WORCESTER POLYTECHNIC INSTITUTE
ROBOTICS ENGINEERING PROGRAM

Ibex: Robotic Mining Platform

A Major Qualifying Project

Submitted to the Faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the

Degree in Bachelor of Science
in
Robotics Engineering & Computer Science

April 26, 2018

TEAM MEMBERS

Nikhil Castelino, Thomas Hagen, Rayyan Khan,
Kushagra Kumar, Parmenion Patias

FACULTY ADVISORS

Michael Ciaraldi, Kenneth Stafford

Abstract

The use of resources in foreign environments is essential to the success of manned missions to Mars. This project explores the different ways a rover can mine and deliver resources in a simulated Martian environment. This robot is capable of autonomously excavating the simulated ice chunks 30 cm (11 in) below the surface and driving to a collection station to unload the material it has collected. This project was inspired by the NASA Robotic Mining Competition which established a set of rules for how the robot was to be constructed.

Acknowledgements

The team would like to thank the advisors Professors Michael Ciaraldi and Kenneth Stafford for their constant input and advice throughout the project cycle. Special thanks to Tom Gravel at Hydrocutter Inc. for generously providing his water jet services free of charge and Worcester Sand & Gravel for providing us with free testing material. The team would also like to thank Julie Chapman for donating her used treadmill to the team for use on our robot. Finally, the team would like to extend our gratitude towards Torbjorn Bergstrom, Ian Anderson, and James Loiselle of Washburn Shops for their help in machining parts of the robot and providing a workspace.

Contents

1 Background 1

1.1 Mars . . . 2

1.2 Markhor . . . 3

1.3 NASA Robotic Mining Competition . . . 3

1.4 Excavation Methods . . . 4

1.4.1 Bucket Wheel . . . 5

1.4.2 Auger Drill . . . 5

1.4.3 Clamshell Bucket . . . 6

1.4.4 Trencher . . . 7

1.4.5 Markhor's Excavation System . . . 8

1.5 Drivetrain Options . . . 9

1.5.1 Rigid Chassis . . . 9

1.5.2 Rocker Bogie Suspension System . . . 9

1.5.3 Wheels vs. Treads . . . 10

1.6 Electrical Components . . . 11

1.6.1 Raspberry Pi . . . 11

1.6.2 Talon SRX . . . 12

1.6.3 Arduino . . . 13

1.7 Software and GUI Design . . . 14

1.7.1 Programming Languages . . . 14

1.7.2 Previous Team's Programming . . . 14

2 Methodology **16**

2.1 Improvements to Markhor 16

2.1.1 Excavation robustness 16

2.1.2 Autonomy 17

2.1.3 Weight 18

2.2 Milestone: System Requirements 19

2.2.1 Planned Size 19

2.2.2 Planned Mass 19

2.2.3 Payload Capacity and Gravel Collection 20

2.2.4 Control 20

2.3 Milestone: Robot's Plan of Operation 20

2.3.1 Finding Robot Orientation 20

2.3.2 Going to Border of Starting Zone and Obstacle Area 21

2.3.3 Driving to the Excavation Zone 21

2.3.4 Mining Regolith and Gravel 21

2.3.5 Drive to Collection Bin 21

2.3.6 Align to the Collection Bin 22

2.3.7 Deposit Collected Material 22

2.4 Prototyping 22

2.5 Modeling and Design Phase I 22

2.6 Milestone: Preliminary Design Review 23

2.7 Budgeting 23

2.8 Modeling and Design Phase II 24
2.9 Manufacturing 25
2.10 Testing 25
2.11 Milestone: Critical Design Review 26

3 Mechanical System **27**
3.1 Locomotion 27
3.2 Excavation 28
3.2.1 Simultaneous Carriage Motion 28
3.2.2 Scoop Design and Connection 29
3.3 Material Deposit 31

4 Electronics Control System **33**

5 Software Control System **35**
5.1 Robot Controller 35
5.2 Control Station 36
5.3 Controller Board Communications 38
5.3.1 HERO Motor Controller Communication 38
5.3.2 Arduino Sensor Controller Communication 39

6 Performance Evaluation **40**
6.1 Mechanical System Testing 40
6.1.1 Testing the Scoop Designs 40

6.1.2 Testing the Excavation System 41

6.1.3 Testing the Conveyor Belt 43

6.1.4 Testing the Vision Processing 45

7 Future Work **48**

7.1 Mechanical Design 48

7.2 Electrical and Computer Engineering Design 48

7.3 Software Design 49

8 Conclusion **50**

Appendix A Gantt Charts for A, B, and C Terms **51**

Appendix B Autonomy Flow Chart **52**

Appendix C UML Class Diagrams **53**

Appendix D Motor Calculations for Sub-systems **55**

List of Figures

1 Sojourner Rover 1
2 Mars Landscape 2
3 Markhor Robot 3
4 Diagram of the Field 4
5 Virginia Tech's 2017 NASA RMC Robot Utilizing a Bucket Wheel 5
6 An Auger Drill in Action 6
7 Clamshell Bucket 7
8 Trencher 8
9 Markhor's Excavation System 8
10 Tank Drivetrain 9
11 Rocker Bogie Drivetrain 10
12 Raspberry Pi 12
13 Talon SRX 12
14 Arduino Mega 13
15 Markhor's GUI For Robot Operation 15
16 Early Render of Ibex Rover 23
17 Rendering of Final Ibex Design 25
18 Ibex Testing Field 26
19 Ibex Drive System Side Rendering 27
20 Driveline Tension Modules 28
21 Excavator in Retracted and Extended Positions 29

22 Connection Between Prototype Scoop and Chain 30

23 Final Scoop Design 31

24 Ibex Unloading system 32

25 Electrical Design Diagram 33

26 Electronics in Ibex 34

27 Software Communication for Ibex 35

28 GUI Screen Shot with Autonomy Queue and Recovery Stack 36

29 GUI Screen Shot for Commands Panel 37

30 GUI Screen Shot with Real-Time Robot Data 37

31 Scoop Design Ideas a) First Iteration, b) Second Iteration c) Third Iteration d) Final Iteration 40

32 Prototype Excavation System 41

33 Prototype Backplate 42

34 Aluminum Tab Connections 42

35 Testing the Digging Mechanism 43

36 Initial Material Deposit Test 44

37 Initial Material Deposit Test with Grousers 44

38 Final Material Depositing System Testing 45

39 Target Design and OpenCV Output 46

40 Final Formula Used for Perceived Area Calculations 47

41 B-term Gantt chart 51

42 C-term Gantt chart 51

43 Current Term Gantt chart 51

44 Flowchart for Autonomous Robot Control 52

45 UML Class Diagrams for Control Station 53

46 UML Class Diagram for GUI and Robot Control 54

47 Calculations for Conveyor System 55

48 Calculations for Gas Spring Mechanism 56

49 Calculations for Depth System 57

1 Background

Since the 1960s, space agencies worldwide have focused on Mars, its theorized ability to host life in the future and the probability of having hosted life in the past. Mars is the second closest planet to Earth (approximately 225 million kilometers away) and even though the two planets may be very different when it comes to temperature, size and atmosphere, they share similar geological processes. This is why many space agencies have attempted to gather as much information on Mars as possible through telescopes, orbiting probes and exploration programs.

Sojourner was the first rover to ever land on Mars in 1997 and since then many have followed.

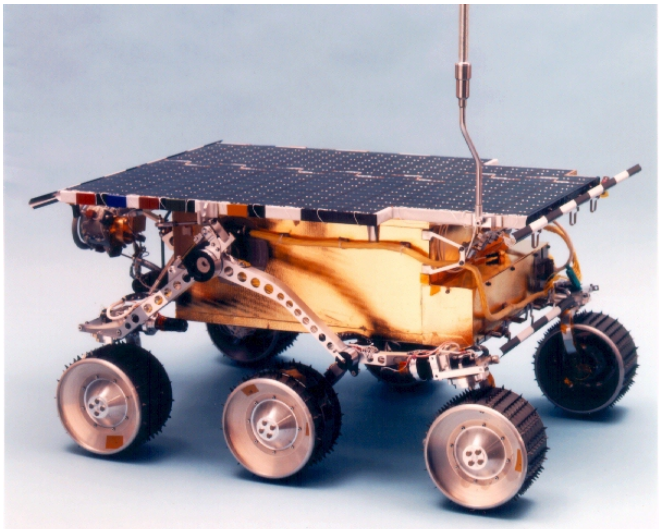


Figure 1: Sojourner Rover[1].

The most recent rover sent to Mars is Curiosity. Its purpose is to explore the Gale Crater and determine if it ever offered environmental conditions favorable for microbial life. A secondary aspect of its mission is to investigate for the presence of water in preparation for human exploration. Curiosity has traversed Mars for over three years past its expected lifetime. Currently, there are many planned rover expeditions to Mars, and the European Space Agency (ESA) has stated that they plan to have manned missions to Mars by 2035. Mars contains carbon dioxide which can be transformed and used as an energy source. In order to harness this energy, an automated process to gather resources and produce energy needs to be created. The National Aeronautics and Space Administration

---

[1] `http://beatty-robotics.com/wp-content/uploads/blue_up1.jpg`

(NASA), in an effort to encourage the manufacturing of a rover with these capabilities, has created the annual Robotics Mining Competition (RMC). The goal of the competition is to build a rover capable of traversing through a simulated Martian field, to avoid collisions with objects, and to collect and return samples back to the collection site. In order to better understand the scenario, areas of research included the planet Mars itself, the competition and its rules, the previous team's project, and existing excavation methods.

1.1 Mars

The fourth planet in our solar system has approximately half the equatorial radius than that of Earth and 36.7% of Earth's volume. Even though Mars is smaller than Earth, Mars' deepest canyon is 7 km, about 3 times deeper than Earth's deepest canyon, the Grand Canyon, which is only 1.8 km deep. Martian gravity is only 0.375 times that of Earth and has harsh topography, which presents a challenge for a robot to traverse. That wasn't a problem for Sojourner that had to travel in total 100 m but created many design requirements for Curiosity. The atmosphere on Mars, or lack thereof, can be a problem which makes it impossible for any electronic system to operate unless it has been properly shielded. Another issue can be the temperature, which on average is -55°C, making some components of the robot require shielding to operate when the temperature drops.

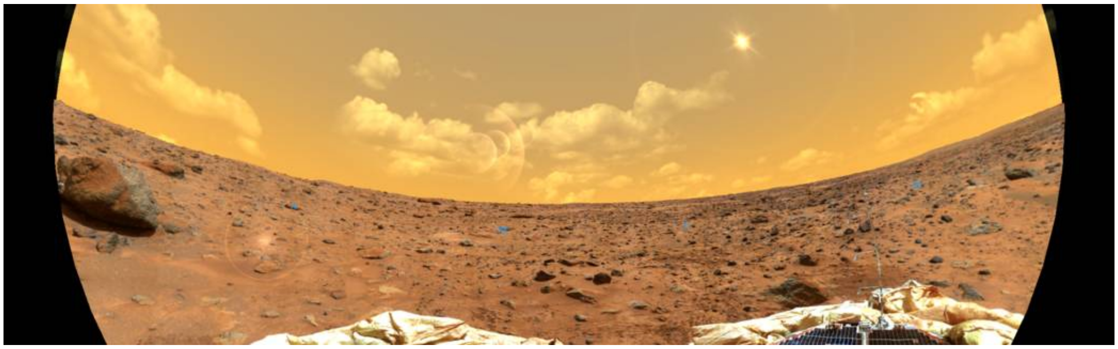


Figure 2: Mars Landscape[2].

---

[2] `https://www.nasa.gov/sites/default/files/styles/ubernode_alt_horiz/public/thumbnails/image/mars_acidia_planetia.jpg`

1.2 Markhor

WPI has had teams participate in NASA's RMC the past 2 years, making this team the third consecutive team to build a robot for the competition. As a result, getting to know how last year's team, Markhor, performed is crucial. To do this, the team spoke with the members of Markhor and studied their robot. Markhor used a novel digging system which rewarded the team with the Regolith Mechanics Award. The robot was able to dig well below the 30 cm threshold to get to the gravel. Its collection system was a bucket capable of depositing over 100 kilograms of regolith simulant. Markhor focused most of their efforts into making a mechanically sound robot, which resulted in them being unable to solve some of the electrical and software aspects of the competition. The final robot was mainly composed of thick aluminum and lexan which unfortunately makes it quite heavy.

Figure 3: Markhor Robot[3].

1.3 NASA Robotic Mining Competition

Started in 2009, NASA's RMC (sponsored by Caterpillar Inc.) focuses on excavation as a prerequisite to visiting the nearby planet. Each team is to present a rover that is under 80 kg and is within the dimensions of 1.5 m length, 0.75 m width and 0.75 m height. When the run starts, the team has five minutes to place the robot in the arena and turn it on. The arena, shown below, is 7.38 m long, 3.88 m wide, 1.0 m deep and has less than 2.50 m of clearance above it. Once the robot has been placed in the arena and the run

---

[3] `https://web.wpi.edu/Pubs/E-project/Available/E-project-042617-133846/unrestricted/MARKHOR_Robotic_Mining_Platform.pdf`

starts the team has ten minutes to transverse through the simulated Martian environment, handle obstacles, dig and get at least 1 kg of simulated ice, and then return and deposit it to the collection area. NASA's rules also don't allow any technology that wouldn't work on Mars (e.g. Global Positioning System). Another significant rule includes a bandwidth limit to simulate the communication limitation between Earth and Mars. [4]

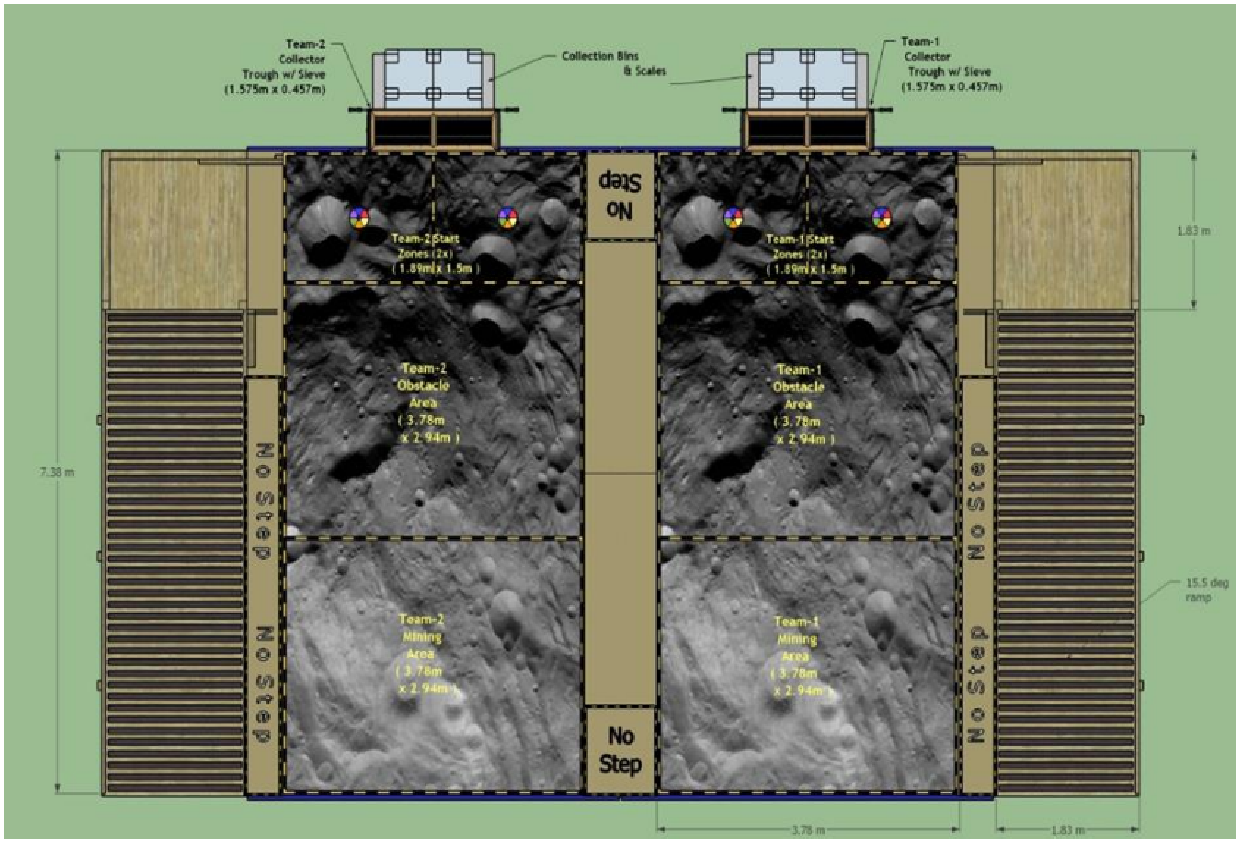


Figure 4: Diagram of the Field[5].

1.4 Excavation Methods

The main aspect of this competition is to mine the gravel which simulates the ice beneath the Martian surface. To understand the best way to go about this challenge, the team researched existing excavation methods and evaluated the pros and cons of each. Below is a collection of excavation systems that were identified as possible paths to take in the construction of the robot's excavation system.

---

[4] https://www.youtube.com/watch?v=s9HEK0IrkjY

[5] https://www.nasa.gov/sites/default/files/atoms/files/rmc2018_rules_rubrics_partiv.pdf

1.4.1 Bucket Wheel

A bucket wheel system, shown below, consists of a large wheel with a continuous pattern of buckets used to scoop material as the wheel turns. The material is then turned over to a belt system and carried away. [6] This method of excavation is used in large-scale mining operations. Due to the fact that the wheel needs to be higher than the chassis itself, there would need to be a modification to the bucket wheel to make it applicable for NASA's RMC.

In the previous year's RMC, Virginia Tech used the bucket wheel system to collect the regolith simulant. In their proof of life demonstration video, they were successfully able to perform the excavation and the dumping system. Even in the actual competition, they were able to successfully collect the regolith simulant although they were unable to deposit the collected regolith simulant.

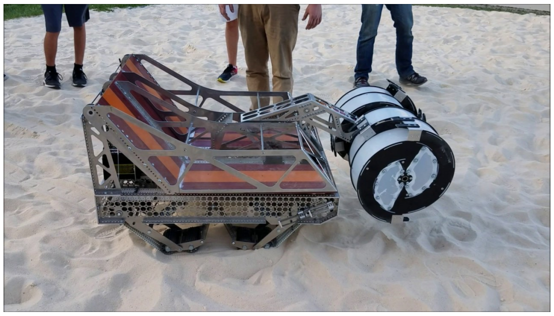


Figure 5: Virginia Tech's 2017 NASA RMC Robot Utilizing a Bucket Wheel[7].

The rotating bucket wheel system was bidirectional in nature. They would rotate in a clockwise direction during collection of the regolith simulant and in a counterclockwise direction to offload the regolith simulant to the collection platform of their robot. Although the system behaved well, the robot seemed unstable when more regolith simulant was collected beyond a point. This caused the center of mass to shift from the center of the robot and make it more prone to tipping over.

1.4.2 Auger Drill

An auger drill utilizes the principles of Archimedes' screw to bore out material from the ground. The continuous flight auger is the most common type of auger that

---

[6] `http://acumen.lib.ua.edu/content/u0015/0000001/0001851/u0015_0000001_0001851.pdf`

[7] `https://www.youtube.com/watch?v=jSRdcl76avA`

lifts material with its helical edges. The system requires immense torque as it needs to overcome the friction of the material it is lifting. The torque requirements are typically met by directly mounting a hydraulic motor or a gear drive system. The main advantage of this technique is the ability to collect material vertically, allowing it to reach greater depths faster than most other techniques.

During the NASA RMC of 2016, the University of Nebraska Lincoln attempted to use this system. The robot was capable of quickly breaking through the top layer of regolith simulant and reaching the gravel. However, the implementation of the design fell short while attempting to unload the material. The auger was unable to lift the material reliably thus causing all of the samples to be lost out of the bottom of the drill.

Figure 6: An Auger Drill in Action[8].

1.4.3 Clamshell Bucket

A clamshell bucket uses two buckets bound together at the middle to form a claw-like system. This system can be used to pick up and at the same time, store material inside it. Thus, a robot using a clamshell bucket may not require an additional storage system. Additionally, the clamshell bucket system can be used to unload the collected gravel at the deposit site. However, a significant disadvantage is that this system requires a high amount of torque to dig into the soil and then to close the two buckets together.

While this system can collect a big load per each dig, the change in competition

---

[8] `https://www.youtube.com/watch?v=trJ1oqJ44wA`

rules can result in gaining very few or zero points for the first dig. For this year's competition, no points are awarded for the collection of BP-1 (regolith simulant). This reduces the need to collect excessive amounts of regolith. A clamshell bucket would easily address this by simply not storing the material gathered in the first excavation. The following loads would have more gravel and result in the team gaining more points.

Another solution to this issue could be to implement another storage container which can be used to hold multiple loads resulting in a robot that has a higher storage capacity. However, this solution will require the storage container to unload the collected gravel and will result in a more complex design compared to a design in which the clamshell bucket can be used to collect and unload gravel.

Figure 7: Clamshell Bucket[9].

1.4.4 Trencher

A trencher is a mechanism that is primarily used to dig channels for drainage or piping. Resembling a chainsaw, the chain of the trencher is held at an angle and wrapped around a metal frame. This mechanism is best suited for hard terrain that a bucket type excavator will not be effective on. Material removed by an excavator is not collected by any mechanism and is instead shot out to the side of the trench being dug.

Additions to the mechanism must be implemented in order for a trencher to be viable in the NASA RMC. The main component that would be necessary is a collection

---

[9] `http://www.kinshofer.com/en/images/djmediatools/km603fkm502.jpg`

system for the cut material. Instead of being shot out of the side of the channel, the material should be stored for later deposit. If a trencher were to be used on a robot for the competition, it would be important to consider the width and depth of the dug trench in order to maximize its potential amount of material.

Figure 8: Trencher[10].

1.4.5 Markhor's Excavation System

The Markhor excavation system, which refers to the digging mechanism used by last year's team, is a chain-driven bucket ladder system. Because of its dynamic chain mechanism, it has the ability to plunge into the ground and extend deeper. The system redirects most of the torque that would be on the chain system onto a guide rail to form a force couple. As material is collected, the buckets follow the chain and deposit material into a collection system.

Figure 9: Markhor's Excavation System[11].

---

[10] `https://www.ditchwitch.com/sites/default/files/styles/hd_scale/public/Trenchers_Hero.jpg?itok=dQooTOh3`

1.5 Drivetrain Options

In order to make the robot move in the most optimal way, considering size limit, field conditions, and obstacles, a variety of drivetrains were looked at. Additionally, when considering drivetrains, possible methods of movement researched included wheels and continuous tread drivelines.

1.5.1 Rigid Chassis

Rigid suspension chassis drive systems, shown below with a six wheel drive base, may be the most simple drive train to construct for a robot. Robots with a rigid chassis can have many wheels on each side, but the minimum would be one wheel on both sides with a caster wheel behind for balance. Drive trains such as the one below are simple to design, build, program, drive, and can be made to be agile.

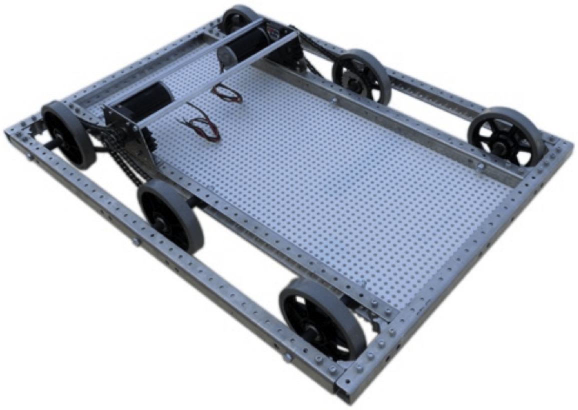


Figure 10: Tank Drivetrain[12].

1.5.2 Rocker Bogie Suspension System

In terms of space exploration, every rover that has been on Mars has used a rocker bogie suspension system. The purpose of this suspension system is to allow

---

[11] `https://web.wpi.edu/Pubs/E-project/Available/E-project-042617-133846/unrestricted/MARKHOR_Robotic_Mining_Platform.pdf`

[12] `http://www.simbotics.org/sites/all/themes/zen/redcub/images/resources/mobility-ben-11.png`

the rovers to climb obstacles and descend into valleys while keeping all wheels on the ground. This driveline is more complex compared to tank drive but adds accessibility to areas where a tank drive system might bottom out and get stuck [13].

Figure 11: Rocker Bogie Drivetrain[14].

1.5.3 Wheels vs. Treads

Wheels have primarily been the choice for all rover missions undertaken by NASA. The Mars rovers Spirit, Sojourner and Opportunity have all used wheeled systems. Rovers are typically designed using six wheels on a rocker-bogie suspension system with four motors. This standard design allows for sufficient maneuverability across Martian terrain. This commonly used system is a consideration for the project, however the use of treads is also a viable option especially in the scope of the competition. Both systems contain different features that are important when designing a robot's drivetrain. Choosing the most suitable system depends on several factors including the traction, ground pressure, suspension, and steering.

The benefits of using wheels revolve around cost, speed, maneuverability, weight, and maintainability. In terms of cost, a wheeled system generally costs less compared to a track system. Wheels also require less torque in order to begin moving from a stationary position. Higher maneuverability is also possible given treads are more prone to slipping. Wheels are a lightweight solution especially in cases such as the competition where the mass of the robot is a critical property. Additionally, wheels are easier to maintain due to

---

[13] `http://dpm.kipr.org/papers/robotics02.pdf`

[14] `http://www.simbotics.org/sites/all/themes/zen/redcub/images/resources/mobility-ben-11.png`

fewer moving components, thus fewer components that can get damaged. However, in terms of obstacles, wheels are not a suitable solution because the wheel's radius must at least be equal to the height of an obstacle [15]. As the competition is to be held on a simulated Martian terrain, thin wheels would not be suitable as they would likely dig into the ground.

In comparison, a continuous track or tread system offers a power-efficient solution, generally with more traction and the ability to travel more smoothly on rough terrain. Treads generally have a higher performance and more optimized system than a wheeled one. On a Martian surface, treads are beneficial due to the ability to surmount obstacles and cross ditches, whereas wheels will often get stuck. However, treads fall short when it comes to speed, maneuverability and weight. The system also has many moving parts which can break, thus must be maintained more frequently[16]. Treads also have a shorter lifespan than wheels and can be more difficult to repair.

1.6 Electrical Components

In order to have an autonomous robot, having durable, robust and reliable electrical components is necessary for a successful operation. The team researched various electrical components that could be used in order to integrate autonomy.

1.6.1 Raspberry Pi

A Raspberry Pi can be used as a miniature computer. It can use a Linux based operating system, handle complex processes, and perform multiple tasks simultaneously [17]. The current Raspberry Pi, Model B+, has four USB ports that allow multiple USB devices to connect to it. It also has a standard ethernet port that can be directly connected to a router or a wired network to enhance performance in terms of connectivity. It comes with a camera connector and has the option of mounting an extra SD/microSD card to increase built-in storage. For this project, a Raspberry Pi could be a great platform for assembling and acting as a control unit for all robot operations.

---

[15] `https://robotics.stackexchange.com/questions/541/wheels-vs-continuous-tracks-tank-treads`

[16] `https://www.intorobotics.com/wheels-vs-continuous-tracks-advantages-disadvantages`

[17] `https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/`

[18] `http://media.rs-online.com/t_large/F8111284-01.jpg`

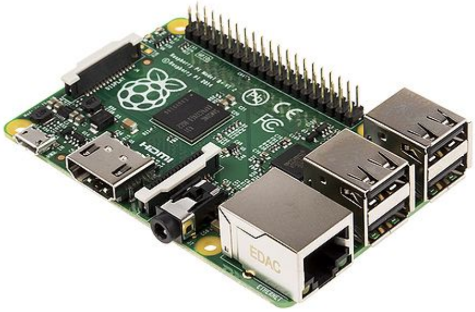


Figure 12: Raspberry Pi[18].

1.6.2 Talon SRX

The Talon SRX is typically useful for driving the motors of a robot as it has an integrated PID controller. The Talon SRX can perform the PID control by reading input values of encoders and passing it to the onboard PID algorithms that can improve the control system of the robot. Moreover, the Talon SRX has additional support for controller area network (CAN) integration[19]. The Talon SRX has a fast control system and can improve the locomotion of the robot in terms of speed without having to handle the additional complexity of performing PID calculations for each of the motors. Additionally, they allow us to monitor the current for every device which helps to recover the robot from hazardous situations during autonomous operation.

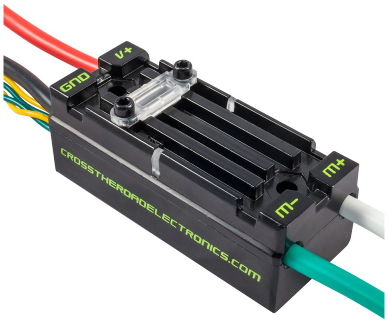


Figure 13: Talon SRX[20].

---

[19] http://www.ctr-electronics.com/downloads/pdf/Talon%20SRX%20Software%20Reference%20Manual.pdf

[20] https://www.vexrobotics.com/media/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/2/1/217-8080.jpg

1.6.3 Arduino

The Arduino is an open-source electronics platform that helps with sensor integration. Arduino is a low-cost solution that has its own IDE which is available on many platforms [21]. For example, AVR-C code can be used to program the Arduino. There are many available shields and other hardware available that are Arduino compatible. Arduino also has serial communication capabilities which can be used to display data and confirm coherent communication protocol. Arduino is essentially a platform for fast prototyping because of the vast available libraries in their software.

Being an open source platform, there are many available resources online with a vast presence of forums and blogs which helps in debugging any common issues that may come up. Arduino is a large part of the maker community and many projects have been created that are available online [22].

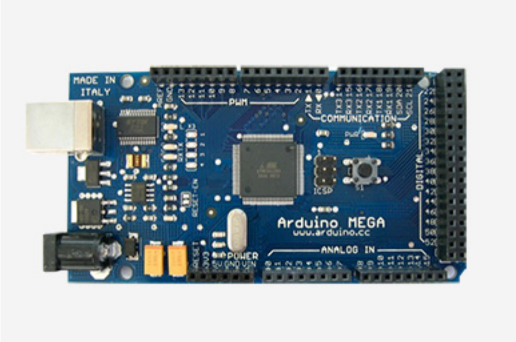


Figure 14: Arduino Mega[23].

---

[21] https://www.vexrobotics.com/217-8080.html

[22] https://www.arduino.cc

[23] https://www.arduino.cc/en/uploads/Main/ArduinoMega.jpg

1.7 Software and GUI Design

Incorporating intelligence and autonomy is only possible if well-written software is developed for the robot. Having a graphical user interface (GUI) helps to understand and ease the debugging process.

1.7.1 Programming Languages

When it comes to programming, most object-oriented languages can be used on the robot, given that the right framework is set up so that different languages can exchange data effectively. Languages that have been used by other teams include C/C++, Python and Java. Due to Java being an interpretative language, it is sometimes not as fast as code that is compiled into machine code. Java has many uses for robotics, and one popular application involving Java is GUIs. Java has many libraries that make it easy to create a robust and editable GUI. One language that is very similar to Java is C#.

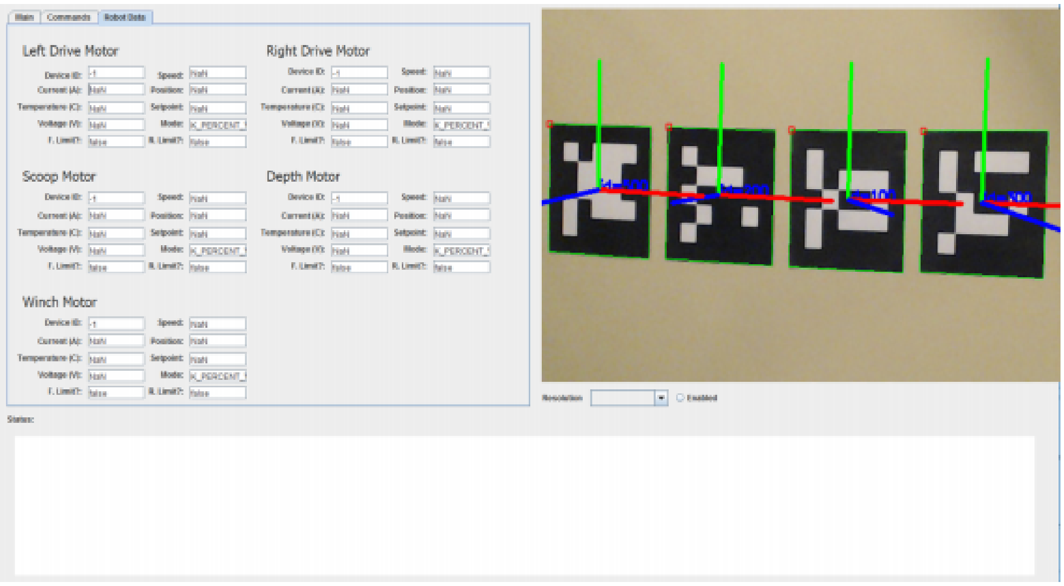
Python is becoming increasingly popular in use for multiple reasons. Like Java, Python is an interpretative language; however, Python's focus is on ease of use. It is a relatively easy programming language to learn and is often the first language someone learns when they start coding. Additionally, Python is included in Robotic Operating System (ROS) which is used in many applications and projects.

C and C++ are two of the most widely used languages for programming robots. Essentially, C++ is an extension of C and both are compiled into machine code. One downside of C and C++ is that they are not as simple to use as Python and similar languages. It can take longer to implement the same functionality using C and will require more lines of code. However, as robotics is very dependent on real-time performance, C and C++ are probably the closest thing to a standard language.

1.7.2 Previous Team's Programming

The three programming languages Markhor used were Python, Java, and C#. Their GUI was built with Java and its Swing library, while the controllers for the GUI were built with C#. All coding of the actual robot was done in Python. Additionally, there is a part of code in C++, which was used with the onboard Arduino Mega to collect data from the sensors.

Shown below is a screenshot of the GUI that was developed by the previous year's team. In the left side of the GUI is the robot's control center. Commands can be given to the robot and at the same time, the team can look at current, temperature and voltage readings of the motors, thus knowing what is happening at every moment. On the right side of the GUI, one can see the robot running a camera tracking algorithm that identifies specific patterns.



Left Drive Motor

Device ID: -1
Speed: NaN
Current (A): NaN
Position: NaN
Temperature (C): NaN
Setpoint: NaN
Voltage (V): NaN
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Right Drive Motor

Device ID: -1
Speed: NaN
Current (A): NaN
Position: NaN
Temperature (C): NaN
Setpoint: NaN
Voltage (V): NaN
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Scoop Motor

Device ID: -1
Speed: NaN
Current (A): NaN
Position: NaN
Temperature (C): NaN
Setpoint: NaN
Voltage (V): NaN
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Depth Motor

Device ID: -1
Speed: NaN
Current (A): NaN
Position: NaN
Temperature (C): NaN
Setpoint: NaN
Voltage (V): NaN
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Winch Motor

Device ID: -1
Speed: NaN
Current (A): NaN
Position: NaN
Temperature (C): NaN
Setpoint: NaN
Voltage (V): NaN
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Figure 15: Markhor's GUI For Robot Operation[24].

---

[24] https://web.wpi.edu/Pubs/E-project/Available/E-project-042617-133846/unrestricted/MARKHOR_Robotic_Mining_Platform.pdf

2 Methodology

For this project, the team planned to continue to develop Markhor. The robot had a robust driveline with an excavation system that had the potential to collect and unload gravel well within the time limit of the round. Under ideal conditions, Markhor was able to collect the gravel under the regolith simulant, however, due to mechanical issues, when under a lot of stress their excavation system tended to jam. The team identified areas of Markhor, that when improved upon, would create a robot that would be very competitive in the NASA RMC. Outlined below are these planned improvements, goals, and identification of different phases in a round for the competition.

2.1 Improvements to Markhor

Markhor laid the framework for a successful robot with a focus on its mechanical design. However, certain aspects of their robot could be improved upon for a better performance in the competition. Markhor's excavation system, as stated above, tended to jam when under heavy loads. The team believed that if this issue is solved, the robot should be able to collect gravel while under load. Another issue that needed to be addressed was the weight of the robot. Markhor was the maximum weight allowed for this competition, and while inherently this isn't a negative, the score of the robot is deducted based on how much it weighs as per the NASA RMC rules[25]. Lastly, an area the team identified as something necessary to improve upon is the autonomy of the robot. While Markhor was designed with autonomy in mind, the robot was completely teleoperated for the competition. According to the NASA RMC rules, autonomy garners extra points toward the robot's rankings, and overall makes the robot more suitable for Martian conditions due to the communication delay between Mars and Earth.

2.1.1 Excavation robustness

Markhor's system had the potential to excavate the regolith simulant as well as gravel. However, since this year's focus is only to collect gravel, the team can improve on Markhor's robustness. An issue that Markhor faced was that the flexible chain kept jamming. This problem occurred because the guide rails, which guides the scoops on the

---

[25] `https://www.nasa.gov/sites/default/files/atoms/files/2018_rulesrubrics_partii.pdf`

excavator, were machined to be too loose causing the guide rail system to buckle. This issue can be fixed by having a closer fit, ranging between 0.005 in or 0.010 in, creating more tension leading to smoother operation of the guide rail system.

Markhor's scoops were manufactured with a focus on regolith simulant collection. Each scoop had a capacity of 0.91 kg of the simulant. The shape of the scoop system needed to be modified to add robustness to collect more gravel. The teeth of the scoop could be made sharper to decrease the torque applied when excavating gravel. Hence, the total power required for gravel excavation can be reduced. Moreover, the chains were prone to get damaged with extensive use due to the accumulation of dust. Using belts is a possible solution to this as it does not require continuous greasing or oiling. Moreover, it will decrease the total weight of the entire robot.

2.1.2 Autonomy

Autonomous operation rewards the team with a maximum of 500 points in the competition[26]. The process of making Markhor autonomous had only just begun and remained unused during the competition because it was incomplete. For this reason, the team aimed to make the existing robot fully autonomous for the competition. In the scope of the competition, the robot should be able to orient itself at the beginning of each run, drive to the excavation zone, collect the gravel, return to the deposit site and unload the gravel. For the self-orientation process, the team decided to use a camera that detects one of three different markers. The markers would be placed to the left, right, and back of the robot and the rear facing camera would pick up a specific marker to determine its initial position. The robot would then rotate accordingly if the left or right markers was picked up, and then center and align itself with the marker behind it. Once correctly oriented the robot will begin moving toward the excavation zone.

Driving to and from the excavation zone requires the use of encoder sensors fused with the existing camera used for orientation. While driving away from the deposit site, the camera will be used to actively center the robot. The encoders, used in combination with a closed loop control system, would help make the adjustments necessary to maintain straight travel. Once the robot has reached the excavation zone, the robot will begin its excavation routine before returning in a similar manner.

---

[26] `https://www.nasa.gov/sites/default/files/atoms/files/2018_rulesrubrics_partii.pdf`

The excavation routine requires some method of determining when the robot's bucket is sufficiently filled. One method is using strain gauges under the bucket which would convert the force of the contents of the bucket into electrical resistance that the robot sense. Once a set threshold is met, the robot will end its routine, return to the deposit site and begin unloading.

Depositing the material is the last event in the sequence of autonomous operations before making another run. In this case, the robot would need to determine when it is aligned correctly with the collection bin before unloading. An idea is that when the robot is within a certain distance using the camera, the encoders will be used to cover the remaining distance.

Even though the robot should be able to complete its tasks autonomously, the team will be able to tele-operate at any point when necessary. This is to allow correct operation and ensure safety.

2.1.3 Weight

Markhor weighed in at 80 kg which is the maximum mass permitted at the competition. Eight points are lost for each kilogram of the robot, meaning Markhor lost a total of 640 points. As a result, the team focused on reducing the current weight of the robot in every possible way.

The heaviest component of Markhor was the drive system. The module had 0.635 cm thick aluminum frames to provide enough strength to support the weight of the robot with an additional load of 100 kg. However, the robot did not need to carry 100 kg. By changing the thickness of the robot from 0.635 cm to 0.3175 cm, the weight of the frame was halved which had a significant impact on the overall weight of the robot.

Another solution to reducing the weight of the frame could be to change the material being used. One possible material to use in place of aluminum could be sheet metal or lexan with reinforced supports. In order to implement this solution, the frame would need to include aluminum supports linearly distributed throughout the frame while the remaining sections of the frame would be either sheet metal or lexan.

2.2 Milestone: System Requirements

While deciding upon the final ideas to be prototyped, special care had to be given to the competition rules and assess the feasibility of the ideas. The team decided on the following specifications to dictate the design of the robot:

Maximum Size	1.5 m x 0.75 m x 0.75 m
Maximum Mass	70 kg
Storage Capacity	50 kg per payload
Gravel Collection	12 kg of gravel per 10 min
Control	Fully Autonomous

2.2.1 Planned Size

The maximum allowed size of the robot at the beginning of the round is 1.5 m in length, 0.75 m in width, and 0.75 m in height. The robot is allowed to extend all its dimensions after beginning the round. While there is no limitation to extending the length and width of the robot, the height is not allowed to extend over 1.5 m except when depositing the collected material where it may extend up to 2.5 m, as per the NASA RMC rules[27].

2.2.2 Planned Mass

To pass the initial check and qualify, the robot and navigational aid system combined cannot weigh more than 80 kg. Additionally, the navigational aid system cannot weigh more than 9 kg. The team loses 8 points for every kilogram of the robot with a maximum of 640 points. By limiting the dimensional and weight requirements, NASA reduces the competitive advantage gained by teams with a higher budget and allows for bigger participation and flow of ideas as more teams are able to participate. As the mass of the robot increases, the power required for travel also increases. As a result, the team has set the maximum mass of the robot to be 70 kg.

---

[27] `https://www.nasa.gov/sites/default/files/atoms/files/2018_rulesrubrics_partii.pdf`

2.2.3 Payload Capacity and Gravel Collection

According to the NASA RMC rules, the robot has to excavate and deposit at least 1 kg of gravel to qualify. Contrary to the previous years, this time teams gain no points when excavating regolith simulant. In previous years, robots had to excavate and deposit at least 10 kg of simulant and/or gravel. NASA has changed the rules signifying its interest in the collection of gravel.The planned storage capacity for the collection bucket is 50 kg and the planned gravel collection capacity in a single run is 12 kg.

2.2.4 Control

It is difficult to tele-operate the robot on Mars due to the communication lag. Hence, incorporating autonomy is a better choice to be more compliant to the real-life scenario. The robot should also be able to perform recovery subroutines in case of any performance issue or abnormalities. For example, if it gets stuck during excavation mode, it should be able to recover by executing pre-programmed actions.

2.3 Milestone: Robot's Plan of Operation

Having a robust plan for operation is imperative for a successful and efficient collection of the ice-simulant. The team first laid out a concrete plan for the autonomous operation and divided it into seven stages, described below:

2.3.1 Finding Robot Orientation

There are two possible zones where the robot can start. Moreover, in each of the possible zones there are 4 possible orientations that the robot can start at. The first important step is to find out the orientation of the robot in its starting state. To achieve this task a target was designed with two rectangles of equal area, one with a vertical orientation and one with a horizontal orientation. A camera will be mounted on a servo which will rotate to find the two contours and return the angle for the orientation. Using OpenCV and GRIP engine the robots camera will able to determine the two rectangular contours and compare the two areas. It will then perform an area check to determine the starting zone and its pose (orientation). Thus, with both the information of the starting zone and orientation the robot can locate itself in the field.

2.3.2 Going to Border of Starting Zone and Obstacle Area

Once the robots complete orientation is determined, the robot will turn towards the center of the border between the Starting zone and Obstacle Area. After turning towards that point, the robot will drive and re-orient the robot perpendicular to the target or Collection Bin. After achieving this step, it will be ready to drive towards the Excavation zone through the Obstacle Area.

2.3.3 Driving to the Excavation Zone

After the robot is oriented correctly at the border of the obstacle zone and starting zone, the robot will drive straight into the mining area by using encoders. Any obstacles in the path of the robot will be pushed away by the front cow-catchers of the robot. The robot is only allowed to start mining when it reaches the Excavation Zone.

2.3.4 Mining Regolith and Gravel

This step mainly involves the robot digging and collecting as much gravel it can up to its maximum capacity. Once the robot reaches the Excavation Zone, the robot will run the scoops along with the dynamic chain and keep on digging until the scoops are below 40 cm. Once the desired depth is reached, the robot will drive backward slowly towards the target while collecting the gravel.

2.3.5 Drive to Collection Bin

As the robot collects and drives backward to the target zone, the robot performs three checks simultaneously. If the buckets capacity is reached or if the robot has reached the obstacle area or if there is not enough time to complete the run, the scoops will stop running and the dynamic chains will retract back to its original position. The robot will then drive straight towards the target and proceed to the next step to dump the contents of the bucket.

2.3.6 Align to the Collection Bin

Once the robot is driven back to the collection bin, the robot will try to align itself correctly to the bin. To facilitate the alignment process, the robot will use two bump sensors on either side of front of the robot. Both the bump sensors will have to be engaged to the target to make sure the robot is perfectly aligned and ready to unload the bucket contents into the collection bin.

2.3.7 Deposit Collected Material

Once both the bump sensors are engaged, the robot will start running the conveyor belt to unload the contents from the bucket to the collection bin. The conveyor belt will be running at an adequate speed for complete disposal of the contents. After the contents have been disposed, the robot will stop running the conveyor system and conclude the run.

2.4 Prototyping

As with the development of every engineering system, the prototyping phase was extremely important to validate ideas before investing time and money into systems that might not even work. The team used various techniques to make prototypes including laser cutting, 3D printing, milling and lathing along with cheap and easy to get materials like plywood, sheet metal and aluminum. Also, multiple aspects of last year's robot were used in the prototyping process, such as the excavation and drive systems.

2.5 Modeling and Design Phase I

During the prototyping phase of the project, the team started creating CAD models of each subsystem in SolidWorks. Much of the design of the drive system and dynamic chain system of the excavator were adapted from last year's robot, with modifications to reduce weight. This model was primarily made for the Preliminary Design Review (PDR) to be used as a way to convey how the team planned to address the main processes the robot needed to accomplish. Main components of this design included the drive-line adopted from Markhor, a conveyor belt unloading system, and a depth mechanism utilizing four lead screws, eight bevel gears, and two gas springs to tilt the excavator to the desired angle. This design was expected to undergo changes based on feedback received from the PDR and any discoveries found during the manufacturing process.

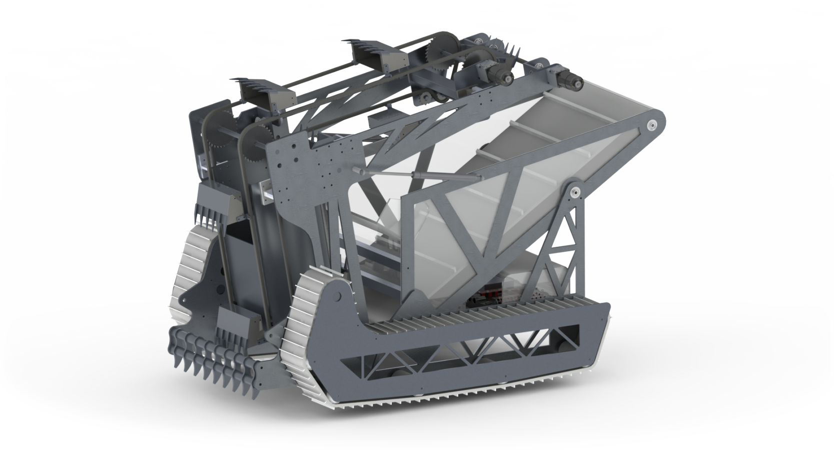


Figure 16: Early Render of Ibex Rover

2.6 Milestone: Preliminary Design Review

The PDR took place on December 7, 2017. The project advisors, Robotics Engineering Program faculty and fellow WPI students attended the hour long session. Ibex's design was discussed and validated for optimal performance during the competition. The PDR helped to better identify possible issues the team missed to address. Overall, it allowed for constructive comments of the robot's design that were addressed afterwards.

2.7 Budgeting

An estimate of $5000 was determined before the project started. At the time of writing, the team has spent $1423.57 on purchasing raw materials and estimates to spend another $3450 on robot shipping, additional manufacturing, traveling, and housing.

Budget	Cost
Manufacturing Spent	$1433
Future Manufacturing	$50
Future Robot Logistics	$500
Traveling	$2900
Total Project Cost	$4883

The team performed budget reviews bi-monthly in order to ensure there were funds left for estimated future costs. Finally, the team's net project cost ($4833) was below the estimated budget allocated at the beginning of the year.The other expected sources to fund the MQP were from various Engineering Departments at WPI increasing the budget by another $2000.

The team was also sponsored by Worcester Sand and Gravel, who provided the team with sand and gravel for testing, and Hydrocutter Inc., which graciously offered their water-jet services. The team actively looked for other potential sponsors that could fund the remaining manufacturing and travel costs.

2.8 Modeling and Design Phase II

The above figure is the final rendering of the Ibex rover. At first glance, there might not be much difference in terms of looks, but significant changes were made during the entire manufacturing process and after the PDR. The most significant change to the design was scrapping the lead screw system to facilitate the depth extension of the digger. This system was replaced by a tape drive system, similar to what can be found in car window mechanisms. A side result of changing this system was that there no longer needed to be gas springs to tilt the digger, which helped to reduce the weight. Other changes included small weight reducers throughout the initial design and adding more holes to mount sensors.

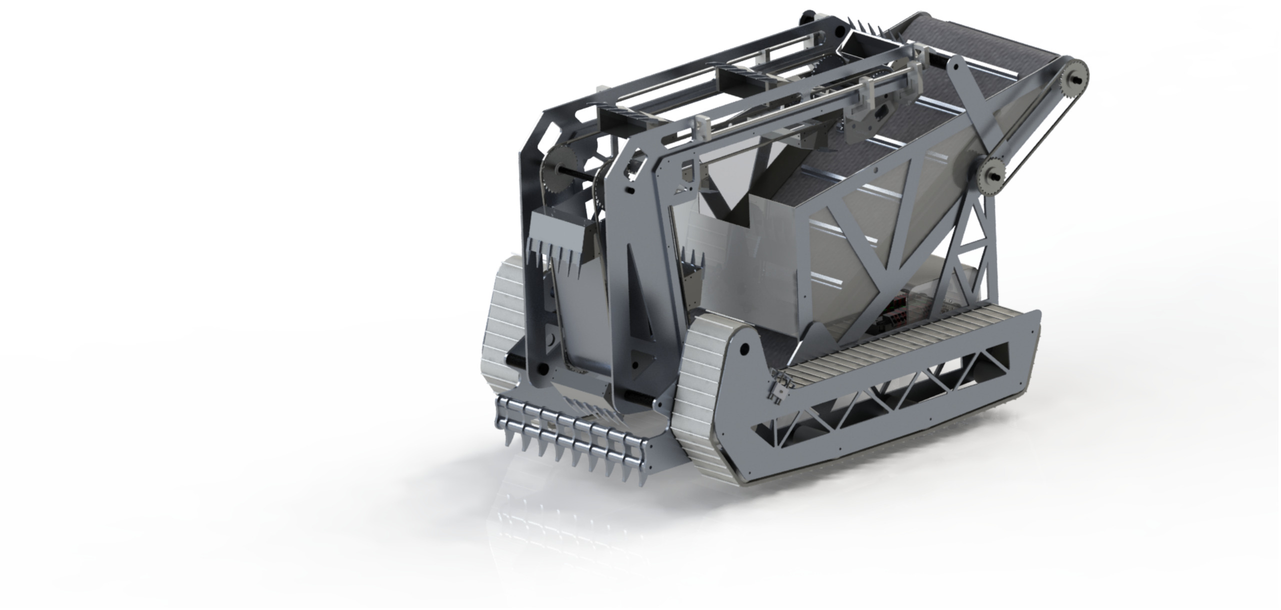


Figure 17: Rendering of Final Ibex Design

2.9 Manufacturing

As the design of the robot was being finalized, manufacturing began. This included the drive base, which was very generously water-jet for free by Hydrocutter Inc. As more subsystems were assembled, changes to machined parts were necessary. For this we used the machines in Washburn Shop such as the milling machines and lathes. All mechanical components aside from parts that were water-jet were machined and assembled by the team.

2.10 Testing

During the manufacturing phase, extensive testing on each subsystem had to be undergone to ensure the creation of proper parts that fit the design specifications set above. When the drive base was assembled, there was opportunity to test autonomous functions regarding localization. To do this a prototype of the competition arena was laid out. The testing area was made with using the real dimensions of the actual field. The testing area of the starting zone is as shown below in Figure 18.

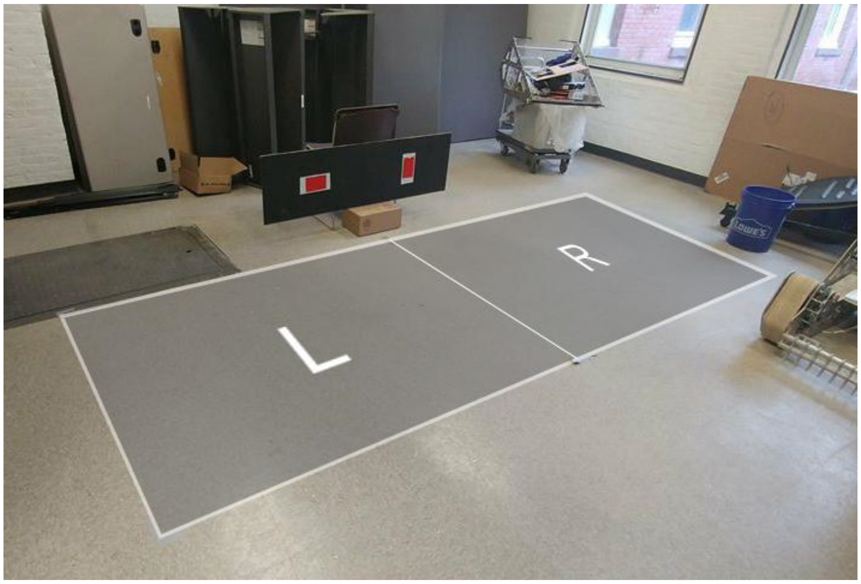


Figure 18: Ibex Testing Field

2.11 Milestone: Critical Design Review

The Critical Design Review was on April 20th, 2018. There were two phases of the presentation, a poster presentation, where people freely walked around and asked for explanations on various subsystems, and a presentation in front of faculty of the Robotics Engineering department and anyone who wished to view the presentation. These presentations provided an opportunity for the team to share the whole process of the MQP and receive insight into what could have been done better throughout the project.

3 Mechanical System

3.1 Locomotion

The drive system implements a tread based design consisting of two BRECOflex ATK10K13 timing belts. The tread has a herringbone pattern for sufficient traction to drive without stalling the motors while turning and evenly distributes the weight of the robot across the treads in place of several points by using wheels.

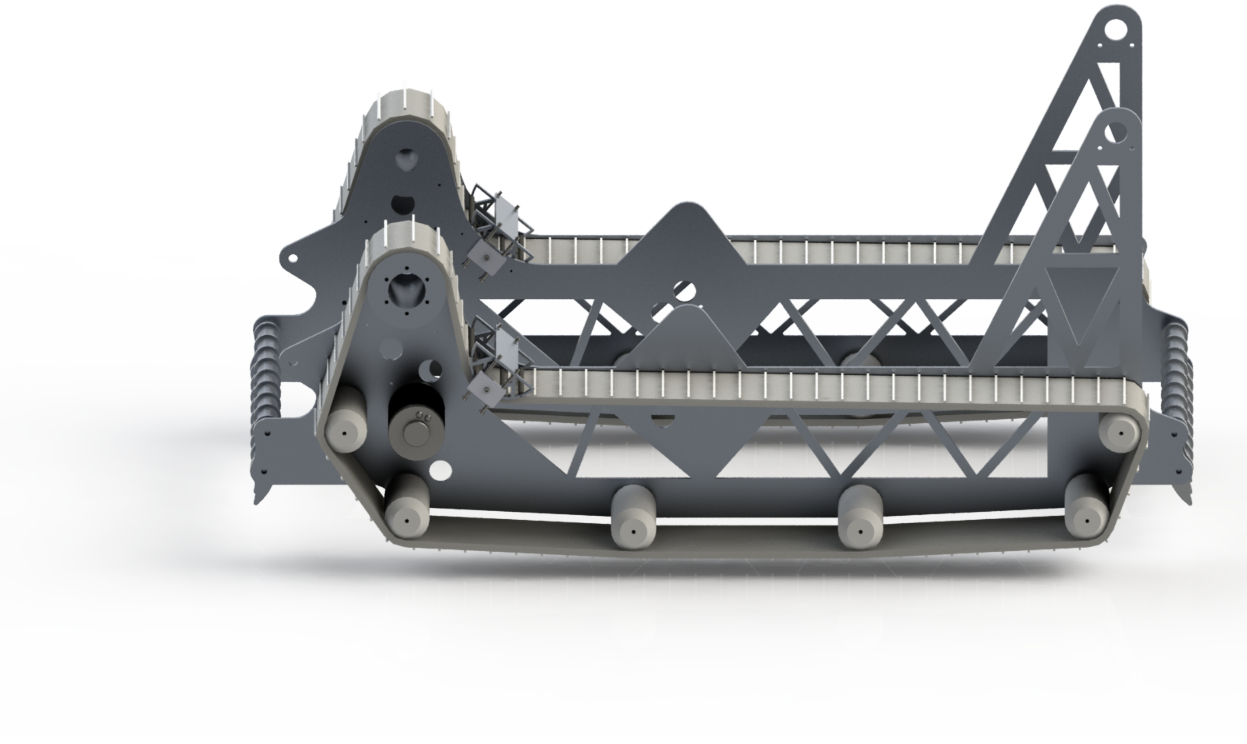


Figure 19: Ibex Drive System Side Rendering

This drive system was adopted from last year's robot. The driven pulley is elevated off the ground to minimize the amount of debris that could get stuck between the pulley and the belt. There is a tensioner module on either side of the driveline. The modules use springs to keep the tensioner tight against the belt, and allow rocks to pass through the system if one were to get stuck. The tension modules can be seen in the image below.

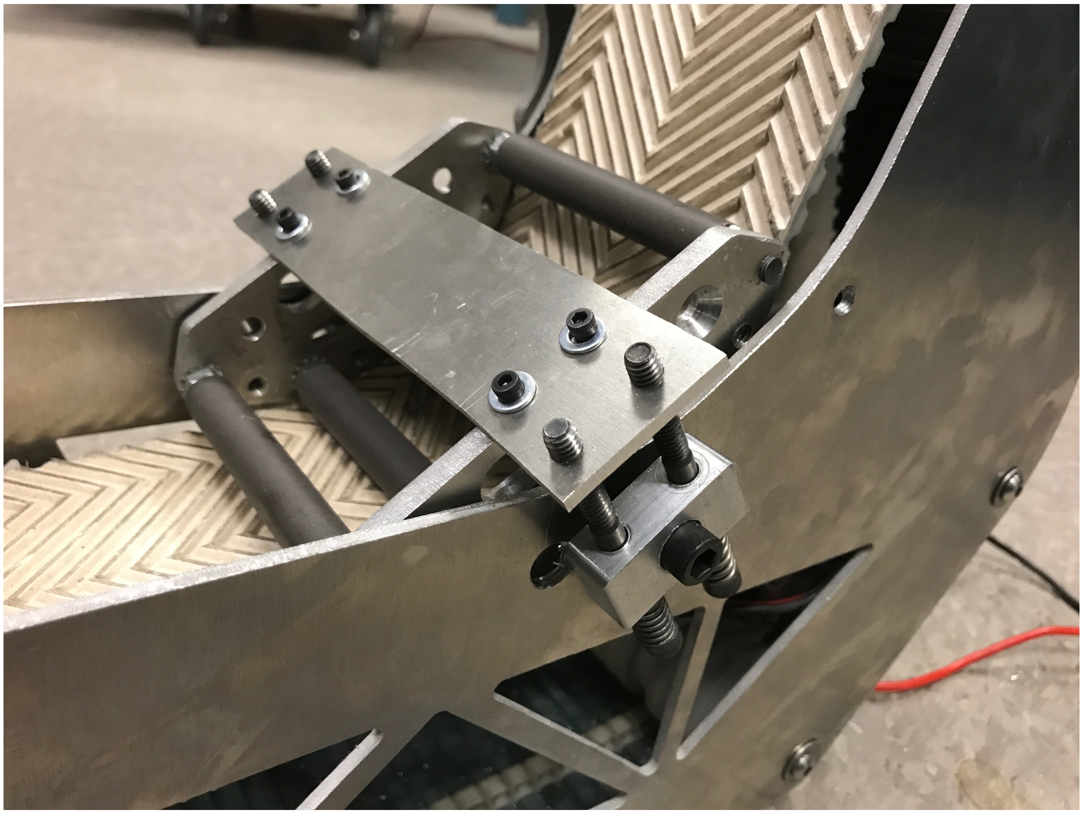


Figure 20: Driveline Tension Modules

3.2 Excavation

Due to the change in competition rules, excavation became one of the most important systems to improve. The team decided to implement the bucket ladder design used by Markhor and make improvements on it.

3.2.1 Simultaneous Carriage Motion

The excavation system consists of two carriages: the bottom carriage travels vertically into the ground while the top carriage travels horizontally over the bucket. These carriages have a 1-to-1 travel relationship that is controlled using car window regulator tapes. This system simultaneously pushes one carriage and pulls the second carriage driven by a globe motor with a gear ratio of 1:5. This is the same way last year's team controlled the depth of their excavation system, however their system extended only 35 cm. This was extended by 10 cm allowing the excavation to reach the 45 cm the project team set out as system requirement.

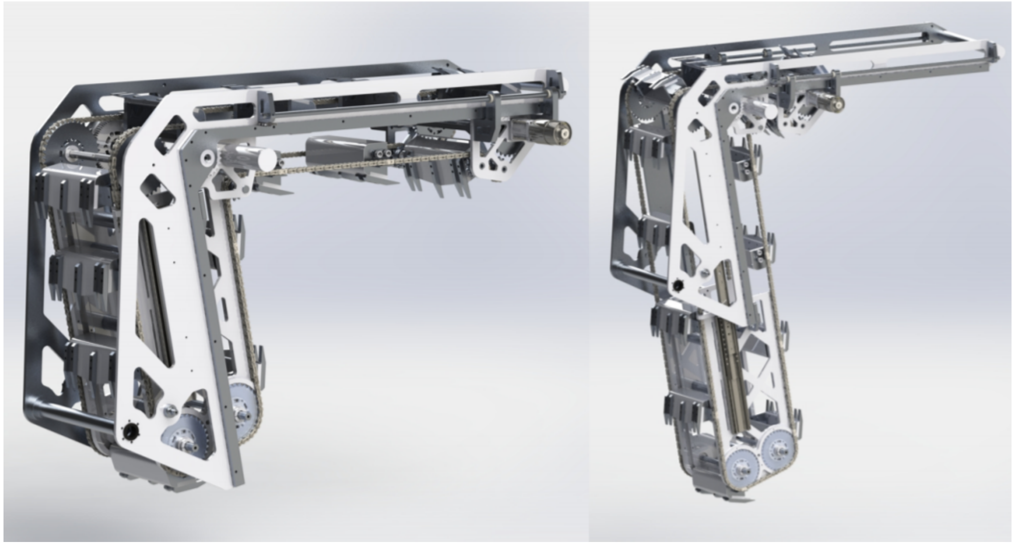


Figure 21: Excavator in Retracted and Extended Positions

3.2.2 Scoop Design and Connection

In order to excavate material from the ground, scoops are connected to two #35 chains on both sides. The way in which the scoops were attached to the chain had to be redesigned to be able to excavate the icy regolith buried 30 cm under the BP-1. By attaching each scoop to the chain at two points on each side the force applied to the chain during excavation would be reduced. One issue arising from this configuration is that the scoops are unable to move smoothly through different angles of the excavator. To resolve this problem, the connections between scoops and the chain were non-rigidly attached.

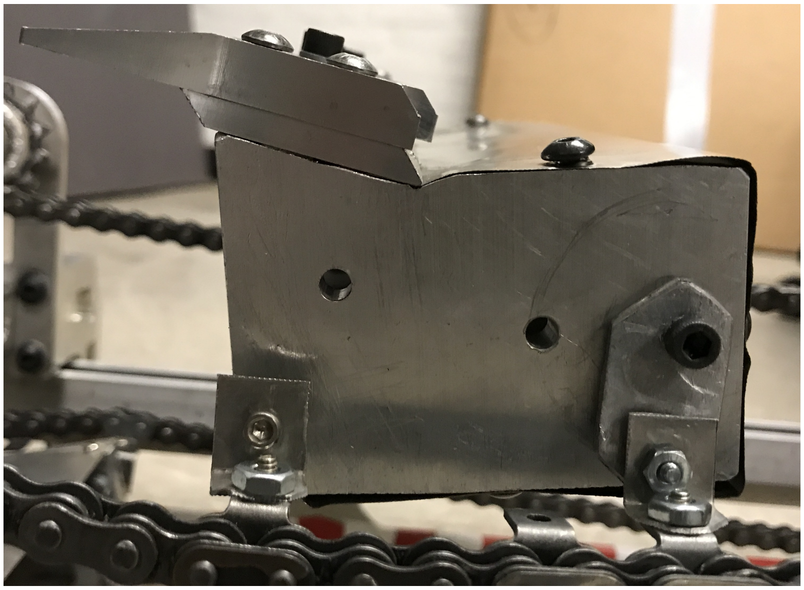


Figure 22: Connection Between Prototype Scoop and Chain

A backplate was added to prevent the scoops from twisting and causing the chain to buckle whenever a scoop hit the ground. The design of the scoop was also changed to efficiently utilize the backplate system. Therefore, the back of the scoop has a flat surface. This enables the scoop to always be in contact with the backplate when it is digging into the ground. Additionally, the scoop has a 30 degree outward draft from back to front enabling the scoop to unload all its contents with ease. The scoop design is shown below:

Figure 23: Final Scoop Design

3.3 Material Deposit

The unloading system of the robot underwent significant changes from last year. It aimed towards implementing a conveyor belt as the base of the bucket to unload the excavated regolith into the collection bin. The conveyor belt was a repurposed belt from an old treadmill. Since the angle of repose for regolith is 45 degrees, the conveyor belt is at an angle of 32 degrees with respect to the horizontal. To ensure that the conveyor belt is able to unload the collected material, grousers were attached to the belt. The average diameter for gravel is 2 cm and so, the height for the grousers was set to 1 cm. To keep the belt taut, a tensioner was designed. The system consists of the upper conveyor pulley situated in a slot and screws able to precisely adjust the pulley's position in the slot. A rubber door strip was mounted where the back wall of the bucket met with the belt. The purpose of this was to have something strong enough to not let the gravel through, but flexible enough to allow for the grousers to pass. A globe motor with a gear ratio of 117:1 is used to drive the conveyor belt and unload all material into the collection bin.

Figure 24: Ibex Unloading system

This system was much simpler than the unloading system used by the team last year, which included a winch to pivot the bucket around the back of the robot and simultaneously decoupled the excavation system on a four bar linkage. The unloading system was a major point where weight could be reduced, as it would eliminate a CIM motor, a gearbox, and gas springs.

4 Electronics Control System

The previous electrical system utilized a fanless microbox PC running Linux, a CTRE HERO motor controller, and an Arduino Uno to interface with sensors and control servos. Power from an AndyMark 12V 18Ah battery was distributed to all of the components using a CTRE Power Distribution Panel. Although a Raspberry Pi was considered earlier, the team chose to use the fanless microbox PC for extra computational speed and storage space. To control the motors, the HERO motor controller interfaced with Talon SRX's to make use of their internal PID control. Dust proofing the ECE system was crucial. Due to time constraints, the previous electrical system was not dust-proofed and the only protection was tape covering exposed outlets. Due to the onboard PC having no ventilation, there was a need to determine the best ways to dissipate the considerable amount of heat generated during operation. Sensors such as limit switches, IR sensors, an IMU, two cameras, and encoders would also need to be incorporated. The preliminary electrical system design can be seen below:

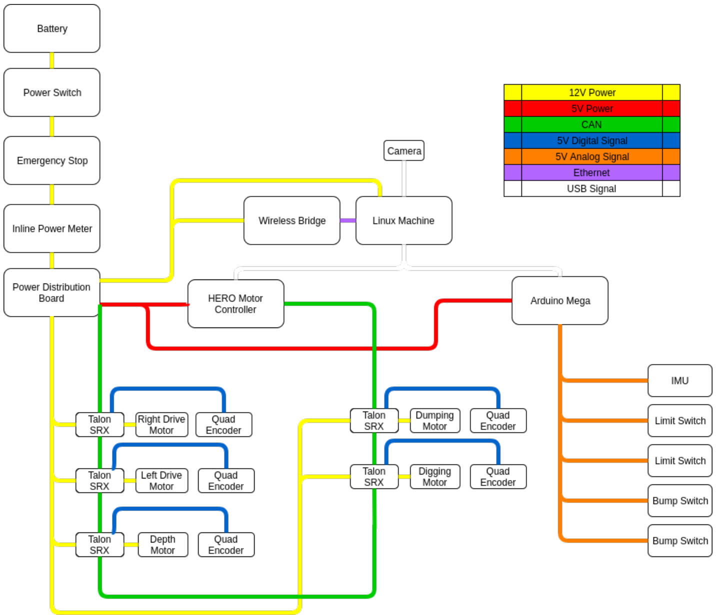


Figure 25: Electrical Design Diagram

Addressing the issues that persisted with last year's electrical system was of high importance. Ways to dissipate heat and mount the electronics in an optimal way were researched and discussed. Since the microbox PC is the only component that got significantly warmer after the 10 minute runs, it was determined that this would be placed above most other electronics on an aluminum mount. This ensures that the heat will dissipate upwards without affecting any other electronics. Another change in electronics was more effective dust-shielding. It is essential that the electronics are shielded from the electrostatic BP-1. A picture of how the electronics are mounted, albeit before any significant shielding, can be seen below.

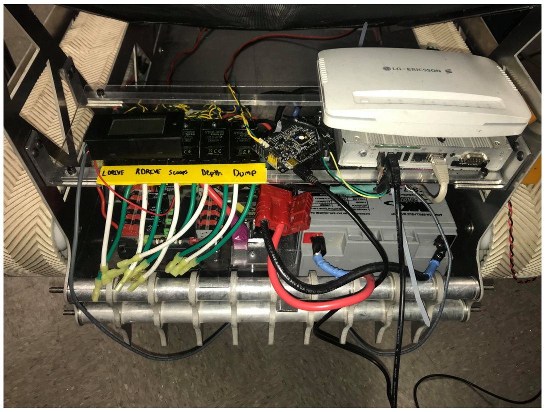


Figure 26: Electronics in Ibex

5 Software Control System

The team plans to operate the robot autonomously and will be controlling the robot remotely. A python based Flask server was developed that was be able to host the live feed from the robot's cameras and read the real-time robot motor and sensor data. The live feed was good in quality but there was a delay in the video feed and robot system data feed as well. Thus, the team decided to discard the idea of hosting the live feed via Flask mainly due to the bandwidth usage penalty rule, and instead build on MARKHOR's Java based GUI leveraging the Swing framework. The robot data was more instant in this case with a more optimal implementation of the network communication architecture.

The mode of communication between the main four software components are described as below-

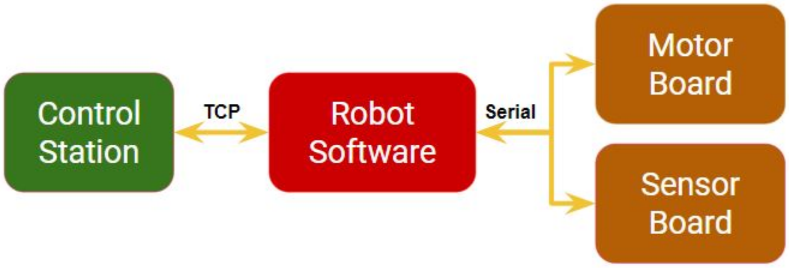


Figure 27: Software Communication for Ibex

5.1 Robot Controller

The robot software contains the main state machine that processes the messages received from the control station and executes relevant functions. The main function parses the incoming messages and determines the message type sent via the control station. Once the robot determines the type of the message, the robot executes the corresponding state in the state machine. The states are responsible for telling the robot to send the correct messages to the motor and sensor boards to perform the correct actions. Upon task completion, the robot updates its own program, lets the control station know that the message was performed and waits for the next message to be sent. A serial handler, sensor handler and motor handler were programmed to maintain and update the device status internally.

5.2 Control Station

The control station can be divided into front-end and back-end. The front-end provides a visual display of the robot's status and messages. To develop the Graphical User Interface (GUI) for the control station, Java's Swing framework was utilized and the screen was divided into three main sections. First, the message status display panel prints the current message queue and recovery stack so that the user can quickly understand what task the robot is performing. The next panel provides the ability to add new messages to the main queue. The final section of the GUI, robot data, displays real-time data of each motor and sensor on the robot. It also visually warns the operator by highlighting values when any abnormal data is detected to help make faster decisions.

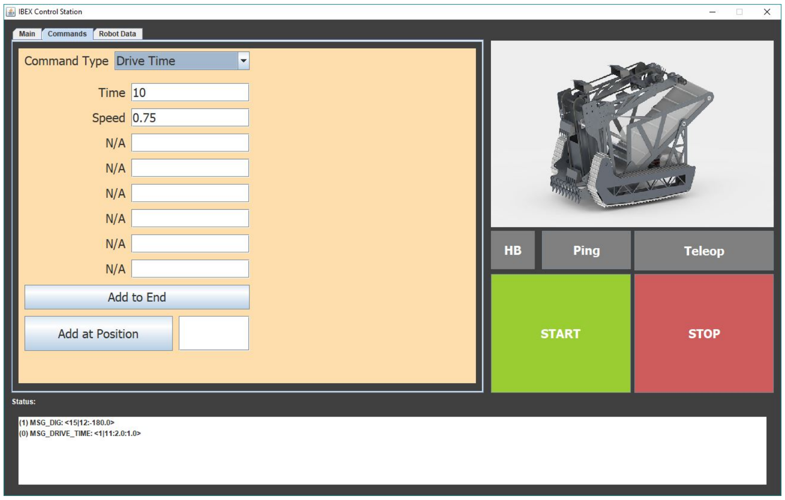
The 3 main sections for the GUI sections are as follows-

```
(0) MSG_RETRACT_DIGGER: <16|13>
(1) MSG_GO_TO_DUMP: <17|14>
(2) MSG_DUMP: <18|15>
```

Status:

```
(1) MSG_DIG: <15|12:-180.0>
(0) MSG_DRIVE_TIME: <1|11:2.0:1.0>
```

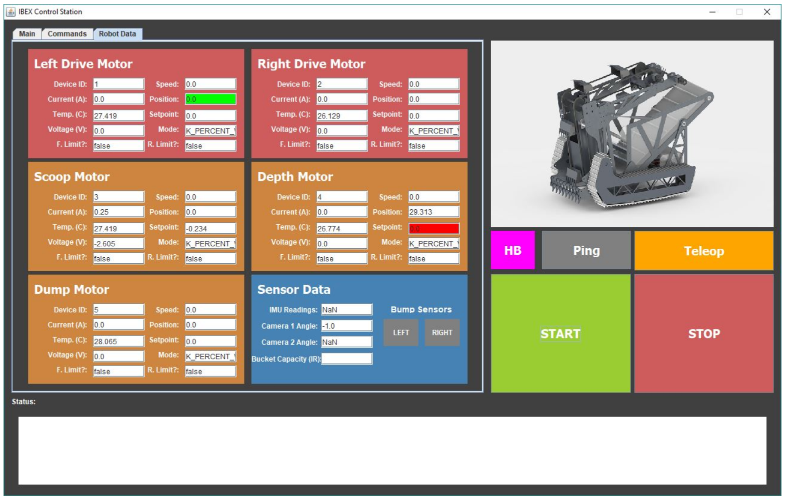
Figure 28: GUI Screen Shot with Autonomy Queue and Recovery Stack



Command Type: Drive Time
Time: 10
Speed: 0.75
N/A:
N/A:
N/A:
N/A:
N/A:
N/A:

Status:
(1) MSG_DIG: <15|12:-180.0>
(0) MSG_DRIVE_TIME: <1|11:2.0:1.0>

Figure 29: GUI Screen Shot for Commands Panel



Left Drive Motor
Device ID: 1
Speed: 0.0
Current (A): 0.0
Position: 0.0
Temp. (C): 27.419
Setpoint: 0.0
Voltage (V): 0.0
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Right Drive Motor
Device ID: 2
Speed: 0.0
Current (A): 0.0
Position: 0.0
Temp. (C): 26.129
Setpoint: 0.0
Voltage (V): 0.0
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Scoop Motor
Device ID: 3
Speed: 0.0
Current (A): 0.25
Position: 0.0
Temp. (C): 27.419
Setpoint: -0.234
Voltage (V): -2.605
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Depth Motor
Device ID: 4
Speed: 0.0
Current (A): 0.0
Position: 29.313
Temp. (C): 26.774
Setpoint: 0.0
Voltage (V): 0.0
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Dump Motor
Device ID: 5
Speed: 0.0
Current (A): 0.0
Position: 0.0
Temp. (C): 28.065
Setpoint: 0.0
Voltage (V): 0.0
Mode: K_PERCENT_
F. Limit?: false
R. Limit?: false

Sensor Data
IMU Readings: NaN
Camera 1 Angle: -1.0
Camera 2 Angle: NaN
Bucket Capacity (IR):

Figure 30: GUI Screen Shot with Real-Time Robot Data

The main back-end components include the Message Queue, Message Handlers, and Recovery Stack classes. The Message Queue provides a way of sending a sequence of tasks to the robot. Once the tasks are performed on the robot's side, the robot sends a "Finished" message which signals the control station to send the next item in the queue. In addition, the message that has been executed is stored in the Recovery Stack. In case of any abnormalities, the control station would send the corresponding recovery functions for every message contained in the stack until the robot is normalized. Next, there are Message Handlers, which properly format the message to be sent to the robot. They are also responsible for parsing the messages and updating the elements used in the GUI.

5.3 Controller Board Communications

Both controller boards, the HERO and the Arduino, communicate to the onboard PC over serial. There are two threads that facilitate constant communication from the PC to the microcontrollers. These threads continuously poll the serial port that each microcontroller occupies and sends select information to the board.

5.3.1 HERO Motor Controller Communication

The HERO board is manufactured by Cross The Road(CTR) Electronics and is programmed by leveraging the .NETMF framework that uses the C# API. The role of the HERO Motor Board is to communicate with the onboard Talon SRXs which are also manufactured by CTR Electronics. The entire program written in C# is able to get the real-time data for every motor connected to each Talon. The HERO board communicates via serial connection with the Robot controller at 115200 bauds per second.

The team set up a proper messaging format that is used to send over motor messages to the HERO board.

<DEVICE_ID:MODE:SETPOINT>

Multiple such messages (as shown above) can be sent to the HERO board for all the possible motors in our robot. Each set of messages is followed by a newline character to denote the end of the the messages. These messages are matched with the corresponding motors (via the DEVICE_ID) on the robot. Once the match occurs, the corresponding device is updated with its setpoint and mode. In case a certain Device ID does not match with any of the on-board device the message gets discarded.

Once the message is parsed and updated to the corresponding device(s), the HERO Board reads the device status messages from the Talon SRXs and sends back to the robot in a format as shown below-

<DEVICE_ID:CURRENT:TEMPERATURE:VOLTAGE:SPEED:
POSITION:SETPOINT:MODE:FWD_LIMIT:REV_LIMIT>

This status message is further sent via the robot software to the Control Station (GUI) which displays the real-time data of the on-board devices. This helps to monitor data and detect any abnormalities in the devices that occur during autonomy.

A major challenge faced by the team was to handle the System exceptions thrown by the HERO board program as they would stop our entire communication to the Talon SRXs. Hence, to enable an uninterrupted communication the team continuously monitored and updated the HERO Board program so that all possible exceptions are properly handled.

5.3.2 Arduino Sensor Controller Communication

The Arduino is used to relay sensor data and control the camera servos. The Arduino is programmed to parse the incoming message, update the servos accordingly, and output the current value of the servos and sensors. The current values are formatted and added to an outbound message. The outbound message format is as shown below-

<SENSOR_1:VALUE_1><SENSOR_2:VALUE_2> ... <SENSOR_n:VALUE_n>

The sensors present on the Arduino are the IMU, two limit switches and two bump switches. All these sensors and its values are constructed into the above message format and sent over to the robot. These messages are further sent via the robot software to the Control Station which displays the real-time data of the on-board sensors.

6 Performance Evaluation

6.1 Mechanical System Testing

6.1.1 Testing the Scoop Designs

The team developed several iterations until the scoop design was finalized. This was critical for effective and efficient collection of the ice-simulant due to the rule change. Some of our designs we developed and tested are as follows:

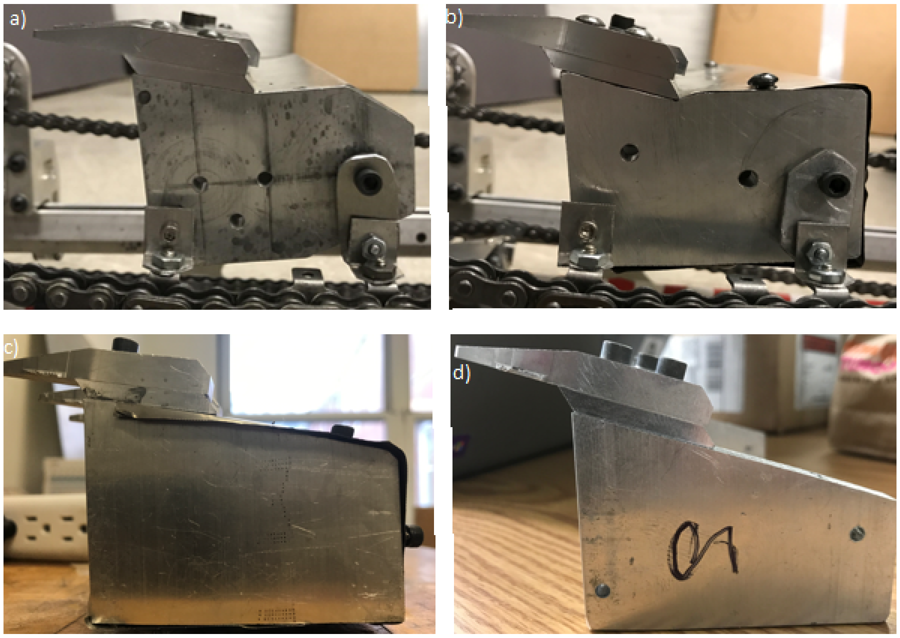


Figure 31: Scoop Design Ideas a) First Iteration, b) Second Iteration c) Third Iteration d) Final Iteration

The final scoop design on the bottom right worked best at directing the forces along the backplate instead of on the chain. Mounting holes for the tabs on the scoop at different heights increases the range of movement of the scoop on the chain, allowing it to make contact with the backplate. Thus, this is the design that the project team went with for the final robot.

6.1.2 Testing the Excavation System

Multiple tests for different iterations of the excavation system occurred throughout the year as new components were developed. In November and December of 2017, the team started prototyping the excavation system using laser-cut wood. The purpose of these tests were to find the optimal mounting arrangement of the scoops. Pictures of this prototype can be seen below. The prototype was driven by a hand drill to easily control the input torque on the driven sprocket. The prototype was able to dig through a cement mix, but the system as a whole was not extremely sturdy.

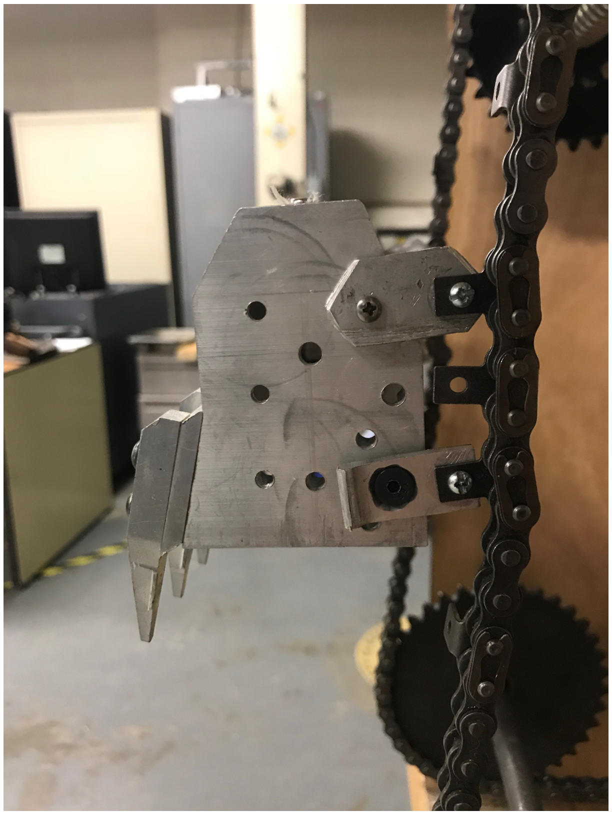


Figure 32: Prototype Excavation System

The team began testing a backplate on the previous years excavation system and quickly machined aluminum tabs to mount the scoops to the chain in early January. Seen below, this system was driven by a single VEX 775 pro motor.

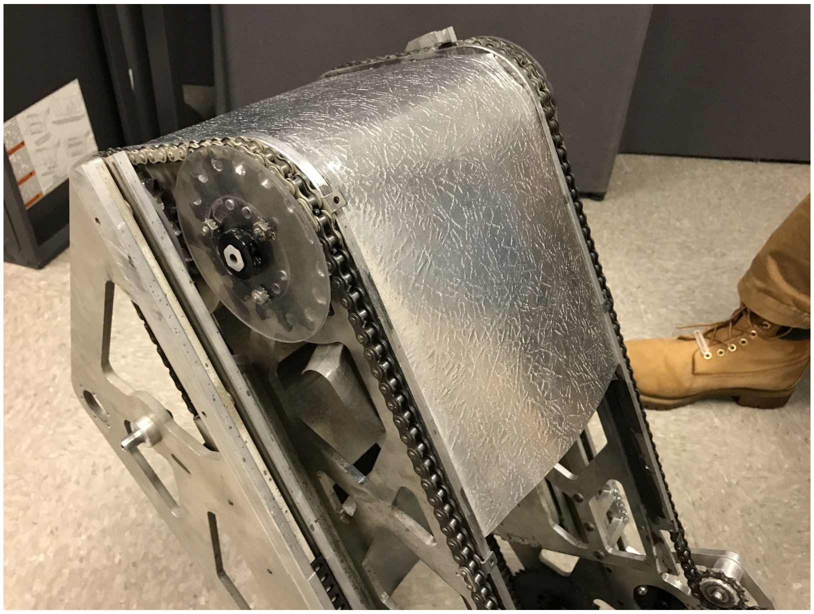


Figure 33: Prototype Backplate

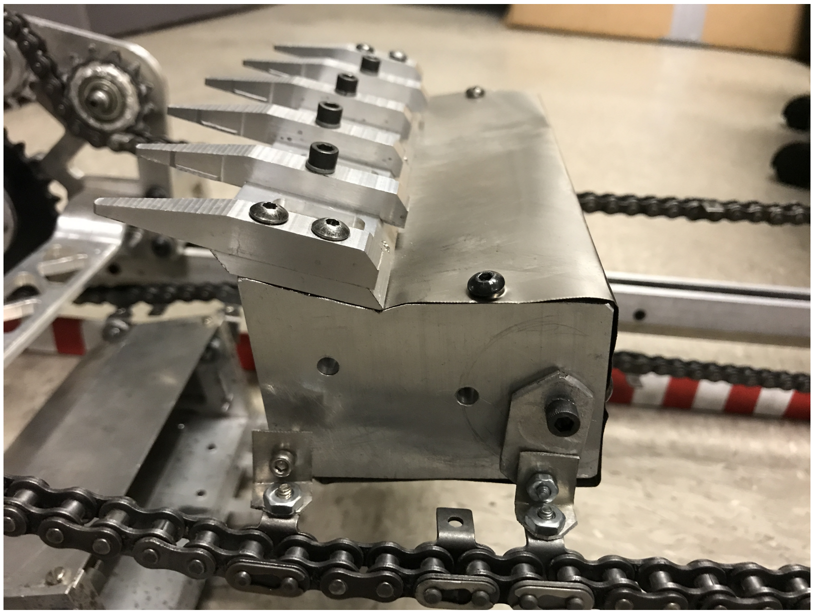


Figure 34: Aluminum Tab Connections

Although this worked well and was able to dig a significant depth in a sand and gravel mixture, the aluminum backplate were extremely beat up. This convinced us to go with a steel backplate to be able withstand the force applied. Also, the tab connections were decided to be made from steel so that the size of the tabs was minimized but provided enough strength.

The final iteration of testing occurred in April 2018. This phase of testing included steel tab connections and a steel sheet backplate. The excavation system was mounted on the drive-line at this point. The scoops were driven by a VEX 775pro motor and the tape drive system was driven by a globe motor with an internal gear ratio of 117:1 and an external gear ratio of 5:1. The robot was able to dig well past the 30 cm mark to reach the gravel.

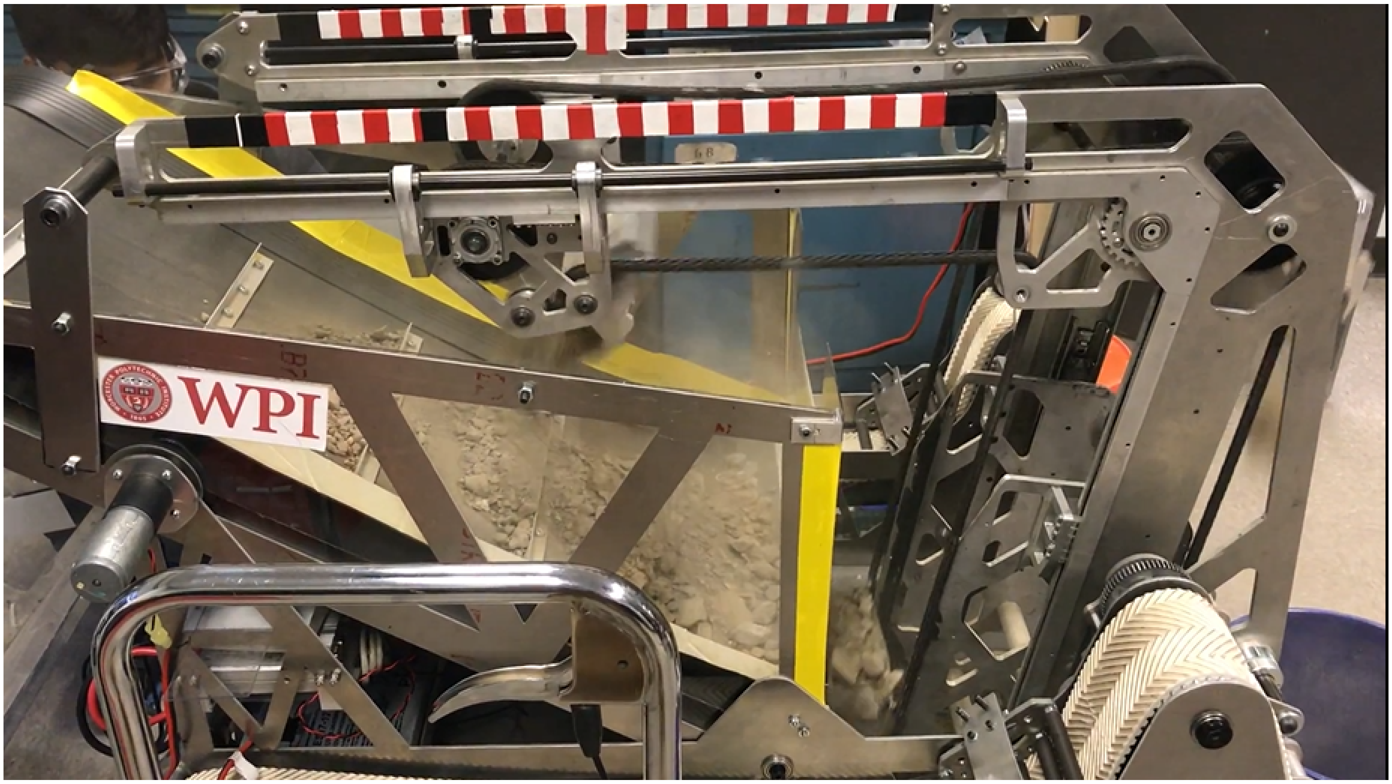


Figure 35: Testing the Digging Mechanism

6.1.3 Testing the Conveyor Belt

The material depositing system began testing in mid-February. The team received a broken treadmill and re-purposed the belt as the conveyor for the robot. To see how the belt behaved with gravel and sand on top of it, the belt was placed at the same angle as the CAD design belt. This initial phase of testing can be seen below.

Figure 36: Initial Material Deposit Test

The material began slipping as the belt moved, which confirmed a suspicion that the belt would need grousers to carry up the collected material. Thus, rudimentary grousers were attached to the belt, seen below. This proved far more effective at depositing collected material, however another issue arose. The material began falling off the side of the belt as it was carried up. To solve this and other foreseen issues, the team began its third phase of testing.

Figure 37: Initial Material Deposit Test with Grousers

Final testing of the material deposit system began in April. This included testing whether or not the bucket could hold the gravel inside with the rubber stops and seeing if there was any dust being let through the sides. Seen below, the material deposit system was successful at containing the gravel inside the bucket and carrying the material high enough to be put in the collection bin. As a result of this testing, the team decided to add more dust protection underneath the sides of the belt as an extra safety precaution.

Figure 38: Final Material Depositing System Testing

6.1.4 Testing the Vision Processing

Vision sensing was tested on a small robot built with VEX components using the target in previous sections. The team devised multiple states depending on the robot's orientation. The two on-board cameras rotated until one of them recognized the two contours. The camera would then enter one of eight predefined states to calculate the angle towards the center of the border of the simulated excavation zone and obstacle area. Once this angle is calculated the robot will go towards the center point of the obstacle area. The prototype worked very well and gave the team a chance to identify possible areas for improvement in the vision system and the target design. A feedback to the user was added to tell the side it detected and the perceived areas of the contours. Also, when the camera sees the two contours, the camera sends the perceived image to the Control Station over a socket along with the contour's respective areas. A few improvements were added to the camera vision after a rigorous series of testing which included the following:

Masking the Image

To remove the redundant noise from the image captured by the camera the team decided to mask the current perceived image with another image that has the top and bottom of the image filled with black in order to remove contour detection from undesired location (i.e., not on the target).

Filtering the Image by Solidity

To remove unwanted detection of contours the team decided to study the OpenCV contour detection process. The team decided to use the solidity attribute which is the ratio of the contour area to its convex hull area. That ratio is set to a very high value (above 0.8) which helps filter the unwanted contours in the perceived image by the camera. Since the contour is almost a perfect rectangle the contour area is very close to the area of the contours convex hull. This feature significantly improved the contour area detection.

Improving the Target

Initially, a white reflective tape was used as our contour. Because the RGB (Red, Green, Blue) characteristics of the white reflective tape would not be so different from the field's regolith color, the team decided to make it red for better detection by providing more contrast from the background objects in the field.

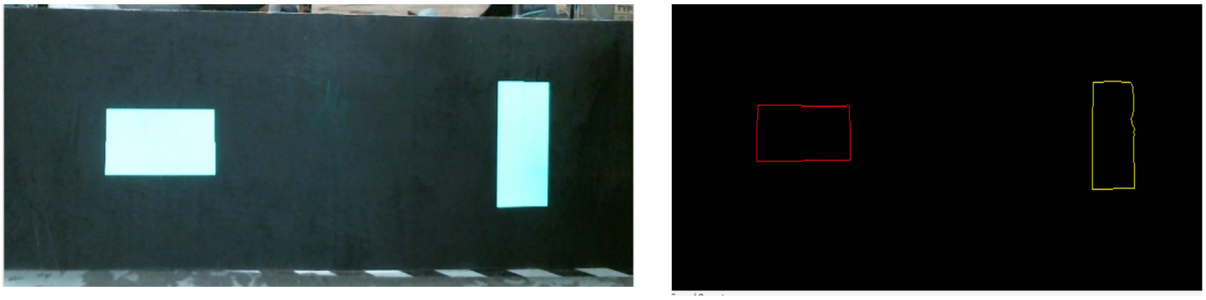


Figure 39: Target Design and OpenCV Output

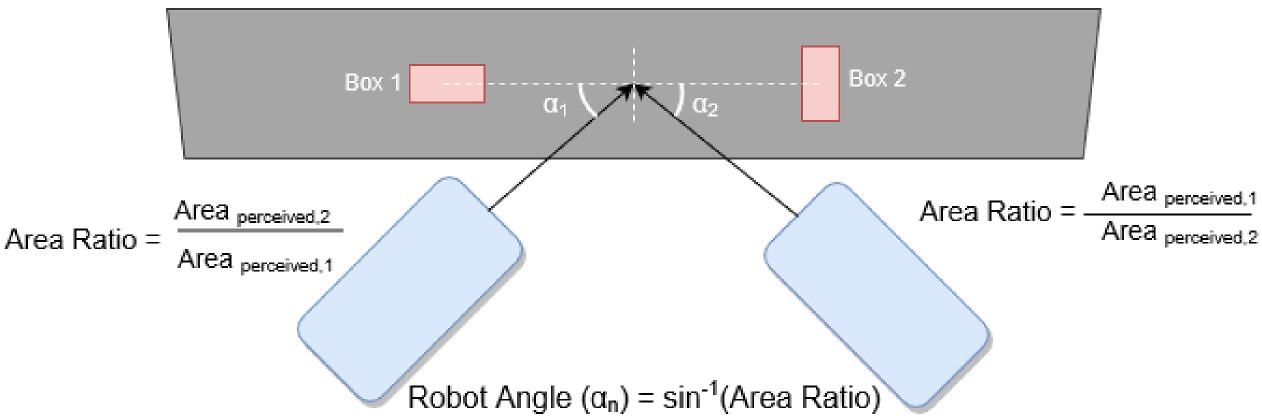


Figure 40: Final Formula Used for Perceived Area Calculations

7 Future Work

Having reached the end of the MQP and being only 3 weeks away from the 2018 NASA RMC there are some changes the project team would like to implement to the robot, but due to time and budget restraints cannot be feasibly implemented at this time.

7.1 Mechanical Design

The main focus on mechanical design should be to maximize the effectiveness of digging deeper than 30 cm while minimizing the weight of the robot. The excavation system's weight could be reduced simply by using thinner aluminum. Further weight reduction could include using lighter and better performing materials such as carbon-fiber.

Additionally next year's team could focus on upgrading the robot to increase its dust-proof operation. This could happen by covering parts of the digging and depositing mechanisms so that BP-1 won't be dispersed in the air to earn more points in competition.

7.2 Electrical and Computer Engineering Design

In terms of the ECE components of the robot, next year's team could focus on creating a thermally efficient and dust-proof ECE box that would allow the robot to operate for more than 20 minutes on the RMC arena under the high-temperature and high-humidity Florida weather.

Next year's team could also invest in high quality wires and plugs that could make the creation of a dust-proof ECE box and the connection of components easier and more secure. As previously described, BP-1 is electrostatic and if any of the material gets into any of the electronics it could short it, thus the security such an upgrade could provide is invaluable.

7.3 Software Design

Future attempts can be made to the software architecture by making it more modular. Implementation of a simulation software leveraging Robotic Operating System (ROS) could help development of various robot functions. This year, the team programmed the robot software directly to the Micro-box PC and faced limitations of not being able to program in an integrated development environment. A significant amount of program development time can be saved once the simulated software is developed.

8 Conclusion

This report serves to detail the entire process undertaken by the project team from August 2017 to April 2018. The goal of succeeding in the competition was a driving force for the team to produce a robot capable of performing all the competition requirements. The table below shows how the robot's specifications stack up to the expectations laid out early in the project cycle.

Parameter	Specification	Requirement Met
Maximum Size	1.5 m x 0.75 m x 0.75 m	Met (1.3 m x 0.74 m x 0 .74 m)
Maximum Mass	70 kg	Met (62 kg)
Storage Capacity	50 kg per payload	Partially Met (30 kg)
Gravel Collection	12 kg of gravel per 10 min	Met
Control	Fully Autonomous	In progress

The project team has been able to meet all but one of the parameters laid out early in the year. The one that has not been fully met yet is the fully autonomous control. This is still being tested and tweaked as the robot is being prepared for competition. Enough autonomy has been developed to at least earn points in a competition run for completing specific tasks autonomously. This project was truly a culmination of all the engineering skills picked up by the team members during their undergraduate years at WPI, mixing in mechanical and electrical engineering as well as computer science. The team hopes that WPI continues to go forward with this as an MQP and improve on performances at the competition.

Appendix A Gantt Charts for A, B, and C Terms

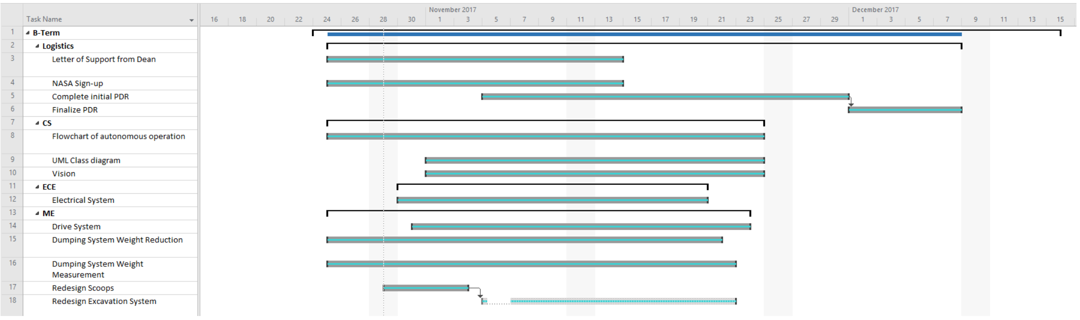


Figure 41: B-term Gantt chart

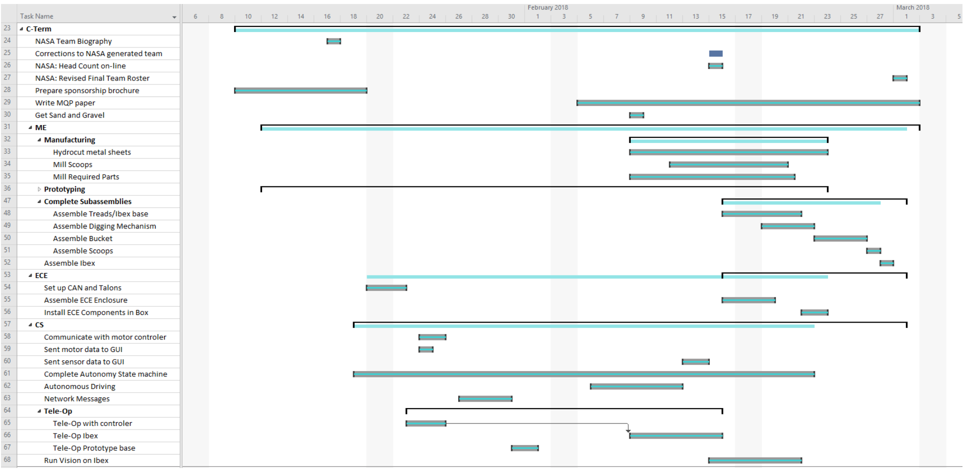


Figure 42: C-term Gantt chart

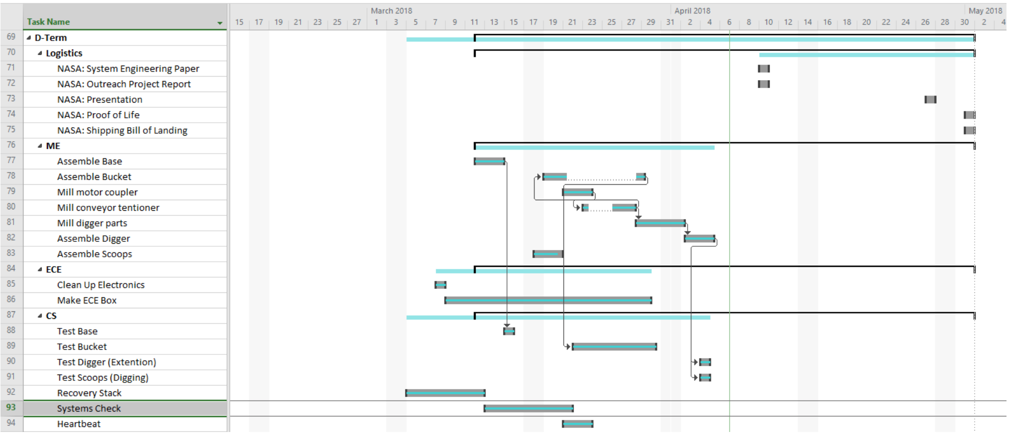


Figure 43: Current Term Gantt chart

Appendix B   Autonomy Flow Chart

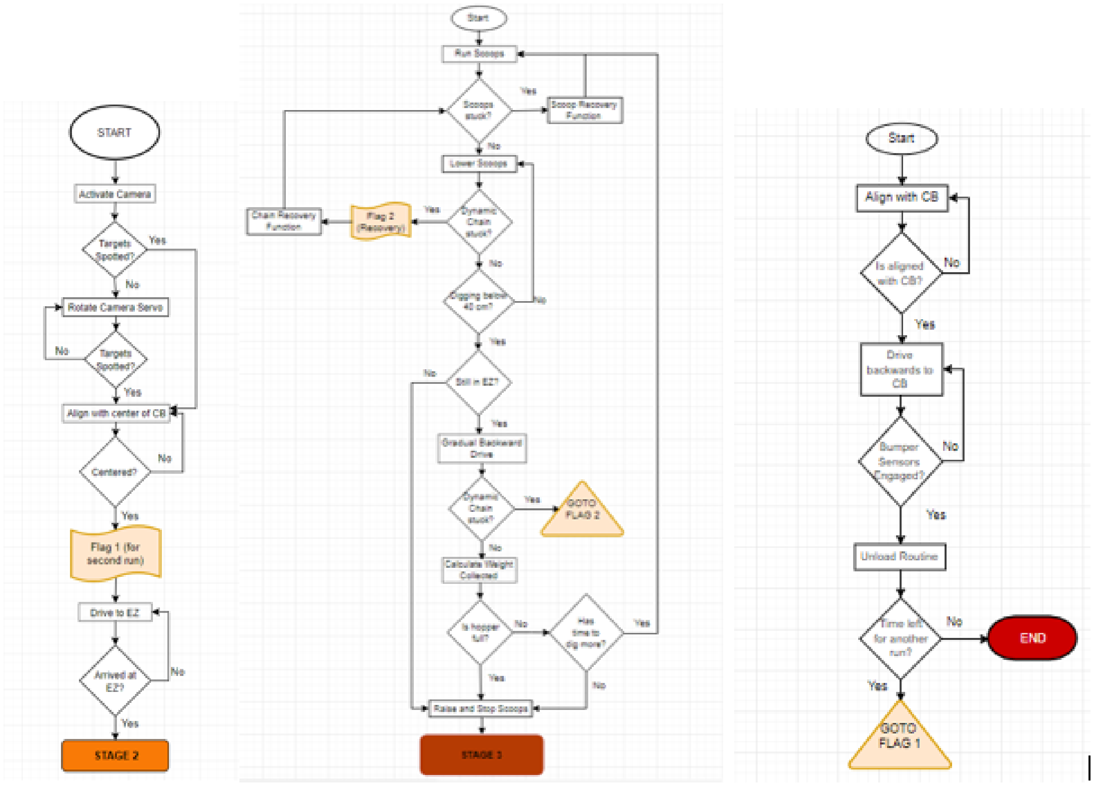


Figure 44: Flowchart for Autonomous Robot Control

Appendix C UML Class Diagrams

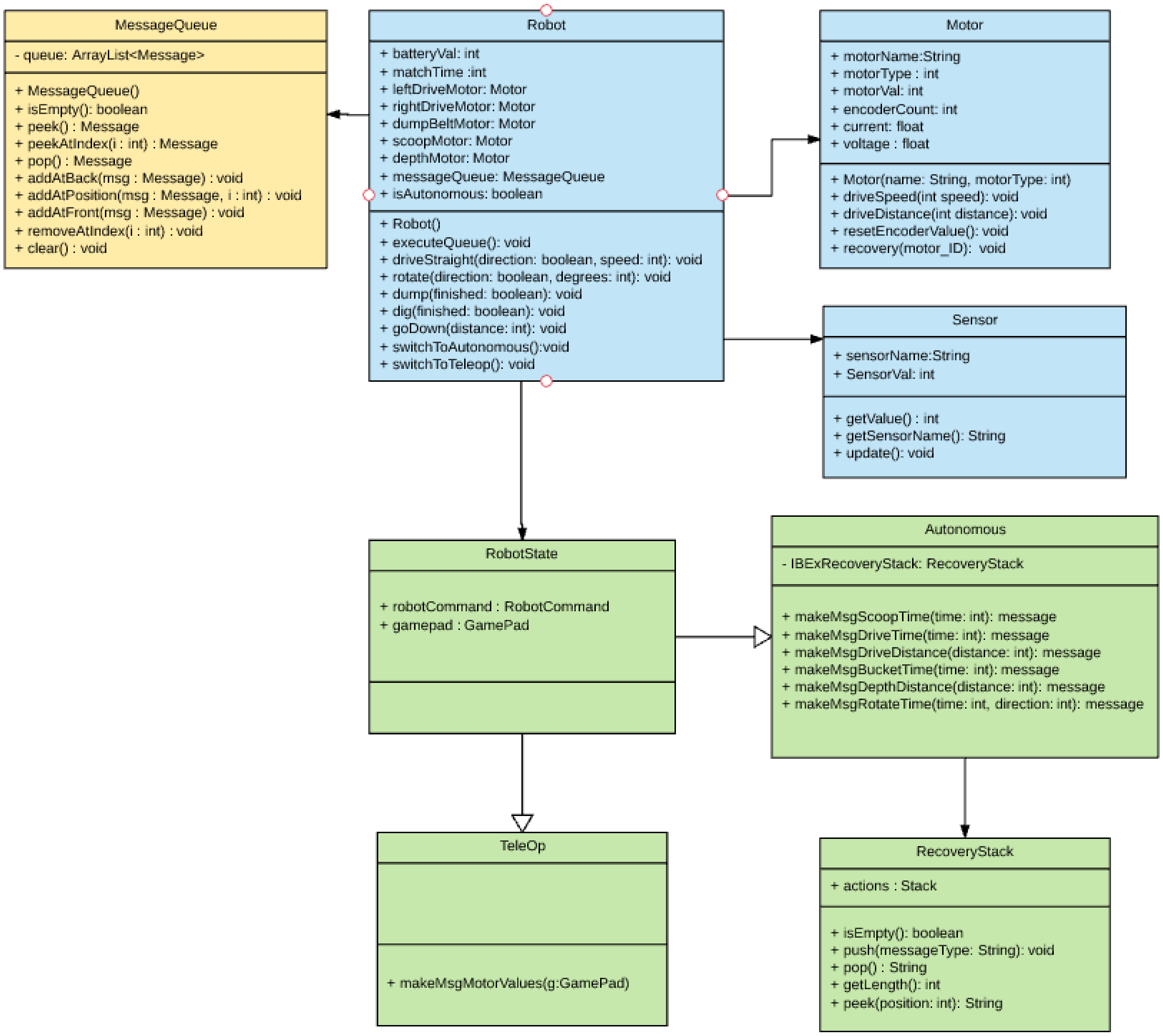


Figure 45: UML Class Diagrams for Control Station

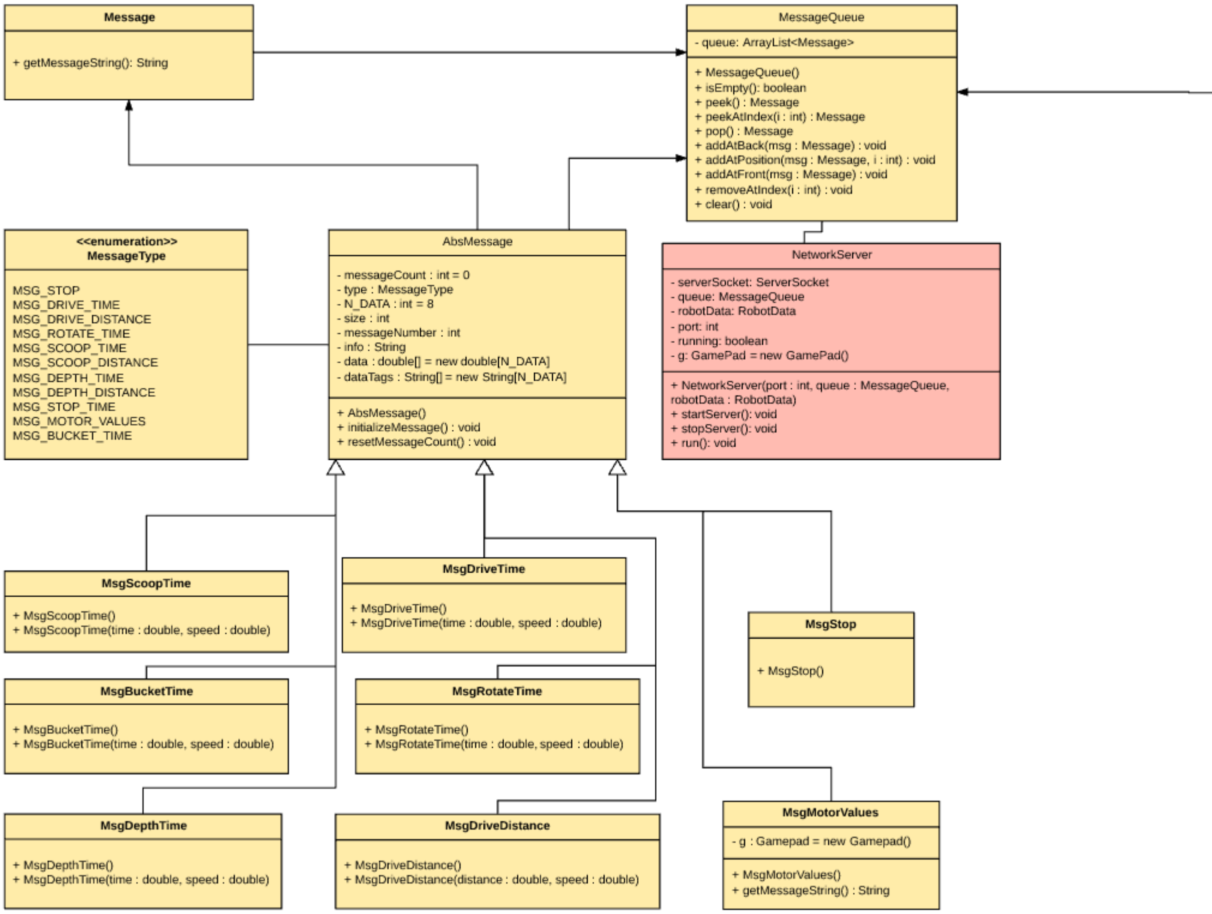
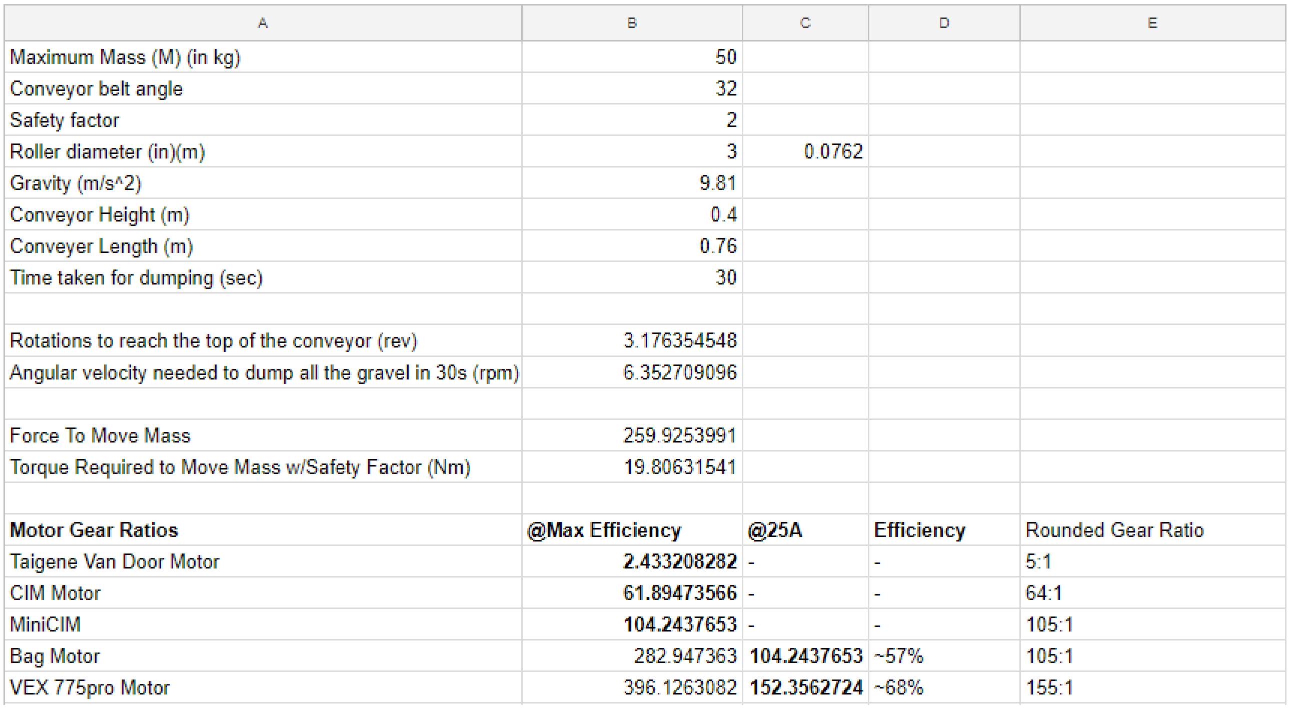


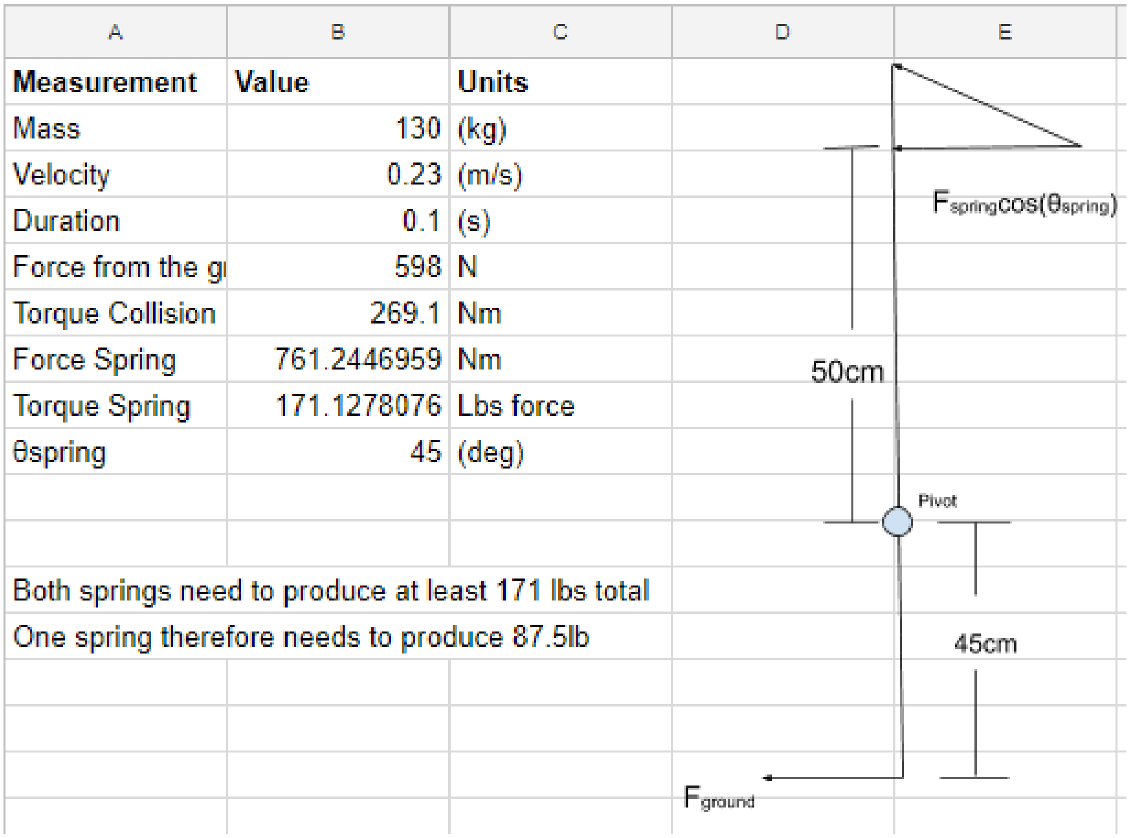
Figure 46: UML Class Diagram for GUI and Robot Control

Appendix D Motor Calculations for Sub-systems



A	B	C	D	E
Maximum Mass (M) (in kg)	50			
Conveyor belt angle	32			
Safety factor	2			
Roller diameter (in)(m)	3	0.0762		
Gravity (m/s^2)	9.81			
Conveyor Height (m)	0.4			
Conveyer Length (m)	0.76			
Time taken for dumping (sec)	30			
Rotations to reach the top of the conveyor (rev)	3.176354548			
Angular velocity needed to dump all the gravel in 30s (rpm)	6.352709096			
Force To Move Mass	259.9253991			
Torque Required to Move Mass w/Safety Factor (Nm)	19.80631541			
Motor Gear Ratios	**@Max Efficiency**	**@25A**	**Efficiency**	Rounded Gear Ratio
Taigene Van Door Motor	**2.433208282**	-	-	5:1
CIM Motor	**61.89473566**	-	-	64:1
MiniCIM	**104.2437653**	-	-	105:1
Bag Motor	282.947363	**104.2437653**	~57%	105:1
VEX 775pro Motor	396.1263082	**152.3562724**	~68%	155:1

Figure 47: Calculations for Conveyor System

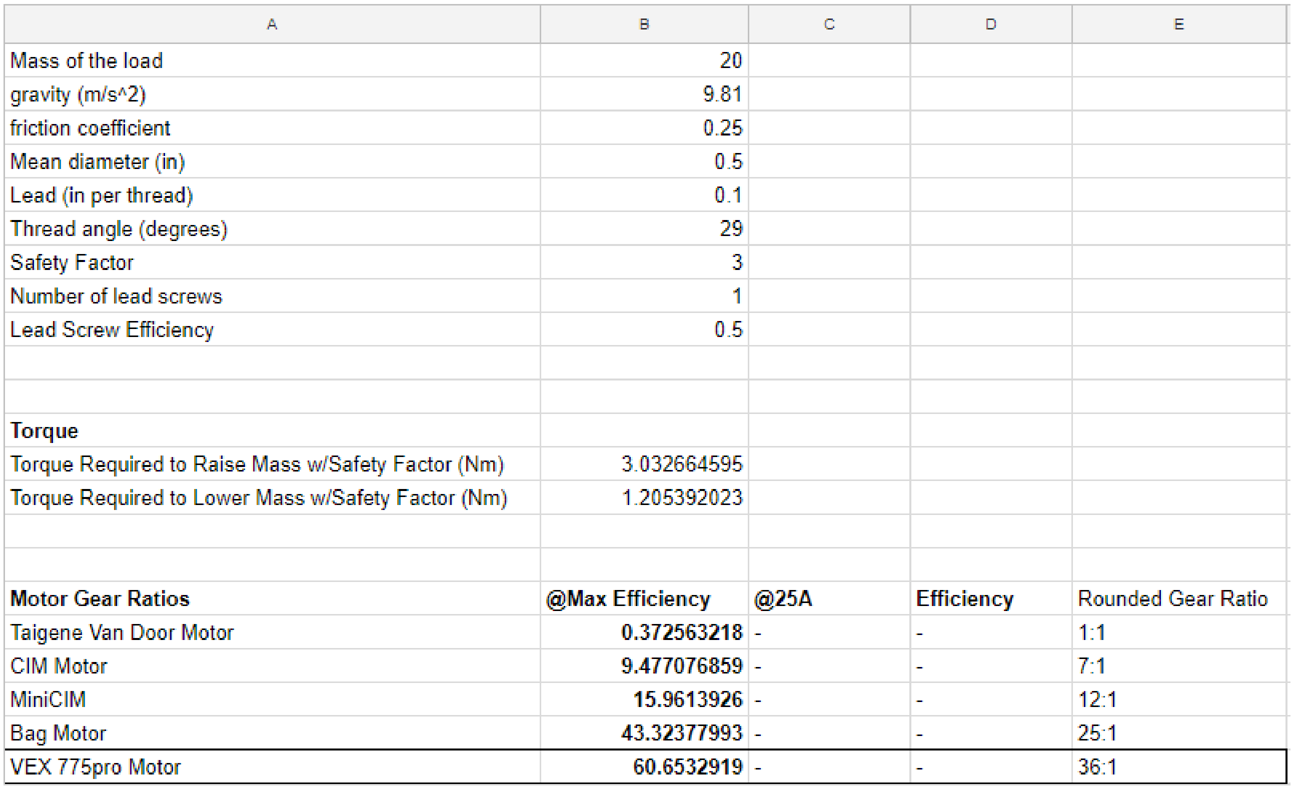


A	B	C
Measurement	**Value**	**Units**
Mass	130	(kg)
Velocity	0.23	(m/s)
Duration	0.1	(s)
Force from the g	598	N
Torque Collision	269.1	Nm
Force Spring	761.2446959	Nm
Torque Spring	171.1278076	Lbs force
θspring	45	(deg)

Both springs need to produce at least 171 lbs total

One spring therefore needs to produce 87.5lb

Figure 48: Calculations for Gas Spring Mechanism



A	B	C	D	E
Mass of the load	20			
gravity (m/s^2)	9.81			
friction coefficient	0.25			
Mean diameter (in)	0.5			
Lead (in per thread)	0.1			
Thread angle (degrees)	29			
Safety Factor	3			
Number of lead screws	1			
Lead Screw Efficiency	0.5			
Torque				
Torque Required to Raise Mass w/Safety Factor (Nm)	3.032664595			
Torque Required to Lower Mass w/Safety Factor (Nm)	1.205392023			
Motor Gear Ratios	**@Max Efficiency**	**@25A**	**Efficiency**	Rounded Gear Ratio
Taigene Van Door Motor	**0.372563218**	-	-	1:1
CIM Motor	**9.477076859**	-	-	7:1
MiniCIM	**15.9613926**	-	-	12:1
Bag Motor	**43.32377993**	-	-	25:1
VEX 775pro Motor	**60.6532919**	-	-	36:1

Figure 49: Calculations for Depth System