
 & = 23.324 \text{ J} \sim 24\text{J}
 \end{aligned}$$

$$\text{Power Required to raise bucket in } 4 \text{ s} = 24\text{J}/4\text{s} = 6 \text{ W}$$

Fig : Calculations for Work Done to Raise Bucket

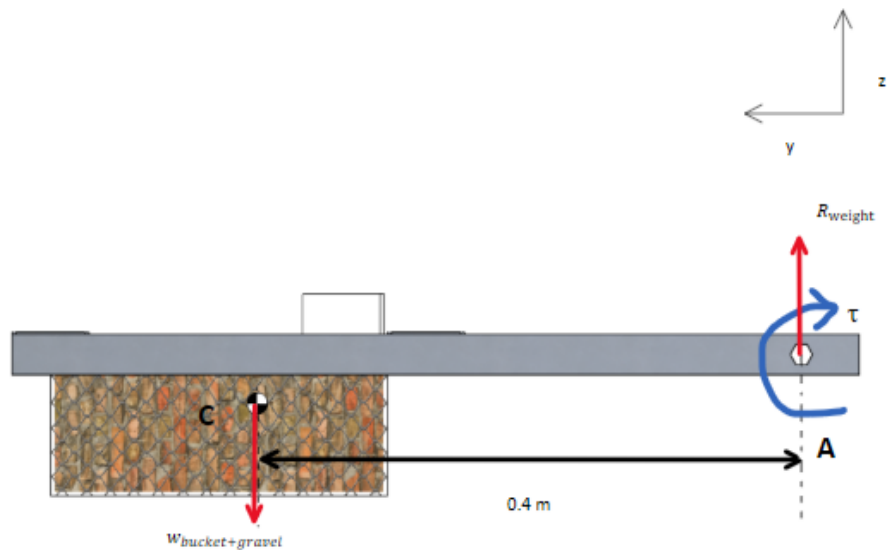


Figure 28: Free Body Diagram used to calculate the minimum torque required to raise a fully loaded bucket

#### Calculating the torque

$$\text{Minimum torque required to raise a fully loaded bucket} = F \cdot d = 7 \text{ kg} * 9.8 \text{ N} * 0.4 \text{ m} = 27.44 \text{ Nm} \sim 30 \text{ Nm}$$

30 Nm torque is required to raise the bucket. We want the bucket to cover 120 degrees in 4 seconds, therefore our desired output speed is 5 RPM.

Desired speed of the bucket = 120 degrees in 4 seconds = 5 RPM

$$\begin{aligned} \text{Gear ratio required} &= \frac{\text{Output Torque}}{\text{Input Torque}} * \eta = \frac{\text{Input Speed}}{\text{Output Speed}} \\ &= \frac{30 \text{ Nm}}{6.11 \text{ Nm}} * 0.95 = 4.65 \\ &= \frac{25.5 \text{ RPM}}{5.4 \text{ RPM}} = 4.65 \end{aligned}$$

(Input torque and speed values obtained from the datasheet and curvesheet as shown in Figure )

Fig : Calculations for Gear Ratio

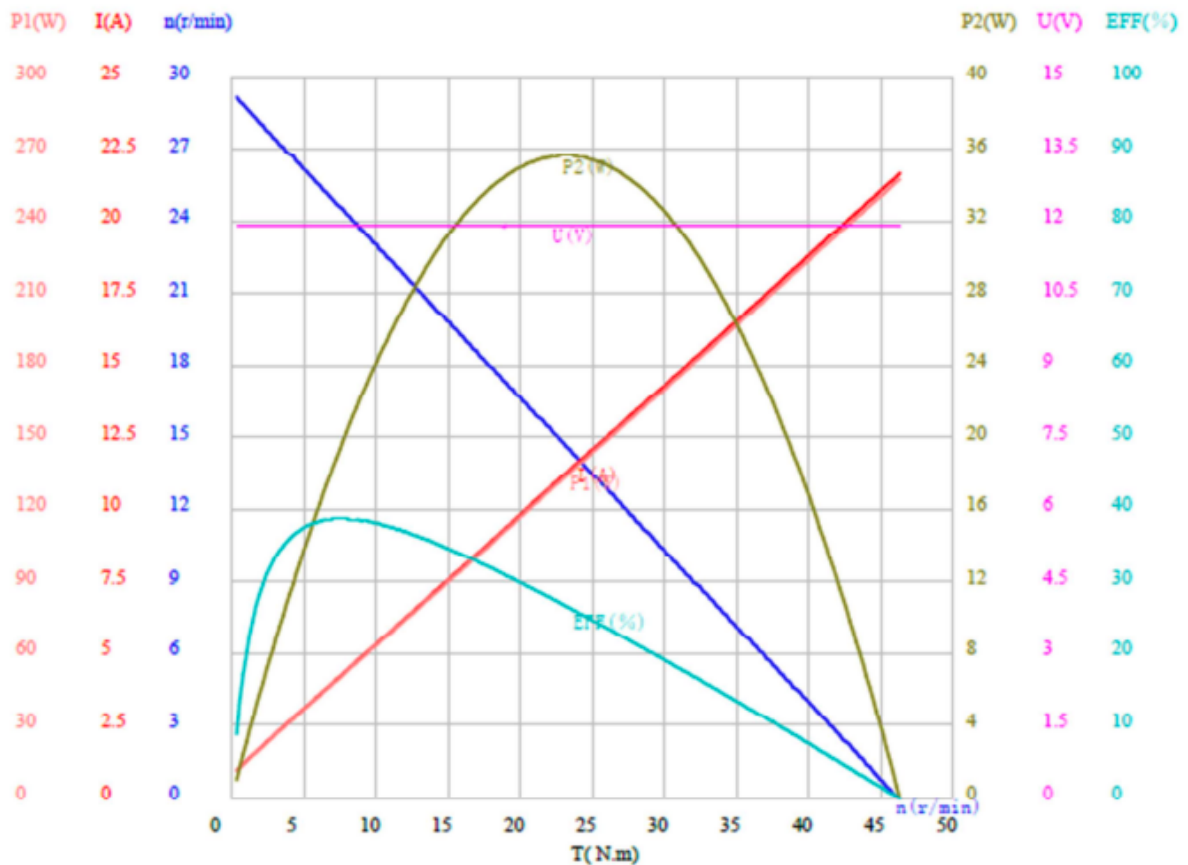


Figure 29: Curve Sheet for PG188 Motor (*AM-3656 Motor curve sheet*)

With the design having been finalized for the most part, the team moved to begin manufacturing the bucket but realized a significant problem with the design during manufacturing. Transmitting torque through a hex shaft going through 1/8" thick aluminum walls would create stress that would exceed the yield strength of 6061-T6 aluminum. To remedy this, a set of 5/8" square inserts were designed to go inside the arms where the hex shaft would go through to effectively increase the thickness of the material the torque was transferred through. Another issue the team encountered was that the bucket arms were not long enough to bring gravel into the bucket when the excavator dug at the depths where most of the gravel would be encountered. A 3D printed wedge made out of PLA was added to extend the front of the bucket forward, which can be seen in Figure 30.



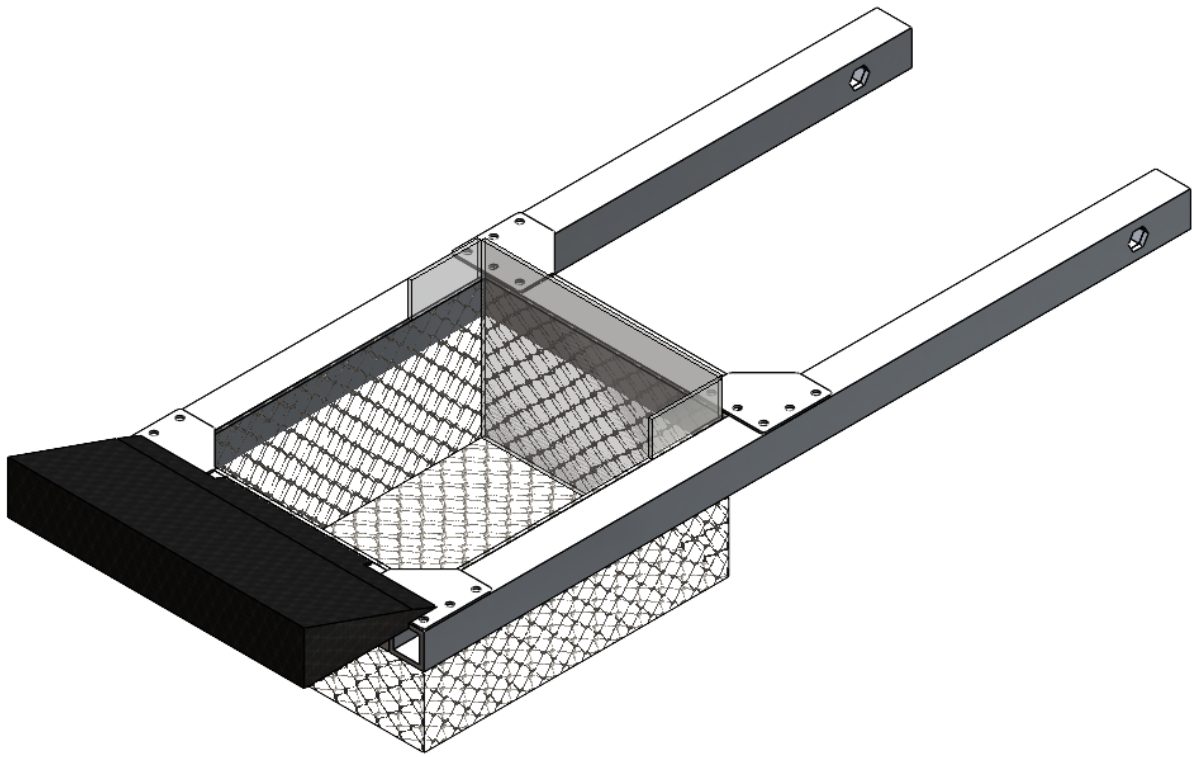


Figure 30: The bucket with the front wedge attached

### 3.3.3 Vision/Navigation

Based upon the initial overview of the computer vision and navigation systems in the previous sections, there were three significant subtasks that this system had to accomplish. The system had to be able to quickly, consistently, and accurately identify the robot's location with respect to the fiducial marker on the sieve. The system had to be able to generate a map quickly and accurately. The system had to provide a path in a usable form for the roboRIO (robot controller), a path efficient in terms of time and energy consumption, a path with a built-in margin of safety during navigation, and a computationally quick algorithm. To define each of these broad objectives in a detailed manner, a list of system requirements was developed for each of the sub-tasks, which are represented in the tables below. These requirements are either a core functionality or a performance requirement. Core requirements (CR) represent a function that had to be integrated for the system to work. Performance requirements (PR) evaluate a function being accomplished to a specific, measurable value that was validated with system testing.

Requirement	Type	Description:
#1	CR	The system localized without having to move the robot.
#2	PR	The location estimation was within $\pm 2$ cm of the true location when the distance from AprilTag was less than 2.5m.
#3	PR	The location estimation was within $\pm 3.5$ cm of the true location when the distance from AprilTag was greater than 2.5m.
#4	CR	The system was able to find or relocate a tag within five seconds.
#5	PR	Localization occurred at a rate of 4-8Hz.
#6	CR	The system was able to consistently track the AprilTag

Table 9: Localization system requirements

Two major design choices were required for the localization subtask to achieve the system objectives in Table 9: obtaining localization data and actuating the camera. The first significant design choice was selecting a method to obtain localization information (translation and orientation) of the robot with respect to the environment. Based on the regulation of the Lunabotics challenges, there were two primary techniques: the use of fiduciary markers or infrared beacons. After researching both methods, using an AprilTag as a fiducial marker was chosen because of its ability to generate both translation and rotation data from a single marker and our team's familiarity with the technology. The first AprilTag-based localization method investigated a single AprilTag and a mathematical approach known as homography. Homography is a projective transformation and is a concept often used in computer vision to correlate points in one plane to another (Figure 31). In this case, homography was used to correlate the three-dimensional points in the robot's environment to the two-dimensional picture taken by the Logitech c920 webcam. The corners and center of the tag were identified and

mapped using direct linear transformation and information regarding the camera's intrinsic parameters. From this information, the translation and rotation of the robot were ascertained with respect to the AprilTag and returned in a transformation matrix.

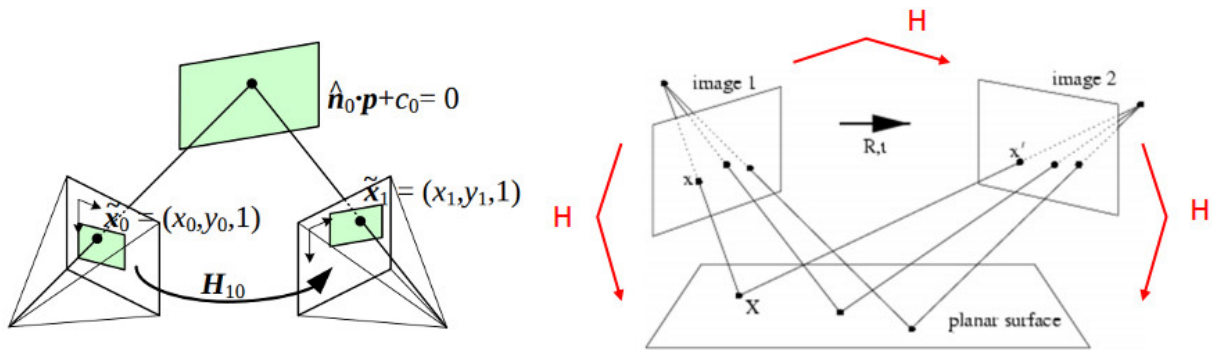


Figure 31: Visual representation of a projection transformation (Adapted Hartley and Zisserman 2023)

The second methodology was investigated using two AprilTags, the distance between the tags was known, and the distance from each tag was calculated and used to create a pair of circles (visualized in Figure 32). These circles intersected twice, once inside the playing field and once often outside of the playing field. The intersection that was found within the playing field represented the location of the robot. Based upon early rudimentary testing, it became evident that the dual AprilTag method was an obsolete and unnecessary solution. Initially, the two-tag process was designed due to concerns about the reliability of a single tag at further distances (defined as further than two meters). However, during testing, our team found that localization with a single tag was within two percent accuracy over a range of four meters. Based on this information, the team shifted focus to the single tag method. This decision was driven by three factors: the single-tag method proved to be of similar accuracy to the two-tag method, the identification of two tags was more computationally intensive and slowed the cycle time from six-eight Hz to two-three Hz, and attempting to fit both April tags within the frame of the webcam proved challenging when close to the sieve.

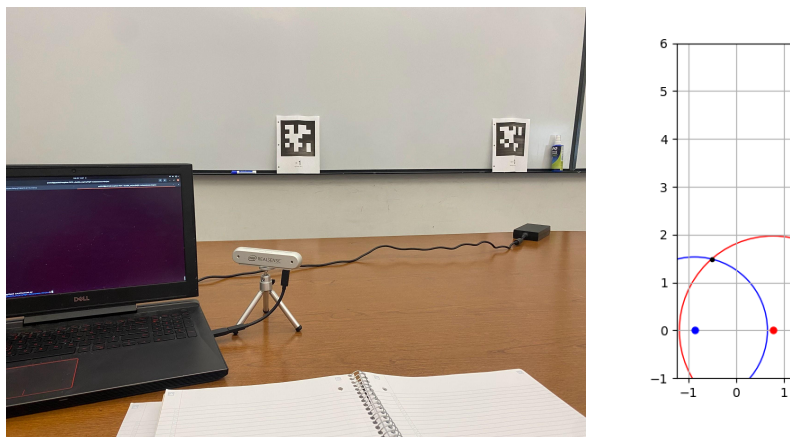


Figure 32: Visual representation of the two-tag localization method

After selecting the single tag method for obtaining localization data, the implementation of the second major design decision needed to be addressed. The next step was actuating the camera to enable it to rotate 360 degrees while being aware of the camera's orientation in relation to the robot. This actuation was needed to allow the camera to localize and track the AprilTag without the actuation of the robot. There were two solutions analyzed for the actuation of the camera. The first option was a 360-degree continuous rotation servo with feedback. The second solution involved the implementation of a small stepper motor to actuate the camera. While there is no direct feedback by knowing the starting position and steps taken, the orientation of the robot could be calculated. The stepper motor was selected because it provides high torque and little vibration at low speed, which fits the applications of our actuated camera system. The motors can be configured to increase their resolution by micro-stepping. Microstepping increases the resolution by modulating the current being passed through the motor windings, allowing the motor to move into positions between complete steps. This improved upon the low-vibration characteristics of the stepper motor and its resolution, allowing the team to estimate the robot's orientation based on the cumulative number of steps the motor had taken. Additionally, continuous rotation servos with feedback proved very limited and often had multiple weeks of lead time before receiving them. The 17HS4023 Nema motor used 1/8th of a full step achieving a resolution of 0.255 degrees (See figure 33). This motor was combined with basic proportional control to relate the number of steps the motor takes to the displacement of the AprilTag. Using this technique, AprilTag tracking was successfully implemented, and the system was able to search and find a tag if the tag was lost.



Figure 33: Left image is the 17HS4023 Stepper Motor and the right is FEETECH continuous rotation servo (Adapted FEETECH and GEMS Motors, 2023)

Requirement:	Type:	Description:
#1	CR	Remove statistical outliers from point cloud data to improve the efficiency of point cloud operations by decreasing the size of the working set.
#2	CR	Filter the resulting point cloud to only contain points that are relevant to the robot's navigation.
#3	CR	Project the three-dimensional point cloud onto a two-dimensional plane by generating an occupancy grid on which navigation can occur.
#4	PR	Locate objects within the map accurately within two percent of their actual location.
#5	PR	Represent objects within the map accurately, in terms of size, within five percent of their actual size.
#6	PR	Run map generation stably between 3-5Hz
#7	PR	(Reach Req.) Run Map Generation in real-time

Table 10: Map generation system requirements

To accomplish map generation, several major design requirements were considered, as visualized in Table 10. The mapping accuracy and feasibility of integrating sensors into the existing robot were primary areas of concern. For mapping purposes, two main technologies were analyzed, the first being LiDAR systems, including both spinning and fixed systems, and the second approach used a depth-sensing camera. After preliminary analysis, our team selected the depth-sensing camera for several reasons. Multiple LiDAR systems were tested. However, these systems were often large and bulky, with some possessing a mass of over three kilograms. Additionally, it was challenging to fix this system on the robot, as there was not space available for mounting a large device or a position where other parts of the robot would not block the laser scan. While the LiDAR systems offered better accuracy, they consumed more power than the camera and were less effective at indicating crater-like features during mapping. Our team performed preliminary testing comparing the mapping accuracy of the camera to the LiDAR. While the LiDAR was more accurate, the depth camera was within two percent of the actual values. The Intel RealSense D455i camera met the system's needs with reasonably accurate measurements. After selecting the depth-sensing camera, the next step was retrieving the data from the camera and processing it with the Raspberry Pi 4B to accomplish the above objectives.

The Intel RealSense product line was shut down, and the code repositories were archived approximately four years ago. Unfortunately, this made integration with the newer Raspberry Pi 4B difficult. Integration between the camera, Linux distribution (Ubuntu 20.04), and the ROS wrapper for the Intel camera system proved to be a substantial challenge. Eventually, the integration challenges between ROS and the Intel RealSense camera forced our team to write our

own custom ROS wrapper to integrate directly with the camera and publish point cloud data. This process allowed for the implementation of all the point cloud processing and operation in a single ROS node. This removed multiple data type conversions between ROS point clouds and the Point Cloud Library (PCL) types, improving the speed and efficiency of the mapping process. To remove statistical outliers from the point cloud, a standard deviation method compared the average distance of a point to its closest twenty neighbors. If this average distance had a variance greater than the set threshold, that point was removed. After the implementation of this functionality, it became apparent that such processing was computationally expensive and, therefore, not used when the robot was moving and mapping simultaneously. The remaining point cloud was then modified using a passthrough filter, selecting points within two meters of the robot and between  $\pm 0.2$ - $0.4$  meters on the Z-axis. This threshold was selected based on a test of the effective ground clearance of our robot, which was determined to be slightly more than 0.2 meters. This substantially reduced the size of our working data set, which allowed for improved computation time (See images 34 for a visualization of the processing).

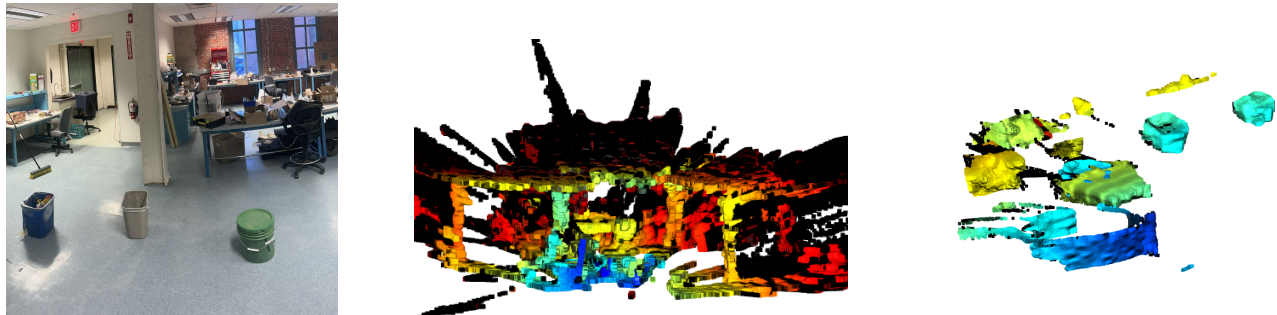


Figure 34: Depiction of the point cloud processing pipeline from the far left working in is a picture of the scene, then the initial point cloud representation followed by the filtered and slice final product

Once the final point cloud had been developed using the custom ROS wrapper, the three-dimensional point cloud data was transformed onto a two-dimensional plane and was represented in a probabilistic occupancy grid. The probabilistic occupancy grid marks the likelihood of an obstacle in a given cell. This probability is based on the number of points within a grid square on the occupancy map determining if it was free space or an object. Table 11 represents the testing data in the system validation section that confirms the system's performance standards outlined in the system requirements.

Requirement:	Type:	Description:
#1	CR	Implementation of a margin of safety (C-Space).
#2	CR	Implementation of frontier detection.
#3	CR	Implementation of a path-planning algorithm that chooses the most efficient path.
#4	PR	Path-planning algorithm takes less than 0.5 seconds to calculate a path from any set of points with the playing field.
#5	CR	The path was transmitted from the Raspberry Pi 4B to the roboRIO in a usable manner.
#6	PR	(Reach Req.) Path-planning algorithm penalizes unnecessary turning and traveling near the margin of error

Table 11: Path-planning system requirements

The final subtask required for navigation was path planning and path selection functionality. After generating the map using depth-sensing technology, modifications must be made to prepare it for the path-planning process. A buffer zone around objects must be applied where the configuration space (C-Space) is implemented. The C-Space was implemented by identifying and selecting the object-filled grid cells and then dilating those objects to make them larger. This was accomplished by calculating all of the valid neighbors of a given object cell and marking them as objects. Unfortunately, this process can be computationally expensive, depending on the size of the map and the number of obstacles. However, by making use of mathematical matrix libraries, such as Numpy, this process was optimized to reduce operation times.

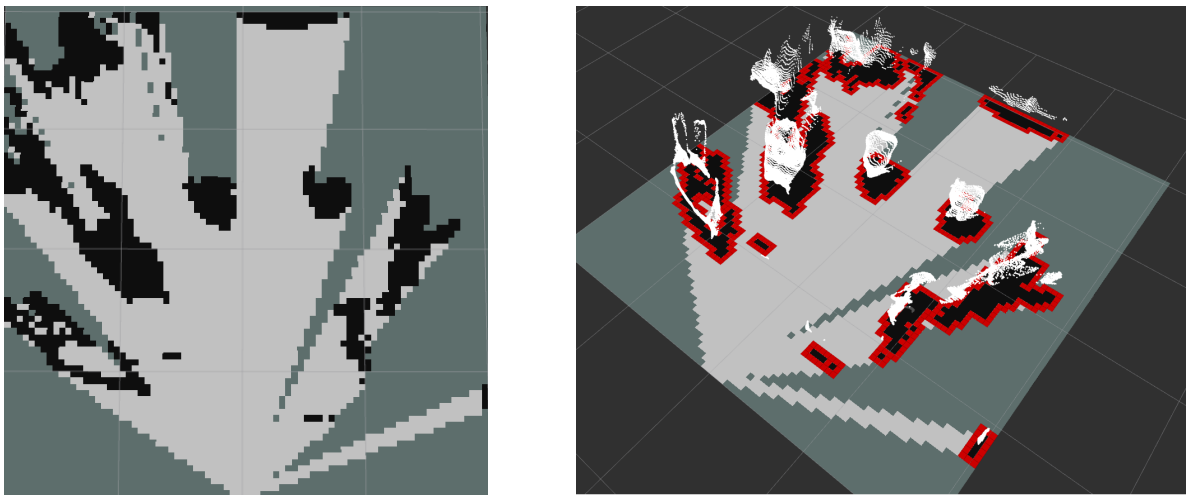


Figure 35: Visualizing the initial mapping and the application of the C-Space

According to Table 11 requirement two demands that the system identify frontier regions within the map, allowing for the robot to identify areas to explore. Frontier space was defined as the border between regions of known free space and space where no data has been collected yet. To detect edges, the matrix that represents the map was converted into an image to use different computer vision approaches. The two methods of image processing that were explored for this particular situation were the Canny edge detection algorithm and contour identification. The Canny edge detection algorithm is a multi-stage algorithm that identifies the edges of a given image using Gaussian blur, filtering, and intensity thresholds. The Canny methodology first applies a Gaussian blur followed by filtering to reduce the noise in the image resulting in more prominent edges. Then the intensity and gradients of the image were found, and two intensity thresholds were applied to bound the gradient magnitude. The edges were then tracked using hysteresis, and edge detection, in theory, was complete. During preliminary isolated testing, this edge detection method was inaccurate and missed marking many of the frontiers within our map. After further investigation, we believe that this was caused due to the relatively small number of pixels that represented the frontier. The Gaussian blur distorted these small areas lowering their intensity beneath the threshold that needed to be marked as an edge. Consequently, large regions of frontiers were expanded and created more inaccuracy. Therefore another solution was required.

Contour identification was the solution that remedied the shortcomings of the Canny edge detection algorithm for the path-planning application. Contour detection was a simpler identification process that found boundary pixels with the same color and intensities. This removed the issue with Gaussian blur causing miss identification of edges and removed many of the steps in the Canny edge detection algorithm. This solution was ideal for this specific application because representing the map in black and white made it easy for contour detection to detect border pixels with the same color and intensity. After isolated testing, the accuracy and speed of the contour detection showed a significant improvement over the Canny edge detection. After the application of contour detection, a dictionary of frontiers was created by recursive iteration through the edge points that are identified. Edge start points are selected by identifying points with only one other neighboring point that was also an edge. The recursion occurs and assembles a list of grid coordinates representing each edge. The edges were then scored based on the size of the frontier, its distance from the robot's current location, and how centralized the point is on the map. These criteria rewarded the robot for selecting frontiers that travel towards the excavation zone, avoiding unnecessary lateral movement, and exploring large portions of the map. The frontier with the highest score is then selected with its midpoint target of the path-planning algorithm.



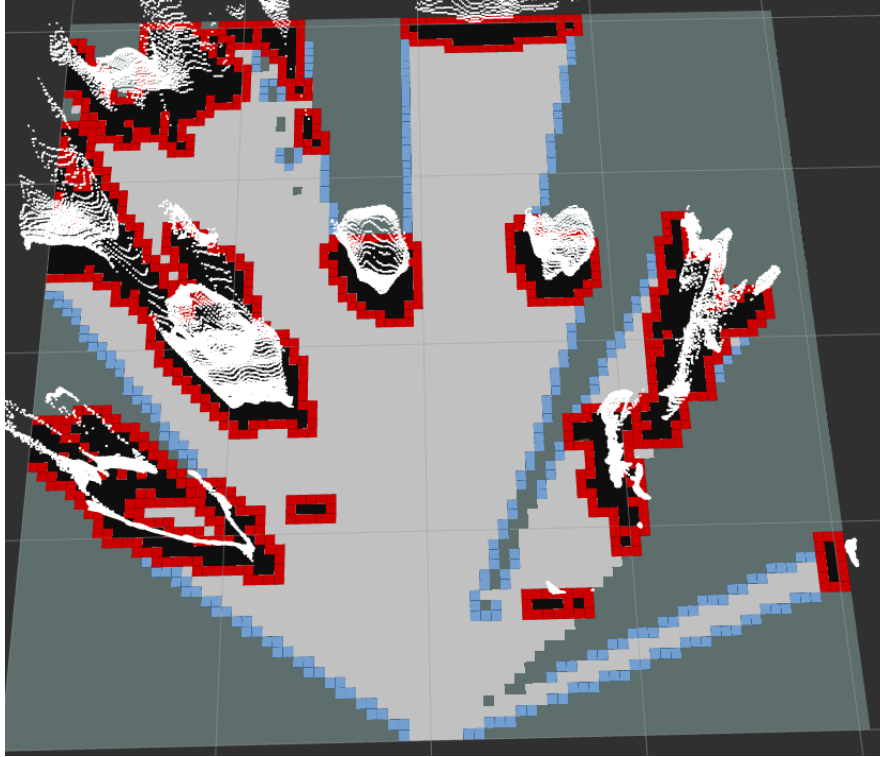


Figure 36: Visualization of edge detection using the contour detection

As detailed above in Table 11 requirements three, four, and six state that the path generated must be found quickly and be an efficient option. The A\* algorithm was selected for path planning based on those requirements. A\* and path-planning implementations have been researched and documented extensively. Therefore this portion of the report will provide insight into the heuristics chosen and the modification made to the algorithm to better execute the robot's needs. In the A\* algorithm, several different heuristics methods were used to estimate the future cost of moving to a position. The most common heuristic methods are Manhattan distance, Diagonal distance (Octile and Chebyshev), and Euclidean distance. Manhattan distance is a heuristic based on the neighbors of four movement constraints where a robot can only move in the four cardinal directions, diagonal movement within an occupancy grid is not possible. The diagonal distance heuristic further develops the Manhattan heuristic that is optimized when diagonal movement is enabled. Diagonal distance calculates the number of steps taken if the diagonal movement was not possible and then subtracts the steps saved by using diagonal movement. See the below equation:

$$Cost = D * (xdistance + ydistance) + (D2 - 2 * D) * min(xdistance, ydistance)$$

Equation to solve for the diagonal distance coefficient where:

$D$  – Cost of a single cardinal direction movement

$D2$  – Cost of a single diagonal movement

$xdistance$  – X-axis displacement from start to goal

$ydistance$  – Y-axis displacement from start to goal

There are two methods of the diagonal distance heuristic, Octile distance and Chebyshev Distance. Octile distance is where the movement cost across the diagonal ( $D_2$ ) is 2, an accurate representation of the increased length of traversing a diagonal. Chebyshev's distance, more commonly known as chessboard distance, sets both the cardinal direction movement cost and the diagonal distance cost equal to one ( $D=D_2=1$ ). This method favors diagonal movement in terms of a real-world mapping application, as the cost to traverse the longer diagonal distance is the same cost as the shorter cardinal direction movement. Given our application and the movement structure of the robot, the Octile heuristic was selected. Additionally, the Euclidean distance heuristic provides the straight line distance between the two points, ensuring the heuristic is always optimistic or exact. This methodology assumes movement along any vector is possible; however, this was not the case in this mapping system. The robot was represented in the center of each cell, and the robot can only move to the center of its eight neighboring cells. Therefore, this heuristic would not be appropriate. As discussed above, the octile heuristic is the most appropriate for our application; however, testing was performed to ensure the implementation and assessment of the algorithm, and heuristic were correct. The test took a random map and calculated the average time to find a path and the number of nodes expanded for every start and end location within the map. This testing confirmed the analysis described in this section.

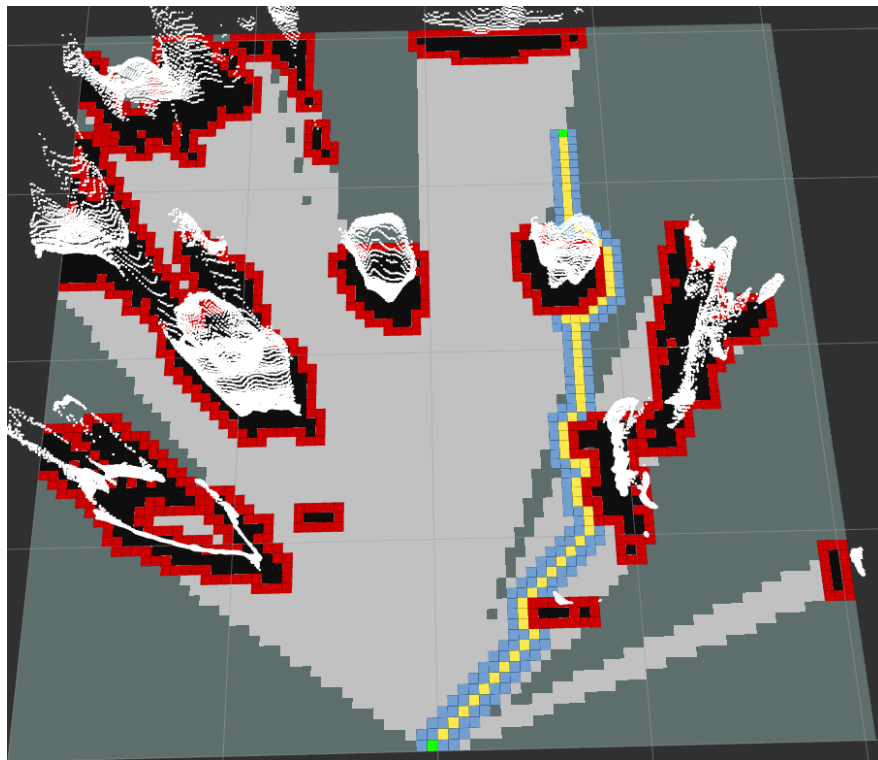


Figure 37: Visualization of path to the target frontier (scene setup can be seen in figure 34 far left)

### 3.3.4 Autonomy Integration

To autonomously control the robot, we implemented various sensors to monitor the state of each moving part in the robot system.

For the drive motors, through the use of the Spark Max Controllers controlled by the RoboRio, we were able to retrieve encoder values from each of the drive Neo motors powering the wheels. Getting encoder position values from the wheels was necessary to monitor the distance traveled. Using the below equation to calculate the number of encoder ticks needed to travel a certain distance, we utilized this in a “drive” function and a “turn” function. The drive function sets the wheel motors to turn initially at a designated set speed and uses the difference between the encoder goal and current encoder count as feedback to a PID loop that reduces the speed of the wheels the closer it gets to its goal. This drive function is able to drive both forward and backward with reliable precision. The robot executes various zero-point turns for the turn function according to the angle value in degrees given as input. It uses the same concept as the drive function except that the encoder goal calculated is halved and negated depending on the direction of the turn. For clockwise turns, the left wheels would move forward while the right wheels move backward and vice versa. The turn is centered around the center of the robot.

$$Encoder\ Goal = \left( \left( \frac{distance(in)}{\pi \cdot wheel\ diameter(in)} \right) \cdot encoder\ resolution \cdot gear\ ratio \right) + current\ encoder\ value$$

Encoders are also utilized on the linear slider, excavator, and bucket motors with speed controllers like Talon SRXs. These encoder values are utilized to send each of these subsystems to designated presets. For instance, with the excavator, there are two main configurations in which it has to be positioned - stowed and digging. We have determined the encoder values these subsystems should get to in order to precisely and accurately get to both of these configured positions every time. With the bucket, there are also two configurations, when it is laid down in the chassis to collect gravel and another for dumping its contents into the sieve. This is accomplished again through specified encoder values the motor must reach.

In addition to the encoders throughout the robot, we also utilized a hall effect and a pressure sensor to monitor the excavator and bucket states, respectively. The functionality and implementation of these sensors will be explored in more depth in the system integration section, but their purpose was to act as a trigger for different autonomous states of the robot. The hall effect sensor was used in conjunction with magnets to monitor the relative positions of the linear slider on the excavator. The pressure sensor was utilized to monitor the force exerted by the bucket due to the mass of gravel it held. We found a threshold value to aim for that would result in an adequately filled bucket.

### 3.4 System Integration

This section goes over how the subsystems were integrated with the robot and the challenges associated with it.

#### 3.4.1 Excavator

Several modifications were necessary to integrate the newly ordered linear screw with the existing mounting system. First and foremost, the linear screw had to be cut to the correct length. Furthermore, the threads on the ends of the screw needed to be smoothed to a 10mm diameter to facilitate free rotation within the bearings that hold it. Due to the lower coefficient of friction on this screw, a small hole was drilled through the linear screw at a distance of 40mm inward from one side. This was done to prevent the screws from fastening the gear that drives the system from slipping on the threads, thereby eliminating the gear slippage that contributed to the failure of the previous system.

In the previous screw system, a hex nut was mounted in place, and the nut was stabilized using two circular aluminum fittings with the shape of the nut cut out in the center. These circular fittings were placed over the nut and secured to two mounts on either side of the nut. The mounts prevented the circular fittings from turning, and the circular fittings prevented the nut from turning, thus locking the entire system in place. However, the newly acquired system had a cylindrical-shaped nut, rendering the previous mounting system inapplicable.

To address this issue, new mountings were fabricated, with the fastening holes aligned with the nut itself. The new nut was then drilled through with the same fastening pattern, and the screws were inserted through the nut to stabilize the system, eliminating the need for outer fittings. These modifications allowed for the seamless integration of the new linear screw with the pre-existing mounting system, resulting in a more efficient and effective system overall. The tool paths for manufacturing the new mounts and the final product can be seen in Figure 38 and Figure 39.

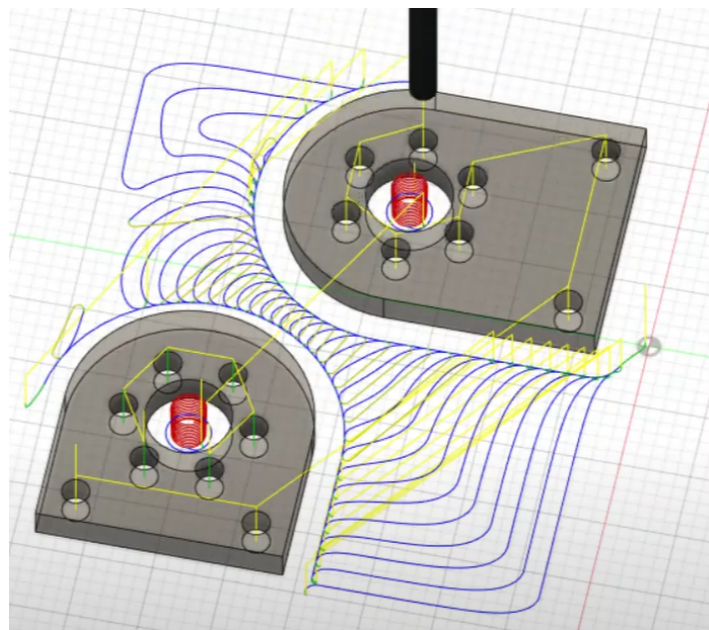


Figure 38: Toolpaths for CAM program to make the new mountings



Figure 39: The new mountings machined on the VM-2

### 3.4.2 Storage and Deposit

Integrating a redesigned storage and deposit system onto the existing chassis proved to be a significant challenge for the team and resulted in several modifications needing to be made to the design of the bucket as well as the mounts for the bucket to the chassis. After disassembling the original storage conveyor, the team discovered a rod going through the middle of the chassis that was used to connect the two rocker bogies. This limited the depth of the bucket as it would not be able to drop into an empty hole in the middle of the chassis. If the bucket was too tall, the excavator would not be able to lie as flat in its stowed configuration as it was last year. This would cause the robot to no longer conform to the dimensional constraints set by the Lunabotics rules.

The second and more significant system integration issue the team encountered with the bucket was mounting it to a chassis not designed for a bucket system. The back of the chassis is about nine inches away from the back of the wheels to allow a larger range of motion with the motor mount design last year's team chose. The cable that connected the rocker-bogie system to the chassis also limited the size of the mounts. The team initially wanted to mount the bucket on an extension attached to the back of the chassis, but this proved not to be a viable solution. An extension could not be put on the edges of the chassis because the drive motors would hit the extensions when the robot was traversing a large obstacle or deep crater. A plate could not be put on the inside of the chassis for more space because it would cover the power board of the robot, which needed to be accessible. Ultimately, the team was forced to design a set of mounts that could be attached in the small space between the two gusset plates that held the chassis together. The robot would also be forced to back under the sieve in some way in order to deposit gravel, similar to last year's robot.



### 3.4.3 Vision/Navigation

Integration proved to be a significant challenge for the vision processing and navigation component of the robot. The integration of each component was substantial before incorporating it with the rest of the robot. The subsystem integration required utilizing the outdated Intel RealSense libraries, with restricted ROS functionality due to its installation on the Raspberry Pi 4B. Unfortunately, the ROS wrapper for the Intel RealSense libraries was not compatible with the Raspberry Pi 4B; therefore a custom C++ wrapper was written for the vision processing. Overall, the time spent on battling integration issues was comparative to the time spent on the development of the computer vision based navigation subsystem.

The primary function of the vision processing and navigation system within the project was providing localization data and a navigation path to the robot controller (roboRIO). For the two systems to communicate with one another, an Ethernet crossover cable was used to enable direct communication between the systems. Without the use of the Ethernet crossover cable which is designed for computer-to-computer communication, a network switch would have been needed to facilitate the communication between the two devices. After the physical connection was made, static IP addresses were configured on each device and a LAN network between the two devices was established. This network provided communication speeds of approximately one gigabit per second which was more than adequate for providing the necessary bandwidth. Unfortunately, the roboRIO and FRC hardware/software were not easily compatible with ROS or other third party software libraries. Therefore, the pynetworktables, an FRC network based communication protocol, was used to allow the Raspberry Pi 4B to communicate with the roboRIO and provide localization and path planning. The internal subsystem integration, physical integration with roboRIO, and communication protocols integration were substantial components of the vision processing development process.

### 3.4.3 Autonomy Integration

#### Driving Autonomy

The drive and turn helper functions described previously were integrated into a “drive\_to” function to fully integrate autonomous driving with camera vision navigation. This “drive\_to” function uses the Smart Dashboard to retrieve updated odometry position values from the camera-raspi system and target positions for the robot. These target positions are intermediary waypoints generated from the path planning algorithm on the camera side. The drive\_to function then calculates the appropriate turns and driving distances necessary to achieve the targeted position. The following equations were used to calculate both the linear distance and angular turning distance required:

$$\text{Linear Dist} = \sqrt{(target\ x - current\ x)^2 + (target\ y - current\ y)^2} = \sqrt{(diff\ x)^2 + (diff\ y)^2}$$

$$\text{Angular Dist} = \arctan 2((target\ x - current\ x), (target\ y - current\ y)) = \arctan 2(delta\ x - delta\ y)$$

These calculated distances were then sent to a send\_speed function that called the turn and drive functions, respectively, to orient and translate the robot appropriately. Together, this

drive\_ function allows the robot to move to any cartesian coordinate in any quadrant and direction when given its current odometry and target location in cartesian (x,y) coordinates.

### Bucket Autonomy

For the bucket we used the Hall effects two channel encoders (Figure 40) already installed in the Andymark PG188 motor used to actuate the bucket. We used the Encoder class in WPILib Library used to manipulate and operate quadrature encoders. We first initialized the encoders and set up the encoder resolution, which can be done using the SetDistancePulse() method. This ensures we are not dealing with raw encoder values but getting distance readings in degrees that we can visually determine by looking at the bucket. We set the distancePulse to 0.05, we obtained this by the formula below:

$$\text{Encoder Resolution} : \frac{360 \text{ degrees}}{188 * 4.65} = 0.05 \text{ degrees per pulse}$$

As the documentation of the function mentioned in the fr:: Encoder class , in order to take into account the gear ratios, we must include the gear ratio from the end of the motor shaft. The gear ratio from the end of the motor shaft includes the gearbox connected to the motor of ratio 188:1 and the external gear ratio provided by the VEX pro gears of 4.65:1. We used these values and our desired units, which in our case were the units of degrees.



Figure 40: Hall effects two channel encoder in the PG188 motor

This helped us make a function called 'bucket\_actuator(int targetDegrees)', which takes in the target degrees that the bucket has to move to relative to its starting position. This function

reads the current and moves the bucket motor until it reaches the target distance. We used the class function `getDistance()`, which measures the relative distance from the starting position of the encoder to the final position. This function helped us measure the relative distance, and we made the bucket move till it covered the relative distance in degrees. The `'bucket_actuator(int targetDegrees)` function covers both raising the bucket and bringing it back into the initial position to ensure that gravel dumping occurs as a single process.

To control the bucket, we implemented a pressure sensor (Figure 41) with a threshold value of 3500. There was an inverse relationship between pressure and the output reading, so lower values corresponded to a higher load on the pressure sensor. Through multiple tests, we determined that 2500 was an appropriate threshold to get a 3kg load of gravel in our bucket.



Figure 41: MD30-60 Pressure Sensor

### **Excavator/Linear Slider Autonomy**

The excavation system works in three parts, each of which was autonomously controlled. The first is the “excavator,” which pivots the entire excavation system around its mounting point on the robot chassis. This pivots the excavation system from its stowed configuration into its digging configuration. To control this autonomously, we used a Kayang motor from last year's team to drive the sprocket and chain transmission system. Unfortunately, the Kayang motor does not include encoders for position monitoring. To account for this, we attached a Vex 1/2” hex shaft encoder to the end of the transmission system. Using this encoder, we found that an encoder value of -500 would be optimal for a digging configuration and +500 for a stowed config and wrote reconfiguration functions appropriately to control this autonomously.

The second part of the excavation system is the linear slider, which moves the whole system up and down to dig into the sand. For this, we placed magnets spaced about 3 inches apart all along the excavator plate, with a central magnet in the middle for the beginning of the digging position. We attached a hall effect sensor to the moving linear slider so that it is able to monitor the linear slider's movement by tracking its passage through the magnets. We have written an autonomous function that keeps count of how many magnets the linear slider has passed, and once it has reached a designated number, excavation will stop, and the system will notify that the excavator has reached far down enough so as not to go beyond the limits of the linear slider.



The last part of the excavation system is the belt, which we set to run at a set speed for as long as the linear slider was going and until the bucket was full. The belt was kept in check by monitoring its current draw. If the belt ever exceeded 35A, it was immediately shut off to ensure the safe operation of the belt and prevent damaging the motor if something got stuck in the belt that kept it from moving freely.

### Smart Safety/Autonomy Control

The team used the FRC SmartDashboard to view motor and sensor data as an essential tool for ensuring the safe operation of our robot. With the ability to monitor motor voltages, current, and temperature, we were able to quickly identify any abnormalities that could indicate a potential issue. For example, if a motor current was significantly higher than expected, it could indicate that the motor was working harder and potentially lead to premature wear or failure. Similarly, if a motor temperature was higher than normal, it could indicate that the motor was overheating and could be at risk of burning out.

Part of the data viewed on the SmartDashboard was the ability to monitor the pressure sensor value, bucket angle position, excavator reconfiguration angle position, and battery cell voltages. These sensors provided critical information about the state of the robot and its environment and being able to view them easily allowed us to make informed decisions about how to operate the robot safely and diagnose problems. For example, monitoring the pressure sensor value helped us know when we collected enough gravel, while monitoring the excavator's motor current draw and temperature helped us to identify if the excavator got stuck. The GUI representation for this can be seen in Figure 42.

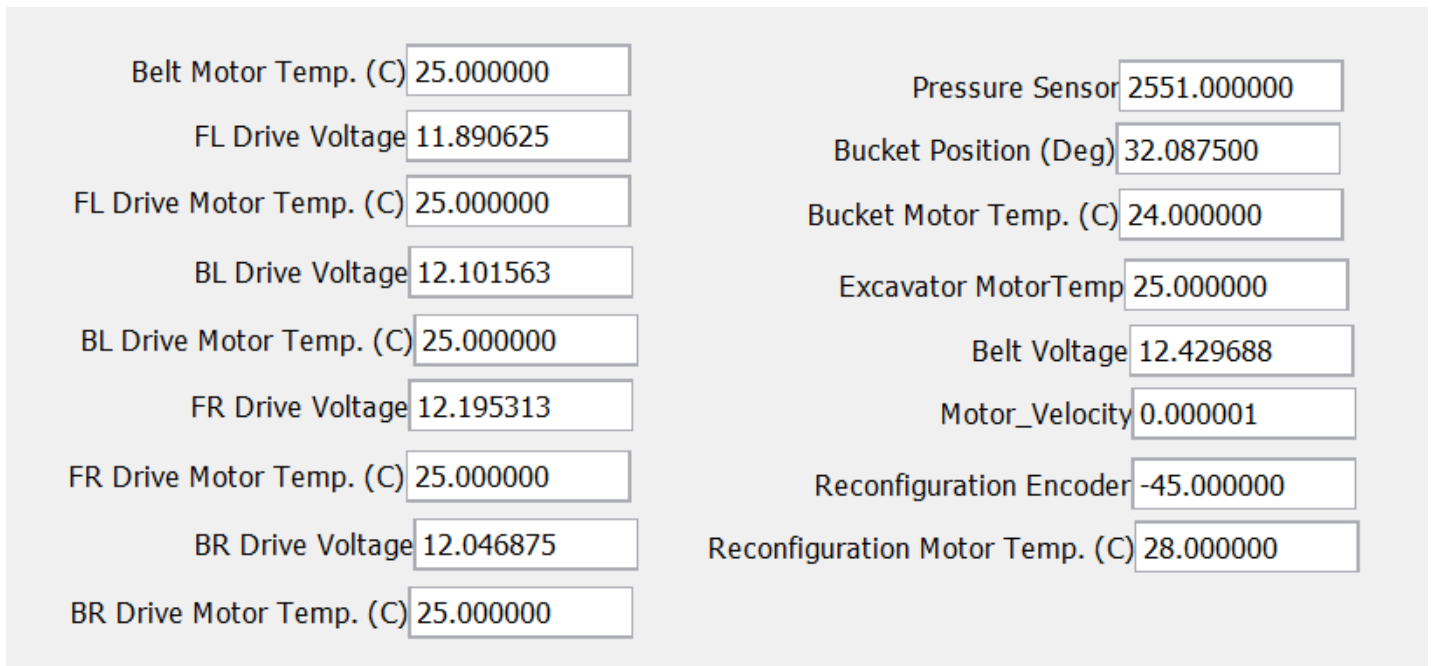


Figure 42: FRC SmartDashboard

## 4 Systems Testing and Validation

This section describes the validation testing done for each of the subsystems to check if they had met the team's design objectives.

### 4.1 Excavator

#### 4.1.1 Lead Screw System

After successfully manufacturing the new lead screw mounting system, a purely observational analysis was conducted by directly connecting the detached linear slider drive motor to the 12V battery that powers the robot. It was quickly observed that the system's speed had significantly increased, even without taking any measurements. Before the system's upgrade, moving the linear screw mount from top to bottom took over 2 minutes. However, with the new screw system, the traversal time was reduced to approximately one minute. It is important to note that this test was conducted with the system perpendicular to the ground. Further testing was required to determine if the system's effectiveness remained consistent when mounted at a 55-degree angle from the ground.

After attaching the linear slide system to the robot, we put it in the 55-degree digging configuration to test the traversal time at the correct angle. This timed test took exactly 56 seconds going downward and 1 minute and 4 seconds coming upward. While the added mass of the excavator did slow the screw system slightly, the design was less massive, so the added load was manageable.

### 4.1.2 Belt System

The new tensioner system met its objectives well. The team was able to fix the belt by moving the tensioner to the opposite side of the linear slider and adding another roller to the top of the friction point between the belt and transmission. The excavator can now run smoothly for a full cycle and is able to dig and deliver sand and gravel to the storage and deposit system. This was tested by running the belt for five continuous minutes (Figure 43) while digging into a bucket of sand and gravel. Because the excavator was able to dig to the gravel layer at a total depth of 40 cm, the excavator design was deemed successful.



Figure 43: Testing the excavator digging down to depths where gravel would have been encountered

### 4.1.3 Dust Protection

In order to protect the excavator from the materials it would be digging, the group needed to apply dust protection material to the gaps in the Lexan plates serving as the main enclosure of the system. In order to select this material, a test was done where each potential material was applied to the prototype acrylic covers (Figure 44). The cover was then contained in a cardboard box with 30g of BP-1 simulant collected on one side. On the other side was a collection plate that would collect dust blown through the protection material. The team then used a hairdryer to blow the material through the gaps in the protection for 1 minute per test. After 1 minute, the collection plate would be dumped into a small paper towel and reweighed to determine how much material went through.



Figure 44: Testing various materials for their dust protection effectiveness

In addition to the material test, the team conducted a separate test. This test measured the friction between the sliding portion of the linear slider mechanism and the material being tested. The team used this secondary result to determine which material could protect against dust while allowing the system to move freely. The results in Table 12 concluded that the black rubber material was the best for these conditions. After the five minute run of the excavator, this material proved to be effective in stopping large collections of foreign material from entering the excavator.

Material	Dust Passed Through (g)	Friction Estimate
Rough Felt	4.8g	Low
Fine Felt	1.9g	High
Gray Rubber	1.9g	High
Black Rubber	1.1g	Medium

Table 12: Dust protection results

## 4.2 Storage & Deposit

This section goes over the testing for both the frame and the depositing ability of the new storage system.

### 4.2.1 Bucket Frame

Once the bucket frame had been constructed, it was placed in the robot's frame to ensure it was small enough to fit in the chassis and not interfere with the excavator. The team observed that the bucket frame was small enough to allow the excavator to lie nearly flat on top of it in its stored position without any interference. The bucket frame's arms also lined up near-perfectly with the back of the robot's chassis, and it was able to complete its full range of motion without hitting the excavator. The inserts also held up well, as the hex bore in the bucket's arms did not fail even when the bucket was fully loaded with gravel. Overall, the new storage system was 5.5 kgs lighter and used over 50% fewer moving parts compared to last year's conveyor system. A test of the bucket arm's strength can be seen in Figure 39, where it holds 6.9 kilograms without failing.

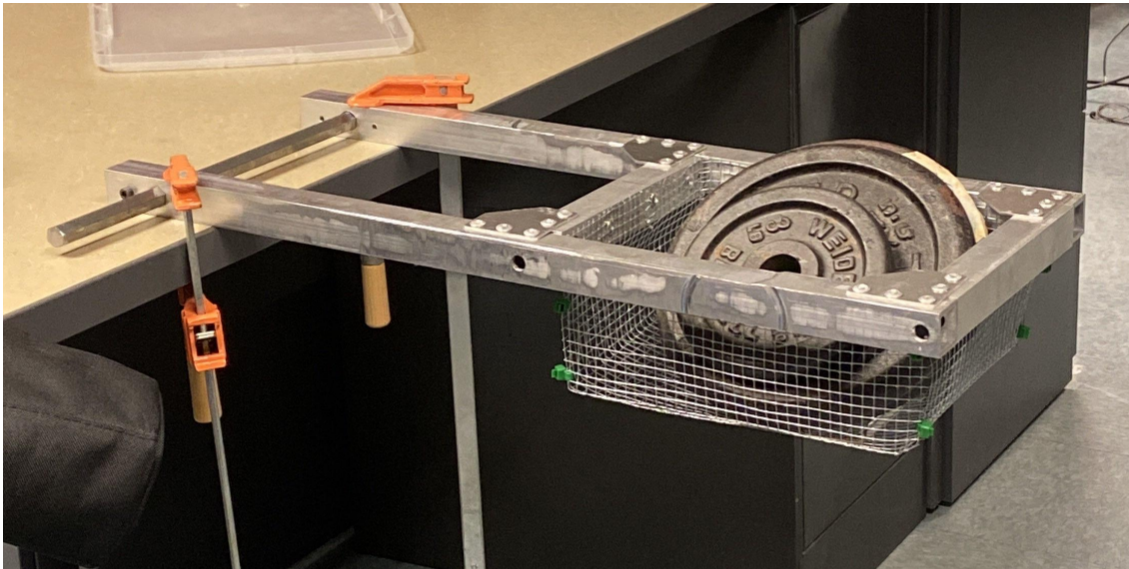


Figure 45: The frame showed no signs of deformation even when holding more mass than it would be holding with a full load of gravel

### 4.2.2 Depositing

The bucket easily solved the ground clearance issue created by last year, as it never had to be actuated lower than its initial position above the chassis. This improved ground clearance by 17.8 centimeters. With the addition of the Lexan backplate, the bucket was able to easily deposit three kilograms of gravel onto the center of the sieve, as seen in Figure 46. The total time to deposit and bring the bucket back was only four seconds, a significant improvement over the depositing speed of last year's storage conveyor, which took over 54 seconds to deposit gravel during its competition run. Gravel also did not get stuck on the Lexan plate of the bucket, with only a few smaller gravel particles sometimes not making it out of the bucket and onto the sieve.





Figure 46: The bucket depositing gravel on the center of the sieve

The results of testing the bucket with about three kilograms of gravel (our reach goal) can be seen below in Table 13.

<b>Before Dumping (kg)</b>	<b>After Dumping (kg)</b>	<b>Difference (kg)</b>
3.038	3.030	0.008
3.038	3.000	0.038
3.004	2.866	0.138
3.002	2.914	0.088
3.038	2.820	0.218

Table 13: Bucket deposit testing results

### 4.3 Vision/Navigation

This section on system verification will review the performance requirements outlined in section 3.3.3, describing the design and testing method used to evaluate the system's compliance with performance-based requirements. Requirements two and three in Table 9 evaluate the accuracy of the AprilTag localization-based method. The table catalogs the data from the accuracy tests. To conduct the test, four set distances were marked from one to four meters, and an AprilTag was placed at each point. The distance was then calculated using the single tag homography-based method, and a set of 25 data points was collected for each distance and averaged. This average was used to calculate the percentage error and average error distance in meters. Based on this testing, the designed subsystem met both of the accuracy requirements defined previously.

Distance Tested:	Averaged Measured Value:	Error in Meters:	Percentage Error:
1.0m	1.0021m	0.0021m	%0.21
2.0m	2.027m	0.027m	%1.35
3.0m	3.087m	0.087m	%2.9
4.0m	4.078m	0.078m	%1.95

Table 14: Localization accuracy testing

In addition to accuracy testing, the rate at which localization data was generated needed to be evaluated. When operating on the Raspberry Pi 4B, the localization code ran at an average rate of 5.6Hz, with the peak sustained performance measure at 9.4Hz, and the lowest sustained rate at 3.4Hz. Although these measurements were on the lower side of the performance goal, they fell within the outlined requirements, and the system effectively performed each of its core functionality goals.

The next set of evaluations pertained to the map generation performance requirements, including object detection accuracy, object size accuracy, and cycle time to generate a map. To analyze the requirements, three different shaped obstacles were placed two meters away from the camera, with one object directly in front of the camera and one placed half a meter to the left and the other a meter to the right (see Figure 34, far left for the setup). Measurements were taken from the leading surface of the object to the camera, and each object's top-down surface area was calculated as the size of the object is represented in two dimensions during the mapping. The results from the location accuracy testing mirrored the results from the localization accuracy. There were slight increases in error with the objects placed on the left and right sides of the robot, but these results were well within the defined requirements. The size representation, however, was initially overestimated by approximately 12%. The mapping system accurately represented rectangular objects, but it overestimated the size of cylindrical objects as the occupancy grid struggled to effectively represent the shape with grid cells. Increasing the mapping resolution of the system reduced the area calculation error to four percent, but the map generation and path-planning calculation times suffered, increasing substantially due to a denser

map of nodes. Although the overestimation did not pose issues within the test field, it is critical to understand that a field more densely populated with objects may become untraversable due to this overestimation. There are two solutions to this problem: using a different mapping approach or increasing the computational power responsible for image processing. The third and final test administered to map generation was cycle time. Using ROS tools, it was calculated that the map was generated at an average rate of 3.8Hz (in isolation). When combined with localization running simultaneously, that rate decreased to 2.9Hz. In an isolated environment, the map generation fulfilled the generation rate requirement. However, when running simultaneously, it did not achieve the requirement. All functional requirements for this subsection were achieved.

The last subtask to be evaluated within this section is the path planning portion, with one major performance requirement: the calculation time of the path. A test simulation was developed using a PLY file to provide a consistent map for several different heuristic versions of the A\* algorithm. This map was used to calculate a path from every set of valid points. After doing so, the average calculation time and the number of nodes expanded were recorded for each heuristic. The conceptual analysis developed in the previous section 3.3.3 selected the Diagonal Distance (Octile) heuristic as the most efficient method, the result in Table 15 confirmed this finding. Additionally the path planning subsystem achieved all of its core functionality requirements.

Heuristic Method:	Average Calculation Time (sec):	Average Nodes Expanded
Manhattan	0.3557	4523.8
Diagonal Distance (Octile)	0.1984	3248.4
Diagonal Distance (Chebyshev)	0.2148	3469.7
Euclidean Distance	0.2498	3987.2

Table 15: A\* calculation times and node expansion in relation to the implemented heuristic

#### 4.4 Autonomy Integration

To validate our various autonomous functions, we worked to test all the different edge cases. For driving in particular, to ensure its completion, we tested by having the robot drive both backward and forward and then chaining commands together to ensure that the robot can execute each effectively without getting caught in one of the functions. For turning, the edge cases were for turning left and right. These functions worked well initially until we attempted to use them with the drive\_to function when we found out that our system for keeping track of the current encoder counts and the goal encoder counts did not translate well for driving in the second quadrant. This was because the initial turn to orient the heading of the robot in the right direction required making the left wheels go in a negative direction. Therefore the current encoder values of those motors would thus be negative. We did not account for an initial negative value for encoder counts in our drive function, so the drive function was not able to execute as a result. To



fix this problem, we instead kept track of an initial start encoder value, and all proceeding calls to the getPosition() function of the encoders was subtracted from the start value to result in a scalar “distance traversed” encoder reading that was not impacted by positive/negative directions.

As discussed in our advisor meetings, it was a concern that if we relied too heavily on zero-point turns, the robot might dig itself into a hole that would prove difficult to escape. To combat this, we implemented a check for the turns. If the robot has to turn for more than 60 degrees at a time, it would first turn those 60 degrees, drive forward 5 inches, and then finish the rest of its turn requirement.

On our excavator, to validate our hall effect sensors and various configurations, we ran multiple tests to ensure that the linear slider would reliably stop every time it reached a magnet. This required repositioning of the magnets and linear slider, as well as the dust protection foam on the side panels, as those were getting bunched up during the movement of the slider, which prevented the sensor from accurately reading the magnets. Furthermore, we started testing all of our excavator moving parts at low speeds first, as manipulating so many moving parts at high speeds could be detrimental and lead to errors. We found that for all of the excavator systems, a base speed of 30% was optimal for a slow enough speed able to accomplish everything it needed to well while being fast enough to overcome the deadband area of the motor.

For validating the bucket actuation function, we checked for the edge cases as well to make sure that the bucket was not moving beyond the angles we wanted it to move. We tested the bucket actuator functions with different angles to ensure the function worked correctly. We tested the function multiple times to ensure consistency and also made sure the bucket arms did not hit the sieve with significant force. We added pressure sensors to the bottom of the bucket and monitored the change in the values of the pressure sensor readings when we put gravel in the bucket. We tested it with a fully loaded bucket as well to find the thresholds we needed to signal the excavator to stop digging and prepare for depositing.

## 5 Discussion

This section takes a closer look at the results and what they mean for accomplishing our design goals. The table below summarizes our objective and what we have achieved this term.

<b>Parameters</b>	<b>Minimum Goals</b>	<b>Reach Goals</b>	<b>Achieved</b>
<u>Level of Autonomy</u>	Obstacle Avoidance	Autonomous Navigation	Strong foundation for autonomy has been established, code has been written and tested on the RaspPi that can be used to localize the robot with April Tag detection and RealSense camera can be used to generate a map and provide a path to robot to avoid obstacles. Team was not able to integrate it with the roboRIO and achieve autonomous navigation due to limited time as we were met with a lot a of compatibility issues.
<u>Dust Protection</u>	IP50	IP60	IP50 achieved properly through dust protection material, transmission system cover and new electrical box.
<u>Amount of Gravel Collected</u>	2 kg in 15 min	3 kg in 15 min	Both achieved, the new storage and deposit system can collect and deposit 3 kg of gravel in 4 secs
<u>Competition Cycle Speed</u>	12 minutes	10 minutes	With all systems requiring significantly less time the competition cycle speed has been reduced greatly and would require around 10 mins.
<u>Maximum Mass</u>	55 kg	50 kg	Final mass was 35.7kg
<u>Competition Defined Cycles</u>	Complete 1 full cycle	Complete 2 full cycles	We were able to test one cycle successfully using teleoperation.

<p><u>Safety Rating</u></p>	<p>Lunabotics requirements (e.g.: switches, handles, e-stop)</p> <p>Smart Safety Telemetry System (e.g.: temp., current draw, battery voltage)</p>	<p>Visual indicator system or GUI to convey issues back to the ground station.</p>	<p>Robot Safety Light added to view if the robot is enabled or disabled. SmartDashboard used to convey (1) voltage, current draw, and temperature of each motor, (2) the angle position of the bucket and excavator, (3) to show the value of the pressure sensor, (4) the voltage of each battery cell.</p>
-----------------------------	--	--	--

Table 16: The team’s goals for the robot this year with descriptions on what was achieved

**5.1 Excavator**

While the excavator itself was not a project objective, it contributed significantly to the overall autonomy and cycle time goals. By adding the hall effect sensor, we were able to move the excavator between 6 key positions in the autonomous process. For the excavation period, we implemented current monitoring of the belt system to prevent stoppages or errors during the digging process. Both led to smooth autonomous processes that could be relied upon in a testing run. Additionally, replacing the linear screw system provided a maximum speed increase of 1 minute and 30 seconds to the reconfiguration speed of the robot. While the calculations for the best output force-to-speed ratio for the linear screw could be further optimized, the team deemed this time reduction acceptable for the system redesign. Further optimization was not critical to the excavator's success.

**5.2 Storage and Deposit**

The new storage and deposit system met all of the objectives it was designed to. It increased the ground clearance in the storage/depositing configuration by seven inches, was able to deposit about three kilograms of gravel with minimal losses, and helped the robot achieve other goals. Notably, the new system was 5.5 kilograms less massive than last year’s conveyor system. It could deposit its load of gravel in four seconds which was over 13 times faster than last year’s storage conveyor, saving time that the robot could use to complete another cycle. For reference, the runner-up in last year’s competition, the University of Alabama, took 25 seconds to deposit a load of gravel with their conveyor. Compared to other teams that used a bucket, SDSU’s robot, as seen in Figure took eight seconds to deposit, and the University of Akron’s robot from this year took 21 seconds to deposit (The University of Akron's 2023 Proof of Life Video, 2023). While this robot will not be tested at an in-person competition, the bucket test results are a good sign that the system would probably do well if the robot were to be brought to an in-person competition.

### **5.3 Vision/Navigation**

The Vision/Navigation system was the subsystem in which we faced the most difficulties. This subsystem was essential to achieving full autonomy. We were successfully able to write and run code on the RaspberryPi, use a web camera, and a stepper motor to detect and follow the April Tag. When integrated with the robot, this will provide localization data and more accurate odometry than what encoders on the wheel motors could offer. A custom ROS wrapper was written to generate a PointCloud from the data we received from the RealSense camera. The mapping and path planning approach made use of the ROS octomap\_server to visualize the map. This provided a snapshot of the obstacles in front of the robot, unfortunately the system was unable to perform the dynamic mapping of obstacles when the robot moved due to computational and memory constraints. The subsystem succeeded in 18/19 system requirements outlined in section 3.3.3 unfortunately integration with the other systems caused substantial challenges and was not completed. There is still some work to be done in order to integrate this subsystem with the robot, by coordinating the Raspberry Pi 4B and the roboRIO so the robot can be localized using AprilTag Tracking. This will improve odometry accuracy and allow the robot to accurately follow the path calculated by the vision processing module.

### **5.4 Autonomy Integration**

The autonomous subsystems all worked as expected. When the robot is manually given a cartesian coordinate, it is able to travel to the specified coordinate reliably by orienting itself appropriately and driving to it. The excavator and bucket systems can be autonomously controlled, and safety checks are in place to ensure those moving subsystems do not go beyond what they are physically constrained. Compared to last year, when all of these systems had to be controlled remotely through teleoperation, our autonomous integration has successfully automated these systems with feedback control. There is potential room for improvement in the driving capabilities of the robot as our software does not effectively account for the slippage of the wheels in the sand. Furthermore, the states of our excavator and bucket are mostly binary, which could be improved to be more modular and accommodate for more needed positions instead of just two. Overall, our autonomous integration has made it easier for anyone to utilize our robot system and understand its functionality.

## 6 Conclusions and Recommendations

By the end of the project, the team realized a robot that was able to meet most of the goals set at the start of the project. After researching previous WPI designs as well as the designs of other teams, the team was able to create a new storage and deposit system that was not detrimental to ground clearance, deposited quickly, was lighter, and utilized fewer moving parts than last year's conveyor system. This was one of our major mechanical redesigns, and despite the size constraints of the previous robot's chassis, we were able to integrate this new design and actuate the system successfully.

We achieved autonomous operation in each subsystem with the supplementation of sensor feedback, increasing the overall efficiency and operation of the robot. The autonomous integration of these subsystems also meant increased reliability in the individual performance of each subsystem.

The team would also like to make the following recommendations for future iterations of the project. The electrical wiring on the robot has a lot to be improved. The CAN wires on the robot are currently connected at its joints with solder. A potential fix would be to use lever wire connectors to ensure a strong connection while making the wiring look more professional and well done. Additionally, the corrugated wire guards could be replaced with cable carriers. For the electrical box, add dust-proof rated connectors to the box to isolate the inside and outside environments.

For the mechanical aspects of the robot, the roller chain tensioners, which prove to get loose on excavator rotation, could be tightened or redesigned to provide the needed tension throughout the excavation process without hindering other assemblies. Additionally, the point of contact from the transmission gear to the lever of the excavator and bucket should be extended so as to reduce the torque loads on the hex shaft, which currently provides some play in the bucket system that is slightly disadvantageous for repeatability. Lastly, there could be considerations made to changing the bucket hex shaft to a round dead shaft to directly transmit power from the gear to the arm instead of losing some efficiency with the current hex shaft in its fitting.

For the software aspects of the robot, especially the vision processing portion of our project, we ran into numerous issues when it came to compatibility between the RealSense Camera and the RaspberryPi. We had solutions for path planning but with every compatibility issue we had to come up with alternate solutions. Our team was able to establish a strong foundation in autonomous operation for future teams as we were able to try and test multiple solutions on the RaspberryPi, but with the limited time we were unable to integrate it with the mechanical systems. We would recommend future teams to use a more powerful image processor to avoid such issues and also integrate the mechanisms with roboRIO.

## References

*Amazon.com: 2-channel isolated can bus expansion hat compatible with ... (n.d.). Retrieved April 27, 2023, from <https://www.amazon.com/2-Channel-CAN-HAT-Compatible-SN65HVD230/dp/B08BFZMHJ4>*

*AndyMark. (2021). AM-3656 Motor curve sheet. [PDF]. [https://cdn.andymark.com/media/W1siZiIsIjIwMjAvMDQvMDMvMTIvMTMvMjQvZTlmMjY2NGEtMmNhYy00OTc1LWJkMGEtZDdjZDk4YzAwMjE5L2FtLTM2NTYgTW90b3IyY3VydmUgc2hlZXQucGRmIl1d/am-3656%20Motor%20curve%20sheet.pdf?s\\_hash=6b7667d3338bd980](https://cdn.andymark.com/media/W1siZiIsIjIwMjAvMDQvMDMvMTIvMTMvMjQvZTlmMjY2NGEtMmNhYy00OTc1LWJkMGEtZDdjZDk4YzAwMjE5L2FtLTM2NTYgTW90b3IyY3VydmUgc2hlZXQucGRmIl1d/am-3656%20Motor%20curve%20sheet.pdf?s_hash=6b7667d3338bd980)*

*BAG Motor - VEXpro Motors - VEX Robotics. (n.d.). BAG Motor - VEXpro Motors - VEX Robotics. <https://motors.vex.com/vexpro-motors/bag-motor>*

*Beech, J. (2017, March 6). *The care and feeding of LiPo packs*. ProModeler. <https://www.promodeler.com/askJohn/How-To-Care-for-LiPo-Packs>*

*Bimonte, K., Burack, H., Freedman, C., Hogan, J., Hogan, M., & Kuberka, N. (2020). NASA Lunabotics 2019-2020 [Major Qualifying Project, Worcester Polytechnic Institute]. Digital WPI. <https://digital.wpi.edu/pdfviewer/4j03d2179>*

*Hartman, C., Parker, J., Hou, K., James, T., Yurcak, J., Rossetti, M., Jagdish, N., Ng, N., Grant, P., Chava, M., Sheats, J., McCarthy, S., Akinyele, J., & Sterrett, T. (2022). *NASA Lunabotics*. : Worcester Polytechnic Institute.*

*Houbre, D., Mushatt, J., Spalding, L., Villafana, S., & Wang, K. (2021). NASA RMC: Lunabotics [Major Qualifying Project, Worcester Polytechnic Institute]. Digital WPI. <https://digital.wpi.edu/pdfviewer/sj139462s>*

*Martin, J. (2015, November 12). *A Crimson Dynasty*. College of Engineering News Site. Retrieved April 26, 2023, from <https://news.eng.ua.edu/2015/11/a-crimson-dynasty/>*

*Montana Tech NASA Robotics. Montana Technological University. (n.d.). Retrieved October 6, 2022, from <https://impact.mtech.edu/project/32279>*

*NASA Lunabotics 2023 Guidebook. (2022, September 9). [https://www.nasa.gov/sites/default/files/atoms/files/lunabotics\\_guidebook\\_2022\\_ver\\_4.0\\_02.23.2022.pdf](https://www.nasa.gov/sites/default/files/atoms/files/lunabotics_guidebook_2022_ver_4.0_02.23.2022.pdf)*

National Academies of Sciences, Engineering, and Medicine 2022. *Origins, Worlds, and Life: A Decadal Strategy for Planetary Science and Astrobiology 2023-2032*. Washington, DC: The National Academies Press. <https://doi.org/10.17226/26522>

PG Series Gearboxes. (n.d.). Www.andymark.com. Retrieved April 27, 2023, from <https://www.andymark.com/products/hex-pg-series-gearboxes-options?Gear%20Ratio=188%3A1&Motor%20%26%20Encoder%20=Installed&Shaft%20Size=1%2F2%20Hex>

Stafford. (2023, January 18). *Motor Projection Template*.

*Studica. roboRIO 2.0. (n.d.). Studica Limited. <https://studica.ca/en/ni-roborio-20>*

*The Artemis Plan: NASA's Lunar Exploration Program Overview. (2020, September). [https://www.nasa.gov/sites/default/files/atoms/files/artemis\\_plan-20200921.pdf](https://www.nasa.gov/sites/default/files/atoms/files/artemis_plan-20200921.pdf)*

*University of Alabama Astrobotics. (2021). Alabama Astrobotics. <https://www.alabamaastrobotics.com/>*

Wamback, C. (2017, May 17). WPI Students are Digging Deep to Understand Mars. Worcester Polytechnic Institute. Retrieved January 23, 2023, from <https://www.wpi.edu/news/wpi-students-are-digging-deep-understand-mars>

YouTube. (2022). *2022 Space Raiders Proof of Life. YouTube*. Retrieved October 6, 2022, from [https://www.youtube.com/watch?v=el\\_QOnVRUi8](https://www.youtube.com/watch?v=el_QOnVRUi8).

YouTube. (2022). *Nasa Lunabotics 2022 Pol | Atlas | University of Illinois at Chicago. YouTube*. Retrieved October 6, 2022, from <https://www.youtube.com/watch?v=qedJUfM2PpY>.

YouTube. (2022). *Proof of Life Team Hades San Diego State University Nasa Rmc 2022. YouTube*. Retrieved October 6, 2022, from <https://www.youtube.com/watch?v=OP11x6xWztk>.

YouTube. (2022). *Purdue Lunabotics Proof of Life Submission 2022. YouTube*. Retrieved October 6, 2022, from <https://www.youtube.com/watch?v=UkdBPDy9G0c>.

YouTube. (2023). *The University of Akron's 2023 Proof of Life Video. YouTube*. Retrieved April 23, 2023, from [https://www.youtube.com/watch?v=3mQcF\\_pKkOo](https://www.youtube.com/watch?v=3mQcF_pKkOo).

## Appendix A

<i>All Expenses Report</i>				
<i>Hardware</i>			<b>Budget</b>	<b>Spent</b>
			\$1,000	\$0
<b>Item</b>	<b>Description/Units</b>	<b>Quantity</b>	<b>Unit Cost</b>	<b>Total Cost</b>
Allen Key Set	Make sure it fits the requires fasteners	1.00		\$0
Screwdriver Set	N/A	1.00		\$0
Wrench Set	N/A	1.00		\$0
Vacuum	N/A	1.00		\$0
<i>Electronics</i>			<b>Budget</b>	<b>Spent</b>
			\$500	\$42.35
<b>Item</b>	<b>Description/Units</b>	<b>Quantity</b>	<b>Unit Cost</b>	<b>Total Cost</b>
Lipo Charger	<a href="#">/Ctoys-and-games%2C80&amp;ref=nb_sb_noss</a>	1.00	\$42.35	\$42.35
XT60 Connectors	<a href="#">Caps%2C128&amp;ref=nb_sb_ss_ts-doa-p_1_4</a>	1.00		
RasPi Power Cords	<a href="#">y+pi+power+wires%2Caps%2C79&amp;sr=8-5</a>	1.00		
Lipos 3S Battery	<a href="#">-bZDvEDr3wyagChXBoCr48QAvD_BwE</a>	3.00		\$0
<i>Software</i>			<b>Budget</b>	<b>Spent</b>
			\$250	\$0
<b>Item</b>	<b>Description/Units</b>	<b>Quantity</b>	<b>Unit Cost</b>	<b>Total Cost</b>
				\$0
				\$0
<i>Misc.</i>			<b>Budget</b>	<b>Spent</b>
			\$250	\$0
<b>Item</b>	<b>Description/Units</b>	<b>Quantity</b>	<b>Unit Cost</b>	<b>Total Cost</b>
				\$0
				\$0



## Appendix B

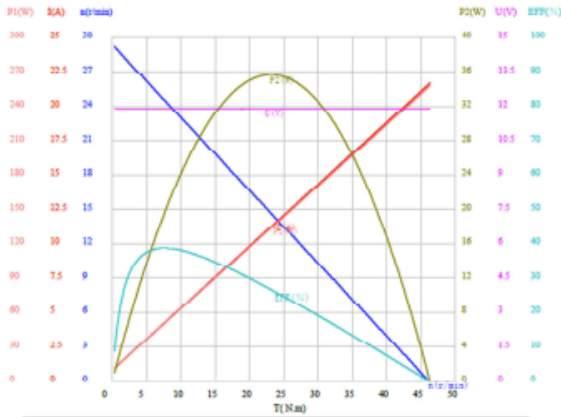
<b>Funding</b>	
Professor Cuneo's PDA	\$2,000.00
Funding from Majors	\$1,750.00
Award Funding from Last Year	\$250.00
Individual Team Contributions (\$200 per student)	\$1,200
<b>Total Funding</b>	<b>\$5,200.00</b>



AM-3656 and AM-3653 Technical Information

PG188 gearmotor, using 12 vdc am-3108 motor and encoder

Date: 4/2/2020



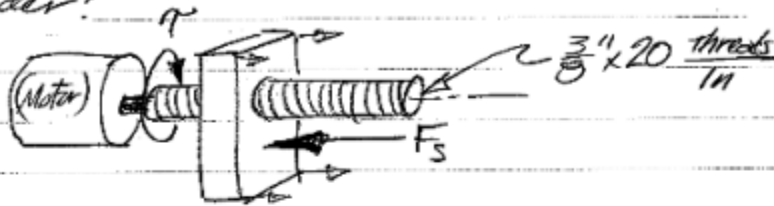
	U(V)	I(A)	P1(W)	T(N.m)	n(r/min)	P2(W)	EFP(%)	P.F
No Load Point	11.89	0.978	11.63	0.33	29.2	1.019	8.765	1.000
Stall Point	11.91	21.65	257.5	46.31	0	0	0.000	0.998
Max Torque	11.91	21.63	237.3	46.31	0.0	0.000	0.000	0.998
Max Output Power	11.90	11.26	134.0	23.21	14.6	35.66	26.61	0.999
Max Efficiency	11.89	4.198	49.94	7.50	24.6	19.35	38.74	1.000
Rated Point	11.89	5.325	63.34	10.00	23.0	24.10	38.05	1.000

U(V)	I(A)	P1(W)	T(N.m)	n(r/min)	P2(W)	EFP(%)	P.F
11.89	0.978	11.63	0.33	29.2	1.019	8.765	1.000
11.89	1.497	17.81	1.49	28.4	4.437	24.91	1.000
11.89	2.017	23.99	3.64	27.7	7.677	32.00	1.000
11.89	2.536	30.17	5.80	26.9	10.74	35.00	1.000
11.89	3.055	36.35	4.95	26.2	13.63	37.48	1.000
11.89	3.575	42.53	6.11	25.5	16.33	38.41	1.000
11.89	4.094	48.71	7.26	24.7	18.86	38.73	1.000
11.89	4.614	54.89	8.42	24.0	21.22	38.65	1.000
11.89	5.133	61.06	9.57	23.3	23.39	38.31	1.000
11.89	5.652	67.24	10.73	22.5	25.39	37.78	1.000
11.90	6.172	73.42	11.89	21.8	27.21	37.06	1.000
11.90	6.691	79.60	13.04	21.1	28.85	36.25	0.999
11.90	7.210	85.78	14.20	20.3	30.52	35.34	0.999
11.90	7.729	91.96	15.35	19.6	31.61	34.37	0.999
11.90	8.249	98.14	16.51	18.9	32.72	33.34	0.999
11.90	8.768	104.3	17.66	18.1	33.65	32.26	0.999
11.90	9.288	110.5	18.82	17.4	34.40	31.14	0.999
11.90	9.807	116.7	19.97	16.7	34.98	29.98	0.999
11.90	10.327	122.9	21.13	15.9	35.38	28.80	0.999
11.90	10.845	129.0	22.28	15.2	35.60	27.59	0.999
11.90	11.37	135.2	23.44	14.3	35.63	26.37	0.999
11.90	11.89	141.4	24.59	13.7	35.52	25.12	0.999
11.90	12.40	147.6	25.75	13.0	35.21	23.86	0.999
11.90	12.91	153.7	26.90	12.3	34.73	22.58	0.999
11.90	13.44	159.9	28.06	11.5	34.06	21.30	0.999
11.90	13.96	166.1	29.21	10.8	33.21	20.00	0.999
11.91	14.48	172.3	30.37	10.1	32.09	18.69	0.999
11.91	15.00	178.5	31.52	9.3	31.00	17.37	0.999
11.91	15.52	184.6	32.68	8.6	29.62	16.04	0.999
11.91	16.04	190.8	33.83	7.9	28.07	14.71	0.999
11.91	16.56	197.0	34.99	7.1	26.34	13.37	0.999
11.91	17.08	203.2	36.14	6.4	24.43	12.03	0.998
11.91	17.60	209.4	37.30	5.7	22.35	10.68	0.998
11.91	18.12	215.5	38.45	4.9	20.00	9.330	0.998
11.91	18.64	221.7	39.61	4.2	17.65	7.990	0.998
11.91	19.16	227.9	40.76	3.5	15.03	6.596	0.998
11.91	19.68	234.1	41.92	2.7	12.34	5.238	0.998
11.91	20.20	240.3	43.07	2.0	9.365	3.856	0.998
11.91	20.71	246.4	44.23	1.3	6.116	2.481	0.998
11.91	21.23	252.6	45.38	0.5	2.789	1.104	0.998
11.91	21.85	257.5	46.51	0.0	0.000	0.000	0.998

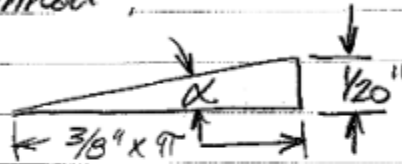
Datasheet and details of the motor used to actuate the bucket.

## Screw Thread Analysis:

Consider:

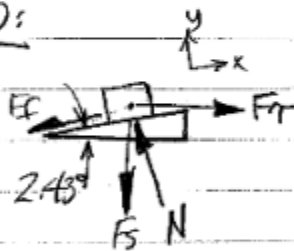


"Unwrap 1 thread":



$$\therefore \alpha = \tan^{-1}\left(\frac{.05}{1.178}\right) = \tan^{-1}\left(\frac{1}{23.6}\right) = 2.43^\circ$$

FBD:



EofE:

$$\textcircled{1} \sum F_x = 0 = F_T \cos \alpha - F_f - F_s \sin \alpha$$

$$\textcircled{2} \sum F_y = 0 = N - F_s \cos \alpha - F_T \sin \alpha$$

$$\textcircled{3} F_f = \mu N$$

Note:  $\cos(2.43^\circ) = .999 \approx 1$

$\sin(2.43^\circ) = .042$

$$\therefore \text{from } \textcircled{1} \Rightarrow F_T = F_s (\sin \alpha) + F_f \quad \textcircled{4}$$

$$\text{" } \textcircled{2} \Rightarrow N = F_T (\sin \alpha) + F_s \quad \textcircled{5}$$

since  $F_T \ll F_s$  (with no friction  $\frac{1}{23}$ )  
and  $\sin \alpha \approx \frac{1}{23}$  then  $F_T (\sin \alpha) \ll F_s$  so...

$$\textcircled{6} \boxed{N \approx F_s}$$

From (4):  $\frac{F_s}{F_q} (\sin \alpha) = 1 - \frac{F_f}{F_q}$

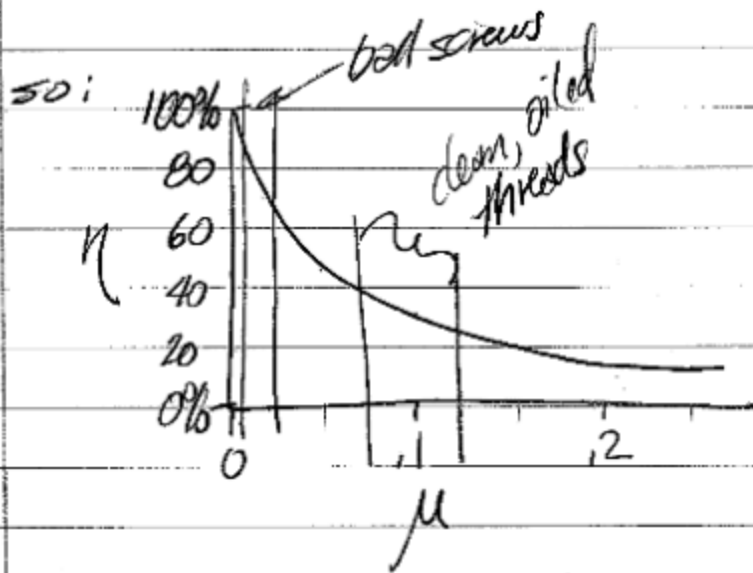
$\therefore$  Mech Advantage  $\equiv \frac{F_s}{F_q} = \frac{1 - \frac{F_f}{F_q}}{\sin \alpha}$  (7)

Efficiency  $\equiv \eta = 1 - \frac{F_f}{F_q}$  (8)

(eg. @  $F_f = 0$ ,  $\eta = 1.0$ ,  $\therefore$  Mech Adv = 23.6)

From (8), (3), (4), (6):

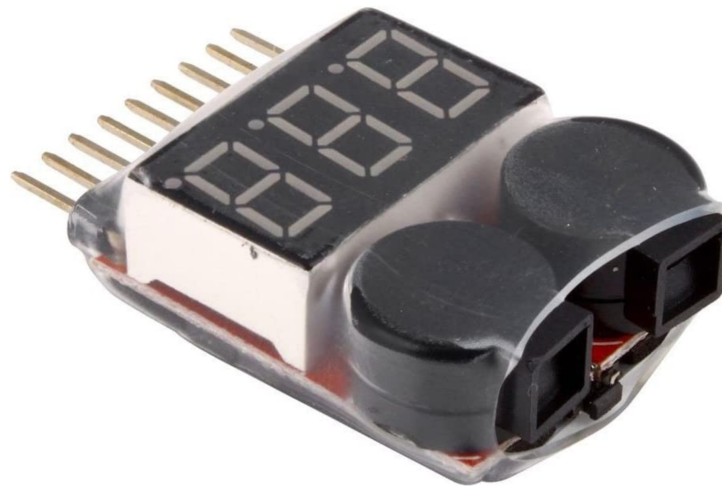
$\eta = 1 - \frac{\mu F_s}{F_s (\sin \alpha + \mu)} = 1 - \frac{\mu}{\sin \alpha + \mu}$  (9)



## Appendix E

As the FRC Driver Station (DS) was used to communicate and enable/disable the robot, the DS can display the supply voltage to the roboRIO. As the roboRIO is typically powered from the Power Distribution Panel (PDP) with a 12V lead-acid battery, the suggested voltage to swap a battery for a LiPo is different at 11.2V – instead of 10.5V.

Early adoption of Smart Safety was the use of a low-voltage alarm for the battery. Since we have used LiPo batteries, their voltage needs to be monitored to not overdraw from the battery's capacity and to warn of any voltage drop so as not to cause damage to any electrical component on the robot. As part of monitoring a LiPo battery's voltage, the individual cell's voltage can be monitored from the balance connector to warn the operator when at least one cell's voltage level drops below 3.75V or about 25% of the remaining battery's capacity. It is recommended by Beech (2017) not to go below 20% of a LiPo cell's capacity.



(CAMWAY via Amazon.com)

Figure E.1: CAMWAY 1-8s LiPo Battery Voltage Tester, RC Low Voltage Buzzer Alarm

A CAMWAY LiPo Battery Voltage Tester was strapped to a battery, and the battery's balance connector was plugged into four of the closest to ground leads of eight to help remind robot handlers and operators when the voltage of at least one of the battery's cells got too low – to then swap out for a new battery. Since the target voltage per cell to stop at was 3.75V, but the device can only increment in 0.1V, the alarm was set to go off if any of the three cells reached about 3.8V. The product advertised voltage detection precision to +/- 0.01V.



Figure E.2: A generic Digital Battery Capacity Tester from Makerfire Store

A digital battery capacity tester, as commonly used with RC Hobbyists, was used along with a Mastech MAS830 Multimeter to validate voltage readings from the CAMWAY LiPo Battery Tester. The values in the Table E.1 below were recorded by taking measurements from their respective device.

Comparing Voltage Readouts of a 3S Ovonc Car LiPo

Device:	Battery (V)	Cell 1 (V)	Cell 2 (V)	Cell 3 (V)
CAMWAY Tester	11.2	3.74	3.73	3.75
Makerfire Tester	11.24	3.745	3.748	3.751
Mastech Meter	11.27	3.75	3.75	3.76

Table E.1: Comparing Voltages of Three Voltage Testers

As part of moving forward with monitoring battery cell voltages, a prototype board of a breadboard was put together to plug jumper wires into the balance connector of the onboard battery and into the roboRIO. As two analog pins of four on the roboRIO are taken by the hall effect sensor and pressure sensor, two remaining slots were temporarily used for this task. As the battery's voltage is already monitored within the roboRIO, only two voltages between one cell and a combined voltage or two cells would be needed to monitor the cells' voltages. As the analog pins are recommended for 5V and that two cells charged at 4.2V each would go over 5V at 8.4V, a voltage divider was used. Two 10kOhm resistors were chosen to be used as they are common and provide a balance between keeping the circuit's sensitivity to thermal noise (Johnson-Nyquist noise) low and keeping the final voltage closer to theoretical, though it draws more current.

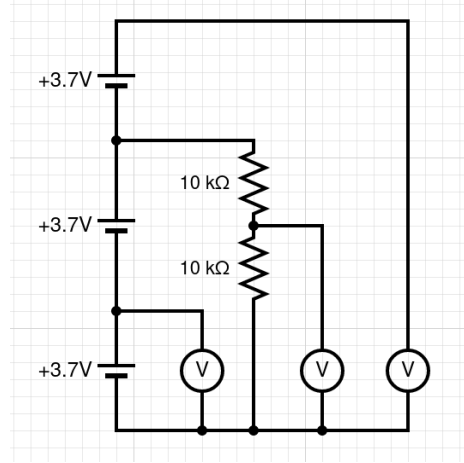
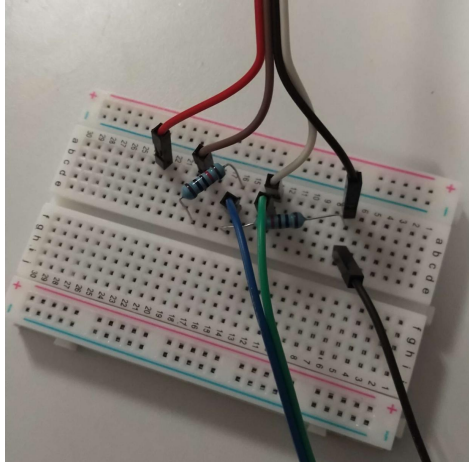


Figure E.3a: Prototype Board with Voltage Divider    Figure E.3b: Circuit Schematic

From Figure E.3a, the red, brown, white, and black wires came in from a LiPo battery. Since the battery's voltage across the main terminals was already monitored by the roboRIO, the red wire is not needed. The blue, green, and black wires went to the roboRIO with the analog port bay. The blue wire is for the voltage across cells 2 and 3 going to Analog Port 3. While the voltage was divided in half – this was compensated for in code to obtain an approximate value. The green wire represents the voltage across cell 3 and goes to Analog Port 4. The remaining black wire goes to ground on the analog port rail on the roboRIO to shorten the distance and reduce excessive resistance in wires to obtain a closer reading.

From Figure E.3b, the three cells at nominal voltage 3.7V are shown. However, when charged, each cell started at 4.2V, so measuring across two cells would go above the recommended maximum voltage of the analog signal pins of 5V. The theoretical max of 8.4V from cells 2 and 3 needed to be reduced to below 5V and two 10kOhm resistors with 1% rated tolerance were used to cut the voltage by about half.

$$V_{out} = V_{in} * \frac{R2}{R1+R2} \quad (\text{Eq. E.1})$$

$$V_{in} = 8.4V, R1 = 10kOhm, R2 = 10kOhm$$

$$V_{out} = 8.4V * \frac{10kOhm}{10kOhm + 10kOhm}$$

$$V_{out} = 4.2V$$

The theoretical voltage was at 4.2V, while measuring with a multimeter showed 4.18V with 8.4V input on the power supply. The analog pin on the roboRIO measures voltage from 0V to 5V and uses an analog-to-digital converter (ADC) with 12-bit. To find the voltage measured on the analog pin, we used the following formula/equation:

$$\text{Voltage} = (\text{Digital Value} / 4095) * 5.0 \quad (\text{Eq. E.2})$$



Equation E.2 was used with the equations below to calculate approximate values for the voltages of each cell. DigitalValue of cells 2 and 3 took in voltage reading after the voltage divider, to compensate the reduction, the calculated value was multiplied by 2.

$$\text{batteryVoltage} = \text{frc}::\text{DriverStation}::\text{GetBatteryVoltage}();$$

$$\text{cellVoltage}_{2\text{ and }3} = (\text{DigitalValue}_{2\text{ and }3} / 4095) * 5.0 * 2.0 \quad (\text{Eq. E.3})$$

$$\text{cellVoltage}_1 = \text{batteryVoltage} - \text{cellVoltage}_{2\text{ and }3} \quad (\text{Eq. E.4})$$

$$\text{cellVoltage}_3 = (\text{DigitalValue}_3 / 4095) * 5.0 \quad (\text{Eq. E.5})$$

$$\text{cellVoltage}_2 = \text{cellVoltage}_{2\text{ and }3} - \text{cellVoltage}_3 \quad (\text{Eq. E.6})$$

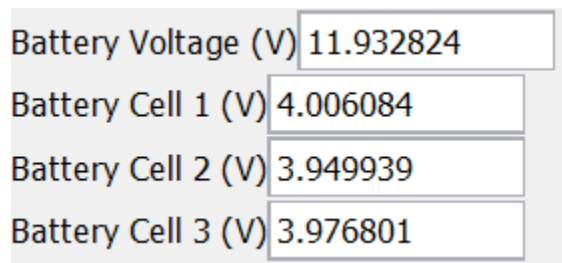


Figure E.4: FRC Smart Dashboard with Battery and Cell Voltages



Figure E.5: FRC Driver Station Battery Voltage Meter

Device:	Battery (V)	Cell 1 (V)	Cell 2 (V)	Cell 3 (V)
FRC Driver Station	11.94	N/A	N/A	N/A
Smart Dashboard	~11.93	~4.01	~3.95	~3.98

Low Voltage Meter	11.9	4.02	3.98	3.97
Battery Capacity Tester	11.97	4.009	3.991	3.972
Multimeter	11.99	4.01	3.99	3.97

Table E.2: Sample 1 Validation Data of Measuring Voltages of Battery #1 within 1%