

April 2019

Autonomous Landmine Detection Rover

Benjamin Paul Wagner
Worcester Polytechnic Institute

Daniel Jacob Wensley
Worcester Polytechnic Institute

Dillon Joseph Arnold
Worcester Polytechnic Institute

Nicholas John Lanotte
Worcester Polytechnic Institute

Follow this and additional works at: <https://digitalcommons.wpi.edu/mqp-all>

Repository Citation

Wagner, B. P., Wensley, D. J., Arnold, D. J., & Lanotte, N. J. (2019). *Autonomous Landmine Detection Rover*. Retrieved from <https://digitalcommons.wpi.edu/mqp-all/6869>

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.



WPI

WORCESTER POLYTECHNIC INSTITUTE
ROBOTICS ENGINEERING PROGRAM

Autonomous Landmine Detection Rover

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 By
Dillon Arnold
Nicholas Lanotte
Benjamin Wagner
Daniel Wensley

Submitted To
Nicholas Bertozzi
Susan Jarvis
Craig Putnam

This report represents work of four WPI undergraduate students submitted to the faculty as evidence of a Degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, please see
<https://www.wpi.edu/academics/undergraduate>

Abstract

Major engagements throughout modern history have left unexploded and unmarked anti-personnel landmines, which result in thousands of casualties each year. Current demining detection methods use sensors such as metal detectors or trained animals such as rats. This project mitigates the threat to human life by replacing human operators with an autonomous robotic system. Building upon prior work, a rover was developed which locates and marks anti-personnel landmines in a user-defined location which a separate octocopter drone will later eliminate.


Acknowledgements

Our team would like to thank our advisors, Professor Bertozzi, Professor Jarvis and Professor Putnam for their guidance and support throughout the project. We would like to thank Irina Lavryonova for helping with early sections of the report and completing small miscellaneous tasks and Richard Wagner for donating material and building much of the four-bar mechanism. Lastly, we would like to thank Adriana Hera, Erica Stults, Dan Rivera, Chris Whimpenny, Alan Kirby, Mike Norton and Andrew Nagal for their time and resources.

Authorship

The writing responsibilities were divided among group members and are detailed below.

Section	Author(s)	Editor(s)
Abstract	Benjamin	Daniel
Introduction	All	All
2.1	Nicholas & Dillon	All
2.2	Nicholas & Dillon	All
2.3	Benjamin	All
2.4	Nicholas	All
2.5	Dillon	All
2.6	Benjamin & Irina Lavryonova	All
3.1	Nicholas	Dillon & Benjamin
3.2	Dillon	Benjamin
3.3	Benjamin	Dillon
3.4	Benjamin	Dillon
3.5	Dillon	Benjamin
3.6	Nicholas	Dillon & Benjamin
3.7.1	Dillon	Daniel
3.7.2-3.7.4	Daniel	Dillon

4.1	Nicholas	Benjamin
4.2	Dillon	Benjamin
4.3	Dillon	Benjamin
4.4	Benjamin	Dillon
4.5	Nicholas	Benjamin
4.6	Daniel	Nicholas
4.7	Dillon	Benjamin
5.1	Nicholas	Benjamin
5.2	Nicholas	Benjamin
5.3	Dillon	Benjamin
5.4	Benjamin	Dillon
5.5	Nicholas	Benjamin
5.6	Daniel	Nicholas
Conclusion	Daniel	Nicholas
References	Nicholas, Dillon, Benjamin	None
Appendix A	Nicholas	None
Appendix B	Dillon	None
Appendix C	Benjamin	None
Appendix D	Benjamin	None
Appendix E	Nicholas	Dillon, Benjamin, Irina Lavryonova 

Contents

1	Introduction	1
2	Background	3
2.1	Social Implications of Landmines	3
2.1.1	Affected Regions	3
2.1.2	Current State of Landmine Deployment	3
2.2	Types of Landmines	4
2.2.1	Anti-personnel Landmine	4
2.2.2	Anti-Tank Landmine	6
2.2.3	Unexploded Ordnance	7
2.3	Non-robotic Landmine Detection and Detonation Methods	8
2.3.1	Mechanical-Based Methods	8
2.3.2	Biology-Based Methods	9
2.4	Robotic Landmine Detection/Detonation Methods	10
2.4.1	Metal Detection	10
2.4.2	Ground Penetrating Radar	11
2.4.3	Infrared Sensors	12
2.4.4	Acoustic/Seismic Sensors	12
2.5	Methods of Marking Potential Landmines	13

2.5.1	Marking Materials	14
2.5.2	Material Application	14
2.5.3	Shaping the Mark	14
2.6	Current State of Previous Projects	15
3	Methodology & Experimentation	17
3.1	Sensor Platform	17
3.1.1	Landmine Detection Sensor Selection	17
3.1.2	Environment Sensor Selection	18
3.1.3	Preliminary Platform Design	18
3.1.4	Landmine Sensor Testing	21
3.1.5	Environment Sensor Testing	21
3.1.6	Sensor Platform Mechanical Redesign	22
3.1.7	Sensor Platform Control Software Design	25
3.1.8	Sensor Platform Testing	46
3.2	Rover Drivetrain	49
3.2.1	Common Drivetrain Designs	49
3.2.2	Drivetrain Selection	51
3.2.3	Development	52
3.3	Four-Bar Lifting Subsystem	53
3.3.1	Previous Four-Bar Mechanism	53
3.3.2	Link Selection	54
3.3.3	Drive System	56
3.3.4	Support	58
3.3.5	Assembly	59

3.3.6	Initial System Design	61
3.3.7	System Redesign	62
3.3.8	Programming	64
3.4	Four-Bar Testing Procedures	64
3.4.1	Operation	64
3.4.2	Programming Verification	65
3.5	Slider Mechanism	66
3.5.1	Slider Components	70
3.5.2	Bending Deformation Calculations	73
3.5.3	Manufacturing Procedures	75
3.5.4	Assembly Procedures	75
3.5.5	Programming Procedure	76
3.5.6	Complete Slider Design	78
3.6	ROS Network	79
3.6.1	Computer Vision System	79
3.6.2	Sensor Platform Control Node	80
3.6.3	IIC Flow Control Node	80
3.6.4	Landmine Detection Node	81
3.6.5	Maxbotix Ultrasonic Sensor Node	81
3.7	Mapping Algorithm	81
3.7.1	Map Definition Protocol	81
3.7.2	Navigation Protocol	85
3.7.3	Obstacle Logging Protocol	85
3.7.4	Low Battery Protocol	86

4	Results	87
4.1	Sensor Testing	87
4.1.1	Metal Detector	87
4.1.2	Infrared Camera	90
4.1.3	HC-SR04 Ultrasonic Sensor	91
4.1.4	Maxbotics EZ1 Sonar	91
4.1.5	LIDAR	92
4.2	Slider Testing Procedures	92
4.2.1	Manual Movement and Interference Checking	92
4.2.2	Limit Switches Engaged	92
4.3	Slider Initial Results	92
4.3.1	Design modifications	93
4.3.2	Finalization	93
4.3.3	Programming Fixes	94
4.4	Four-Bar Results	95
4.4.1	Initial Results and Observations	95
4.4.2	Component Redesign	96
4.4.3	Result of Redesigned System	99
4.5	Sensor Platform Testing	99
4.5.1	Individual Component Testing Results	100
4.5.2	Combined Component Testing Results	101
4.5.3	Full System Testing Results	102
4.5.4	Sensor Platform "Slim" Redesign	102
4.5.5	Integration with Slider Results	105
4.5.6	Subsystem Controller ROS Node Creation	105

4.5.7	Outdoor Testing Results	106
4.6	Navigation Protocol	107
4.6.1	Straight Line Test	107
4.6.2	Back and Forth Test	107
4.6.3	Simple Minefield Test	108
4.6.4	dGPS Integration Test	108
4.6.5	Google Maps API Integration Test	109
4.6.6	Full System Test	109
4.7	Drivetrain Results	109
4.7.1	Failed Analysis	109
4.7.2	Materials Testing	111
5	Discussion	113
5.1	Problems with Equipment	113
5.1.1	Husky Drivetrain	113
5.1.2	Operating System Upgrade	114
5.1.3	Computer Damage	114
5.1.4	Raspberry Pi Hardware Issues	115
5.2	Improved System Architecture	115
5.2.1	Network Communication to the Base Station	116
5.2.2	Computer Architecture	116
5.3	Slider Discussion	117
5.3.1	End Result	117
5.3.2	Success	117
5.3.3	Improvements	118

5.4	Four-Bar Discussion	118
5.4.1	Improvements	118
5.5	Sensor Platform Discussion	120
5.5.1	Results Discussion	121
5.5.2	Final Cosmetic Improvements	122
5.5.3	Future Improvements	123
5.6	Navigation Discussion	125
5.6.1	End Results	125
5.6.2	Improvements	125
6	Conclusion	126
A	Final ROS Node and Topic Graph	132
B	Slider Calculations	134
C	Four-Bar Free Body Diagrams	142
D	Four-Bar Winch Drive System Force Calculations	151
E	Landmine and Environment Sensor Testing	155

List of Figures

2.1	Image of Cambodia plain (Mottl, 2015)	4
2.2	Image of PMN-1 landmine (PMN mine)	5
2.3	Image of PMN-2 landmine (PMN mine)	5
2.4	Image of PMN-4 landmine (PMN mine)	6
2.5	Image of an M15 anti-tank landmine(Bonsor, 2018).	7
2.6	Example of unexploded ordnance in dirt (Irin, 2013).	7
2.7	Image of landmine flail tank (GRB ₁ 933, 2010)	8
2.8	Image of landmine sniffing dog (Police Dogs Centre, 2012)	9
2.9	Pulse induction metal detector (Metaldetectorlist.com)	11
2.10	Ground Penetrating RADAR (Radiodetection.com)	11
2.11	FLIR infrared camera (Grainger, 2018)	12
2.12	Seismic sensor (Alibaba, 2018)	12
3.1	Preliminary sensor platform 3D Model	19
3.2	Ultrasonic sensor placement	20
3.3	Preliminary marking system placement	20
3.4	Redesigned sensor platform	22
3.5	Marking system rotation	23
3.6	Marking system assembly	23

3.7	Wiring Diagram for I/O between the Arduino microcontrollers	27
3.8	Wiring diagram for I/O for the Raspberry Pi	28
3.9	Wiring diagram for communication between the Arduinos and RaspBerry Pi	29
3.10	Flowchart of Arduino Mega main loop	31
3.11	Flowchart of Arduino Uno main loop	32
3.12	Flowchart of the IIC ISRs	33
3.13	Flowchart of the process IIC messages helper function	34
3.14	Flowchart of the metal detector ISR	35
3.15	Oscilloscope measurement of PWM waveform from metal detector PCB . . .	36
3.16	Flowchart of the check metal detector helper function	37
3.17	Flowchart of the zero metal detector helper function	38
3.18	Flowchart of the measure orientation helper function	40
3.19	Flowchart of the set motors helper function	41
3.20	Flowchart of the motor ISRs	42
3.21	Flowchart of the home orientation ISR	43
3.22	Flowchart of the mark landmine helper function	45
3.23	Example Rocker-Bogie suspension system	50
3.24	Airless tire design	53
3.25	Vertical deflection based on link dimensions	55
3.26	Weighted decision matrix for DC motors	56
3.27	Tetrix Max 20:1 Worm gear set	57
3.28	MDD10A Motor Driver	58
3.29	Tetrix max axle hub	59
3.30	80/20 h-Truss	60
3.31	Tetrix TorqueNADO in preliminary mounts	61

3.32 CAD model of redesigned four-bar mechanism	63
3.33 Slider design 1	66
3.34 Slider design 2	67
3.35 Soft and hard stops	69
3.36 Slider Design	70
3.37 Bearing decision matrix	71
3.38 Free body diagram of slider (front view)	73
3.39 Free body diagram of slider (Side-View)	74
3.40 Complete Slider System CAD	78
3.41 ROS network diagram	79
3.42 Initialize map object	83
3.43 Example listener	84
3.44 Minefield Definition	84
3.45 Graphical display of rover traversing a minefield	85
4.1 Graph of depth detection test results	88
4.2 Landmine heating 0.25in below the surface	91
4.3 Assembled Slider	94
4.4 Assembled four-bar mechanism with redesigned drive system	96
4.5 Torque required for drive system with support height of (a) 7 inches and (b) 8 inches	97
4.6 Constructed Sensor Platform	100
4.7 Solidworks model of sensor platform "Slim"	103
4.8 Sensor platform "Slim" attached to slider mechanism	103
4.9 Simulation Results	110
4.10 Simulation Results	111

4.11	Simulation Results	112
5.1	Example of long range WIFI Antenna (Newegg.com)	116
5.2	Weighted decision matrix for material selection	119
5.3	Overhead view of winch and superstructure	120
5.4	Cosmetic Improvements to Sensor Platform	123
A.1	Section one	132
A.2	Section two	133
A.3	Section three	133
B.1	Free body diagram of slider (Front-View)	137
B.2	Free body diagram of slider (Side-View)	138
C.1	FBD for coupler plate	143
C.2	FBD for follower (upper) link	145
C.3	FBD for driven (lower) link	146
C.4	FBD for Head Pulley	147
C.5	FBD for Head Pulley Support	148
C.6	FBD for main winch	150

Chapter 1

Introduction

The use of unmarked landmines is a widespread problem across the world attributing to roughly 6000 casualties a year (Landmine Monitor 2016). The purpose of this project is to design a robotic solution that can autonomously search a designated area and mark the locations of landmines to be later detonated. A robotic solution is safer and more cost-efficient than other current demining methods. Given that there are a variety of landmines present in the world, this project will focus on the detection and marking of the PMN family of mines, especially the PMN1 mine. The PMN1 is one of the most widely used landmines and contains metal that is detectable (PMN mine) through the use of metal detectors.

This project builds upon previous Major Qualifying Projects (MQPs) which developed both a ground rover to identify and mark landmines using a metal detector and spray paint, and a flying drone that detonated the landmines with the use of sandbags. Improving the project consisted of improving the previous method of landmine detection by creating a metal detector sweeper arm and improved four-bar lifting mechanism in order to increase mine detection rates. Sensors for detecting the environment were added so that the capability to avoid obstacles such as trees, bushes and large rocks can be added in the future with improved software. The metal detector and other environment sensors were mounted onto a robotic arm that sweeps in front of the rover, allowing for more complete ground coverage when detecting landmines and obstacles in front of the robot, ensuring the safety of the robot.

Airless tires for better stability and resistance to sharp objects were investigated but never successfully implemented due to budget and time constraints. The previous algorithms for navigating and searching a field were improved upon to allow the rover to traverse a predefined area of terrain. The selection of what terrain to traverse has been defined in a more streamlined manner in comparison to the past iteration of the project. The Google Maps API minefield definition is completely interactive and the data can be stored and sent to the robot to act as the boundary of points that the robot will be allowed to explore. The addition of a user interface on an HTML webpage benefits the operator so that the boundary definition of the minefield will not be left to neither human interpretation nor human error.

These additions to the ground rover enabled it to actively search and mark landmines, which previously was not possible in past MQPs.

Chapter 2

Background

This chapter reviews the background research completed for the project including information on the most affected countries, types of landmines and current methods and techniques of landmine detection and detonation.

2.1 Social Implications of Landmines

This section discusses the research pertaining to social implications of landmines, specifically which areas are most affected, cost to produce landmines, statistics for casualties, and current treaties.

2.1.1 Affected Regions

Various landmines exist in 64 countries around the world with Egypt, Bosnia and Herzegovina, Cambodia and Croatia having among the highest concentrations (Land Mines: Hidden Killers, 1995). The terrain where these landmines exist varies widely between countries and even within a single country. For example, Cambodia is largely covered in plains and rivers with some surrounding low mountains (Overton, 2018), whereas Croatia varies greatly with mountains, barren plateaus at higher elevations and dense forests at lower elevations. Egypt has an estimated 23 million landmines, the highest number of any country, and the 5th highest number of anti-personnel landmines per square mile. Egypt's terrain makes detecting the mines difficult due to shifting sand and mud that can bury mines deep underground (Facts About Landmines, 2018).

2.1.2 Current State of Landmine Deployment

The cost to produce landmines is anywhere between 3 and 75 USD (Doswald-Beck, Herby, & Dorais-Slakmon, 1995), given that most are made using plastic, while others can contain wood or metal. However, removing landmines can cost anywhere between 300 to 1000 USD, depending on the disposal method (Facts About Landmines, 2018). The indirect cost of removal comes at the danger of human life. Anti-personnel landmines that are employed during conflicts continue to be a longstanding danger to both military personnel and civilians even after said conflicts have ended. It has been reported that there were “6,461 mine/ERW casualties, marking a 75% increase from the number of casualties recorded for 2014 and the highest recorded total since 2006 (6,573).” On top of this already high count, 78% of these casualties were civilians (Landmine Monitor, 2016). This number remains high despite the global ban for using, stockpiling, producing, and transferring anti-personnel landmines (Convention on the Prohibition, 1997). A convention of this ban, the Ottawa Treaty, aims to eliminate AP landmines entirely. The treaty also states that countries should remove landmines they have previously set (Ottawa Treaty). While many countries have removed their landmines before signing the treaty, sometimes records of landmines get lost or misplaced over time or they aren’t originally marked well due to delivery methods, such as airdrops, leaving little evidence (Landmines). This makes the country increasingly dangerous for the average citizen. This is where landmine removal options, such as demining robots, are necessary.



Figure 2.1: Image of Cambodia plain (Mottl, 2015)

2.2 Types of Landmines

There are two main types of landmines, Anti-personnel and Anti-tank. This section discusses these two types of landmines in depth and the dangers of other explosives scattered around the world.

2.2.1 Anti-personnel Landmine

The Anti-personnel (AP) landmine is designed to injure or kill one or more people. They can be buried in the ground or fixed to trees. AP blast landmines are triggered by pressure, primarily from someone stepping on them, and most are designed to injure a person by amputating one or more limbs rather than kill them. Conversely, fragmentation anti-personnel mines are designed to kill by using their explosive charge to propel metal

fragments. These deadly mines are usually placed above ground on stakes and made to blend into vegetation (Landmines, Explosive). One of the most common families of landmines is the PMN anti-personnel blast mines. These mines were created and produced by the Soviet Union starting in the 1950s (PMN mine). They all contain metal in their trigger mechanisms or structure (PMN mine). The PMN-1 is a small landmine that is about 112 millimeters in diameter and contains 240 grams of TNT explosive, which is more than other similarly sized landmines. It takes 5.8 kilograms of pressure to detonate and is effective at amputating limbs of the being who triggered it (PMN mine). Figure 2.2 shows one of these landmines.



Figure 2.2: Image of PMN-1 landmine (PMN mine)

The successor to the PMN-1 landmine is the PMN-2 landmine, shown in figure 2.3. This landmine is more resistant to methods that detonate mines by exerting a sudden burst of pressure on them. It also contains a total of 420 grams of TNT making it deadlier than the PMN-1 (PMN mine).



Figure 2.3: Image of PMN-2 landmine (PMN mine)

The final member of the PMN family is the PMN-4 landmine. The defining characteristic of this mine is that it is largely comprised of metal, making it easier to detect with a metal detector (PMN mine). It is also significantly less powerful, containing only 55 grams of TNT inside (PMN mine). An example of this mine can be found in figure 2.4.



Figure 2.4: Image of PMN-4 landmine (PMN mine)

2.2.2 Anti-Tank Landmine

Still another type of landmine is an anti-tank mine, which contains much more explosive than an anti-personnel mine and requires more pressure to detonate. They are buried just below the surface with the capability to destroy vehicles such as tanks and kill people in and around the vehicle (Gooneratne., Mukhopahyay., & Sen Gupta, 2008). To detonate one of these mines, usually around 350 lbs of pressure is required so a person cannot set them off. A common anti-tank landmine is the M15 blast mine which has 23 pounds of explosive and a diameter of 13.27in. This is made of steel and contains other metal components in the firing mechanism, so it would be easy to detect with a metal detector (Bonsor, 2018).



Figure 2.5: Image of an M15 anti-tank landmine(Bonsor, 2018).

2.2.3 Unexploded Ordnance

An additional hazard in current and previously active warzones is the existence of other unexploded ordnance. This is an explosive, such as artillery or mortar rounds, grenades, rockets, and missiles, that were used or left on the battlefield that have not exploded (Landmines, Explosive). These explosives can be detonated easily- in some cases by a light touch- making them more unpredictable and deadlier than landmines. Some unexploded ordnance may be harmless, but because of their unpredictable nature, all must be treated as extremely dangerous (Landmines, Explosive).



Figure 2.6: Example of unexploded ordnance in dirt (Irin, 2013).

2.3 Non-robotic Landmine Detection and Detonation Methods

A variety of methods are employed in order to detect anti-personnel and anti-tank landmines that do not rely on sensors. These include the use of mechanical instruments such as prodders or clearing vehicles, animals including dogs and bees, and even bacteria.

2.3.1 Mechanical-Based Methods

The most common technique for the mechanical detection of landmines is through the use of a prodder. This method requires an operator to insert a rigid stick into the ground at an insertion angle of less than 30 degrees from the ground, so as to avoid accidentally triggering the mine. The physical feedback the operator receives when doing so helps to eliminate the false positive results from other detection methods, such as using a metal detector. Originally a soldier bayonet, this prodder has recently undergone development to incorporate additional detection methods, such as acoustic and metal sensing, to assist with the operator's judgment (MacDonald & Lockwood, 2003). In order to perform this, however, the operator must be extremely close to the landmine, meaning they are inherently in danger. Further, this method is exceptionally slow, as only a small section can be assessed at any one time, resulting in a low turnover rate.

Oftentimes, militaries do not have the time to employ time-consuming detection methods like the prodder, and instead opt to use machines to relocate or trigger landmines, clearing a path for troops and vehicles in the process. Such machines can use chain flails, shown in figure 2.7, to impact the ground or metal rollers to detonate the landmines. Alternatively, rakes and plows have also been used to move the landmines out of the way. Although these machines minimize the danger to human life and are far faster than most other detection methods, these machines leave the area of operation largely destroyed and unusable. Further, the machines can still miss some landmines and make further detection more difficult. (Gooneratne et al, 2008)



Figure 2.7: Image of landmine flail tank (GRB₁933, 2010)

2.3.2 Biology-Based Methods

Instead of detecting the physical casing of the landmine, biology-based methods are capable of detecting the explosive compounds that are slowly released over time. Given that 95 percent of these released compounds will be absorbed by the surrounding soil, only small concentrations, between 1 and 200 femto grams of residue per millimeter of air, are present for these methods to detect (MacDonald & Lockwood, 2003). The most common method is through the use of trained animals including dogs, rats, and bees working with a handler. Dogs, which were first used as a detection method in the 1970's, are capable of detecting the explosive residue up to 60 centimeters beneath the surface (Kasban Elaraby, Zahran, Abd El-Samie, & El-Kordy, 2010). These dogs are trained to walk ahead of their handler and sit as soon as they smell the residue. Despite having a keen sense of smell enabling them to detect concentrations of explosive residue far lower than the current best mechanical olfactory sensors (MacDonald Lockwood, 2003), the efficiency of mine-sniffing dogs depends heavily on how it was trained and the capability of the handler. This means that some mines may be missed if there is a lower concentration of released residue than the dog was trained for. Additionally, dogs may inadvertently step on and trigger landmines, endangering those around them (Kasban et al., 2010)

Similar to dogs, the African giant pouched rat and honeybees have been used to detect landmines. The rats used by the University of Antwerp were trained to sniff along a given line and scratch when they detected the residue in the soil. Being far lighter than dogs, the rodents are far less likely to trigger any landmine they may step on (Kasban et al. 2010). Honeybees have been similarly trained by the University of Montana to be attracted to the scent of explosive compounds, a process that takes three or four days. Leveraging the insect's keen sense of smell and swarm dynamic, handlers are able to survey larger areas faster and more efficiently than rats or dogs (Kasban et al. 2010). However, both bees and rats are only able to operate under particular environmental and weather conditions, limiting their use. Additionally, and similar to bomb sniffing dogs, bees and rats can miss landmines if they are not trained to detect the lower concentrations of residue (MacDonald & Lockwood, 2003).



Figure 2.8: Image of landmine sniffing dog (Police Dogs Centre, 2012)

Further detection is possible through the use of engineered strains of bacteria. The bacteria are dispersed over a wide area, through the use of some airborne system, and allowed to grow for several hours, after which operators carefully move through the zone with ultraviolet lights (Kasban et al. 2010). The bacteria fluoresce when they come in

contact with explosive compounds such as TNT or 2,4-DNT, a derivative of TNT, allowing for the operators to easily see areas with explosive compounds indicative of buried landmines. Given that the released residue can be carried far from the original mine, this method is not safe from false positives (MacDonald & Lockwood, 2003). Further, the strains of bacteria are unable to operate in extreme temperatures or dry soil, eliminating their use in areas such as deserts, and are currently unable to detect non-TNT derivative explosive compounds such as RDX (MacDonald & Lockwood, 2003).

2.4 Robotic Landmine Detection/Detonation Methods

Deploying a robot to detect and detonate landmines requires fewer humans to be trained and deployed, saving time and money, and ultimately lowering the potential for the loss of life. Today, landmines can be detected using a variety of sensors that can be implemented on a robot include metal detectors, ground penetrating radar (GPR), infrared imaging, and acoustic/seismic sensors.

2.4.1 Metal Detection

Metal detectors (MD) are cheap and easy to install and thus used by both the military and civilians. Metal detectors can scan for landmines and other explosives containing metal, however, they are not effective against objects that are made of plastic. Additionally, they often end up detecting harmless scrap metal and other metallic substances in the soil, generating false alarms (Gooneratne et al, 2008). The three main styles are beat frequency oscillation, very low frequency, and pulse induction metal detectors. A beat frequency oscillator metal detector consists of two coils, one close to the ground and the other by the control box. Pulses of electricity are sent through both coils generating radio waves, which are picked up and converted into an audio signal for the operator to hear. Metal in the ground interferes with the radio signals and changes this audio signal. Although this is the simplest and cheapest method, beat frequency oscillator detectors are not reliable (Metaldetectorlist.com). Similar to the beat frequency oscillator detectors, very low-frequency metal detectors use two coils for detection, a transmitter and a receiver (Metaldetectorlist.com). The receiver coil is shielded from the signal of the transmitter but not from the return signal that is influenced by objects in the ground (Tyson, Metal Detectors).

This type of metal detection can give an approximate depth for a given object based on the strength of the magnetic field returned to the receiver coil. Since each has a different induction value, different types of metals can also be distinguished using this metal detector by measuring the phase shift in the produced magnetic field. By tuning to a particular phase shift, this metal detector will only alert the operator if the desired metal is detected (Tyson, Metal Detectors). This style of metal detection is more powerful and accurate but is ultimately more costly than a beat frequency oscillator.

Unlike the previous two styles, a pulse induction metal detector uses the same coil as a transmitter and receiver. This metal detector sends a pulse of current through the coil creating a magnetic field such that any metal in the ground induces an opposing magnetic field. The amount of metal present beneath the coil dictates the time it takes for the opposing magnetic field to dissipate, with more metal generating a longer lasting field. This metal detector measures this elapsed time and alerts the user when it is over a certain threshold (Tyson, Metal Detectors). This style of detection is able to ignore mineralized water and iron in rocks while maintaining a high level of sensitivity to large amounts of metal (Sergei, 2018). It also is more efficient than other methods at detecting deep objects such as gold (Metaldetectorlist.com).



Figure 2.9: Pulse induction metal detector (Metaldetectorlist.com)

2.4.2 Ground Penetrating Radar

Ground Penetrating Radar (GPR) uses radio waves to make images of materials under the soil and are also lightweight and easy to operate compared to metal detectors. This allows the sensor to detect landmines of varying material and casings (Gooneratne). GPR can reach depths of up to 100 ft depending on the frequency of radio waves used and the composition of the soil (GeoModel, 2014). Ground Penetrating Radar can be sensitive to certain metals in mines, soil moisture, and smoothness of the ground surface, potentially interfering with readings (Gooneratne). This sensor technology is expensive with used units selling from \$4000 to \$20,000 on Ebay as of late 2018 with other online sellers like Tiger Supplies selling them for as much as \$23,500 (Tiger Supplies, 2018).



Figure 2.10: Ground Penetrating RADAR (Radiodetection.com)

2.4.3 Infrared Sensors

Another method of detecting landmines utilizes infrared sensors. These sensors rely on the fact that all objects emit infrared radiation proportional to their temperature. This radiation is measured and displayed for an operator to visualize (Tyson, Night Vision). If there are distinguishable thermal differences between landmines and the material surrounding them, this visualization can allow an operator to see where landmines reside beneath the surface. Implementing infrared sensors is a safe, lightweight and capable approach to scanning large areas. However, the performance of these sensors varies greatly with differences in terrain (Gooneratne). The cost of an infrared camera is relatively low, starting at around \$200 on Amazon and other marketplaces.



Figure 2.11: FLIR infrared camera (Grainger, 2018)

2.4.4 Acoustic/Seismic Sensors

A final detection method implements Acoustic/Seismic sensors. These sensors use sound or seismic waves to vibrate the ground. Landmines are detected by analyzing the resulting vibration frequencies, given that the landmines vibrate at a different frequency than the surrounding soil. Unlike several of the previous detection techniques, this method is not affected by moisture or weather.



Figure 2.12: Seismic sensor (Alibaba, 2018)

However, it is unable to deeply penetrate the ground, making it only effective for landmines that are located close to the surface (Gooneratne). This method is more commonly used as a passive perimeter intrusion detection system in order to protect infrastructure. The sensors can detect the impact of footsteps and vehicles. The military uses this to protect military bases and quickly set up a perimeter in the field (AIS). The

sensors themselves cost anywhere from hundreds to thousands of dollars each and require an additional device to generate the sound waves, increasing the overall cost. This section is summarized in Table 2.1 below, detailing the advantages and disadvantages of each sensor and their relative cost.

Table 2.1: Landmine Detection Sensor Comparison

Name	Advantages	Disadvantages	Cost
BFO Metal Detector	Simple to design and produce	Readings are often unreliable. Only detects if metal present or not	\$
VLF Metal Detector	Detect depth and type of metal. Accurate readings	More expensive	\$\$
PI Metal Detector	Only requires 1 coil. Can ignore minerals and metal in rocks for much more accurate readings. Better at detecting deep objects	More expensive. Can't distinguish between metals or estimate depth	\$\$
Ground Penetrating Radar	Detect landmines without metal. Detect structures up to 100 feet in the ground. Creates image of underground	Very expensive. Requires a lot of computing power. Sensitive to moisture in soil, limiting its range	\$\$\$\$\$
Infrared Camera	Detect landmines without metal. Creates image of underground	Relies on landmines having different thermal signature than surroundings. Sensitive to moisture and composition of soil as a result	\$\$\$
Acoustic/Seismic	Detect landmines without metal.	Requires a producer of sound waves. Expensive. Hazardous around unexploded ordnance.	\$\$\$\$

2.5 Methods of Marking Potential Landmines

In order to properly display the location of buried landmines, demining solutions that employ marking systems need to be able to potentially mark a variety of surfaces,

including dry dirt, wet dirt, grass, gravel, leaves, and even snow in some cases. Different materials and dispensing methods can be used to effectively mark this variety of terrain but should take into account certain characteristics of the terrain, such as whether it contains a high amount of loose particulates, which may affect the marking method used. The pressure-sensitive nature of landmines eliminates the use of physical stakes or flags, which can potentially operate as mechanical prodders, triggering landmines and endangering nearby operators.

2.5.1 Marking Materials

Three common materials used for marking fields and dirt are spray paint, spray-on chalk, and powdered chalk. Spray paint is effective on grass or on ground that does not contain loose particulate, as the liquid solvent bonds to particulates which may be blown away due to the compressed air used (Henkler, 2015). Comprised of a powder instead of a liquid solvent, spray-on chalk is more effective at marking both grass and ground containing loose particulate, such as dry, dusty dirt, given that the powder resides on top of the particulate instead of bonding to it (Krylon). Powdered chalk is most effective at marking loose particulate, as it does not involve the use of compressed air in the application process. It can be used on grass but may allow grass blades to still show depending on the amount dispensed onto the grass. Out of the three common materials, spray chalk is the most versatile, being able to be used on a wide range of encountered surface that may be concealing a landmine.

2.5.2 Material Application

There are many different methods of dispensing a marking material. One way is to have an auger driven dispenser. This uses an Archimedes wheel to feed loose powder or liquid material through a chute. This low-pressure system dispenses material softly, in a sprinkling type of motion. A conveyor belt can similarly be used to transport powder material, but not liquid materials, given that the belt is not an enclosed space, containing the liquid. For sprayable liquids and powders, compressed air, such as from an aerosol can, is used. This delivery method can be more compact than a conveyor system, requiring only an aerosol can and an actuator. However, as discussed previously, the force of the compressed air upon release can scatter loose particulate, creating an incomplete or messy mark.

2.5.3 Shaping the Mark

There are two main ways that a mark can be created to be detectable using the previously discussed materials: through the use of a stencil or by moving the outlet of the dispenser a particular motion. The stencil is a fixed solution, that sits between the dispenser and the surface, creating the same mark each time. However, this solution, requiring an ad-

ditional component, is bulky, which may hinder lifting mechanisms or reduce the portability of the robot. Alternatively, manipulating the dispenser outlet allows for a more compact and versatile design. This method requires instead actuators to manipulate the orientation of the dispenser outlet with respect to the surface. For example, to create a square using a fanned tip spray paint can, a robot can move back and forth while dispensing the material, or an end effector can rotate to move the can through a given arc, effectively creating that same square. Both solutions allow for a variety of shapes and designs to be used. However, the latter allows for a wider range of designs that a demining robot can employ when scanning a minefield, which saves time and resources, given that stencils need to be swapped out to do the same.

2.6 Current State of Previous Projects

Demining robots employ a variety of sensors, some of which are discussed in Section 2.5, in order to identify active landmines for appropriate disposal. One such robot, developed by previous iterations of this project, was a Clearpath Husky A100 Unmanned Ground Vehicle (UGV) outfitted to detect and mark AP landmines. In order to detect the mines, the robot used an inductor constructed from enameled copper wire as a metal detector. This method was cheap to build and required little processing power, which met the previous design goal of maintaining a low overall cost for the robot. The metal detector also enabled landmines to be found in most environmental conditions, however, the sensor was only tested to be effective at depths of one inch. Reliably sensing AP landmines was a stretch goal of last year’s project so the metal detector was intended to be expanded on in future projects (Casey & Rocks, 2018). The Clearpath Husky UGV visibly marked detected mines using a field marking spray paint and a stencil activated by a servo. Two parallel two-foot-long four-bars made out of cedar wood supported the marking and sensing systems. Additionally, the Robot Operating System was used onboard the UGV to run navigation software. This software required a Laser Imaging Detection And Ranging (LIDAR) sensor, which was not available to the previous team, meaning that simulated LIDAR data readings had to be created for testing. The navigation and mine detection algorithms were never successfully tested together, and thus required further development to allow the UGV to autonomously and reliably navigate and detect landmines (Casey).

Previous iterations of this project also developed a robot capable of autonomously disposing of the AP landmines. This robot was a DJI Spreading Wings S1000 octocopter Unmanned Aerial Vehicle (UAV) modified to navigate to a defined GPS coordinate, identify markings indicating the presence of a landmine, and deploy weighted payloads. This robot achieved localization by use of an off-the-shelf GPS unit. However, the resolution of this did not offer sufficient accuracy for locating a point in space as precise as a single landmine. Thus, the GPS was supplemented by a computer vision system capable of detecting physical markers left by the UGV. The payload subsystem consists of a customized storage mechanism and payloads. The 3D printed storage mechanism was comprised of four bays with solenoid-actuated doors. This design sometimes caught the payloads, altering their trajectory or

preventing them from deploying entirely, and warranted a redesign in future iterations. The custom payloads were constructed from plastic wrap and sand, which was suitable for the project at the time. However, other materials were proposed as only the weight and form-factor of the payloads play a role in their performance (Lockman, Tighe, & Hayden; Casey & Rock). The robot was validated for the successful completion of phase of flight, localization, and payload deployment tasks. However, just like the previous UGV, these processes were never integrated and tested in unison. Integration had to be delayed due to damage sustained by the robot from a crash which could not be repaired in-house. In addition to the repair of the UAV and integrated testing, further proposed work on this UAV-based system consisted of improving existing systems as well developing new ones. Foremost among these, the target detection algorithm, as well as the design of the target itself, can be optimized for the expected conditions of operation. Furthermore, the detonation of a landmine must be verified, as marking an undetonated mine as safe is a safety concern. A previous iteration suggested that the sound of the detonation can be used as an indicator, but this was never developed any further (Lockman et al; Casey & Rock).

Chapter 3

Methodology & Experimentation

3.1 Sensor Platform

Choosing sensors for detecting landmines and obstacles in the path of the robot was a major component of the electrical engineering portion of the project. Selection of the landmine detection sensors was completed first, which called for the design of a platform to house the them and improve their functionality. This section discusses the design process undertaken to develop and test the sensor platform for integration with the other components of the project.

3.1.1 Landmine Detection Sensor Selection

The first step was to select the landmine detection sensors. A variety of factors were taken into account, including types of landmines that are detectable, reliability of readings, power requirements and operating conditions but due to the limited budget for this portion of the project, cost was the main deciding factor. A metal detector was chosen as one sensor due to the low cost and because all of the PMN landmines contain metal. The previous iteration of this project had proven a metal detector could be used to detect metal in landmines at shallow depths so another goal of this project was to increase the capability of a metal detector. Another sensor to accompany the metal detector was also selected in an effort to allow the system to detect landmines that do not contain enough metal for detection. A ground penetrating radar was the best choice of sensor because it may be able to detect landmines many feet below the surface more reliably than other sensors but purchasing such a sensor was out of the project budget due to the cost of the sensor and the need for a powerful processor to analyze the raw sensor data. An infrared camera was chosen for testing instead because it was relatively cheap, easier to implement with basic machine vision and may be able to detect non-metal mines within within the depth goal of 3 inches.

The metal detector selected was the Surf Pulse Induction metal detector kit because of the relatively low cost compared to other fully assembled metal detectors and because pulse induction metal detectors were found in research to be better at adjusting for interference and had greater detection depth. This metal detector being a kit makes it easier to integrate into the rest of the system since disassembling was not required and the schematics were well known. The Flir Lepton Infrared Camera was selected as the infrared camera because it is a development camera with software libraries and customer support from the manufacturer. Other hand-held infrared cameras cost less but disassembling and figuring out how to communicate with the camera would be difficult as documentation on these cameras could not be found.

3.1.2 Environment Sensor Selection

One stretch goal of the project is to implement obstacle avoidance so sensors for detecting obstacles in the environment were selected. Detecting obstacles such as trees or bushes in front of the robot is important for preventing crashes so The MaxBotics EZ1 ultrasonic sonar was selected because of its affordability and large sensing area over a standard ultrasonic sensor. Turning and reversing of the robot to avoid obstacles and landmines requires some capability to sense 360 degrees around the robot at a distance of at least five feet so the Scanse outdoor LIDAR was selected for detecting these obstacles.

3.1.3 Preliminary Platform Design

Before any of the sensors were tested, a preliminary design for the metal detector housing was made based on the research of metal detectors. This housing would be held up by the four-bar mechanism and slider. The metal detector would operate best if the coil was kept parallel to the ground due to the magnetic field extending perpendicularly to the direction of the current flowing through the coil so a platform capable of rotating the coil yaw and roll was developed to adjust the coil in response to changing soil heights. A linear actuator was also selected to raise and lower the coil a distance of three inches. This was deemed necessary so that the coil could avoid hitting small mounds of soil or other material on the ground. The HC-SR04 ultrasonic sensors were chosen to sense the ground and provide feedback so that a micro controller could control motors that would keep the coil parallel with the ground. Five of these sensors were placed around the coil enclosure with 4 around the perimeter of the coil and the last one in the center. The outer ultrasonic sensors would be used to sense upcoming changes to the ground surface so the coil height and orientation could be adjusted while the ultrasonic sensor in the center would prevent the coil from hitting the ground. The placement of these sensor can be seen in figure 3.2.

The sensor platform also needed to be designed to house the landmine marking system so that the ground could be marked in the center of the metal detector coil for accurate and precise marking of the landmines. Many marking system options were considered

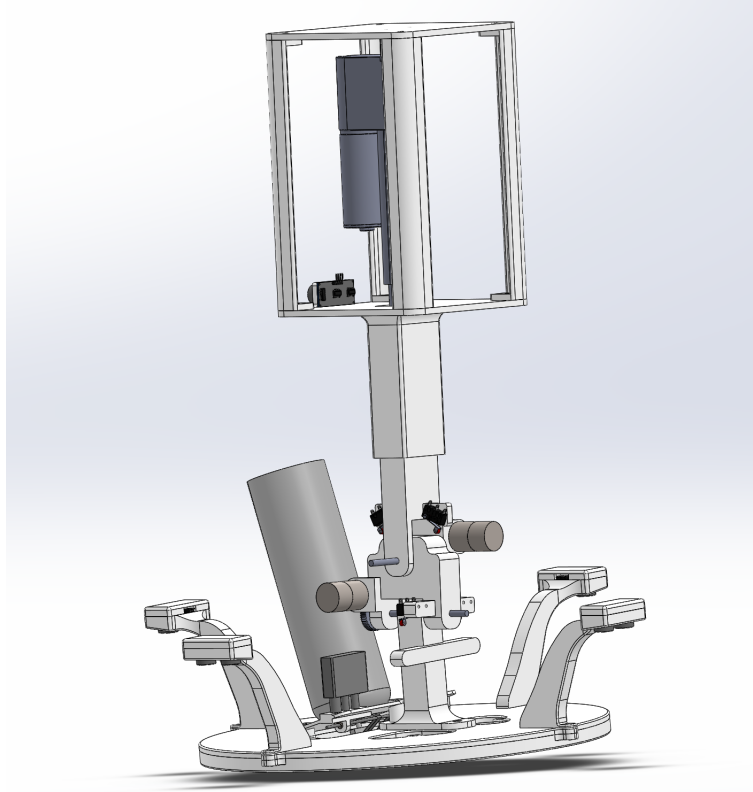


Figure 3.1: Preliminary sensor platform 3D Model

including an auger that would dump chalk on the surface, but a field marking spray can was chosen because it would be cheap, lightweight and easy to design a mechanism to press the spray nozzle. Field marking cans spray paint in a straight line so combining this with the rotating platform for the metal detector would allow shapes like a square to be drawn with small rotations of this platform. The placement of the spray can along with the trigger mechanism can be seen in Figure 3.3. The spray can is highlighted in blue. It is tilted at an angle to allow a variety of spray can sizes to be used while not causing interference with the sensor arm movement. It is actuated by a lever controlled by a servo and is held in place by sliding the can into a slot that holds a lip at the top of the can. This method of securing the can was meant to be revised later as it would likely come loose but was sufficient for a preliminary design.

The last step was to calculate the necessary torque to rotate the weight of the metal detector coil, housing and spray can. A method of detecting when the rotation was at a limit angle was also required. The ClearPath Husky A100 can traverse a maximum incline of 30 degrees so the maximum rotation for the joints of the sensor platform was set to 30 degrees from the vertical. Limit switches were placed so that they would be pressed at these extremes and alert the robot controllers that the sensing platform had encountered a steep incline that that the robot cannot tolerate. Assigning material to the sensor platform model allowed for an estimate on weight to be calculated which was a total of 6 lbs. The linear actuator is rated for 35lbs of force at maximum so the 6lb weight was far under this limit.

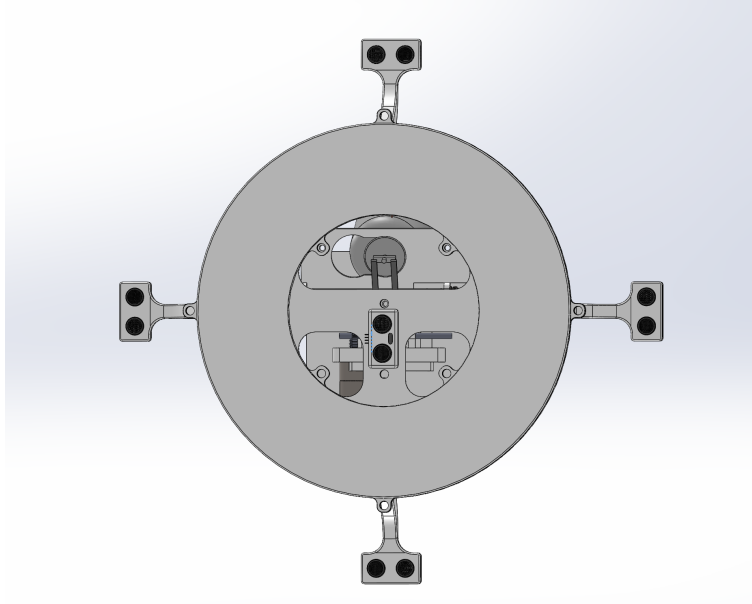


Figure 3.2: Ultrasonic sensor placement

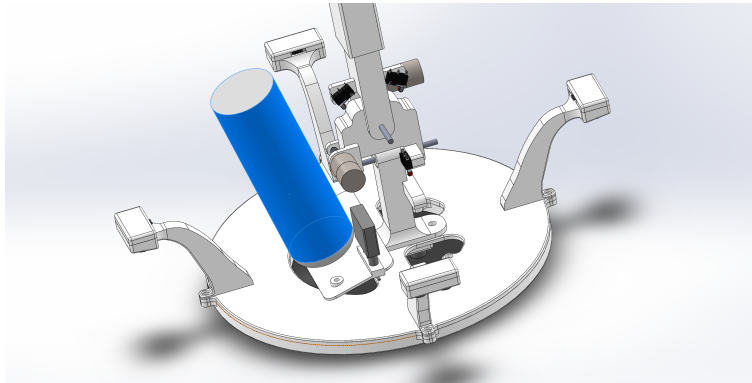


Figure 3.3: Preliminary marking system placement

Estimating the center of mass of the sensor platform allowed for the torque each motor would need to provide to rotate it. The worst case scenario was approximately 10in*lbs of torque needed by each motor to rotate about the yaw and roll axes. To meet this requirement, a 12V motor capable of 2in*lbs of torque geared down 25:1 with a worm gear transmission. The worm gear would also prevent the motors from being back driven, allowing for power to be cut to the motors when not needed to rotate the sensor platform.

The MaxBotics Sonar was placed on the sensor platform in front of the linear actuator as it would be able to scan a large radius for obstacles out in front of the robot. The location of the infrared camera on the sensor platform was not selected due to testing being required to understand where it would best be placed. The LIDAR was placed on the four-bar mechanism instead so that its position would not move relative to the husky A100. At this point, testing of all of the sensors was required to finalize a design for the sensing platform. The metal detector was of significant concern because the preliminary

design includes a large amount of metal that may cause significant interference with the metal detector. Testing for each of the landmine and environment sensors were written and conducted to provide the data necessary to finalize the design and begin building the system.

3.1.4 Landmine Sensor Testing

The goal of the landmine sensor testing was to determine the capability of each sensor to detect the PMN family of landmines in a variety of operating conditions. The full procedure for each of these tests can be found in the *Landmine and Environment Sensor Testing* document, which is attached as Appendix E. Many tests were conducted with the metal detector to determine optimal coil inductance, detection depth, minimum amount of metal needed for detection, detection related to shape of the metal, and detection with varying levels of interference. Many coils of various inductance were created and tested under the same conditions to determine if there was a range for optimal detection capabilities. The best coil was then used to test the depth of detection of various masses of steel. Masses ranging from 0.5 g to 200 g were tested in ideal conditions meaning low interference and through air. The affects surface area had on detection were tested by obtaining a few pieces of steel similar in mass but with different surface areas and testing in ideal conditions. The sensor was tested for detection ability through soil mediums consisting of dry soil, wet soil, sand and grass since these mediums will be present in-between the metal detector and the landmines. The last test was to determine the affects nearby metal on the robot would have on detect-ability of the targeted metal. The motors being used in the sensing platform and the spray can being used for the marking system were introduced into the magnetic field and the affects were observed on some of the samples tested previously in ideal conditions.

The infrared camera testing was less in-depth due to limited resources and time. Ideally, the infrared camera would be tested throughout the day in a variety of soil conditions with varying temperatures but this level of testing was out of scope and budget of the project. The infrared camera was tested by burying a dummy landmine in bins full of various soils which was then heated with a heat lamp. The camera image was then observed over a period of a few minutes to see if any change in temperature was detected by the camera. The landmine was tested at various depths to determine when it is no longer detectable in different soil conditions.

3.1.5 Environment Sensor Testing

Testing of the Maxbotix EZ1 and LIDAR was conducted with the goal of verifying that these sensors were functional and to help with deciding final placement of these sensors on the robot. The HC-SR04 ultrasonic sensor was also tested to ensure they could be used as planned. The distance measurements were tested against pavement and grass to verify that the readings were generally accurate or could be filtered to eliminate noise.

3.1.6 Sensor Platform Mechanical Redesign

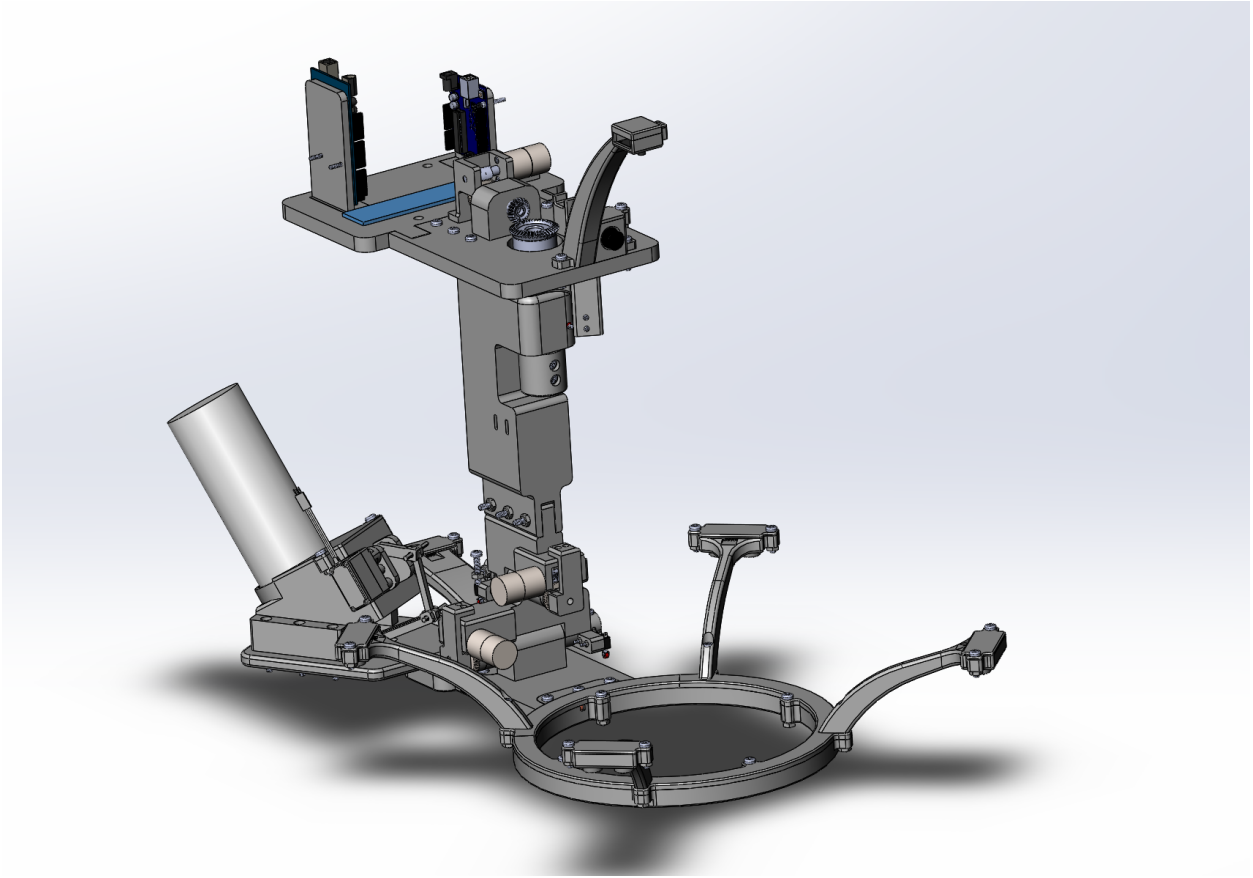


Figure 3.4: Redesigned sensor platform

After sensor testing was completed, the data collected was used to make the final design of the sensor platform. Refer to Section 4.1 in Results or to the *Landmine and Environment Sensor Testing* for full result data. The test results had shown that the spray can and motors will cause interference with the metal detector and must be removed from within the metal detector's area of detection. Further testing while designing the new sensor platform indicated that all detectable metal must be out of the field of detection or the metal detector may frequently report false positives. This called for a complete overhaul of the preliminary design. The metal detector coil was placed extended out front of the joints of the platform so that the motors would not be inside the magnetic field. The ultrasonic sensors also had to be placed four inches out from the metal detector coil to avoid detection with the sensor placed at the center of the coil in the previous version removed. The spray can being anywhere inside the metal detector prevented detection of any other metal so it was moved onto the other side of the sensor platform. This placement required the spray can to be moved into the center of the coil to accurately mark a landmine. The solution was to have an additional joint that controls pitch of the sensor platform so that the marking system could be rotated over the detected landmine. Figure 3.5 shows the rotation of the

sensor platform while detecting metal and then while marking a landmine.

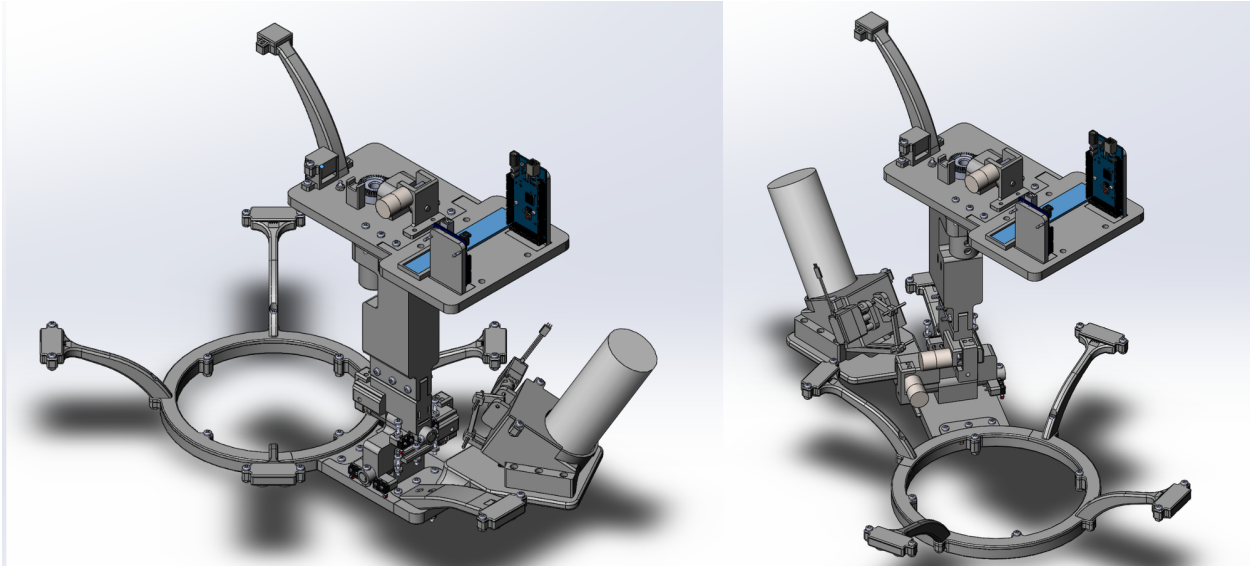


Figure 3.5: Marking system rotation

The marking system is securely attached to the rest of the sensor platform assembly but can be detached with the removal of three screws. Currently the marking system was designed for use with a field marking spray can but additional marking system assemblies could be made for other types of cans and be swapped so that the user could use a spray can that they see fit or is more available to them. The field marking spray can marking system is shown in figure 3.6.

An additional ultrasonic sensor had to be extended from the back of the sensor arm to ensure that the marking system would not hit the ground while adjusting the orientation of the platform. The marking system was also placed at a 150 degree angle to the metal detector coil. This reduced how far out the sensor platform needs to be extended by the four-bar so that the platform will not hit the Husky A100 while rotating pitch. The motor torque calculations were also redone using estimates of weight and center of mass. The estimated weight of the platform was 3.5 lbs with 2.5 lbs being from the marking system and 1 lb coming from the metal detector coil. The center of mass of the metal detector coil was 8 inches out front while the marking system was 7

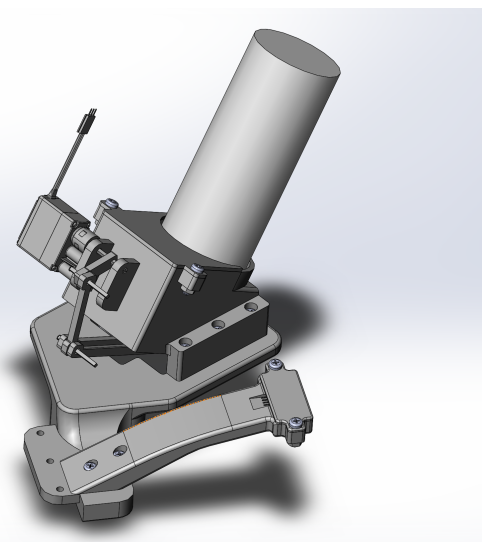


Figure 3.6: Marking system assembly

inches out back. Balancing the torques about the yaw axle brings a maximum torque of 9.5 in*lbs. The maximum output torque from the gear assembly is 50 in*lbs so the maximum torque under normal operating conditions is well below the maximum value.

$$Torque = \Sigma(Force * Distance) = 2.5lbs * 7in - 1lbs * 8in = 9.5in * lbs \quad (3.1)$$

The same process was conducted for the roll axle. At an angle of 30 degrees, the metal detector coil is going to exert a force of 0.86 lb, about 2 inches away from the roll axle while the marking system will add an additional 2.17 lbs of force 5 inches away from the axle. This brings the total torque to about 11in*lbs.

$$Torque = 1lbs * \cos(30) * 2in + 2.5lbs * \cos(30) * 5in = 12.56in * lbs \quad (3.2)$$

The worst case scenario is when the coil is being held at a 0 degree angle. The metal detector coil will not exert any torque but the marking system will exert about 2.5 lbs at 6 inches, bringing the maximum torque experienced to be about 15 in*lbs. This is still about 1/3 of the maximum output torque from the motors. Even if the maximum torque experienced in testing is greater than these calculations, the motors and gear box are suited to take on twice the load while still operating well under their stall torque. The motor responsible for pitch will only need to overcome friction forces to rotate the sensor platform if the robot is on a level surface. The worst case scenario would be if the robot were on a hill of 30 degrees. In this scenario, it was over estimated that a force of about 2 lbs extended 6 inches out from the center of rotation would be acting on this joint. This would be as if the entire weight of the marking system, metal detection coil and arm assembly were acting at the spray can's center of mass. The total weight is about 4 lbs so at an angle of 30 degrees, the force on the pitch joint would be 2 lbs. This creates a torque of 12 in*lbs of torque maximum on the pitch joint.

$$Torque = 4lbs * \sin(30) * 6in = 12in * lbs \quad (3.3)$$

The same motor used for the other joints is used here but the gear reduction is 50:1, making the maximum output from the transmission 100 in*lbs of torque. The gear reduction is larger for this joint, mainly because the same 25:1 worm gear drive is being used to prevent back drive of the motor, and then an additional 2:1 reduction is multiplied in for the bevel gears. The bevel gears serve no other purpose except to rotate the output of the transmission by 90 degrees so that the motor could be oriented horizontal while the output is vertical. The large size of the bevel gears was selected so that they could be 3D printed attached to the axle that will be supporting the entire weight of the sensor platform. The gear reduction was a result of trying to reduce the size of the mount that the motor would be attached to.

Shear calculations were done for the nylon gears on the worm gear drive and the 3D printed axles for yaw and roll. The 3D printed axles were printed with 30% infill resulting in an estimated yield strength of the PLA of 3045 PSI. The roll axle would have 12.56 in*lbs of torque applied. The force would be applied at about 0.19 inches from the center of the axle where the key slot of the axle beings producing a shear force of 66 lbs. The area in

which this shear force is applied is 0.182 square inches resulting in a pressure of 363 PSI. The factor of safety is 8.38 so shear due to the key slot failing should not occur under expected conditions. The yaw axle had a shear force of 50 lbs acting on an area of 0.252 square inches for a pressure of 198 PSI and a factor of safety of 15.37. If one of these axles were to fail it would be due to the layers of the part delaminating over time due to a constant force. This makes the axles suitable for testing but with extended use under load the part could fail. This is a property of 3D printed parts that would not exist with an injection molded part. The nylon gears have a shear strength of about 6500-11000 PSI. The area at which the shear force would be applied was estimates at 0.0084 square inches and the shear force would be about 23.83 lbs on the roll axle, making the pressure 2837 PSI. The factory of safety could be anywhere from 2.3 to 3.87. The yaw axle gear would have a shear force of 18.02 lbs making the pressure 2146 PSI with a factor of safety of 3.02 to 5.13. These are lower than ideal but the gears were already purchased so they were used anyway with the intent on stronger gears being recommended in future work.

The final component of the sensor platform is the addition of the infrared camera being placed on an arm facing downward into the center of the metal detector coil. The field of view from the infrared camera was found to be very small so the placement in the model is the height required to view inside the open space in the metal detector coil holder without seeing much of the coil holder itself. This also ensures that any landmines the infrared sensor picks up are in the center of the coil.

Some additional design challenges were designing the system to easily be assembled and taken apart, using nylon hardware close to the metal detector and making sure each part could be 3D printed with the printers available for the project. The nylon hardware is necessary to prevent the metal detector from falsely detecting metal. The other two additional challenges were to ensure new parts could be quickly made in the event one breaks or changes to the design are needed.

3.1.7 Sensor Platform Control Software Design

Once the mechanical design was completed, development of the software to control the many functions of the sensor platform was designed. To offload processes from the Raspberry Pi, which was selected to be the main controller for the robot, an Arduino Mega was chosen to control processing the metal detector signals, adjusting the orientation of the metal detector coil and marking the landmines. The software relies on a main function loop with several interrupt service routines to read the metal detector, control the motors and communicate over IIC. After some system analysis on the worst case scenarios, it was found possible for many deadlines to be missed and ISRs to be skipped due to the primitive interrupt controller on the Arduino Mega. The solution was to offload the metal detector processing to an Arduino Uno. All software for both Arduinos was done in C++ using the Eclipse Arduino Plugin. The software was designed to be modular so reorganizing tasks between the two boards was a simple process. The following sub sections discuss this design

process in further detail.

Arduino Peripheral Configuration

Figure 3.7 is a graphical diagram of the peripheral components coming off of both Arduinos. The Arduino Mega has five ultrasonic sensors connected. Each sensor has their signal and sense pins bridged as shown by the green wires. This allows each ultrasonic sensor to only require one wire for sending and receiving data, as opposed to needed two wires per sensor. Each limit switch has a voltage divider around it so that the limit switches do not create a direct short to ground and to also limit the voltage on the arduino GPIO pins to 2.5V to prevent damage to the pins. This was chosen over activating the internal pull-up resistor to reduce the number of wires needed to run all of the switches.

The current sense for each motor consists of a fuse and a voltage divider with a total resistance of 1M ohm. Current to the motor flows from the power rail, through the fuse and to the motor (labeled "To Motor" in the diagram). If a motor stalls, the current through the fuse increases to about 0.6Amps. The fuse is rated for 0.5Amps so it will trip when the motor stalls. Normally, the resistance across the fuse would be negligible resulting in the Arduino reading in 3 volts from the voltage divider (current sense wires are the orange wires in the diagram), which means the motor is not stalled. When the fuse is tripped it can reach hundreds of ohms which is much larger than the motor resistance when stalled. This results in the voltage across the fuse to be close to 12V so that the voltage divider outputs a voltage to the Arduino that is close to 0V. Voltage readings close to 0V signal to the software that a motor is stalled. The voltage divider resistance is one mega ohm so that the sensing circuit does not interfere with the motor operation and it offers some protection from back emf generated by the motors. The Arduino Mega also has the motor signal output pins and the encoder read-in pins. The encoder read in pins are connected to the Arduino Mega interrupt pins so that the encoder data can be handled by an interrupt service routine in the software.

The Arduino Uno is connected to a digital potentiometer, used for automatically removing noisy readings from the metal detector at start up, and a pin to read in the metal detector audio output. There is a voltage divider with a resistance of one mega ohm for the input pin so that the Arduino sensing does not interfere with the operation of the metal detector board and to bring the voltage down to 3 volts. The metal detector audio waveform peaks at 12v which would damage the Arduino Uno if read in directly. The audio output is still connected to a buzzer so the when metal is detected the buzzer will beep. The buzzer function is left in so that operation of the metal detector board can be verified without the Arduino Uno. Clear pin numbers can be found in the Arduino software which can be found on the team's Github page.

RaspBerry Pi Peripheral Configuration

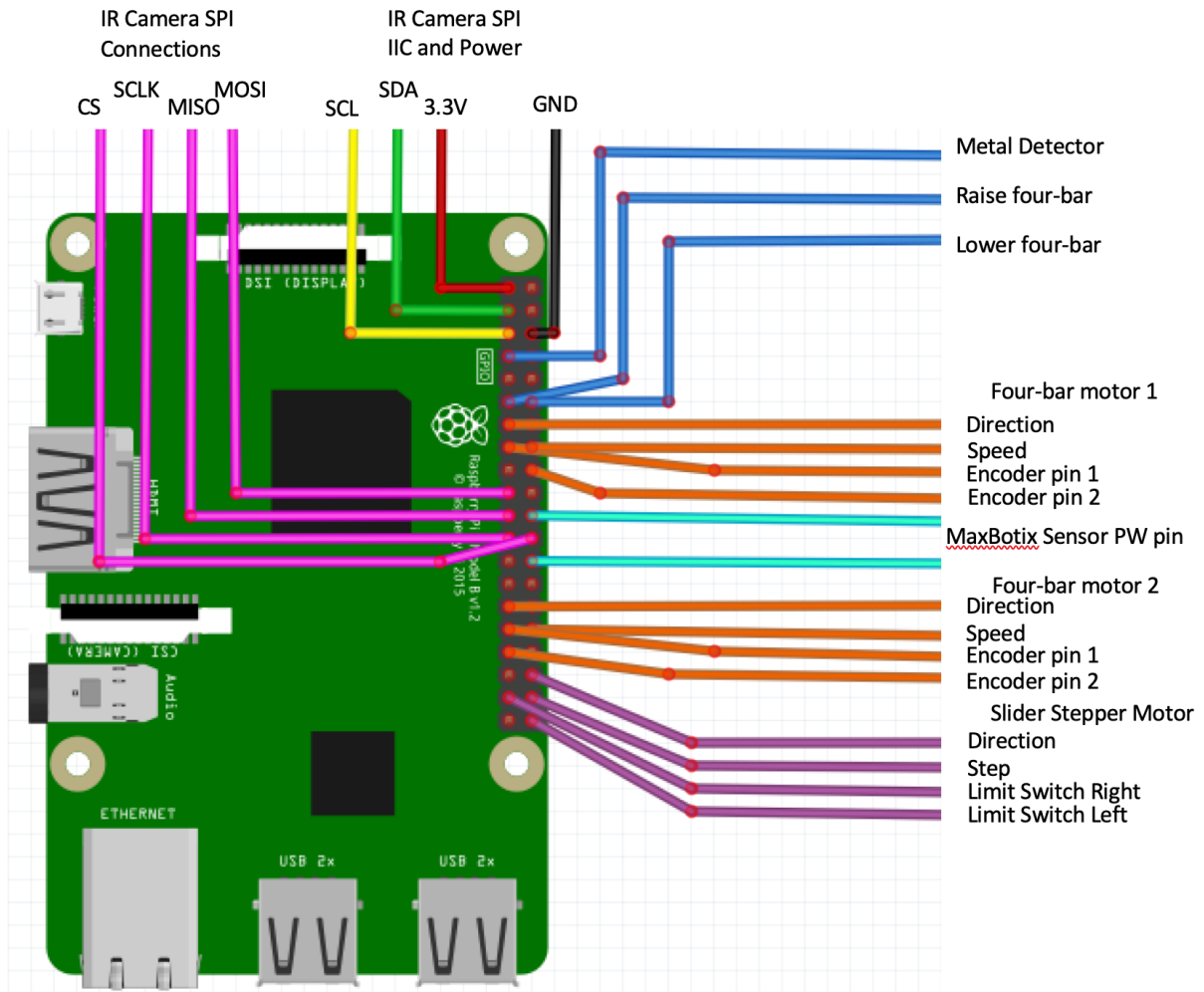


Figure 3.8: Wiring diagram for I/O for the Raspberry Pi

Micro Controller Communication Configuration

A RaspBerry Pi 3B+ was selected as the main control board for the robot. The main mode of communication between the Raspberry Pi and both Arduino boards is IIC since all three boards have the capability and it allows relatively few GPIO ports to be used for communication as many more are needed to connect other peripherals. The Arduino Uno also acts as a metal detector sensor with one of its GPIO ports being used as an output that is set high when metal is detected and low when metal is not detected. This allows for nearly instant communication with the RaspBerry Pi. The Arduino Mega has two GPIO ports that are outputs to the RaspBerry Pi. One is to signal the RaspBerry Pi to raise the four bar mechanism while held high with the other signaling lowering of the four-bar. If both outputs are held high, then this will communicate that the robot has encountered a

steep slope that the robot should not try to traverse. A graphical communication circuit schematic can be found in figure 3.9.

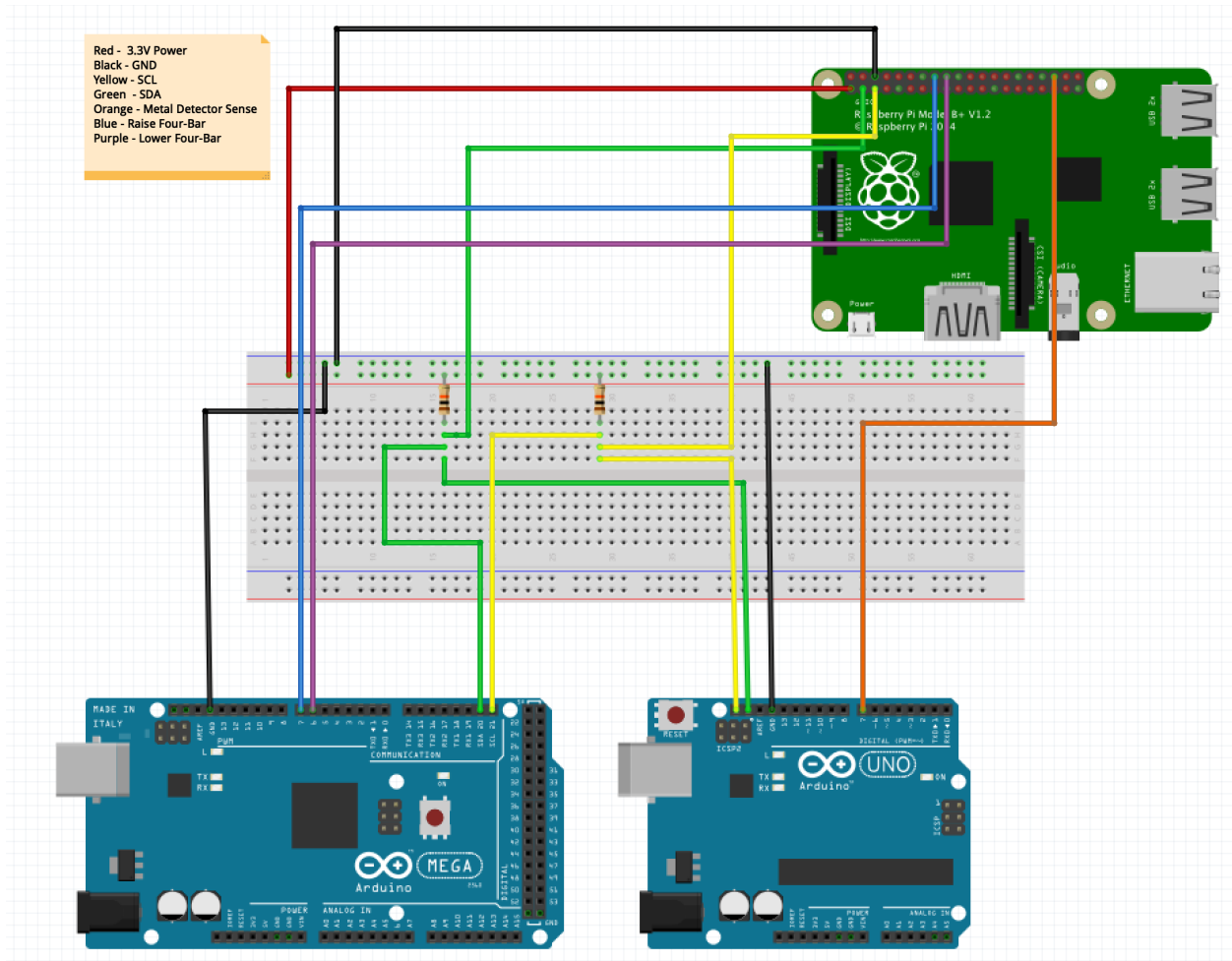


Figure 3.9: Wiring diagram for communication between the Arduinos and RaspBerry Pi

Messages that can be exchanged over IIC are limited to one byte to decrease the time needed to communicate as IIC communication is an interrupt that will halt both Arduinos from executing their main functions. Each Arduino has a state variable and a list of executable commands. The Arduino Mega can be instructed to "Start" and "Stop" adjusting orientation of the metal detector coil, "Home Orientation" of the metal detector coil, meaning that it will rotate the joints of the sensor platform to the home position, "Mark Landmine" or do nothing in response to an unknown command. The Arduino Uno can "Start" and "Stop" reading the metal detector, "Zero Metal Detector" meaning it will try to tune the metal detector to filter out noisy readings, and do nothing in response to an unknown command. The Raspberry Pi can also request a status update from each of the Arduinos so that the main control software will know what each device is doing. The Arduino Mega can be in either the "running", "stopped", "HomingCoil", "ErrorHomingCoil", "Error Motor Stall", "MarkingLandmine", Or "CommandUnknown" state. The Arduino

Uno can either be in the "Running", "Stopped", "ZeroingMetalDetector", "ErrorZeroMD" or "CommandUnknown" state. These states are set at different points in the main loop which is described in detail in the following sections.

Arduino Mega Main Loop

At start up of the Arduino Mega, it will initialize objects to control the motors and the ultrasonic sensors. It will also establish IIC communication and begin in the stopped state. It will wait for a start command from the Raspberry Pi to begin running. The current sense pins will be polled to check for motor stalling during every loop. A reset-able fuse is being used to cut power to the motors in the event of a stall so when voltage across the fuse is measured to be larger than nearly 0 volts then this will be interpreted as a motor stall has occurred. The robot will enter the "ErrorMotorStall" state which will require the robot to return to the base station to be reset. Next, IIC messages will be processed only if the motors adjusting the roll and yaw of the metal detector coil are not moving. A command to mark a landmine could be sent so it is important that the metal detector coil stops moving before this command is executed to ensure the sensor platform is level while rotating pitch. Once this sequence is complete, the limit switches detecting a rotation limit will be polled and if one limit switch is pressed, the Mega will signal the Raspberry Pi that a steep slope has been encountered. Next the Arduino will attempt to adjust the metal detector coil to be parallel with the ground if 200 milliseconds have passed since the last time this was done. This means that adjustments will be made 5 times per second. If an adjustment is needed, the main loop will set the motors to rotate in the appropriate direction and for the appropriate number of encoder ticks. An ISR triggered by the input of the encoder is called to process this movement. This ISR is discussed below. The flowchart of the Arduino Mega main loop can be found in figure 3.10.

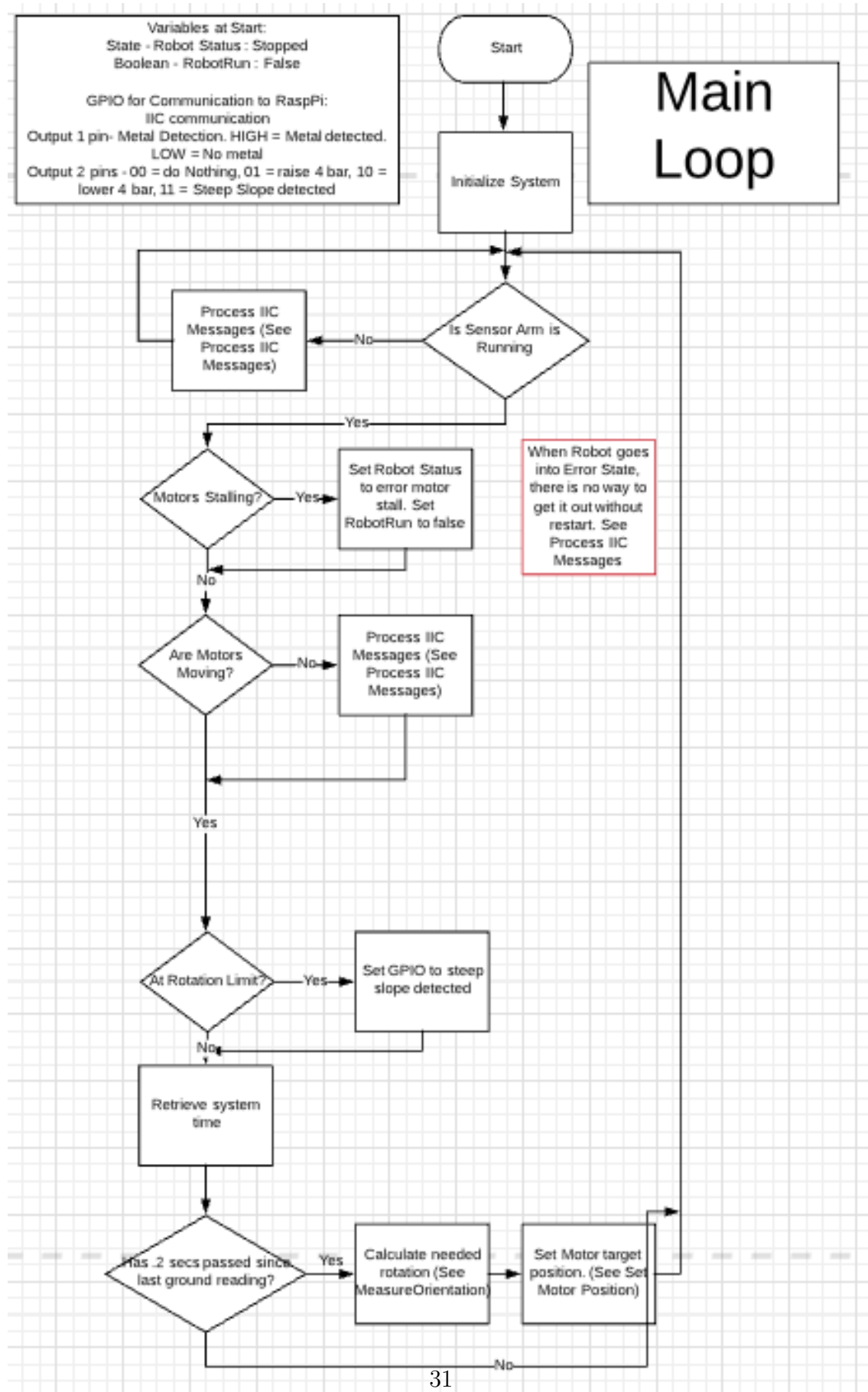


Figure 3.10: Flowchart of Arduino Mega main loop

Arduino Uno Main Loop

The Arduino Uno main loop begins by creating the objects that create the ISR for detecting readings from the metal detector and begins IIC communication. It begins in the "Stopped" state and waits for the Raspberry Pi to tell it to begin. While in the "Running" state, The Arduino then processes the data retrieved from the metal detector every 20ms, and checks for IIC commands every 2 seconds. It checks for IIC commands so in-frequently because communicating over IIC disables interrupts and could potentially miss metal detection readings. The Arduino uno can only be told to stop detecting or zero the metal detector, which are not critical commands. The RaspBerry Pi should only command the Arduino Uno to zero itself at start up and stop detecting once the mission is complete so there is not much reason to check for messages often.

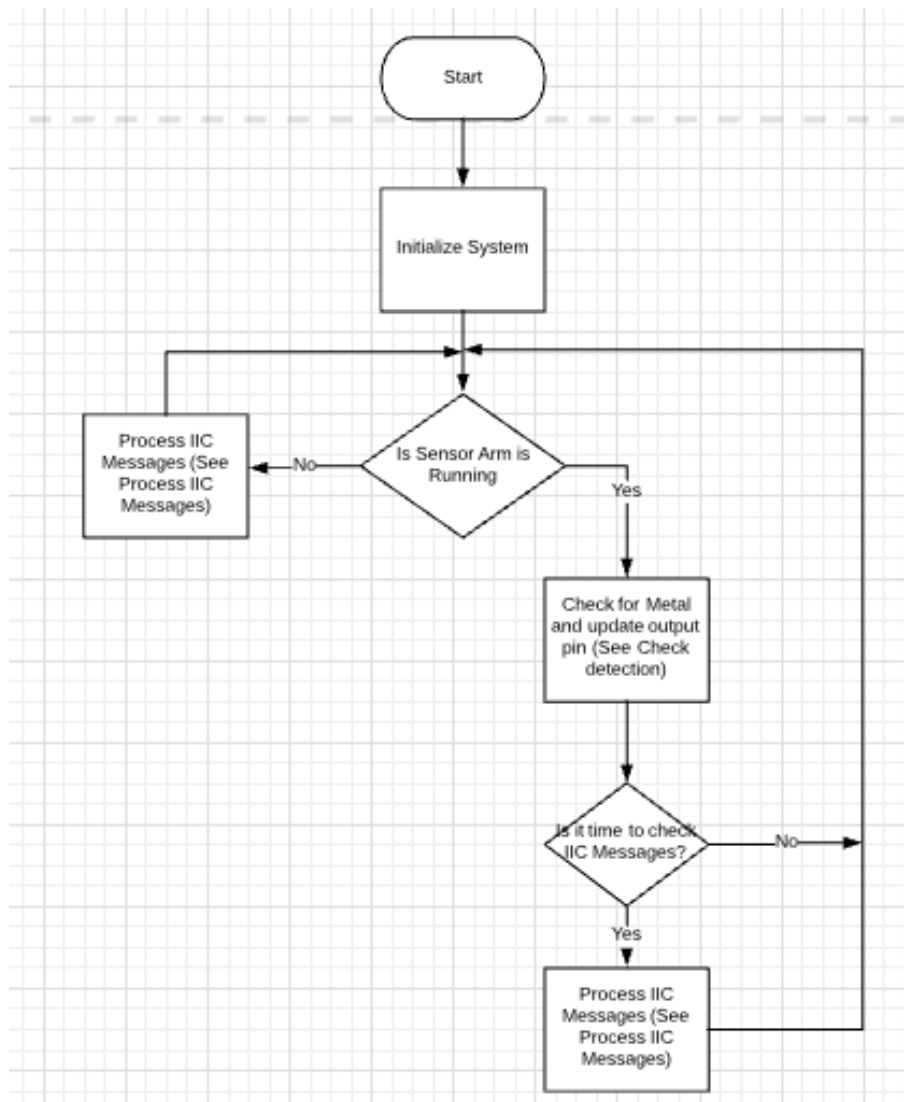


Figure 3.11: Flowchart of Arduino Uno main loop

IIC ISRs

There are two ISRs that each Arduino has to process IIC messages from the Raspberry Pi. The first is the `onReceive()` ISR which is called when a message is received. The ISR will store the message in a FIFO mailbox with a capacity of 10 messages. The Raspberry Pi will likely not send commands frequently enough to fill this but using a FIFO mailbox will prevent shared data issues when the Processing IIC Messages helper function is retrieving a command and gets interrupted by this ISR. The second ISR is the `onRequest()` ISR. Whenever the Arduino receives the request command, instead of storing the message into a FIFO it will immediately report the system status. Flowcharts of these two ISR can be found in figure 3.12

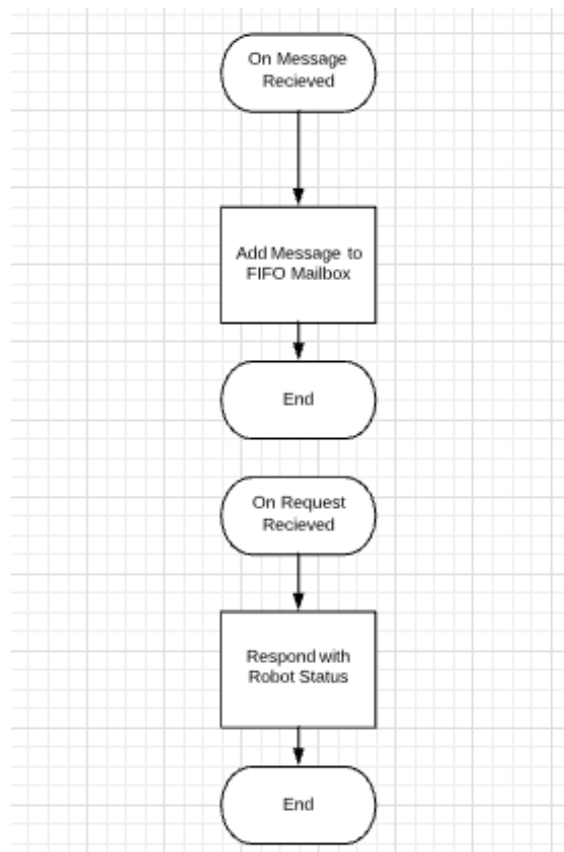


Figure 3.12: Flowchart of the IIC ISRs

Processing IIC Messages Helper Function

This helper function was designed to reduce clutter in the main loop. When a message is processed, the main loop will retrieve this message stored in the command buffer and pass it to this helper function. If the sensor platform is in an error state, it will ignore the command. If the state is ever an error, the sensor platform must be checked over at

the base station because the robot lacks enough sensors to be able to diagnose or correct any problems beyond knowing that either a motor stalled or the metal detector could not remove noise. If the sensors that detected the problem no longer report an issue it is still possible something broke making the sensor platform malfunction for the remainder of the mission. If the robot is not in an error state it will execute the appropriate code. It will also update the robot status variable so that the Raspberry Pi can retrieve this and know what each Arduino is currently doing. Each command and the states associated with it are detailed in figure 3.13.

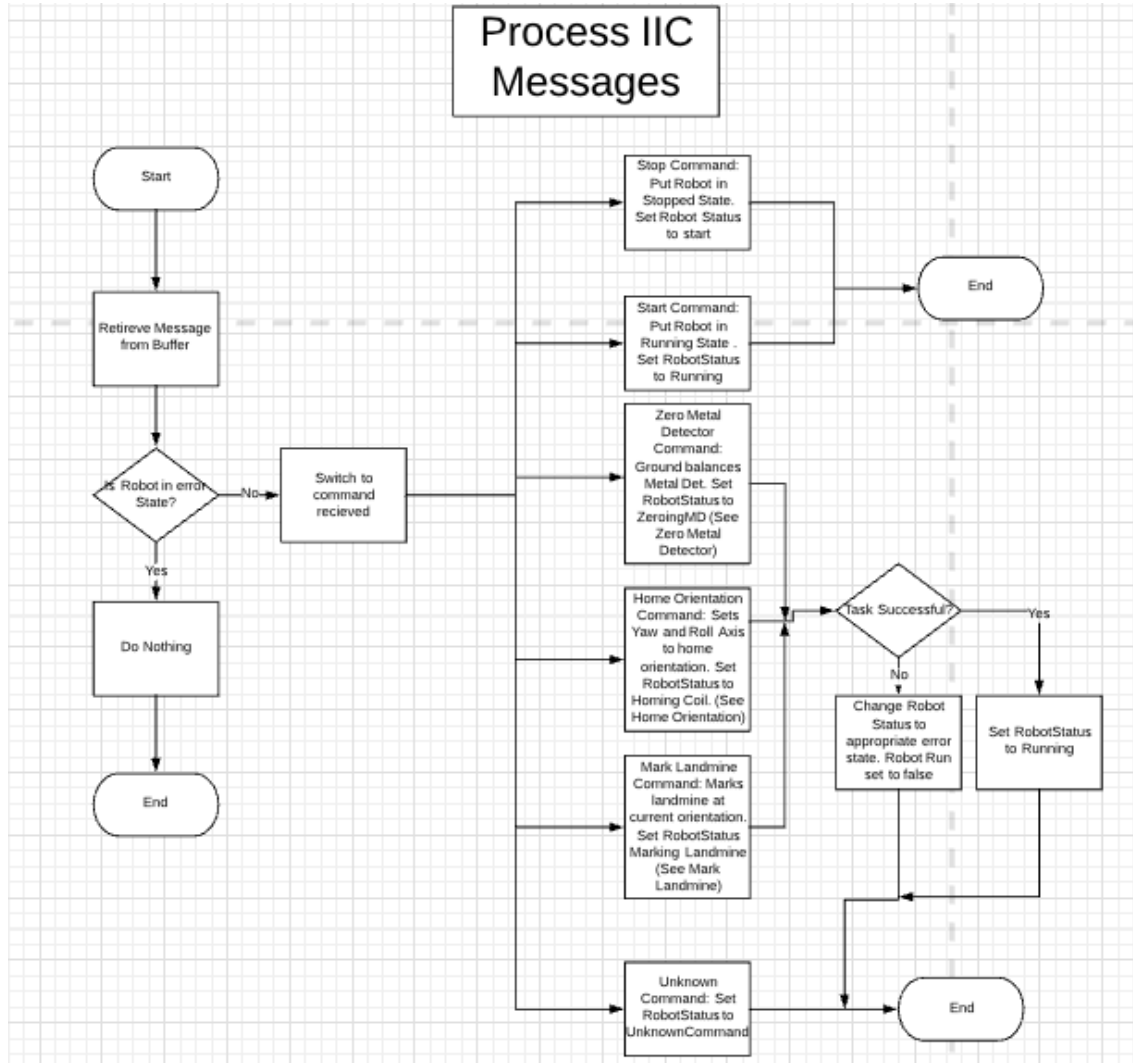


Figure 3.13: Flowchart of the process IIC messages helper function

Metal Detector ISR

The metal detector ISR is only run on the Arduino Uno. This is a simple ISR that increments a count every time a pulse from the metal detector PCB is received. The

metal detector PCB outputs a 555Hz PWM wave whenever metal is detected so the goal is to count the number of pulses so that they can be processed in the main loop to determine if metal is detected or not. Figure 3.14 shows the flowchart of this ISR. Figure 3.15 shows the oscilloscope measurement of the output waveform. The oscilloscope was paused but this does not pause the frequency calculation of the oscilloscope so the frequency was measured using the time axis.

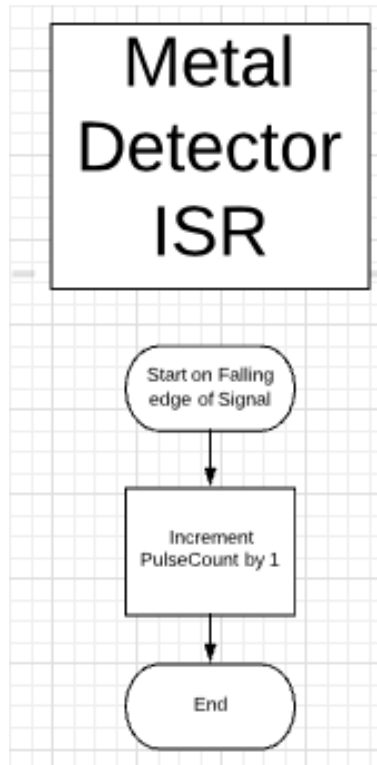


Figure 3.14: Flowchart of the metal detector ISR

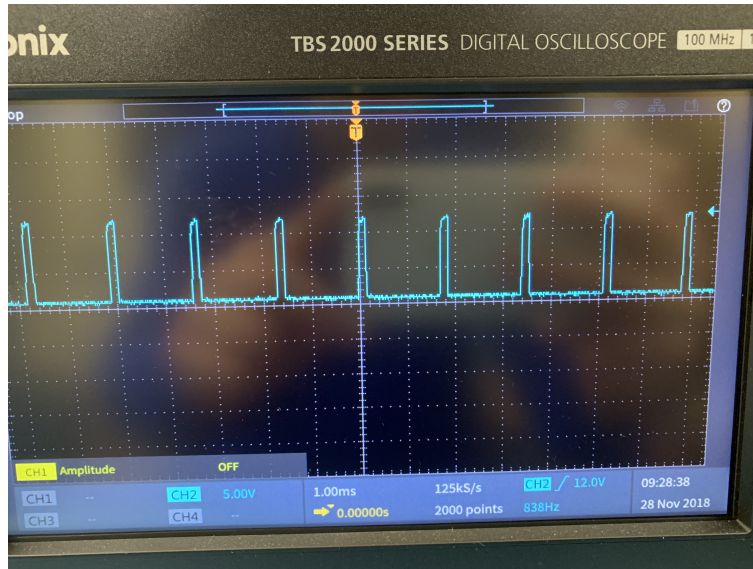


Figure 3.15: Oscilloscope measurement of PWM waveform from metal detector PCB

Check Metal Detection Helper Function

This helper function is called every main loop in the Arduino Uno to check the metal detector. It will only process the pulse count once every 20ms though. This is to allow pulse count time to accumulate. Currently, in order for metal to be considered detected, at least 5 pulses from the metal detector need to be received in the 20ms. If any less than that are received then the metal is considered not detected. This is to remove potential stray pulses from the metal detector although they have never been observed to occur. Typically, if metal is detected, the PWM wave from the metal detector PCB will be generated for at least a quarter of a second when metal is moved through the magnetic field so any real readings of metal should be obvious to this helper function. When metal is detected it will change the output pin to the Raspberry Pi to be HIGH or LOW if metal is not detected. This means that the raspberry Pi will get updated 50 times per second on whether metal has been detected or not.

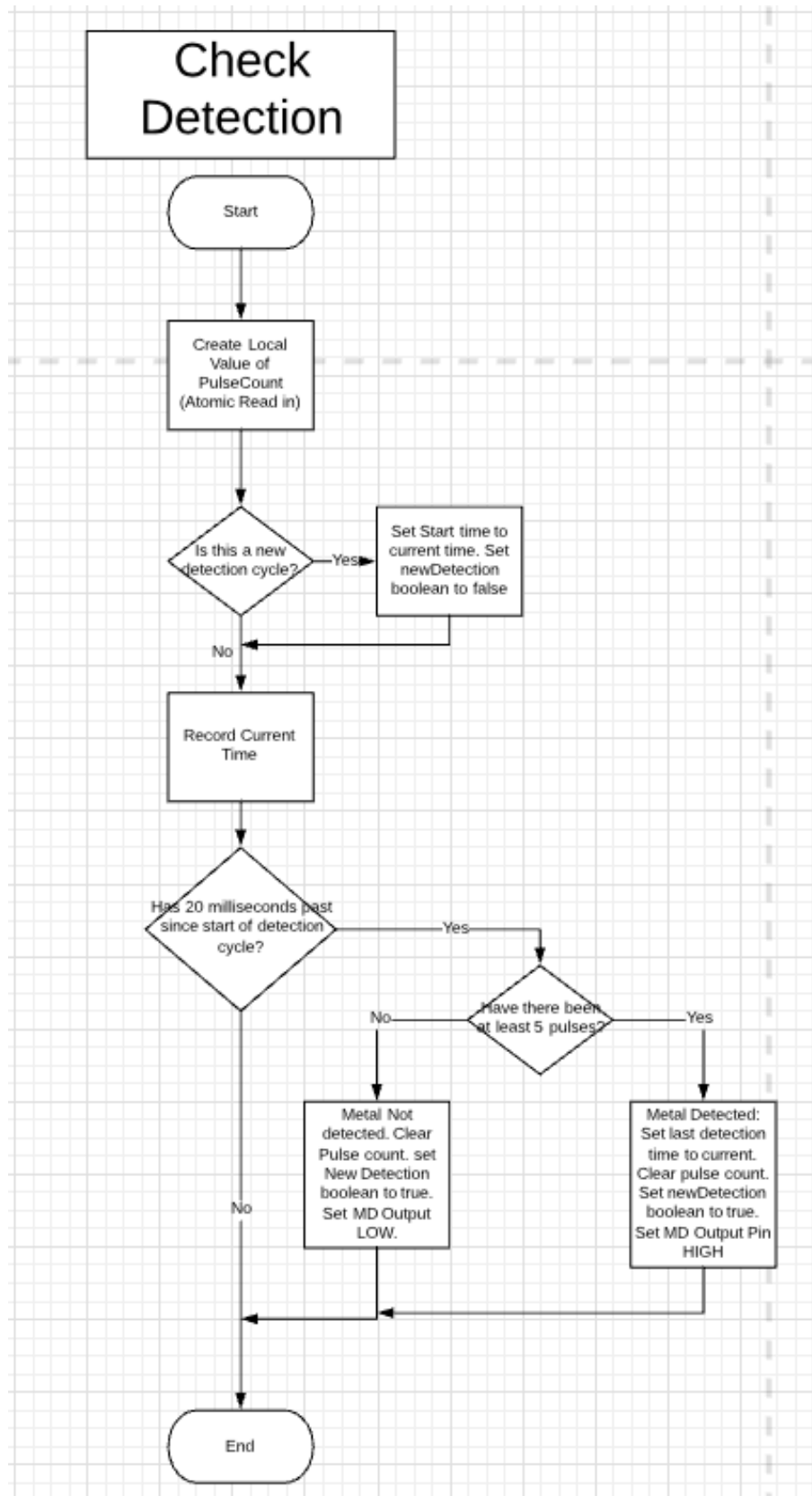


Figure 3.16: Flowchart of the check metal detector helper function

Zero Metal Detector Helper Function

This helper function is only run on the Arduino Uno when the Raspberry Pi requests it. This function will attempt to remove noise from the metal detector, caused by the robot or the ground, by adjusting the metal detector threshold potentiometer, which is an IIC device. In this instance, the Arduino Uno is the master for the digital potentiometer. It will increment the potentiometer resistance until the metal detector stops detecting the noise. If the metal detector has not detected metal in five seconds, it is considered zeroed and the function ends. If the helper function has gone through all of the potentiometer's 128 possible values and could not zero the metal detector, then there is either a problem with the electronics or too much metal in the ground for this system to be used. Another exit condition is if the metal detector has been running for over 30 seconds (subject to be increased with testing). If the process of zeroing has gone on for longer than this time period then this function will end and report zeroing unsuccessful. This is to protect against cases where there is some interference that is periodic with long breaks of not detecting anything, which was observed in testing when there is a lot of metal interfering with the metal detector. A flow chart of this helper function can be found in 3.17.

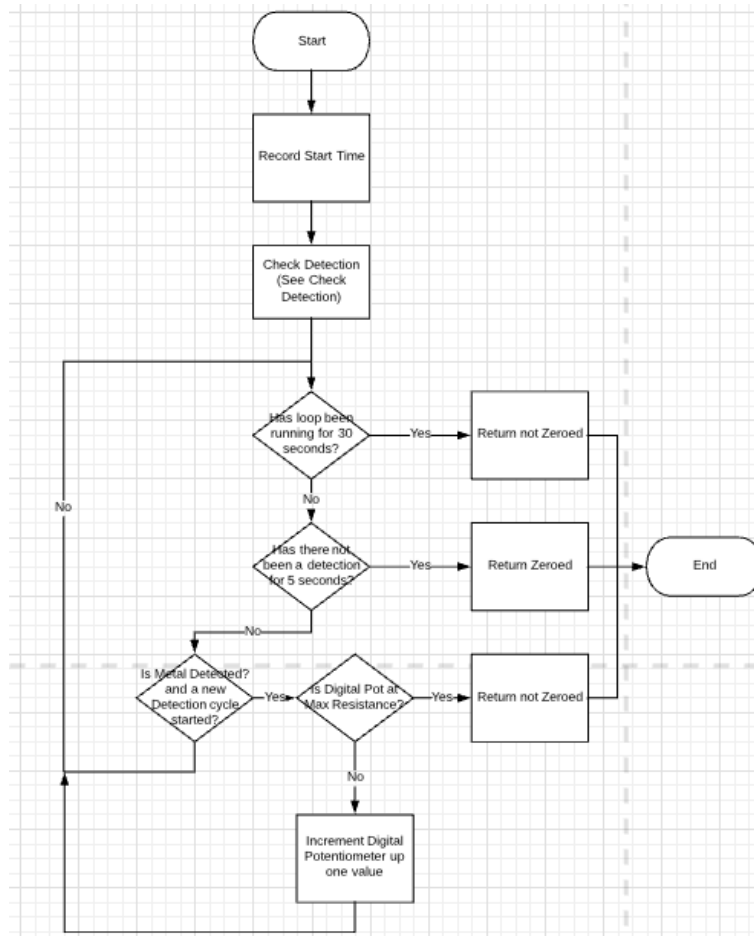


Figure 3.17: Flowchart of the zero metal detector helper function

Measure Orientation Helper Function

One of the main tasks of the main loop of the Arduino Mega is to check the orientation of the metal detector coil and calculate the changes in yaw and roll needed to keep the coil parallel with and one inch above the ground. The first task is to ping the five ultrasonic sensors three times each and add this data to a buffer. Multiple readings are taken and averaged to remove noise. The execution time to ping each sensor three times can take as long as 70ms, which is one of the main reasons why the metal detector functions needed to be offloaded to another Arduino since the check metal detection helper function would have significant latency every time the orientation is measured. The three readings from each sensor are then averaged. Each sensor has a set height that it will want to remain above the ground to prevent collisions of the metal detector coil enclosure with the ground. The change in distance required to meet the desired height at each sensor is calculated. This data is then processed to ensure that the change in height from one sensor will not cause another sensor to become too close to the ground. If the sensor arm would be too close to the ground in order to be parallel, then this function will change the raise four bar pin to HIGH and limit the calculated rotation to try and avoid a collision. Once the four bar is raised the metal detector coil will be correctly adjusted. If the sensor arm is too high above the ground after the rotation, the lower four bar pin will be changed to HIGH. Each pin is brought back low again once the height is optimal again. The changes in rotation angle of roll and yaw are then calculated and outputted to global variables for further processing. It should also be noted that the ultrasonic sensors are only accurate to 1cm and all calculations are done with unsigned integers to avoid floating point math and trigonometry. Observing the source code and all possible height readings that can be calculated shows that there are only 21 possible angle changes that this helper function can calculate for yaw and roll. This level of accuracy will be enough for purposes of keeping the coil level to the ground within a total rotation area of 60 degrees. The flowchart of this helper function can be found in figure 3.18

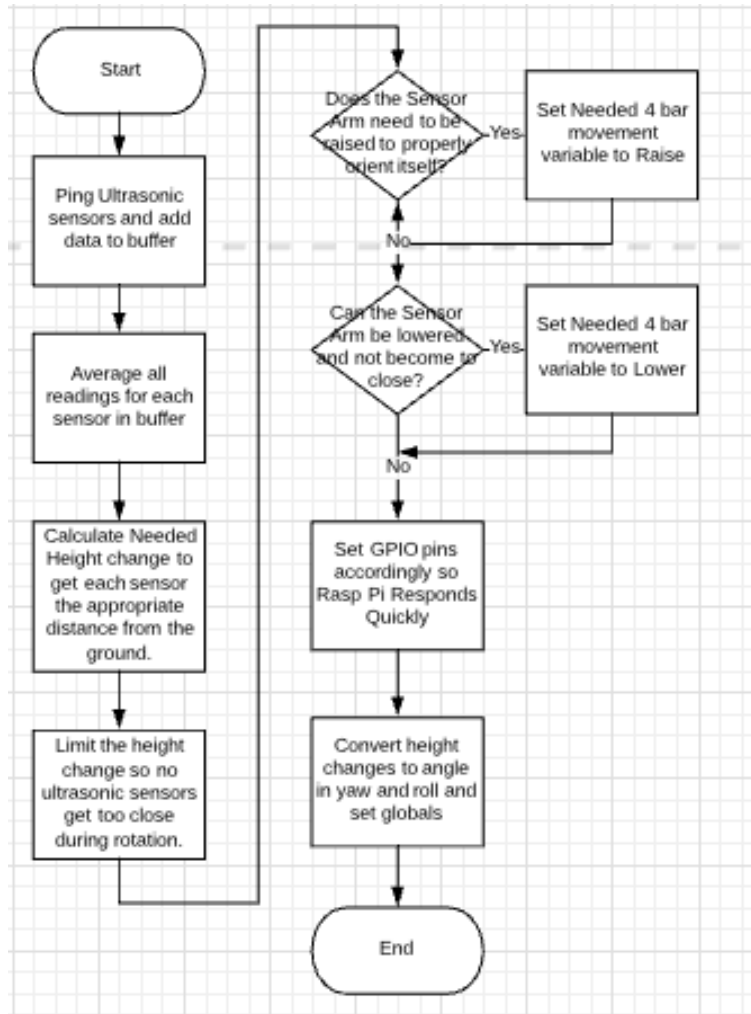


Figure 3.18: Flowchart of the measure orientation helper function

Set Motors Helper Function

The set motors helper function is meant to set the motors into motion in the necessary direction to make the metal detector coil parallel with the ground. This function takes in the angle roll and yaw the coil must be rotated by as arguments. It will determine which direction each motor must rotate based on the sign of each angle. It will then set a value that stores the number of encoder ticks the motor must rotate. This value is limited to the amount of ticks possible for the motor to trigger in 0.2 seconds. This was done because the motors will get set once every 0.2 seconds, and if the main loop ever gets caught up on something, then the motors should stop after 0.2 seconds. How the motors are stopped is covered in the next section. Once the direction and ticks to move are calculated, the motor pins are set appropriately to set each motor in motion. Figure 3.19 shows a flowchart of this process.

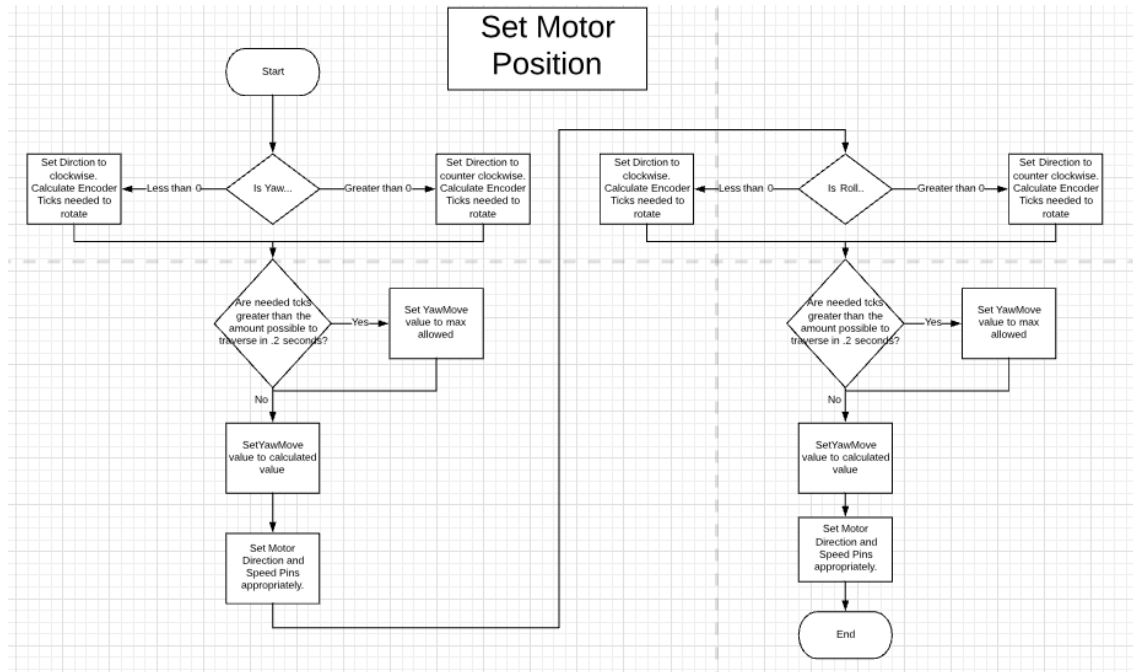


Figure 3.19: Flowchart of the set motors helper function

Yaw and Roll Motor ISRs

The Yaw and Roll Motor ISRs contain the same set of instructions but they are just controlling their respective motor. Every time a pulse from the encoder output of the motor is detected, the ISR will run. This ISR will check to see if a limit switch has been pressed in the direction that movement is intended and if it is pressed, the ISR will remove power from the motor and set the ticks left to move to zero. If the specified limit switch is not pressed, the ISR will update the overall encoder position and decrement the ticks left to move variable for that motor. If the ticks left to move reaches zero then the motor is stopped. This ISR ensures that the motors will always stop even if the main loop gets stuck inside a loop. A flowchart of this helper function can be found in figure 3.20.

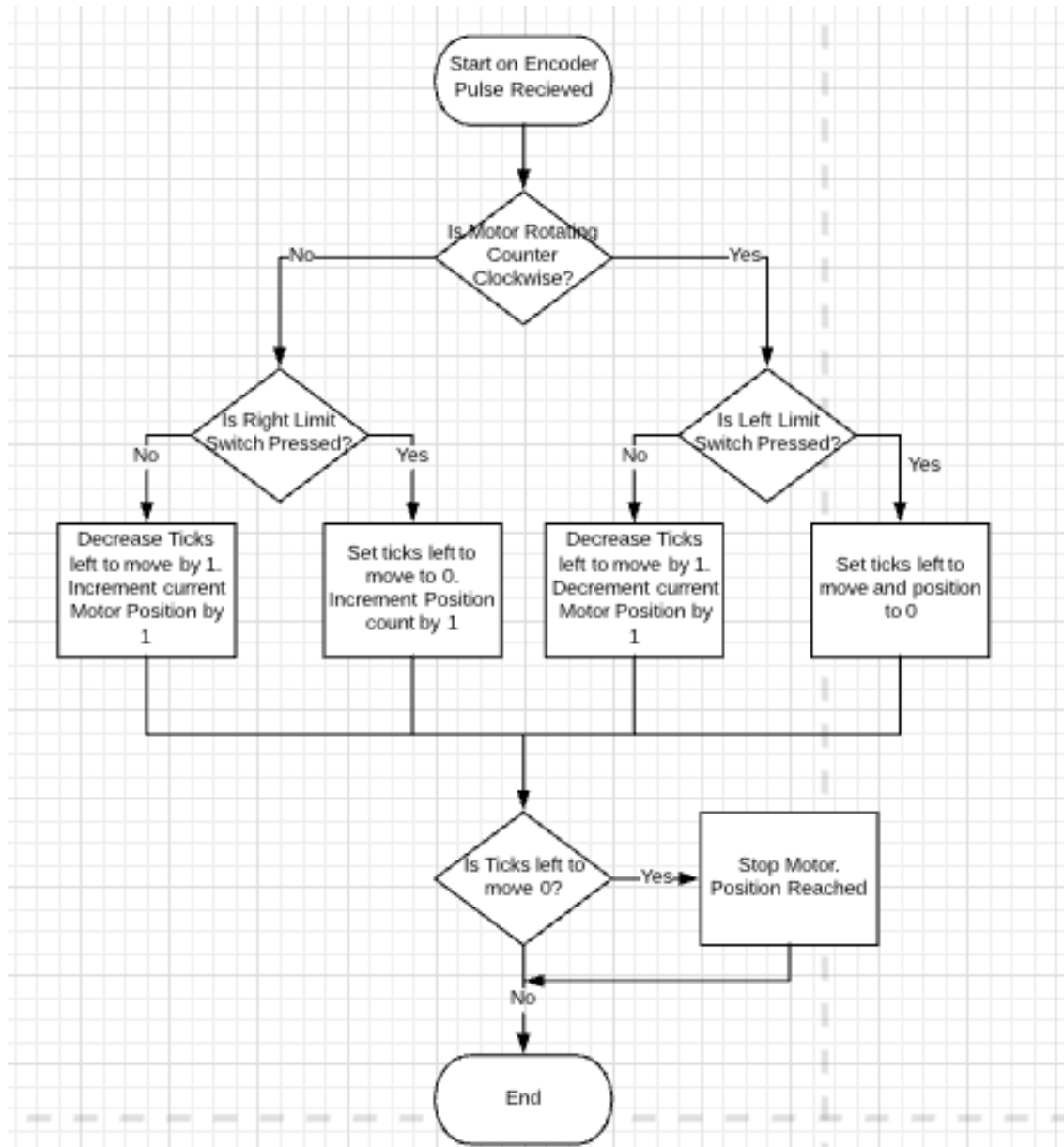


Figure 3.20: Flowchart of the motor ISRs

Home Orientation Helper Function

The home orientation helper function is meant to set each of the the yaw and roll motors to their respective home position so the encoders know where 0 is. This function will set both motors in position counter clockwise. Once they press their respective limit switches, power is removed and the encoder values are set to zero. For safety this function also checks for motor stall and will report if motor stall occurs. The flowchart for this function can be found in 3.21.

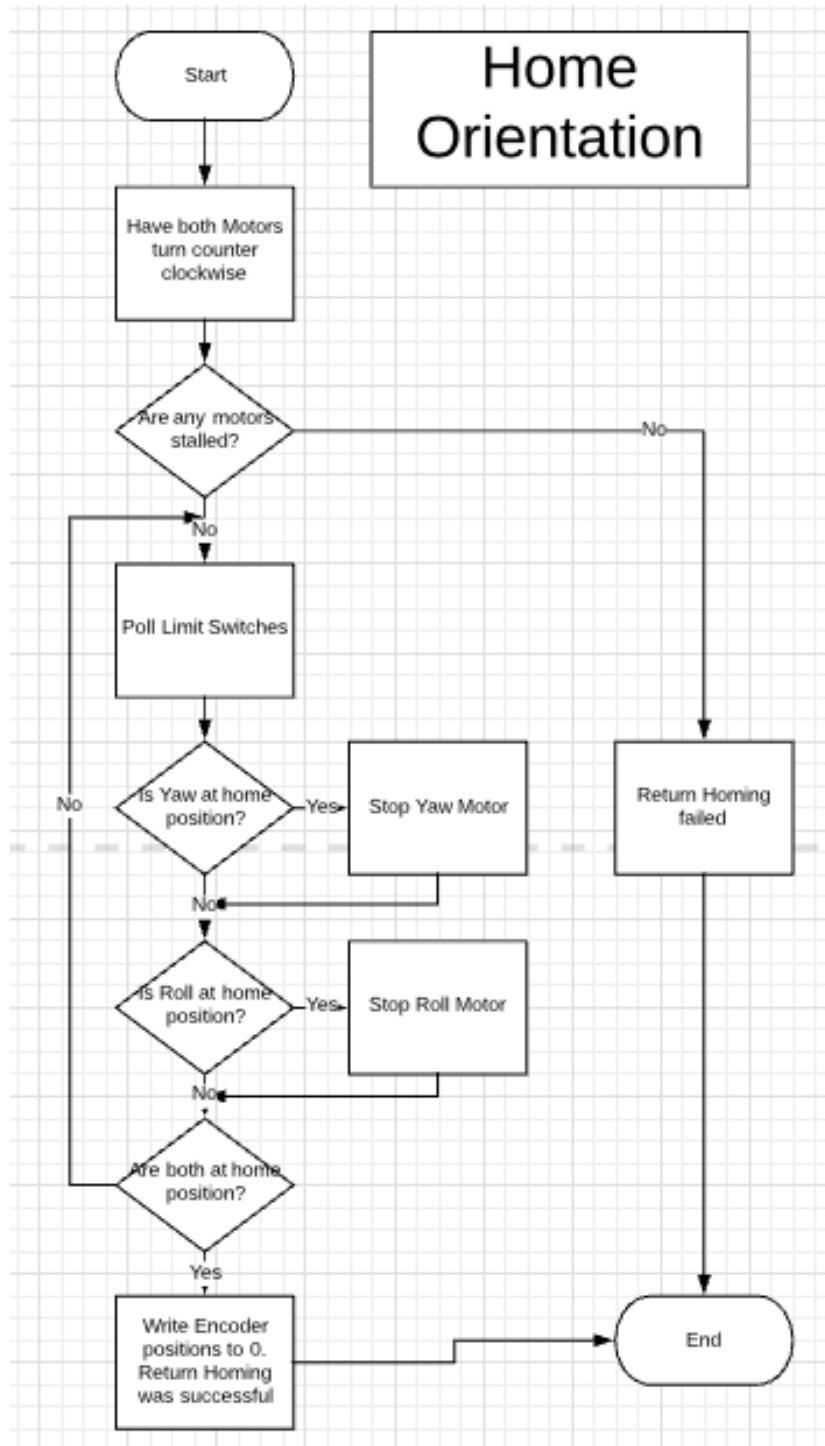


Figure 3.21: Flowchart of the home orientation ISR

Mark Landmine Helper Function

The last main helper function is the Mark Landmine function. This is run on the Arduino Mega when the command to mark a landmine is received from the Raspberry Pi. This function will rotate the marking system out so that it is over where the metal detector had detected metal. If motor stalling did not occur during this process, the servo actuating the spray can will rotate to spray the paint. The sensor platform will then rotate back and forth to create a square marking. The field marking spray can sprays in a line so rotating about the roll axis will draw a square. Once this process is complete, the spray can nozzle is released so paint stops spraying and the motor controlling pitch will rotate back to the original position. If a motor stalled then this will report marking unsuccessful because it may be possible that the metal detector is not extended back out. A flow chart of this process can be found in figure 3.22.

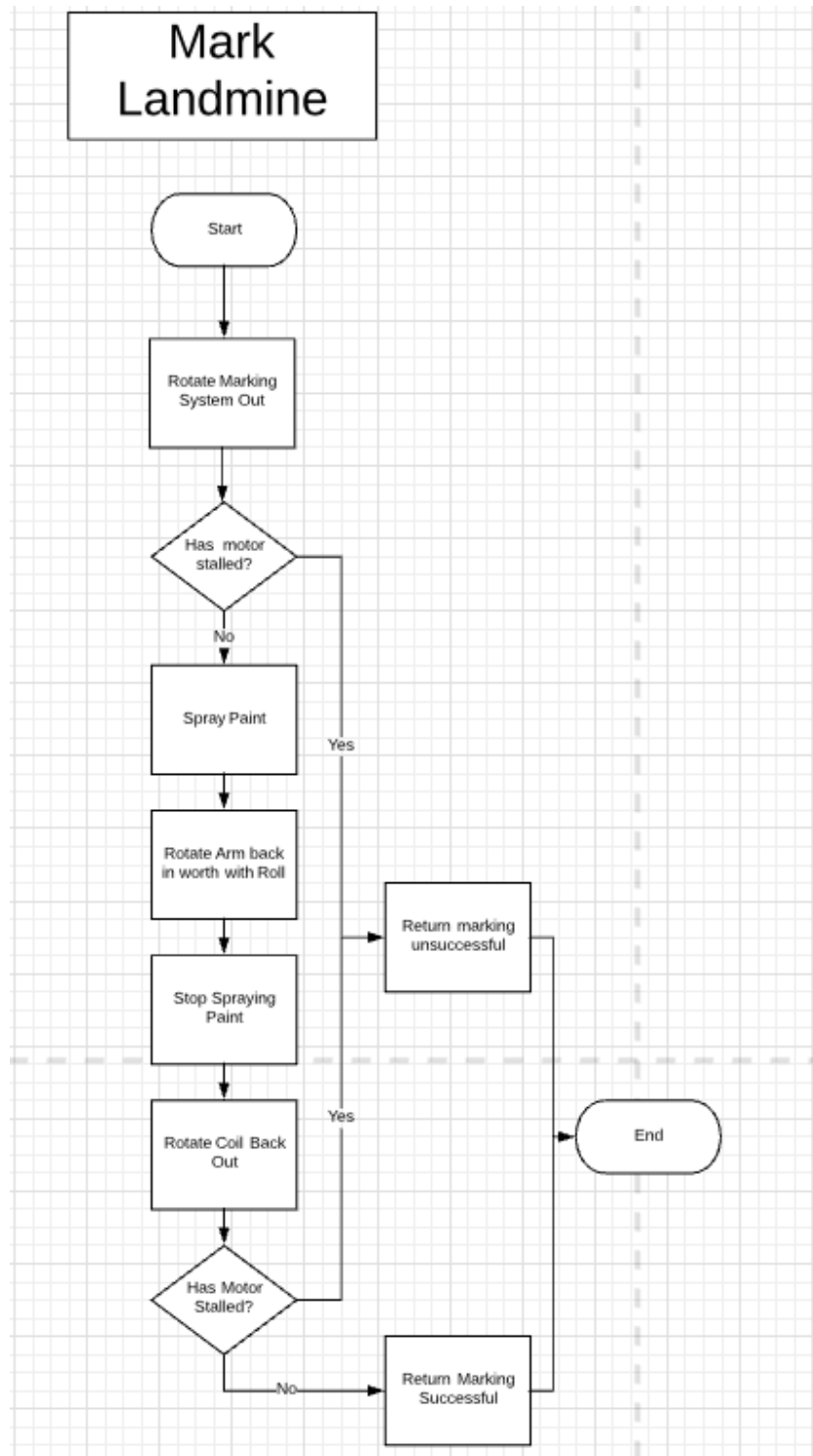


Figure 3.22: Flowchart of the mark landmine helper function

3.1.8 Sensor Platform Testing

This section discusses the testing process for the sensor platform beginning with testing the individual components, followed by testing them working together, and lastly testing while the Raspberry Pi is operating.

Individual Component Testing

Initial testing of the sensor platform began with testing each major function separately to verify that each portion of software was working with their associated hardware. The metal detector was tested first, which consisted of waving metal over the metal detector and viewing the metal detector output pin on the Arduino Uno with an oscilloscope. When metal is being waved over the metal detector coil this pin should be set to output HIGH and when there is no metal it should be set to output LOW. Next the automatic zeroing of the metal detector was tested. This function should consider the coil zeroed when there are no readings for 5 seconds and give up after 1 minute, reporting a failure. Generating noise to mimic metallic soil conditions was not possible so testing was done by waving metal over the coil a few times to make sure the potentiometer was being adjusted to try and remove the readings. After a few attempts the metal was removed to verify that the function does report a successful zeroing. Another trial was done where noise was constantly given to the system and verify that a failure was reported.

Next, the ultrasonic sensors were tested along with the code that calculates the angles to rotate metal detector coil by to become level with the ground. This was done by disconnecting the metal detector coil assembly from the rest of the sensor arm and manually rotating it above a flat surface. The read out of the angles needed to become level was viewed in the serial window and verified. Any errors would then be corrected by ensuring the sensors are wired correctly and then searching for errors in the software. The raise and lower pins were tested to ensure that when the sensor platform was raised too high, the lower pin would be pulled HIGH and when the sensor platform was too close to the ground, the raise pin would be pulled HIGH.

The motor controls were tested beginning with motor direction verification. It was unknown what direction corresponds with a HIGH or LOW output to the direction pin of the motor so each output was tested and then the direction of rotation variables in the software were updated. The limit switches for detecting a movement maximum on the sensor arm were verified to be working by polling their input in a loop and printing to serial. With these two steps completed, the interrupt driver motor control was tested. The Yaw and Roll motors were given a set angle to rotate by and then observed to spin and stop appropriately. The interrupt service routine for the motor encoders is the only way to stop the motors so if the motor stopped spinning and spun the appropriate amount then the software was working. Movement commands that have the motor rotate forward and backward in a loop were used for testing the endstop limit switches. While the motor was spinning, a limit switch would

be pressed by hand to check that the motor does not move in that direction. This was done for all directions. Then the motors were given commands to intentionally reach the endstop to verify that the motors are turned off while the button is pressed instead of causing motor stall.

The pitch motor was also tested but this motor is not controlled with an interrupt. Since this motor only needs to rotate the metal detector and marking system between two positions, the motor is set to run and then stop once the appropriate limit switch is pressed. The motor was disconnected from the drive train, set to move and then the limit switch was manually pressed to ensure the motor stops in the software. This was done for both positions. The last motor component is the servo control to actuate the paint can. The positions associated with pressing and releasing the spray can were found through trial and error and then set appropriately in the software.

Combined Component Testing

The next step in testing was to combine the individual parts into the loop sequences described previously. The angles calculated by the ultrasonic sensors were passed into the motor controller code to test how capable the auto orientation function was able to keep the metal detector parallel to the ground. This was tested on a dense flat lab table, the floor and cardboard. The lab table and floor would be ideal surfaces since the sound would easily bounce back while the cardboard was meant to be a soft surface similar to soil.

The marking system was tested with the pitch motor connected to the drive train to ensure that it could turn the lower assembly adequately. With this working, the full sequence for marking a landmine was tested without the spray can so that paint would not be sprayed indoors.

The last step was to run all of the components at once to verify both Arduinos are functioning independently and do not interfere with each other electrically. With all of these systems tested, the sensor platform was ready to be integrated with the raspberry pi and ROS to accept ROS commands.

Full system testing

With the sensor platform working on its own, the last round of testing would be to verify that the Raspberry Pi is able to communicate with both Arduinos over IIC while the infrared camera was also running as a parallel process. The infrared camera and Arduinos use IIC for commands being sent from separate processes (this is further explained in following sections) so testing both of these devices running in parallel was essential to ensure that no collisions on the IIC bus would occur. This test would consist of streaming video from the infrared camera while also trying to send commands, such as "start" or "stop" to the

Arduinos. Additional software was developed to regulate access to the IIC bus so that only one process could access it at once.

The last portion of testing was to test the sensor platform's ability to adjust to changing ground heights while the slider mechanism is moving it (see Section 3.4 for more information on the slider). If the sensor platform is able to adapt fast enough then the auto-orientation feature would be considered a success.

3.2 Rover Drivetrain

An integral portion of any ground-based system is the drivetrain. As the primary source of locomotion, the drivetrain is responsible for physically moving the system from one point to another. It is also responsible for ensuring the stability of the main chassis during locomotion. Given that there are a variety of terrains to traverse and that ground-based systems can have a plethora of functionalities and weight distributions, there are a variety of drivetrain systems that development teams must consider, including shock suspension and Rocker-Bogie systems as well as treads and non-pneumatic tires.

3.2.1 Common Drivetrain Designs

One of the most common systems seen in ground-based systems is the shock suspension system. This system serves to reduce extraneous vehicle movement and the associated fatigue effects by combining spring and damper subsystems together. The spring subsystem ensures that the wheel maintains a solid contact with the ground by providing a downward force. Meanwhile the damper subsystem ensures that the spring does not expand too quickly after compression. Without this damping effect, the potential energy generated by the spring compression would be dispersed after several successive expansion-compression cycles, resulting in instability for the main chassis. With larger vehicles, such as cars and trucks, there are two main styles of suspension systems that can be used on the front and rear axles: dependent and independent. With the former, a fixed axle connects pairs of wheels on each side of the vehicle while the latter has a separate axle for each wheel, enabling self contained movement. Dependent systems are typically seen on larger, heavier vehicles such as trucks or trailers, while independent systems are more commonplace on smaller vehicles such as cars and robots. Independent systems allow for one wheel to encounter and deal with obstacles without seriously impacting the orientation of the main chassis or affecting the performance of other wheels attached to the frame (Harris, 2005). Given the independent nature, these systems are ideal for robots given that the rest of the suspension system can operate if one wheel or shock absorber system is damaged. Typically, the damper and spring components of this system are combined to save space. This suspension system is highly modifiable, given that there are multiple styles of spring or dampeners, and thus is used across a wide variety of ground-based platforms.

The favoured suspension system of NASA projects, rocker-bogie systems offer advanced maneuverability in more rocky terrains. The rocker-bogie system is made up of six wheels, each independently driven, a differential, and two linkages: the rocker and the bogie. An example of such a system is seen in the image below. The term “rocker” refers to how the larger link on each side of the system rocks with respect to the center of the differential; as one side goes up, the other goes down. This motion, obtained through the use of the differential, helps to keep the main chassis of the vehicle stable and level. The term “bogie” refers to the smaller links on each side of the system that have a driven wheel at each end. The combination of these two links enable a ground-based system such as the

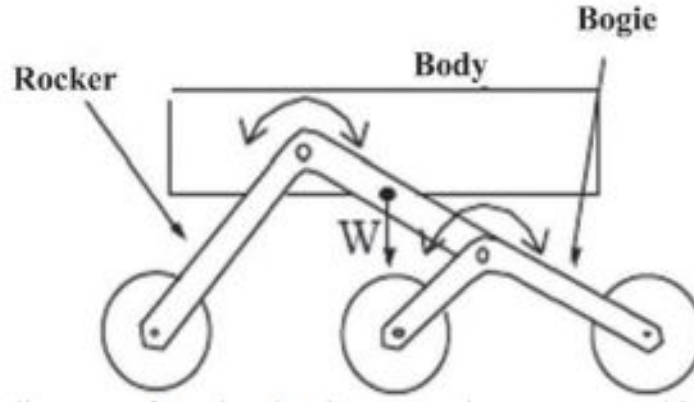


Figure 3.23: Example Rocker-Bogie suspension system

Mars Pathfinder rover to climb obstacles up to two times the wheel diameter and slopes up to fifty degrees. This is done by driving the foremost wheel up by the rear wheels and then dragging the following wheels up by the front wheels. Due to the use of the differential and this climbing cycle, the rocker-bogie system is an ideal suspension system for ground-based vehicles that will be traversing rocky terrain.

Offering a high level of mobility across a wide variety of terrain, continuous track systems, also referred to as tank treads, can be seen on construction, agriculture, and military vehicles alike. This system consists of track runners that are fixed at the sides of the vehicle. The tracks are independently powered on each side. Due to this mechanical restraint, the tracks cannot independently turn left and right relative to the vehicle. Instead, to turn the vehicle, the track speed is limited and sometimes reversed based on the turning radius required. As the magnitude of the speed of one track relative to another increases, the point at which the robot is turning about gets closer to the center of the robot. For example, to turn the robot to the left in a large arc, the left track would be somewhat slower than the right track. To obtain tighter turning radii, the speed of the left track would decrease further until it reaches the same magnitude of the right track, but in the opposite direction. At this stage, the robot would be turning about its center. This style of locomotion is aptly known as “tank drive.”

Given that a large surface area is in contact with the ground at any one point, tracks increase a vehicle’s traction. In addition to this, tracks eliminate gaps or spaces that would be present with a traditional wheeled system, considering that tracks are comprised of a single continuous loop. The continuous loop enables traversal over objects such as logs, sticks, or small rocks that would otherwise get caught in between wheel axles, potentially rendering the vehicle immobile. While the ability for tracks to roll over obstacles prolongs the life of the drivetrain, the rigid structure of the treads causes the vehicle to pitch or roll in response to the obstacle.

A further system which has been implemented by the military and NASA within the last few years is the non-pneumatic tire. As opposed to traditional rubber tires, this type

of tire does not require air to be contained inside of the tire in order to function. This enables the tire to traverse sharp objects such as nails or rocks without compromising its structural integrity or shape. Specifically with the military application of these non-pneumatic tires, a polyurethane membrane is molded in a geometric tessellation to create the necessary support that allows the tire to maintain its shape. The polyurethane membrane is designed in a way such that it elastically deforms under a certain load when encountering an obstacle. This deflection results in a greater surface area coming into contact with the obstacle as it is overtaken by the tire, increasing the traction. Similar to how the shock suspension system reacts to an obstacle, the crumple of an airless tire reduces the disturbance to the rest of the chassis providing smoother locomotion over any rough or unpaved terrain encountered by the vehicle.

3.2.2 Drivetrain Selection

The Clearpath A100 Husky, as renovated by a prior iteration of this project, has a typical drivetrain. Supported on six rubber wheels each with their own independent axle, the Husky drivetrain is powered by two CIM 12-volt DC motors. These motors are connected to each axle via a belt drive. With this system, the Husky is configured with a tank drive system, meaning that the wheels on one side of the rover will all be driven at the same speed. With this configuration, the rover is able to turn in with a range of arcs and even in place, making it ideal for navigating a landmine field with the projected navigation algorithm.

Despite this, the current drivetrain system is not without its faults. Foremost among these, the fixed axles that each hold one wheel transfer any disturbance from the traversed terrain to the rest of the rover platform. This causes inherent instability and rocking for all connected systems to compensate. Further, the design of the wheels, being rubber air-filled tires, are prone to deflation or popping if punctured by a sharp object, which can place the rover in a dangerous position should this occur in the middle of operation. Along with this, the spacing of the axles means that an object such as a tree branch or rock can be caught, impeding the motion of the rover. These facts were taken into consideration when selecting the final drivetrain system

In addition to these faults, the projected conditions the final rover will be able to operate in dictated the selection of the final drivetrain. Given the above faults, the fixed axle-rubber tire drivetrain system was deemed inadequate for the task at hand. From this starting point, the Rocker-Bogie, shock-suspension, and tank tread designs were considered. The Rocker-Bogie system offers the desired amount of maneuverability, given the correct design. In order to appropriately climb six inch obstacles and a 30 degree slope as originally planned, the design of the system would not work with the A100 chassis. While the horizontal distance between the front and rear tire would be similar to the current, fixed axle design, the vertical distance from the wheels to the pivot points would be relatively far above the top of the rover. Further, each wheel of this system would have to be independently driven. These two factors would necessitate extensive modifications to the Husky rover, delaying the implementation

of further systems and straining the limited budget available for this project. The shock suspension system, as discussed previously, would similarly require extensive modification to the rover; each axle would need to be detached from the main chassis, with the shocks for each wheel being attached to the outside of the rover. The tank tread system, on the other hand, would not require as significant of a modification to the current drivetrain. The current wheels and tires would be swapped out for sprockets and tracks. This design would improve traction and load distribution, but does not solve the original stability problem, and is therefore not suitable for the task at hand.

After evaluating these designs, it was decided that the current, fixed axle design would be augmented, instead of replaced, with non-pneumatic tires. The current wheels will be replaced with non-pneumatic tires of equivalent size. A modified tessellation will be designed so that driving over small objects will not impact the rover's orientation significantly. The tires will crumple and recover close to their original shape. They will allow for the rover to travel over sharp objects safely with no risk of inhibiting the rover's ability to continue onward. This will increase the rover's ability to travel over a multitude of terrains.

3.2.3 Development

The patterns that were created will be simulated in ANSYS, a dedicated simulation software, to test for strain and deformation magnitude and directions. One design is pictured below. Applying the load of the robot to each wheel will be done once the final designs are completed and the final weight is calculated. For now, the applied force will be large and out of proportion. This is to visualize the deformation of the pattern that makes up the tire's wall.

The preferred method of creating the designs that were drafted is injection molding polyurethane using a mold. This method is very effective, however, it becomes a very expensive process. With a limited budget, the plan is to 3D print the tires. Further testing must be done in order to effectively and accurately print flexible material in the correct pattern.

A few parameters need to be figured out which is an ongoing challenge faced by the team and a faculty member that knows how to use ANSYS. Once the simulation runs correctly, the pattern can be altered in terms of shapes involved in the tessellation, and wall thickness of the spokes. Many solutions can be found for this issue of what the pattern should look like. Simulations will be used as the primary justification of the design pattern.

The material will be decided upon after the simulation data in ANSYS completes a successful run. This will help the team decide whether a stiff or a more flexible material is needed.

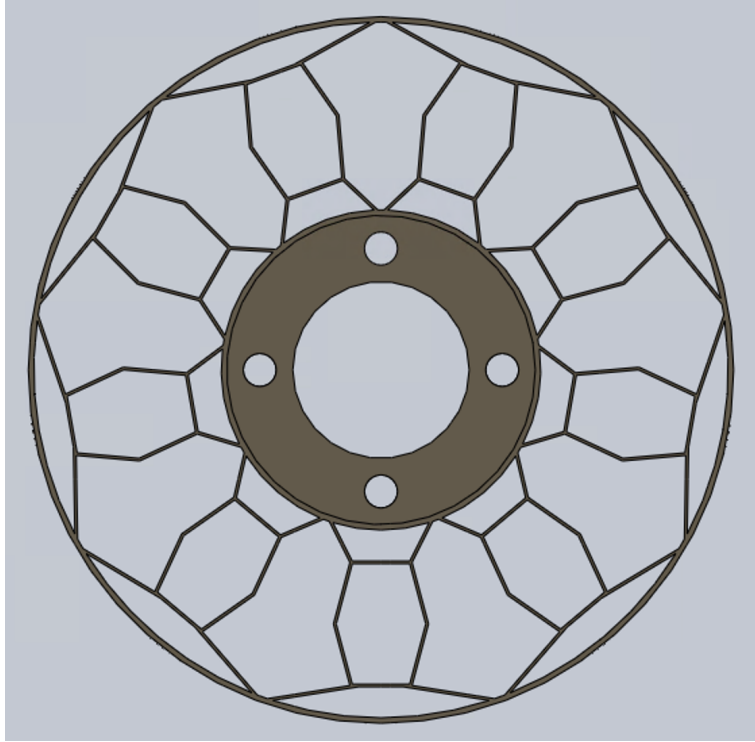


Figure 3.24: Airless tire design

3.3 Four-Bar Lifting Subsystem

Another major component of this project is the support mechanism for the sensor platform. In order to avoid irreparable damage to the Clearpath Husky due to an unplanned explosion, a four-bar linkage system was developed. The length of this mechanism allowed the primary metal detector to scan approximately two feet in front of the rover, mitigating this danger. Comprised of polycarbonate links and powered by two Tetrix TorqueNADO motors, the four-bar system provides the primary vertical locomotion to overcome small slopes and obstacles that may be encountered during normal operation. The following section details the development of the linkages and drive system.

3.3.1 Previous Four-Bar Mechanism

The previous iteration of this project implemented a comparable four-bar mechanism. Similarly concerned with the danger to the robot, the previous system consisted of two parallel four-bar linkages of two-foot long links. The links, which connected the robot to the sensing platform, were made of cedar wood and powered by two independent GB37Y3530-50EN DC motors. Cedar wood was chosen as the material for the links due to its rigid, yet flexible nature and that, similar to the motors, it was the most cost efficient choice. The motors were then augmented with a 1:39.27 gearbox, amplifying the initial 4.4

Newtons of output from the motor to approximately 173 Newtons, more than exceeding the force necessary for the sensor platform at that time. The use of two separate motors in this configuration allowed the sensor platform some limited range of roll, enabling the metal detector to deal with small disturbances in the scanned terrain. (Casey & Rocks, 2018)

While the four-bar system developed for the previous iteration suited the task at hand, it was deemed inadequate for the updated system. Since the disparate components of the system described herein were developed in parallel, an estimated weight of approximately 16 pounds, or 50 Newtons, was used during the design process. This increase in weight over that used for the previous iteration prompted several necessary improvements including a more appropriate material for the linkages.

3.3.2 Link Selection

The foremost step in the development of this mechanism was deciding the appropriate dimensions and material for the linkages. A MATLAB script was developed to generate data on possible configurations. This script iteratively combined support heights and link lengths ranging from zero to twelve inches and from 18 to 36 inches, respectively, each in quarter inch increments. With a given configuration, the “rest” and “rise” angles were then calculated. The “rest” angle corresponds to the angle relative to the horizontal at the top of the rover that the links would be positioned when scanning flat, level ground. The “rise” angle corresponds to the angle the links need to go to in order to effectively scan a thirty degree slope. After running the script, the nearly 4000 test cases were filtered using Excel macros to determine optimal conditions. This filtering ruled out any test cases in which the value for one or both of the angles was negative. Raising the mechanism from a negative angle such as these meant that the metal detector would move forward, away from the robot, until the mechanism passed horizontal. These conditions were ruled out as the sensor arm could potentially be forced into an unseen obstacle in front of the rover. Further, the filtering procedure ruled out excessively high “rise” angles, which were designated as any angle value greater than approximately 75 degrees. These conditions would lead either to the metal detector colliding with the front of the rover or to the metal on the rover causing interference during scanning.

There were several concerns to take into account when deciding the final length of the four-bar and support height from the final 400 test cases. Foremost among these concerned the weight distribution over the ground. A 2009 study at the University of Nebraska found that a 17,000 pound trailer produced only 20 psi of pressure at a distance of two feet from the end of the tire (Olmstead, Fischer). Given that the Clearpath Husky A100 rover is only 90 pounds, this pressure is greatly reduced, well below the pressure threshold for the PMN-1 mine at the same distance. With that in mind, the final test cases were manually parsed to remove any test cases with a link length less than 20 inches. Similarly, any test case with a link length of over 30 inches was eliminated. These cases were eliminated due to torque considerations. As the link becomes longer, the necessary torque output from the

gearbox increases proportionally. In order to avoid an excessively large and complex gearbox or overly strong motor, configuration considerations were kept to the 24- to 30-inch range. The final manual parsing stage eliminated any test case with a “rest” angle close to zero or a very high support height. The former criterion was used in order to avoid needing maximum torque output from the motor, while the latter was used in order to reduce the cost of the entire mechanism. From this elimination procedure, an approximately 22-inch link with a 2.5-inch support was chosen. This support was later increased to three inches after stress analysis of the system showed a potential 0.3-inch deflection at the end of the link. If the system was left unaltered, such a deflection would result in the metal detector scraping the ground and not scanning effectively.

After determining the length of the linkages, an appropriate material was selected. Initially, oak or maple woods were considered given their sturdy, unbending nature and overall cost efficiency. However, polycarbonate and polyoxymethylene - otherwise known as Lexan and Delrin, respectively - were investigated due to their versatility and durability. Polycarbonate was chosen as it was found to be more resistant to deformation and bending than polyoxymethylene. This characteristic was then used to determine the size of each link. Using the below formula, the deflection of one link was calculated:

$$\delta = \frac{F * L^3}{3 * E * I} \quad (3.4)$$

In the above equation, the force, F, was set to the full eleven pounds. This was done in order to ensure that the four-bar mechanism would remain in place and not break. In reality, this weight would be distributed somewhat between all four links in the mechanism. Initially, the link- being $\frac{1}{2}$ inch by 1 inch by 24 inches - deflected by approximately 0.5 inches under this load. The dimensions were subsequently altered to be $\frac{3}{4}$ inch by 2 inches by 24 inches. This new configuration resulted in a deflection of 0.3 inches. Greater changes in the thickness or width of the link would have subsequently increased the cost of the piece as well as the difficulty to machine it, and thus the aforementioned dimensions were used.

Deflection (in)		Thickness (in)						
		0.5	0.75	1	1.25	1.5	1.75	2
Width (in)	0.125	4.19E+00	1.24E+00	5.24E-01	2.68E-01	1.55E-01	9.78E-02	6.55E-02
	0.25	2.10E+00	6.21E-01	2.62E-01	1.34E-01	7.77E-02	4.89E-02	3.28E-02
	0.375	1.40E+00	4.14E-01	1.75E-01	8.95E-02	5.18E-02	3.26E-02	2.18E-02
	0.5	1.05E+00	3.11E-01	1.31E-01	6.71E-02	3.88E-02	2.45E-02	1.64E-02
	0.625	8.39E-01	2.49E-01	1.05E-01	5.37E-02	3.11E-02	1.96E-02	1.31E-02
	0.75	6.99E-01	2.07E-01	8.74E-02	4.47E-02	2.59E-02	1.63E-02	1.09E-02
	0.875	5.99E-01	1.78E-01	7.49E-02	3.83E-02	2.22E-02	1.40E-02	9.36E-03
	1	5.24E-01	1.55E-01	6.55E-02	3.36E-02	1.94E-02	1.22E-02	8.19E-03

Figure 3.25: Vertical deflection based on link dimensions

3.3.3 Drive System

Following the selection of the four-bar links, the drive system was designed. This process centered on the selection of a motor of appropriate strength. From this, a gearbox to amplify the torque and a motor controller to control the speed and direction were chosen. Finally, the initial design for the component was reconsidered.

Motor

In identifying a motor and subsequent gearbox, it is essential to know the operating conditions. Given the estimated eleven pounds of the slider mechanism and sensor platform and that the four bar linkages are about two feet long, the output torque from the gearbox had to be 22 foot-pounds at a minimum. Further, the motor had to run on 12 Volts at a maximum, not draw a significant amount of current, be cost effective, and space efficient. With those specifications in mind, a series of DC motors were investigated. As seen in the weighted matrix below, the CIM, Mini CIM, Pololu 99:1, 227:1 and the Tetrax TorqueNADO motors were evaluated based on their prices, size, stall torque and free running speed. Although CIM motors have traditionally been used by FIRST Robotics teams for such applications, the impressive output torque of the Tetrax TorqueNADO, along with its competitive price and size, made it an ideal candidate for lifting the four-bar mechanism.




Motor	Price (USD)	Size (in)	Free Run Speed (rpm)	Torque (ft-lbs)	Weight
 CIM Motor	32.99	2.5 x 4.5	5330	1.78	0
 Mini CIM motor	29.99	2.5 x 3.86	5840	1.03	1
 Pololu 99:1 Gearmotor	36.95	0.98 x 2.60	100	1.56	1
 Pololu 227:1 Gearmotor	34.95	0.98 x 2.68	33	1.67	1
 Tetrax TorqueNADO	29.95	1.46 x 5.28	100	3.64	2

Figure 3.26: Weighted decision matrix for DC motors

To best drive the mechanism, it was decided that two motors would be necessary,

one for each four-bar set. This means that, at stall, the two TorqueNADO motors would generate approximately 7.3 foot-pounds of torque. Given that this is insufficient, a gearing the motors to increase the output torque is necessary.

Gearbox

In order to ensure that the motors would be able to lift the projected load and not reach stall, it was decided that the motors would operate at seventy percent of the specified stall torque. This mean that a minimum of a 3.5:1 gearbox was needed per motor. Additionally, it was determined that a worm gear would be employed at the last stage of reduction to prevent any high torque or back driving scenarios. With this comes a characteristic inefficiency; compared to the 80 percent efficiency for most other types of gears, worm gear reductions operate anywhere from 50 to 95 percent efficient. With that in mind, a theoretical 70 percent efficiency was applied to the gearbox, increasing the minimum reduction from 3.5:1 to 5:1. This minimum ratio determined that only a single stage reduction would be needed. With that, the Tetrix 20:1 Worm Gear, seen below, was selected. In addition to providing more than enough torque output, the gears are made of stainless steel and copper, ensuring they won't easily shear.

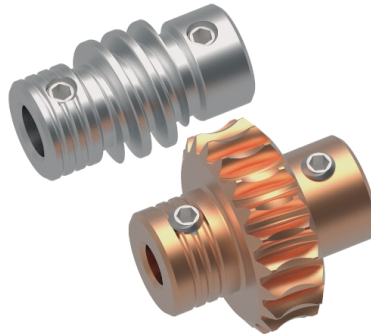


Figure 3.27: Tetrix Max 20:1 Worm gear set

Motor Controller

The final component needed for the actuation of the four-bar was an appropriate motor controller. A proper motor controller needs to be able to supply not only the correct signal but also the the appropriate amount of power to the motor or motors it is connected to. Considering that the TorqueNADO requires 12 Volts and approximately 8 Amperes, the MDD10A Dual Channel 10A DC Motor Driver was selected. This controller employs an NMOS H-Bridge to supply a PWM signal up to 20 KHz to the connected motors. The controller can supply anywhere from 5 to 30 Volts and 10 Amperes continuously, with a limited, 30-Ampere maximum current.

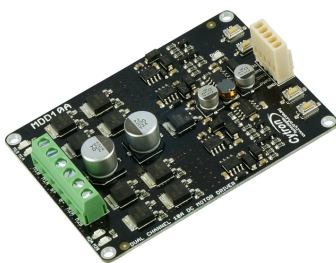


Figure 3.28: MDD10A Motor Driver

The MDD10A was later replaced with the Cytron SmartDriveDuo, a more powerful version of the MDD10A, after the initial controller was compromised due to a current surge. To prevent this same incident from affecting the new motor controller, a 10 Amp fuse was included between the rover battery and the controller. This way, if the motors approached stall torque or another short circuit scenario were to occur, the fuse would trip before irreparable damage to either the controller or the entire mechanism. Further, it was found that the SmartDriveDuo will continue to send the last signal received. To prevent the activation of the four-bar after shutting down the Raspberry Pi, a 10 Amp switch was also wired in series with the 10 Amp fuse. This switch also acted as the emergency stop for the four-bar, given that the primary emergency stop button only stopped main rover locomotion.

3.3.4 Support

Following component selection, it was necessary to identify appropriate support structures and materials for both the links and motors.

To support the polycarbonate links, two sets of axles were supported between 80/20 extruded aluminum channels. The driven and follower links were placed three and eight inches above the top of the rover, respectively. The follower links were supported by half-inch brass bolts, each secured by a Cotter pin on the outside of the channel. Meanwhile, the lower links were supported by a $\frac{3}{16}$ rod made of hardened steel. The small diameter axle was chosen in order to utilize Tetrix components, such as the worm gears discussed previously and the axle hub, seen below. This component was used to augment the torque from the worm wheel by radially distributing the force to each of the four 10-32 screws

The 80/20 extruded aluminum channels used were one inch square, T-slotted channels. These were arranged in a series of h-shaped trusses seen in figure 3.30. These trusses mitigated any deflection towards the front or rear of the rover. To further strengthen the entire mechanism support and mitigate sideways deflection, each truss was secured to the others by a $\frac{3}{16}$ aluminum flat bar running the width of the mechanism. This flat bar also served as a hard stop for the mechanism, limiting the maximum rise angle to approximately



Figure 3.29: Tetrix max axle hub

40 degrees above horizontal.

The final support components to be developed were the motor mounts for the Tetrix motors and the L brackets to attach to the coupler. Initially, two mounts were made for each motor. The primary mount contained a large hole for the main axle and 6 smaller holes arrayed in a circle to match up with the set screws on the face of the motor. The second mount was a simple support that described a half circle around the underside of the motor. The latter mount primarily served to prevent undue strain on the former, as the motor would behave similar to a cantilever beam if left unsupported. Finally, a set of four, large L-brackets were developed to attach the links to the coupler plate. Each bracket contained two, half-inch holes spaced five inches apart. It was important to maintain this dimension as it matched the distance between the driven and follower axles. This ensured both a constant speed and angle as the links were raised. A constant angle meant that the metal detector would not need to constantly readjust when being raised while constant speed ensured that a velocity controller was not needed.

3.3.5 Assembly

The assembly of the four-bar mechanism combined each of the aforementioned components into one subsystem. To secure each together $\frac{3}{8}$ -inch aluminum plate was manufactured with through holes to attach each appropriate support. This base plate spanned the width of the top of the rover such that the maximum distance between the links could be obtained. This separation mitigated the torsional effects of the translating mass of the slider. Additional holes were included to mount the plate directly to the top of the rover, utilizing the 10-32 threaded holes located in a grid pattern on the rover top plate. Considering that the top plate needed to be attached to the rover frame, sections were taken out of the four-bar base plate to provide ease of access. Each 80/20 support was attached with with a $\frac{1}{4}$ -20 bolt screwed directly into a tapped end. Each motor mount was attached with a 10-24 bolt secured by a nut attached to the underside of the base plate. To ensure that the hardware for each mount did not interfere with the rover top plate, a $\frac{3}{8}$ -inch polycarbonate sheet was manufactured with the same dimensions but larger holes to act as a spacer.

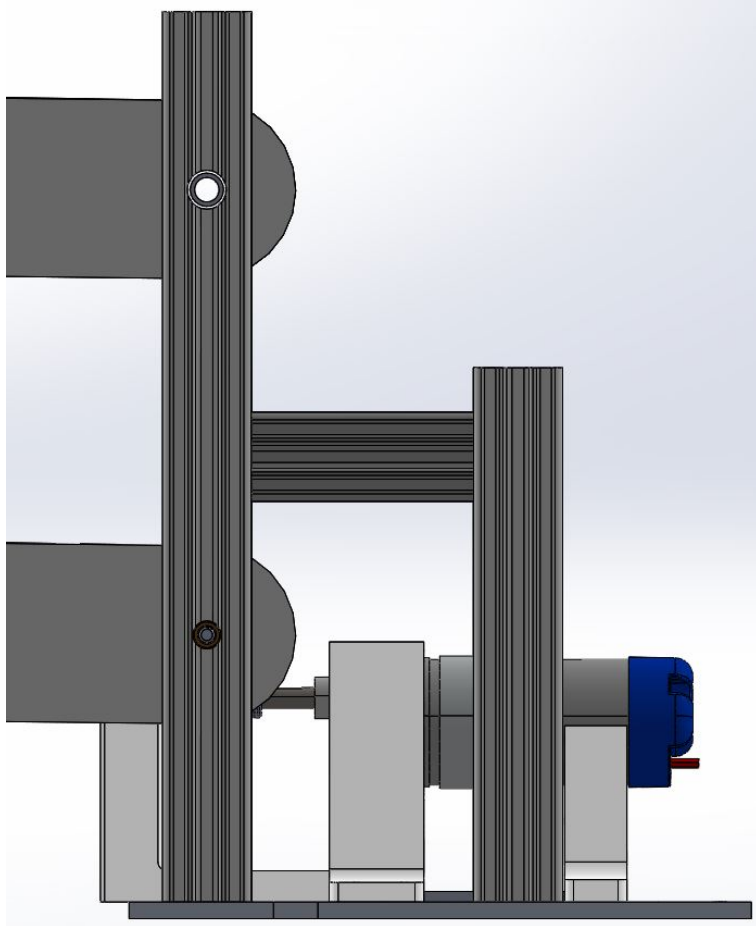


Figure 3.30: 80/20 h-Truss

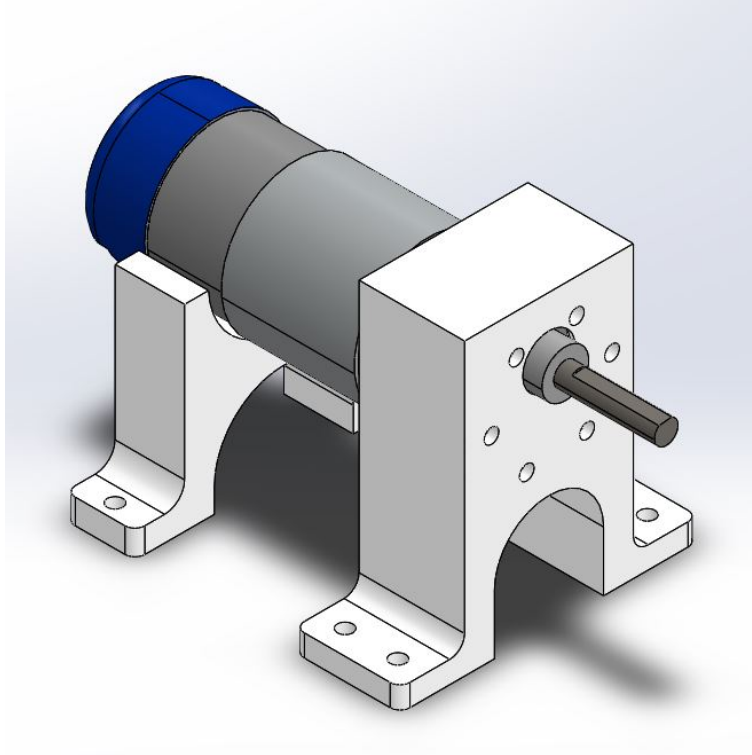


Figure 3.31: Tetrix TorqueNADO in preliminary mounts

After attaching the supports, motors, links, and other components to the base plate, the coupler plate was attached. This $\frac{3}{16}$ -inch aluminum plate contained eight, $\frac{3}{16}$ -inch holes. Each pair of mounting holes served to attach the 3D-printed L-brackets. To keep the links from becoming misaligned, 0.5-inch aluminum tubes were inserted through both sets of brackets and links.

3.3.6 Initial System Design

The initial design of the entire system was inspired by previous work with four-bar mechanisms as well as the mechanism implemented by the previous iteration. With this design the torque supplied by the motors was transferred directly to the driven links, as discussed previously. In this configuration, the two parallel linkages would be independently driven, and could offer a limited amount of control over the roll of the coupler plate and sensor arm. A further benefit of this configuration was also the ability to calculate the distance of the metal detector from the front of the robot using the following equation:

$$Dist = 24 * \cos\left(\frac{N}{7200}\right) + X \quad (3.5)$$

In the above equation N is the number of ticks recorded by the Hall effect encoder

attached to the TorqueNADO motors, while 7200 is the number of encoder ticks for one full revolution of the shaft out from the gearbox and X is the horizontal offset distance from the end of the driven link to the center of the metal detector.

3.3.7 System Redesign

However, despite the control that the initial design offered, several issues, discussed at length in Results, forced the redesign of the transmission system. The new configuration leveraged a simple winch and pulley system to lift the links and load rather than rest the entire weight on the two $\frac{3}{16}$ inch axles. To develop the exact specifications for this redesign, another MATLAB script was developed and can be seen in Appendix D. This script calculated the force at each joint and the required torque out from the motor for various winch diameters, head pulley heights, and cross member support distance. From the value calculated, it was decided to implement a 1.75-inch diameter winch, with a head pulley 14 inches straight above, and a cross member support 17 inches along the bottom links - or approximately two thirds of the link length. Mule tape was used to connect the cross member to the winch via the head pulley. This flat rope was used given its high tensile strength - approximately 2500 foot-pounds- and that it would lie flat around the winch, mitigating any twisting. A sample of the outcome of the development script as well as the performance of the initial and modified designs can be seen in the four-bar results section.

To include this design, several modification to the initial assembly were required. First, the $\frac{3}{16}$ axles were replaced with the same half-inch bolts and cotter pins that supported the follower links. These thicker bolts were more capable of sustaining the loads of the entire system. Next, the head pulley was installed above the existing superstructure. Two additional, 8-inch 80/20 extruded aluminum channels were fastened one inch apart on the existing $\frac{3}{16}$ aluminum plate that aligned the larger channels of the h-shaped trusses. Additional supports were attached to the plate that aligned the lower channels of the truss. The additional supports mitigated the deflection of the head pulley supports.

Finally, the motor mounts were overhauled entirely. Instead of relying on 3D printed components, the new mounts were comprised entirely of metal components. The main structure of the revised mounts consisted of 1.5-inch hollow steel square channels. These were chosen for their strength as well as the fact that the worm gear and wheel used to augment the torque could fit inside, providing necessary support that prior printed mounts could not. Four six-inch channels were welded into two L-shapes, with through holes for the motor and axles. Two $\frac{1}{4}$ -20 nuts were also welded to the interior of the bottom link such that the mounts could be secured to the aluminum mounting plate. Finally, Tetrix two-piece DC motor mounts were acquired and attached to the top of the lower channel. These mounts clamped down on the TorqueNADO motors, providing far more support than the previous mounts.

A number of benefits were immediately clear after implementing this new design. First, the two motors were both used to power the winch, ensuring that there was more

than enough output torque, by using a single axle attached to the winch flanges in the same fashion as the driven links had previously. Next, because there was no force applied by the motors to lower the four-bar, as in the previous design, potential damage to the metal detector was reduced. Further, the four-bar could still be manually re-positioned for testing without running the motors by simply pulling up on the mule tape. Finally, given that the parallel mechanisms would be moving together and not influencing the roll of the coupler plate, additional polycarbonate cross-links were installed. These decreased the deflection of the coupler plate as the sensor arm moved from one end of the slider to the other. The CAD model for the redesigned system can be seen in figure 3.32.

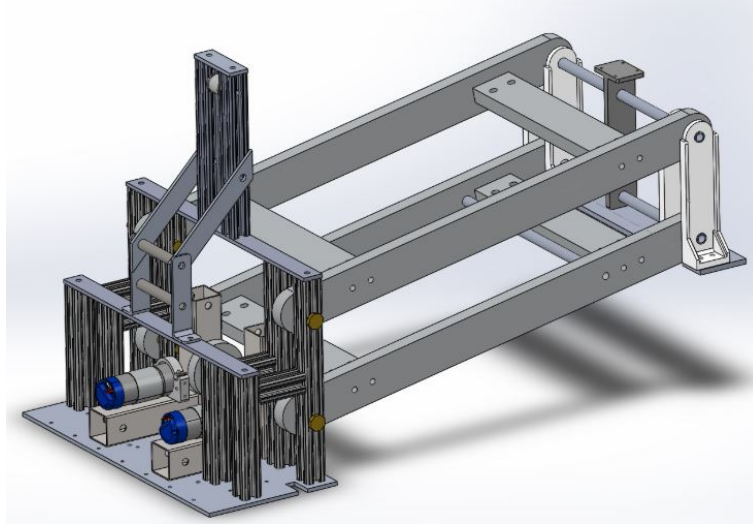


Figure 3.32: CAD model of redesigned four-bar mechanism

Despite these benefits, a few drawbacks were noted. First and foremost, the exact position of the metal detector could not be accurately determined. Instead, the following equation was used to estimate this location:

$$Dist = 22 * \cos(0.0038071 * N) \quad (3.6)$$

In the above equation, N is the number of ticks. This equation was derived using the linear interpolation between the length of mule tape from the head pulley to the cross member support at the maximum rise angle and at horizontal. Since this length does not change exactly linearly as the four-bar raises, the value is not exact. Further, given that the maximum change in the distance from the rover face was approximately five inches, position tracking was ultimately not implemented in the localization of a discovered landmine. Another drawback to this system. Further, given that force was only transferred via the mule tape, there is a risk of unrecoverable failure should the rope break. However, given the high tensile strength, this risk is greatly mitigated.

3.3.8 Programming

The programming for the four-bar was comprised of a set of basic commands programmed on the Raspberry Pi. These commands originally included the ability to raise, lower, calibrate, and correct the four-bar. However, with the implementation of the re-designed drive system, these features were limited to simply raising and lowering the mechanism.

The raise and lower functions were triggered by an input signal from the Arudino board used to control the sensor platform. When receiving a high signal on pin 12, the four-bar would lift the sensor arm, while a high signal on pin 11 would trigger the mechanism to lower. In the instance that the four-bar reached its maximum lift angle and was still receiving a high signal on pin 12, a ROS message was published, however this was ultimately unused.

In the initial design, the driven links were not connected via the same axle and could potentially be slightly out of sync, affecting the sensor arm. To this end, the four-bar control node would routinely check the encoder values on each Tetrix TorqueNADO motor to ensure they were the same value, within a certain tolerance. If the difference between the two motors was outside the set tolerance, the program would correct the positions. With the winch based design, this was considered unnecessary, as a single axle connected both motors.

Finally, a calibration function was developed that could be implemented at the start up of the entire system. This function would raise the mechanism until the limit switch was triggered. At this point, the encoder value of one or both motors was set to zero. This action would enable position tracking of the metal detector during the sweep of a minefield. Further, this feature enabled an additional soft stop as the four-bar approached the maximum lift angle.

3.4 Four-Bar Testing Procedures

3.4.1 Operation

Given that the four bar is supporting the remainder of the developed subsystems, the movement of the assembled system was thoroughly tested and observed. To do this without endangering the structural integrity of the slider or the sensor platform, a dummy weight consisting of a ten pound dumbbell was periodically attached to the cross beams connecting the links and manually lifted. It was expected that the mechanism should complete the full rotation from below horizontal to the maximum rise angle without any interference in motion. This process was completed several times during the assembly process and observed for any additional issues.

3.4.2 Programming Verification

After assembly, the four-bar control node was tested. First, the lifting functionality was tested in a controlled environment. The mechanism was clamped to a table with the slider and sensor arm attached. Prior to enabling the motors, material was stacked beneath an ultrasonic sensor until that particular sensor reported that the appropriate height was obtained. This was done to simulate normal operation and to prevent the mechanism from attempting to lower the sensor arm too far down. Once the minimum height was obtained, a flat piece of material, such as cardboard, was manually held beneath an ultrasonic to trigger the lift and removed to trigger the lowering.

3.5 Slider Mechanism

The slider mechanism connects the metal detector arm to the four-bar mechanism and allows the metal detector to sweep back and forth in front of the robot. This is important because the metal detector should scan the entire area in front the robot such that a landmine is not accidentally set off by driving over them.

The design was chosen after careful calculations were made. One design has perpendicular c-channels and bearings that are shifted towards the robot slightly. The other design of the slider has parallel c-channels and bearings, creating a coupled moment. Using the calculations below, it can be seen that the first design is a much stronger, more supportive design.

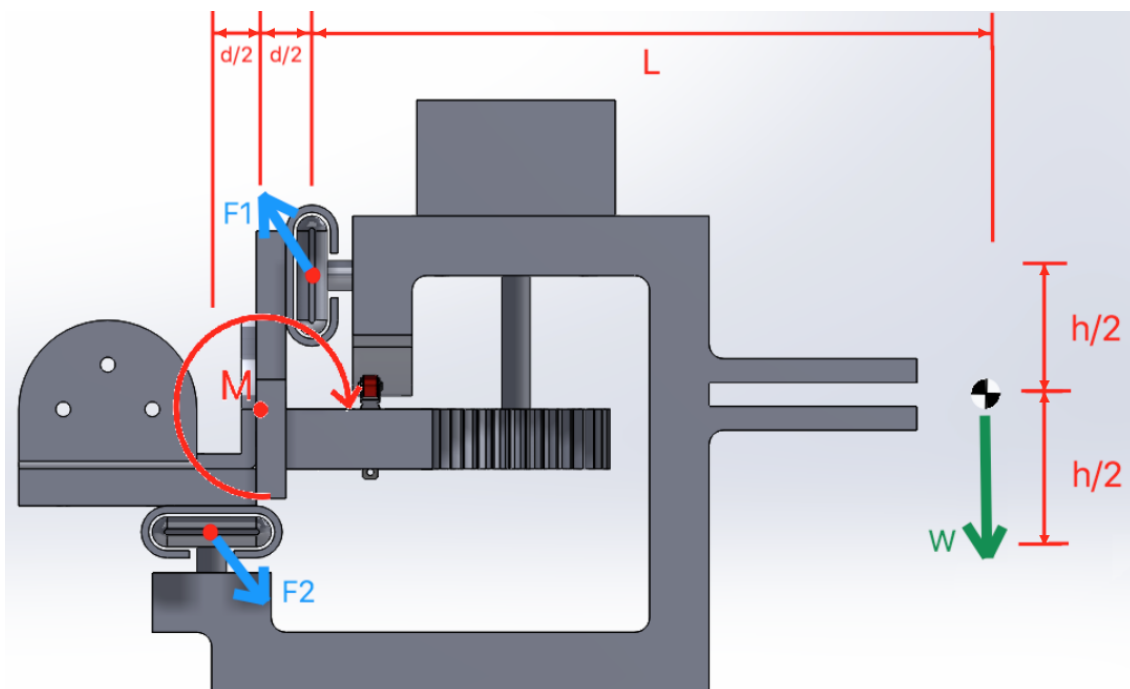


Figure 3.33: Slider design 1

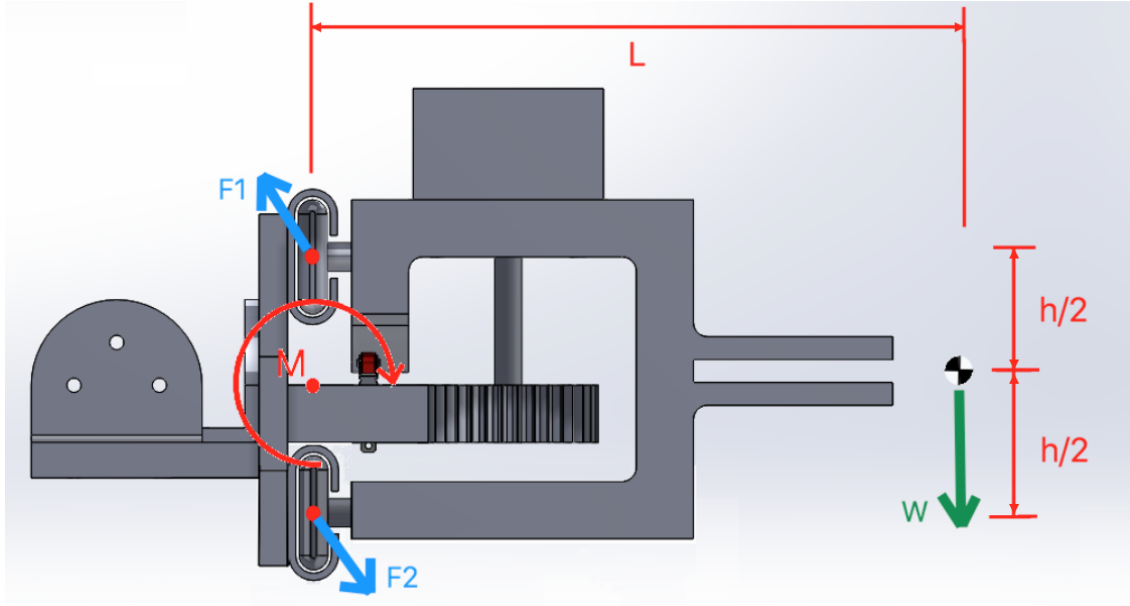


Figure 3.34: Slider design 2

M=Moment F_T =Total Force D=Distance F_1 =Force 1 F_2 =Force 2 W=Force Weight
 D_1 =Total Distances in Design 1 D_2 =Total Distances in Design 2

$$M = F_T * D \quad (3.7)$$

$$\frac{M}{D} = F_T \quad (3.8)$$

$$F_T = F_1 + F_2 + W = 0 \quad (3.9)$$

Where the forces (F) are force vectors

$$D_1 = \frac{h}{2} + \frac{h}{2} + \frac{d}{2} + \frac{d}{2} \quad (3.10)$$

$$D_2 = \frac{h}{2} + \frac{h}{2} \quad (3.11)$$

Fixed distance parameters in both designs: Length to Center of Weight

$$L \quad (3.12)$$

$$\frac{h}{2}$$

Force Totals:

Smaller Force

$$F_{T1} = \frac{M}{D_1} \tag{3.13}$$

Larger Force

$$F_{T2} = \frac{M}{D_2} \tag{3.14}$$

Force Total For Design 2 has a smaller distance, therefore requiring greater total forces on the bearings.

$$F_{T1} < F_{T2} \tag{3.15}$$

The slides are 28 inches long. This allows for 24 inches of travel due to the four inch wide slider bearings hitting the edge. This subtracts two inches off of each side of the slider length of 28 inches long. This is the same width of the robot, meaning the center of the metal detector will detect in front of the tires. This will be propelled via the use of a rack and pinion gear. The bearings and sliders support the load, and the gears propel the slider. For positioning, a stepper motor will be used. The stepper motor will maintain an accurate reading as to where the metal detector is along the rack gear.

In the event that the stepper motor misinterprets the current location, soft and hard stops have been included. A pair of limit switches act as the soft stop. These limit switches reside within the outermost rack gears and get pressed by the round bump on the slider if it drives too far in either direction. Should the limit switches also fail to send a stop signal, the rack gear has a built in hard stop; the teeth for the pinion gear to mesh with run the length of the rack until the end. The tooth pattern stops and becomes solid plastic. This ensures that the motor won't turn anymore. The soft and the hard stops can be seen below.

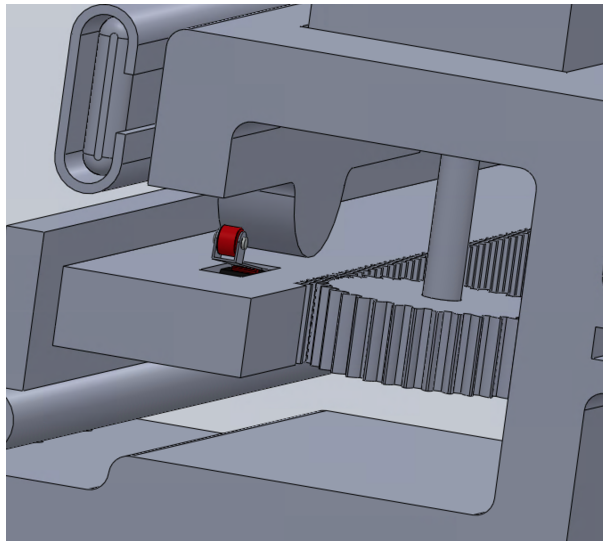


Figure 3.35: Soft and hard stops

A revised design can be seen below:

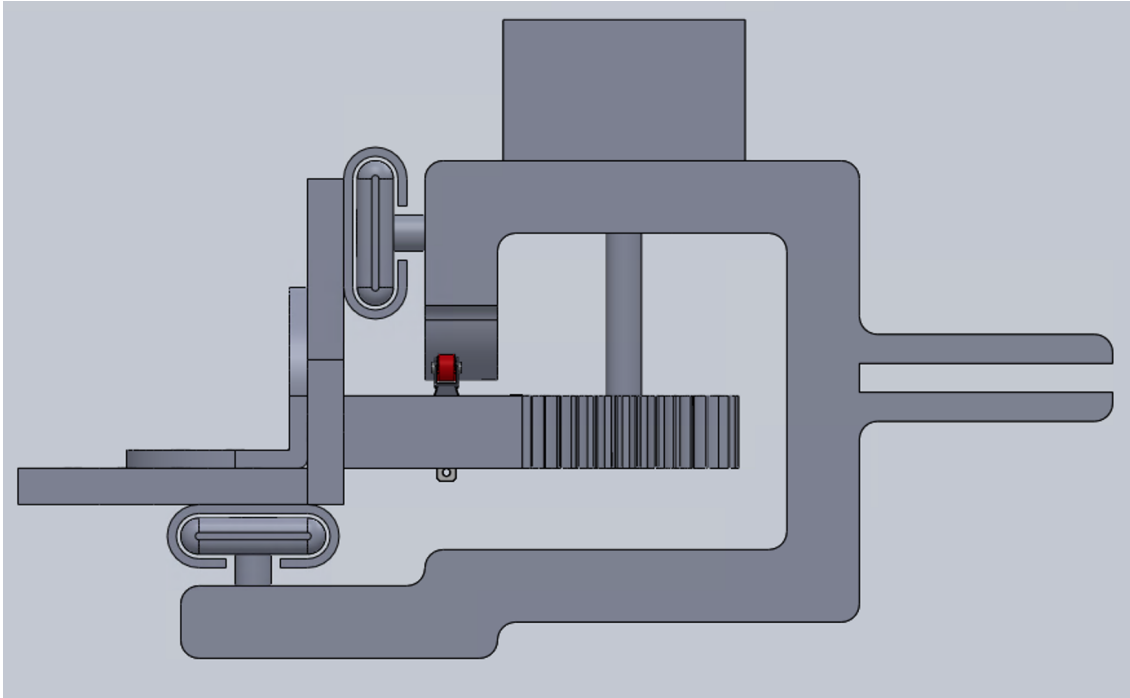


Figure 3.36: Slider Design

3.5.1 Slider Components

Coupler

The coupler connects the metal detector to the four-bar. This piece is responsible for housing the stepper motor, pinion gear, and linear bearings to allow for the linear movement of the slider. Brass threaded inserts will be used for securing the metal detector to the coupler as well as for connecting the linear bearings to the coupler. The NEMA 17 face plate pattern was copied onto the coupler for attaching the stepper motor.

Linear Bearings and Rods

To allow the slider to move back and forth, the use of a low-friction system that can restrict the slider's movement needs to be employed. The system needs to restrict the movement to only move linearly back and forth in front of the robot. The attachment points can rotate around the axis of translation if required. Both bearings and friction less slides were explored for this purpose of restraining the slider. Listed below in Figure 3.37 is the decision matrix illustrating the benefits and downfalls of each system, based off of the research performed.

General Bearing Types	Advantages	Disadvantages	
Plain Bearing	Long lasting	Requires consistent and ample lubrication	
Rolling-Element Bearing	Compact, easy to lubricate, can be sealed	Not for use in heavy-weight applications	
Fluid Bearing	Low vibration, low wear, low friction	For use in high speed and high load applications	
Magnetic Bearing	No points of contact	Requires constant power	
Frictionless Slide Inserts	Stable	Requires consistent and ample lubrication, can get caught on	
Rolling-Element Bearings Types	Advantages	Disadvantages	Track Required
Radial Ball Bearings	Low rotational friction	Low axial and radial load capacity	Slotted Rail
Linear Ball Bearings	Designed to handle side loading on a track system	High precision machining requirements	Tube/Rod
Roller Bearings	High weight capacity	other rolling-element bearings	Slotted Rail

Figure 3.37: Bearing decision matrix

Stepper Motor

A stepper motor will be used to propel the system back and forth. A stepper motor will be used due to the ability to switch directions quickly without having a large current draw when reversing directions on top of being able to count and keep track of motor steps which can be converted to position. Counting the steps that the stepper motor moves is possible because whenever the coils get turned on and the shaft increments, the code can add to or subtract from the variable tracking the step count.

The stepper motor to be used is a NEMA 17 stepper motor. There are many different variants of the NEMA 17 with varying specifications. Any NEMA 17 stepper motor, given the minimum requirements for the system input are met, can be swapped into this design. This is due to the fact that the “17” represents the mounting face dimensions of the stepper motor. The particular stepper motor selected is a low current, high torque motor. The torque required for the stepper motor to operate was calculated using MATLAB. The equations used can be seen in the attached MATLAB script.

The stepper motor will be controlled by an H-bridge stepper motor driver. This chip takes a signal and turns the coils on and off at specified rate in a certain order dependent on which state the “direction” pin is set to.

Axle Coupler

The axle coupler is a set of two pieces that fit together to connect the D-shaft of the stepper motor to the pinion gear. The first piece has the inverse shape of the D-shaft cut into it on one side. On the other side, there is a square impression cut into it to meet the D-shaft. The second piece is a two-part axle. One part is a rectangular prism with a square cross section profile. This connects the stepper motor shaft to the pinion gear to transfer the rotational motion from the stepper motor shaft to the pinion gear. Because of the design

being a square, there is no possible slip between the axle and the pinion gear. Another part of the rectangular prism sits above the pinion gear and connects into the square impression of the first piece. The second part of the axle is a cylinder that rides in a bushing to stabilize the axle and support it on a second end other than at the base of the motor.

Pinion Gear

The pinion gear is responsible for transferring the torque of the motor into a tangential force at the contact point between the pinion and rack gear. The pinion gear is a 36-tooth gear with a pitch radius of 0.75 inches and a diametral pitch of 24 in^{-1} . The teeth are 20-degree involute teeth. This pinion gear meshes with the stationary rack gear and rotates to provide the coupler with linear motion.

Rack

The rack gear is a combination of three separate rack gears to account for ease of manufacturing. The pieces make up one 28-inch long rack gear that spans the front of the robot. This allows the slider to move back and forth in front of the robot and scan in front of each wheel. The rack gear has the same face width and diametral pitch of 24 in^{-1} to ensure optimal meshing of the gears.

At the ends of the outer rack gears, there are slots with retaining pins to house limit switches. These limit switches will act as soft stops in the event that the slider drives itself too far to one side. The slots are cut all the way through the rack so that the wires can be soldered on the underside.

Shear Calculations

To ensure that the gear teeth will not shear under normal operations, a MATLAB code was developed. This code can be seen in Appendix B. This took into account, the tangential force that the gear teeth would be under if the system was travelling along a hill with a roll of 30 degrees. It also accounted for the type of gear teeth being involute teeth and the material properties of a 3D printed gear with a 0.2mm layer height at 100% infill. The code factored in a Lewis Form Factor rating for the teeth as well. This code verified that our gears will not shear under normal operating conditions with a factor of safety of about 6.

3.5.2 Bending Deformation Calculations

In order to see if the coupler will hold to its assembled form and not deflect into the rack gear, a MATLAB code was developed. This code can also be found in Appendix B. This assumes no normal force on the pinion gear and that the rods are handling all of the load of the coupler. The type of deflection problem was assumed to be a point mass in the center of two fixed end points. This accounts for downward deflection. The downward deflection was calculated to be 0.0383 inches. This means, most of the bending came from bending caused by dynamic loading. To solve for the static forces of the system, the code can also be referenced in how the system is fully constrained because the x and y forces equal 0. The Free body diagrams of the system can be seen below.

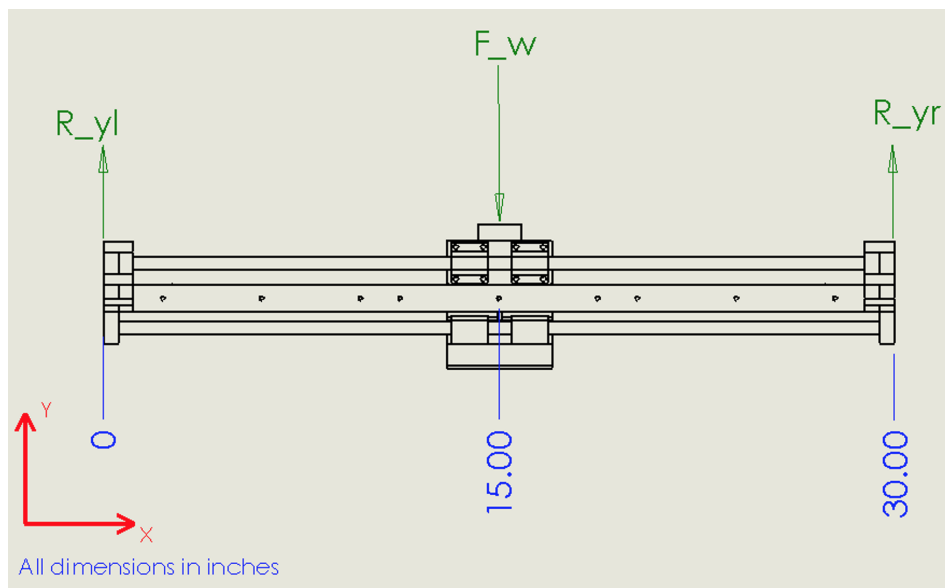


Figure 3.38: Free body diagram of slider (front view)

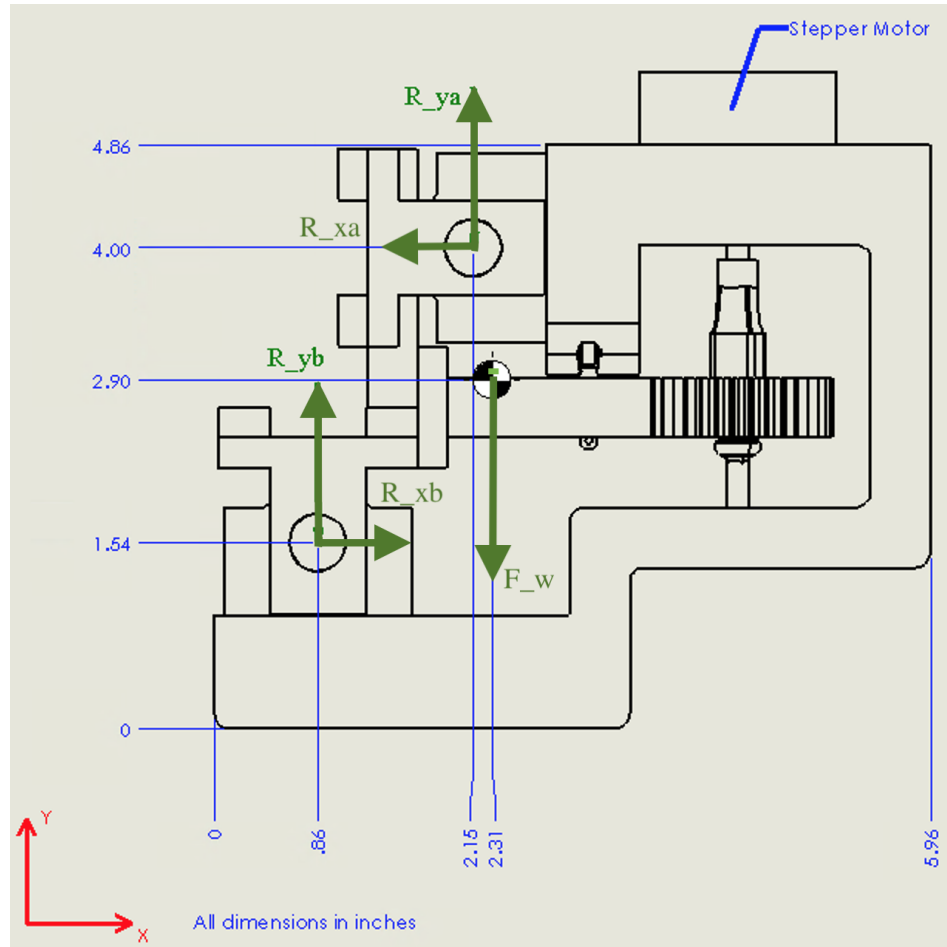


Figure 3.39: Free body diagram of slider (Side-View)

Limit Switches

Normally-open limit switches, attached to pullup resistor activated ports on the Raspberry Pi will help identify the ends of the rack gears. This will be useful when zeroing the slider and in the event of any errors in the normal operation, they will serve as soft-stops to tell the slider to reverse direction,

Four-bar Attachment Plate

To attach this slider system to the four-bar, an aluminum attachment plate will be made. This will be made into an L-bracket so that the rack gear and rods can mount to a solid fixture. The stock will be 0.25-inch aluminum which is lighter and readily available material thanks to Mr. Wagner. This will have attachment holes for the four-bar and the rack gear to attach to. This will join the two systems together, making all of the parts a unified system.

3.5.3 Manufacturing Procedures

FDM 3D Printing

Fused Deposition Modeling (FDM) 3D printing will be used to print the parts for the system. This process is an inexpensive and fast way to produce parts that are not under a large load. The material being used in this process is Polylactic acid (PLA) plastic. This is one of the most readily available plastics to the team as well as it is inexpensive and the material that the team is the most familiar with for 3D printing practices. Nick will be printing the parts for the team using his 3D printers.

Aluminum Stock

Scrap aluminum stock will be used instead of buying new stock in order to cut back on cost. The stock will be cut to size to create the four-bar attachment plate as well as the linear bearing rod holder attachment points. This keeps the system light-weight and strong.

3.5.4 Assembly Procedures

TIG Welding

TIG welding will be done by the Blackstone Valley Tech Welding Shop to make the brackets that will hold the linear bearing rods. TIG welding allows for the securing of the aluminum to occur through high frequency current exciting the metal and melting an aluminum rod to fuse each of the surfaces. One surface fuses to the molten aluminum rod and the other surface also fuses to the molten aluminum rod. This effectively secures the two pieces of metal into a certain configuration.

Intentional Plastic Deformation

In order to screw into the plastic and have a strong hold that is difficult to rip out, the team will employ the use of brass M5 thread inserts. The holes that were 3D printed on the pieces were intentionally slightly smaller than the inserts' outer diameter. Then a soldering iron is used to heat up the insert and push it into its respective hole. When it is heated up enough, the plastic melts around the insert and bonds to its knurled outside surface finish.

Hardware

A DRV8825 dual H-bridge chip is to be used to control the stepper motor. The Raspberry Pi supplies the logic power to the chip that powers the onboard components. The 12- volt power coming from the robot’s power distribution board supplies the rail voltage to be sent to the stepper motor when it is required. Both power sources must be grounded to their respective sources. The Raspberry Pi also sends “Step” and “Direction” signals to the H-bridge chip to tell it when to have the stepper motor move by one step and in which direction it should move. The only other connections off of the H-bridge chip are those that are connecting to the stepper motor coils. There are two coils that connect to the chip for a total of four connections. One coil loop connects to A1 and A2, and the other coil loop connects to B1 and B2.

Wiring/Wiring Harness

All of the wiring will be labeled accordingly and wrapped into as few cable trunks as possible. This makes the cables flow as one when the device is moving. This makes the cables much more manageable as well.

3.5.5 Programming Procedure

Programming the slider mechanism was broken down into three main points that the team wanted to address. ROS message control, and limit switch soft-stops.

Initialization

Initialization will handle zeroing the slider and starting it from the same position every time. When the ROS node initializes, the slider moves to the left limit switch until it triggers the limit switch. Then it marks its current step count (position) as 0, reverses its current direction, and moves back towards the right as the step count increments upward by 1 with every step until the right limit switch is triggered. Once the right limit switch is triggered, the total step count is saved. The left boundary then becomes 100 steps added onto 0 steps (the left limit), and the right boundary becomes 100 steps subtracted off of the total step count. This now becomes the range that the slider can move between. When the scanning begins, the slider will move 1 step then check to see if it needs to reverse based on if it’s at one of the boundaries.

ROS message control

The ability to handle ROS communication will allow for incoming ROS messages to tell the slider whether or not it is allowed to scan freely between set limits. This means that the slider will only start sweeping once the robot starts navigating the minefield and the ROS message tells it to start.

Limit Switch Soft-Stops

Trigger edge cases Having limit switches as soft-stops allows for the system to have a chance to stop itself before it hits the hard stop if it ever overshoots one of the set limits. The code is designed so that each time the stepper motor moves, it checks to see if it has hit the limit switch on the side it is currently headed towards. If the switches are triggered at any point in the scanning process, the direction variable controlling the direction of the stepper motor's rotation will switch. If the slider is moving to the left, and the right limit switch is pushed, nothing will happen because the direction variable is already set to move in the leftward direction. The same logic holds for the slider moving to the right if the left limit switch is pressed.

Dual Switch-Triggered Stop The dual switch-triggered stop functionality is just a safety measure. The functionality will be used to stop the sweeper from moving if an error ever happens or is going to happen within a time-frame where a user can react in time. This prevents the team from shutting down the robot every time the slider's actual output is different from desired output. Instead, the two limit switches can be pressed at the same time by the user and the slider will stop.

3.5.6 Complete Slider Design

The complete slider design can be seen here. This will be produced for the project to satisfy the coupling and sweeping needs of the robot.

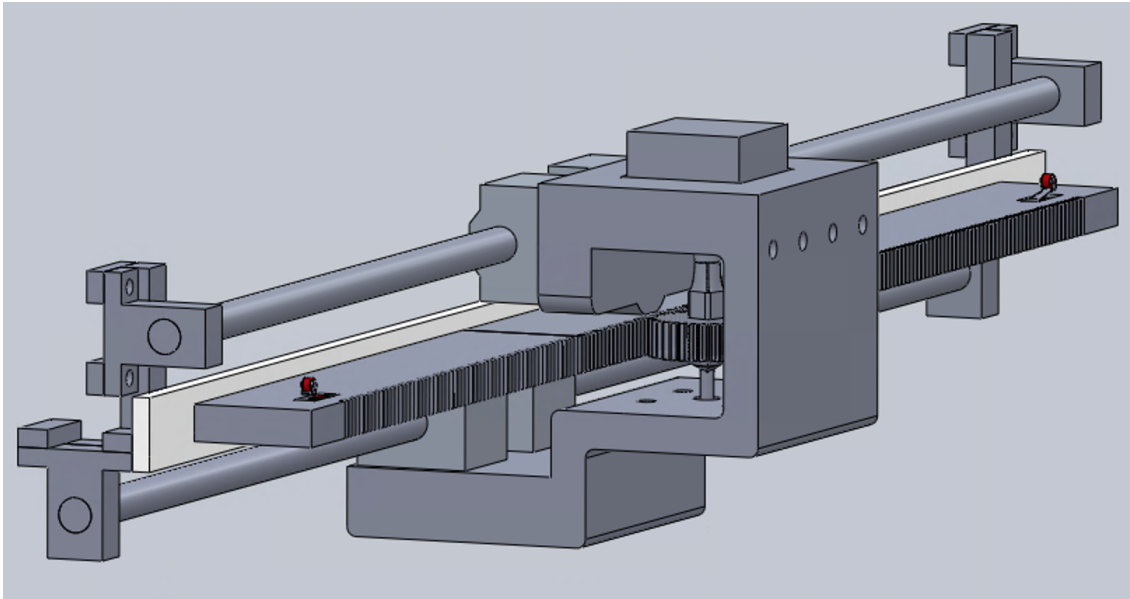


Figure 3.40: Complete Slider System CAD

3.6 ROS Network

The Robot Operating System (ROS) was used to run the main control programs for the robot. ROS allows for modular software development so many ROS nodes were created for each unique function and distributed among the computers on the ROS network. The ROS network consists of three computers including the base station laptop, a Raspberry Pi, and the Husky on-board computer, shown in figure 3.41. The ROS Master computer is the on-board computer and will be running ROS nodes that communicate with the Husky control board, the navigation algorithm (see Section 3.7) and the computer vision algorithm. The Raspberry Pi runs software that communicates with the sensor platform, retrieves images from the infrared sensor, retrieves data from the Maxbotix sensor, and controls the slider and four-bar. The base station computer runs software for viewing the processed images from the computer vision, stores minefield map data and can command the navigation algorithm. All of the software on the base station is not necessary for the drone to complete the minefield sweep so if connection is lost the husky will continue to operate. Each of these programs is discussed in this section.

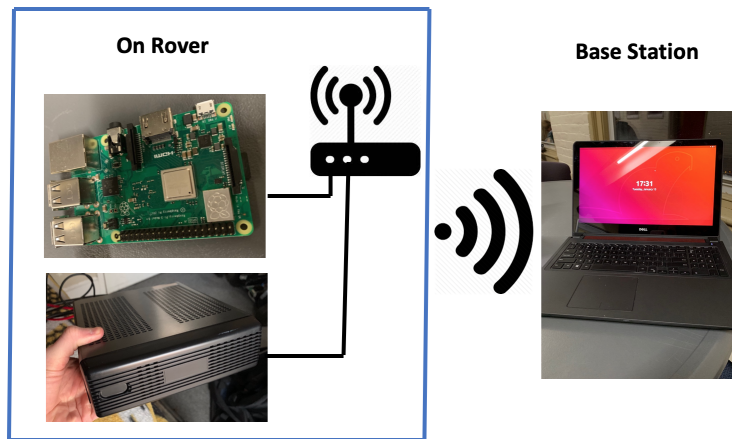


Figure 3.41: ROS network diagram

3.6.1 Computer Vision System

The computer vision consists of two ROS nodes with an additional node for viewing the image. This system was removed from the final design, which will be discussed in the results but it was completed. The first node is called IRCamera, and uses a library called PyLepton to retrieve the image from the camera. This image is then posted to a topic where the CompVision node will process it and search for circles. If any circles are found, the node reports a potential landmine has been detected and adds a black outline around the circle with a square at the center. The image is posted to another topic that the ViewIRCam node listens to. All sent images will be displayed on the base station laptop screen. The landmine

detection message is received by a fourth node called LandmineDetection that will merge that data with the metal detector data and send it to the navigation ROS node.

3.6.2 Sensor Platform Control Node

The purpose of the sensor platform control node is to interface the navigation node with the sensor platform through IIC. The navigation algorithm can send messages to this node and the commands will be automatically forwarded to the appropriate Arduino controller. The messages that can be sent are, "Start", "Stop", "Zero Metal Detector", "Home Orientation" and "Mark Landmine". The "Start" and "Stop" messages will be sent to both Arduinos while the "Zero Metal Detector" is only sent to the Arduino Uno and the last two messages only to the Arduino Mega. Along with the transmission of these commands, the sensor platform controller will request status updates periodically from the Arduinos to confirm that the command was successfully received and is being executed. It can report to the navigation node that the action is complete so that navigation can continue. This is especially necessary so that the robot does not continue scanning the minefield while a landmine is being marked with the spray paint. Additionally, the sensor platform control node will retrieve the status of the Arduinos every 5 seconds, regardless of if a command has been received from the navigation node. This is to verify that the sensor platform is not in an error state and functioning. If the sensor platform is in an error state, this will be sent to the navigation node. The navigation node currently does not do anything with this data, but if implemented, the data could be used to direct the rover back to the base-station for inspection and repair.

3.6.3 IIC Flow Control Node

The Sensor platform control and IRCamera node both use the same IIC bus to communicate with their designated devices. To prevent each of these processes from trying to access the IIC bus at the same time, another node called IICSem was created to control access to that hardware. Similar to a semaphore in a real time operating system, this node has a count variable that counts to 1. Whenever a thread needs to use the IIC bus, it requests permission from the IICSem node and if the count is 1, the count get reduced to 0 and the node receives a key to access the IIC bus. If another thread also requests to use the IIC bus, it will be stalled until the other thread completes its task and returns the key. This request is handled through the use of a ROS service. The service will return an access denied message if the IIC bus is in use. The ROS node in need of the IIC bus will then wait 10ms and try again, repeating until succeeding.

3.6.4 Landmine Detection Node

The landmine detection node is a simple process that retrieves landmine detection data from the Computer vision node as well as the GPIO pin coming from the Arduino Uno carrying the metal detection signal. The data is forwarded to the navigation algorithm for further processing and used to make decisions for stopping and marking a landmine.

3.6.5 Maxbotix Ultrasonic Sensor Node

The Maxbotix Ultrasonic Sensor Node was created to read in the data from the Maxbotix sensor and convert the data to a laserscan message. This is the same type of message that is typically created from LIDAR data and used in ROS navigation cost maps. By having the sensor data be a laserscan message, it will be automatically read by the move_base navigation node and fused with LIDAR data for detecting and avoiding obstacles. This node listens to posts from the slider node to detect when the slider has changed direction. After the slider has changed direction, the Maxbotix sensor node will begin recording data from the sensor. When another change in slider movement direction is detected, the recorded data is sorted, processed and constructed into a laserscan message so that recorded points, when viewed, look like they were created by a spinning LIDAR.

3.7 Mapping Algorithm

The mapping protocol was a significant amount of the programming portion of this project. This was comprised of two major portions, a map definition protocol and a navigation protocol. This section discusses how these protocols were designed to prepare for testing upon the completion of manufacturing.

3.7.1 Map Definition Protocol

An integral component of autonomous mine detection is defining the area to be searched. This can be done with physical markers, such as flags, string, or field paint, or with computer generated global coordinates. The previous iteration of this project used the latter option, manually copying coordinates obtained from an online map directly to the rover. While this method was successful in defining a searchable area, the act of copying coordinates one digit at a time was identified as a source of error. To remedy this, online map code API's were consulted in order to obtain a procedurally generated list that could automatically be copied and used by the robot.

In the development of this list, three code bases were investigated: Google Maps,

Google Earth, and Earth Engine. Although they were all owned by Google, each had subtle differences. For example, the newest API, Earth Engine, was tailored to time-based research concerning the constantly changing state of the Earth, while Google Maps and Earth were more tailored to relative positions and localization purposes. Scripts generated in Google Maps were written using JavaScript while Earth Engine was coded using Python, and Google Earth was a pure user interface. In order to obtain the set of coordinates that defined the searchable area, certain steps had to be taken. Despite including a user interface that allowed for the drawing and editing of polygons, points, and paths, research into the use of Earth Engine showed that it would be unsuitable for the task at hand, primarily due to the fact that it could only be used when connected to the internet. Further, considering that it was released in 2010, the lack of support for the youngest API did not help in determining how best to accomplish this task. Instead, Google Maps and Earth were consulted. Both were able to draw a polygon and output the coordinates corresponding to the vertices, however, Google Earth did so via a user-specified Keyhole Markup Language (KML) file, while a Google Maps script simply output an array of coordinates. Written similar to an Extensible Markup Language (XML) file, KML files are scripts that can be read by a user and a program alike. However, considering that the user specifies where and under what name to save the KML file and that this large file would need to be effectively parsed at startup, the Google Maps API was deemed preferable for this definition task. While not being able to work offline like Google Earth, the simple data structure output by a Google Maps script increases the efficiency of the definition procedure.

In developing this procedure, an HTML file was written using JavaScript. This file allowed a user to access and manipulate a map in order to get the coordinate points of a polygon. The generated polygon was editable by the user up until it was sent out to the robot. Given that points on the polygon can be created and moved, this feature ensured that the correct search area is sent to the robot and that no areas are overlooked or missed.

The team working on the project prior to the current iteration of the project failed to successfully implement the Google Maps API into the project. The points needed to be entered manually for defining the boundary which left room for human error. The goal for this iteration was to create a type of visual feedback that the user could interact with and just drag points around to define the minefield.

An API “Places” key was purchased to successfully generate the map object. Once the key was purchased and put into the verification line of code, jquery 3.3.1 was downloaded. JQuery allows for the functions and object declarations to work with Google. JQuery was installed locally in the same folder that the coding was saved in so that everything could be kept together and locally referenced in the instance of transferring the files between computers.

The first goal was to initialize the map at a specific latitude and longitude and zoomed in at a specific distance. This helps limit the user to only focusing on the area they are in currently. The map is still able to be zoomed in and out of based off of the user’s preference. Initializing the map was as simple as declaring a new map object with the desired

parameters set by the user. This includes a latitude and longitude coordinate as well as the zoom level. The map type was set to “hybrid” which displays the satellite view of the map with an overlay of streets and landmarks for more of a visual reference.

```
function initMaps() {  
    var points = Array();  
  
    var map = new google.maps.Map(document.getElementById("map"), {  
        center: {  
            lat: 42.27548140758946,  
            lng: -71.80486849867248  
        },  
        zoom: 18,  
        mapTypeId: 'hybrid'  
    });  
};
```

Figure 3.42: Initialize map object

Next, the goal was to create a marker that would initialize at the area of interest and be draggable. This was the same as declaring the new map object, however, a marker object was declared instead. The parameters were slightly different as well. The starting position of the marker was declared, but also the “draggable” parameter was set to “true” and the “map” parameter was set to the current displayed map object. This marker would then trigger events when it was done being dragged.

Listeners are functions that wait for a specific event to happen before executing their code contained within. The first listener was set to be triggered when the marker was dropped after being dragged to a position. The listener then logged the latitude and longitude of that point in a positions matrix and on the console screen. This showed the user the point they selected. The next listener waited for a right mouse click. This would trigger the polygon to be generated from the positions matrix that was generated previously. The polygon parameters were set in the listener which were the opacity, color, stroke weight, draggable, and editable parameters. Three additional listeners were created to wait for the polygon to be edited. These listeners would cause the position matrix to update when the polygon was edited and display the new position coordinates. Editing the polygon includes tasks such as adding new vertices to the polygon, removing vertices or moving existing vertices. All three of these listeners were written inside the code that was executed for creating the polygon. This way, the listeners were only able to actively wait once there was a polygon present.

```

google.maps.event.addListener(searchArea.getPath(), 'insert_at', function(){
  points = [];
  var vertices = searchArea.getPath();
  var contentString = '-----';
  // Iterate over the vertices.
  for (var i =0; i < vertices.getLength(); i++) {
    var xy = vertices.getAt(i);
    contentString += 'Coordinate ' + i + ':' + ' lat: ' + xy.lat() + ', ' + ' lng: ' + xy.lng() + '-----';
    position = {lat: parseFloat(xy.lat()), lng: parseFloat(xy.lng())};
    points.push(position);
  }
  console.log(contentString);
  console.log(points);
  console.log('-----')
});

```

Figure 3.43: Example listener

All of this was written within a single `initMaps()` function that gets called when the page loads up. Also, the marker was set to null so that it would disappear once the polygon was created. This way, the marker wouldn't get in the way of editing any vertices. This entire `initMaps()` function was run as soon as the page loaded which then waited for the user's input. Based on the input given, a position matrix was created with the intention that it will be sent to the robot for interpretation about what is the robot's frontier limit. This program must be run with an Internet connection. Offline capabilities have not been discovered as of now for using this without an Internet connection.



Figure 3.44: Minefield Definition

3.7.2 Navigation Protocol

Upon startup, the robot will designate its current location as its home base. Then it will attempt to start the navigation protocol. Within the navigation algorithm, the rover will attempt to explore the closest unknown area by using a greedy breadth first search algorithm. This will allow the robot to always be traversing the least amount of previously scanned area on its way to new frontier nodes. In order to accomplish this task, the robot will set up a digital map based on the information received from the map definition protocol. This map will be a grid of 20cm by 20cm nodes, however the size of these nodes may be modified due to external factors affecting the accuracy of the dGPS (differential global position system). The robot will be constantly scanning for obstacles as described above. When an obstacle is found, the robot will enter its obstacle logging protocol, then return to the normal navigation protocol. After the robot has explored every node in its digital map, it will return to its home base location.

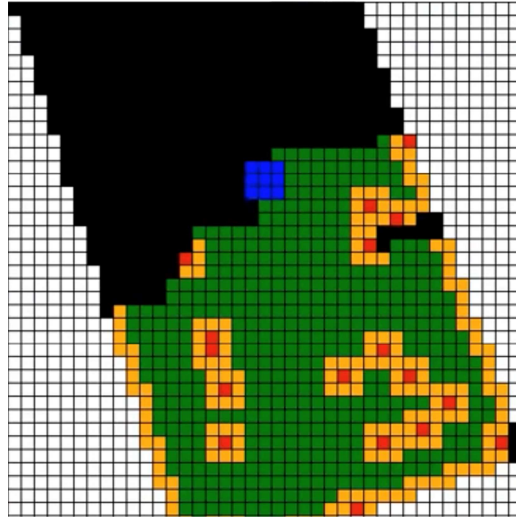


Figure 3.45: Graphical display of rover traversing a minefield

3.7.3 Obstacle Logging Protocol

When the obstacle logging protocol has been invoked, the robot will mark the location of the obstacle on its digital map. If the obstacle in question was a mine the robot will begin to leave a visible marking on the ground above the detected mine and will broadcast its location to the base station. Then, the navigation protocol will expand the obstacle on its digital map. This will allow the robot to have a buffer so that it will not accidentally set off any mines. After this, the robot will return to the navigation protocol. The rover does not currently detect and avoid obstacles besides mines but has sensors capable of doing

so. Avoidance of trees, rocks, bushes, etc could be implemented in the future with software upgrades.

3.7.4 Low Battery Protocol

In the event that the robot's battery is becoming low on power, the low battery protocol will be initiated. The threshold for how low the battery can be before this function is initiated is going to be adjustable by the user. While the robot is in the minefield it will consistently be checking two factors. It will be checking the distance of the shortest path back to its home base and it will be checking the amount of battery power the robot currently has left. If the robot's power level drops below the amount of power the robot thinks it will take to get back plus the margin of error threshold provided by the user, it will immediately start the return back to home base. Its digital map will remain updated so that it may start where it left off once it is fully charged again. The map is currently only stored on the robot but in the future should be sent to the base-station in the event the robot is destroyed during exploration.

Chapter 4

Results

4.1 Sensor Testing

This section summarizes the most important results of the sensor testing. Some of the results have been summarized here but the full results along with testing procedures and testing environment can be found in Appendix E.

4.1.1 Metal Detector

Results from testing if inductance affects the sensitivity of the metal detector are that the inductance does not matter. Four coils with inductance ranging from 54uH to 1305uH were tested. Each was able to detect 50g of steel at 11in away. The exception is that the 54uH coil was not tested as a strange noise was observed coming from the metal detector PCB when connected.

The main results from the depth detection with varying masses was that the detector has a maximum detectable range of about 17inches. This means that even large amounts of metal will likely not be detected if beyond this threshold. Smaller masses require being closer to the coil. A steel mass of 0.5g needed to be 3.5 inches away from the coil to be detected. Other masses fall in-between this range with increases in mass initially causing large increases in detectable distance but gradually level off at around 17 inches. A graph of the collected data can be found in Figure 4.1 with the tabulated results in the results section of Appendix E.

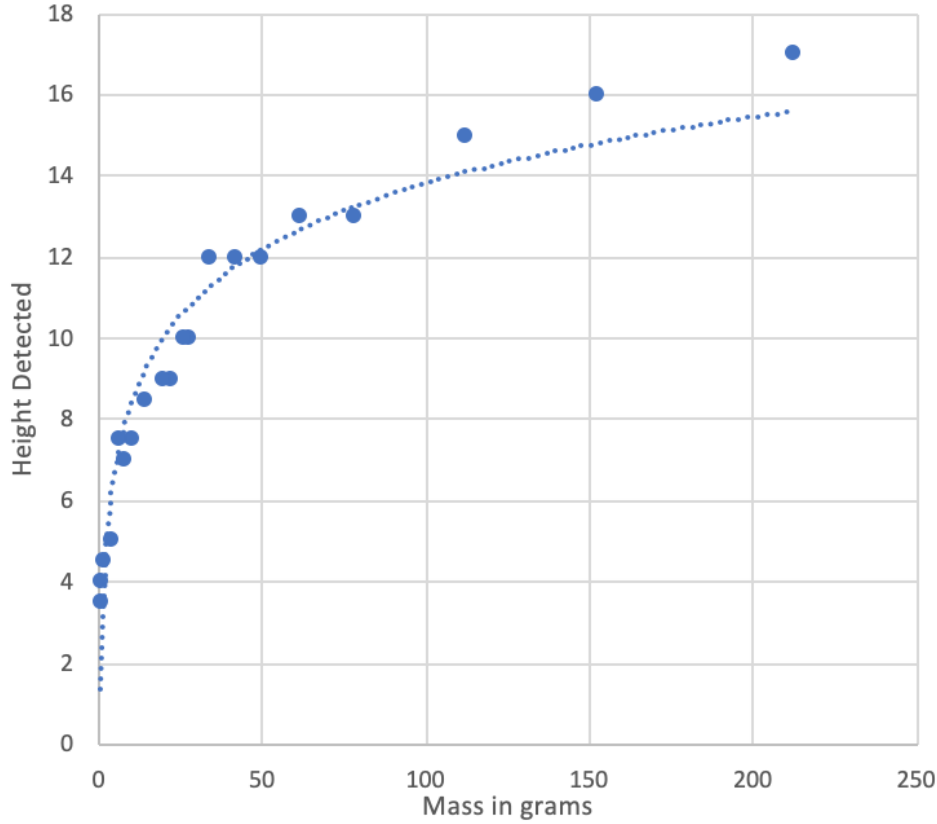


Figure 4.1: Graph of depth detection test results

Varying the surface area of the samples of metal had inconclusive results as more variables affect detect-ability than just surface area perpendicular to the direction of the magnetic field. It was expected that a smaller area perpendicular with the magnetic field would decrease the distance needed to detect the metal but it was found that steel pieces that also had a large surface area parallel with the magnetic field had a maximum detectable range that was further away than when the same piece was held in the orientation with this surface perpendicular to the magnetic field. The tabulated test results can be found in Tables 4.1 and 4.2.

Table 4.1: Detectability With Varying Surface Area

	Surface Area		
	Low	Mid	High
Mass (g)	Height Detectable (in)		
6	9.5	6	8
38	13.5	12	11
50	14	10	11.5
212	14	No Sample	17

Table 4.2: Surface Area Measurements

Mass (g)	6	38	54	212
Surface Area (in ²)	0.09	0.12	0.13	0.21
	0.88	3.75	3.75	-
	2.63	7.97	8.75	30.32

The main result from the interference tests were that aluminum inside the magnetic field prevents the metal detector from functioning at any capacity so an aluminum spray can could not be placed inside the coil. It was found that if the interfering metal (that is not aluminum) is further away than the target metal, the target metal can be detected at the same distance found in ideal conditions. The main problem is that the interfering metal will cause false detection if it is moving with respect to the magnetic field. Any interfering metal that is not stationary relative to the magnetic field is problematic for detection. Table 4.3 summarizes these findings.

Table 4.3: Interference Test Results

	1 Motor	2 Motors	Aluminum Can	All Three Items	1 Motor wrapped in Aluminum
Mass (g)	Height detectable (in)				
212	14	0	0	0	0
6	5	0	0	0	0
0.5	2	0	0	0	0

Lastly, the results from the soil testing was that soil generally does not affect detectability of steel. Soils consisting of dry dirt, wet dirt, dirt with grass and sand were tested and the metal detector was able to detect 6 grams of steel 6 inches deep in each test case. The maximum detectable range of the steel sample was decreased by about one inch compared to the testing done in air but this was consistent across all soils tested and may have been from measurement error. If the steel being detected is around 6g or more then these results suggest the metal detector will detect it beyond the project goal of a 3 in depth.

4.1.2 Infrared Camera

The infrared testing showed that the infrared camera may not be a reliable option for detecting landmines under the surface. The camera was clearly able to identify when a landmine was exposed on the surface since the landmine will heat up much faster than the ground. When the landmine is covered with soil, the changes in soil temperature are too gradual and slow to be detected in testing. Testing was done with a heat lamp and over a short period of time so it is possible that the infrared camera would perform better in a real-world scenario with the ground heating for a longer period from the sun but this was not possible to test with the team's available resources. An example image from the four soil conditions can be found in Figure 4.2.

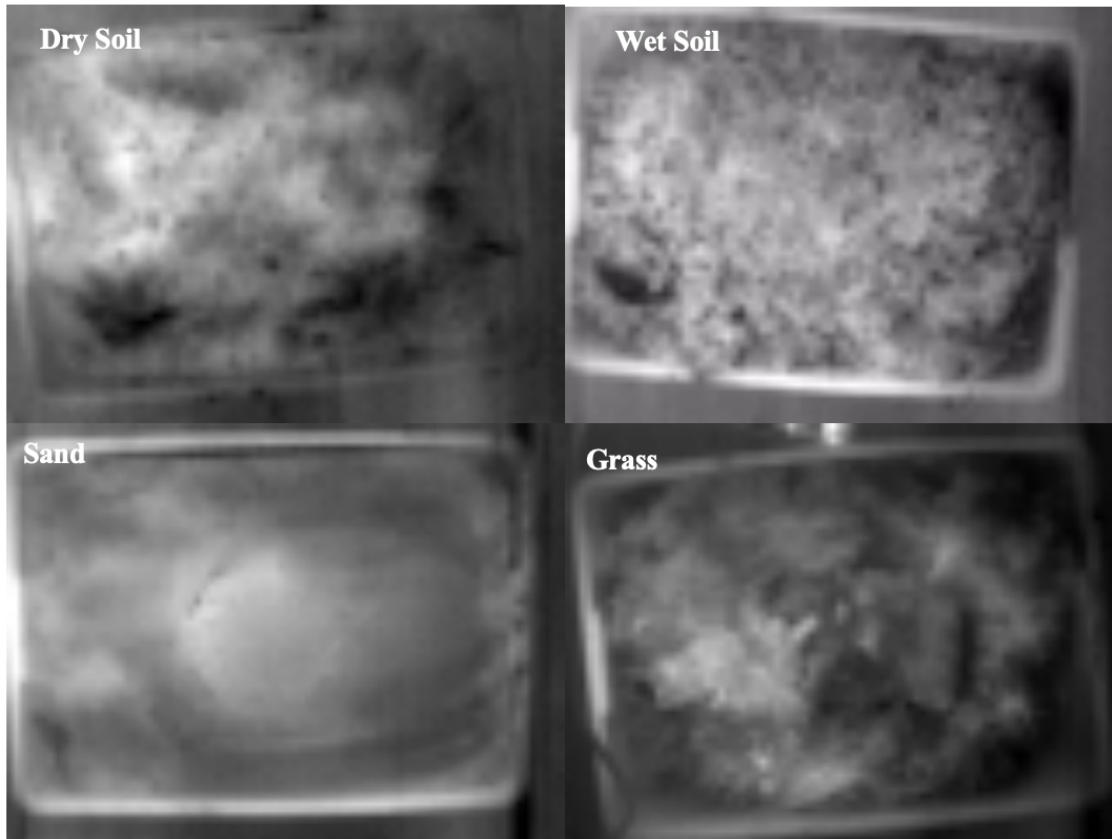


Figure 4.2: Landmine heating 0.25in below the surface

4.1.3 HC-SR04 Ultrasonic Sensor

The results of the ultrasonic sensor measurement on pavement were consistent for distances of 12 inches to 1 inch but the readings were consistently 1 inch greater than the actual. When tested on soft dirt, the measurements were consistent for 12 inches to 1 inch but again 1 inch greater. The measurements over grass had a lot of noise but most of the readings were within 1 inch of the actual amount for distance of 12 inches to 3 inches. Later testing with the robot on grass revealed that over certain areas of grass, especially dead grass, the sensors are unable to get a readings at all. This will be discussed in a later section.

4.1.4 Maxbotics EZ1 Sonar

The MaxBotix Ultrasonic Range finder was successfully able to identify trees and bushes from at least six feet away and overall was very successful at detecting large objects it was pointed at. When tested against the wall, the radius of detection was eight inches around the sensor for objects 18 inches away and 12 inches for objects 30 inches away, verifying that there is a cone area that the sensor can detect obstacles within.

4.1.5 LIDAR

The Scanse LIDAR was successful in detecting objects such as trees, buildings and cars outside in the sun at distances of at least five feet. A different LIDAR was selected to be tested but was defective from the manufacturer. The Scanse LIDAR was tested instead because it was available during the scheduled testing period. It is known to have defects and it did show these defects. After a period of minutes to seconds, the spinning LIDAR would start slipping which caused the orientation of the laser to be lost to the logic circuit inside the LIDAR. This caused the LIDAR to enter an error state. This was solved by slowing the LIDAR down to a rotation speed of 1Hz and tightening screws on the bottom of the enclosure. The LIDAR ultimately was never used for obstacle avoidance due to not having enough time to implement the feature.

4.2 Slider Testing Procedures

4.2.1 Manual Movement and Interference Checking

The system will be moved manually before the code is run. This way the system can be checked for normal operation before letting the code take over. Along with making sure the system is able to move, the system will be checked to see if anything is creating unnecessary friction or if anything is restricting movement. Errors with mechanical movement will be picked up during this process so that they may be fixed before the system is run.

4.2.2 Limit Switches Engaged

The limit switches will be tested to verify that the Raspberry Pi is receiving the correct logic when the limit switches are triggered. The buttons will be pushed manually to ensure that they can close. They will also be checked by printing out what the Raspberry Pi reads as the logic value. Performing these tests rather than running the system to check is a safer way than expecting the device to stop once it reaches a limit switch.

4.3 Slider Initial Results

Upon inspection of the slider's operation after initial assembly, the system did not perform as expected. There was a miscalculation performed by the team when they told the weld shop how to weld the aluminum plate that put the holes in the plate 0.25 inches above where they needed to be and 0.25 inches too close to the four-bar attachment points. Due to this mistake, a large amount of friction was introduced to the system between the coupler

and the rack gear. Along with this, the rack and pinion gears did not mesh because they were separated by that 0.25-inch distance. Because of these issues, at no fault to the weld shop, design modifications needed to be made.

4.3.1 Design modifications

Hole Position

In order to drop the holes downward to make up for the lost 0.25 inches, slots were cut into the aluminum. This also involved cutting a slot in the bottom plate for the screw heads to slide into. Doing so dropped the rack gear enough. However, there was still a small amount of friction present between the bump on the coupler and the rack gear.

Rack Gear and Coupler Switch Bump

The bump in the coupler that is responsible for engaging the limit switch was still rubbing against the rack gear even after dropping the rack gear down as much as possible. To combat this, both components were sanded down with 120 grit sand paper. This removed the remaining layer of plastic that was causing the last of the interference between the two pieces.

Rack Gear and Pinion Gear

To address the issue with the rack and pinion gear not meshing, washers were added to the machine screws used to secure the rack gear. The washers were used as spacers and placed in between the aluminum plate and the rack gear. This pushed the rack gear back out so that it meshed with the pinion gear.

4.3.2 Finalization

Reassembly

Once the modifications were made to the pieces that were not perfect, the system was reassembled. It was reassembled in the reverse order from which it was taken apart. This helps check to see if any parts were left out in between the stages of reassembly.

Verification

After the system was reassembled, the slider was moved back and forth by hand multiple times. This was to verify that the system could move back and forth physically. Verifying this allows for the ruling out of mechanical issues as to why the slider may not move. It also allows the team to see if the system will break itself if it ever tries to move. This verification checks for misalignment in parts and any physical obstructions that may not be visible to the naked eye. The fully assembled slider can be seen below.

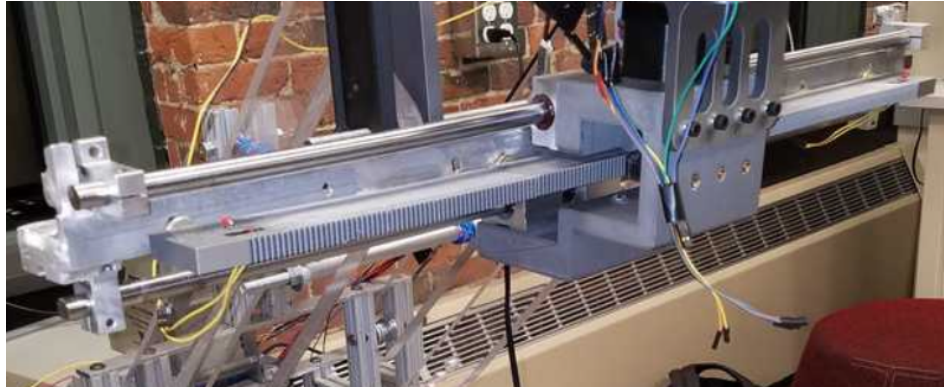


Figure 4.3: Assembled Slider

Greasing

Grease was applied to the bearings and rods to reduce any friction between the two components. Applying grease to the steel rods and bearings also helps seal out moisture. This protects the system from any type of rust and corrosion which can lead to system failures and inconsistencies.

4.3.3 Programming Fixes

Print State Messages for Debugging

After the code was uploaded to the Raspberry Pi and the ROS node was run, nothing happened. To help identify the issues present, print statements were implemented. After every major action was to be performed, a print statement would be implemented. This would help show if logic was being operated on properly. It would inform the team of which direction the slider was supposed to be moving in and what limits were reached, if any.

Setting Direction Pin

One issue that was discovered based off of the print statements was the fact that the motor would never reverse direction. The print statements were saying that the motor was supposed to go in the opposite direction, however, it kept going in the same direction. The reason for this was due to fact that the direction pin variable was switching to the correct logic for the correct direction, “1” for left and “0” for right, but the motor direction pin output was never being set to the variable’s current logic before moving again. This was a simple fix of just adding the following line of code after the motor direction variable was changed which is also written below:

```
self.CurrentMotorDirection = 1
```

```
GPIO.output(self.MotorDirectionPin, self.CurrentMotorDirection)
```

4.4 Four-Bar Results

Ultimately, after implementing the redesign of the drive system, the four-bar was able to lift the combined weight of the slider and sensor platform. The following section presents the observations and data concerning the implementation of both designs.

4.4.1 Initial Results and Observations

The initial assembly of the four-bar presented several problems which delayed testing. First, the large number of holes that needed to be drilled into the base plate caused misalignment issues, considering that it was decided to hand machine the plate due to ease of access. Several times, holes that needed to be in line, such as those to attach the 80/20 supports, were off by mere millimeters, causing binding in the driven axles. Correcting this thoroughly delayed the initial assembly of the mechanism. Next, the inclusion of the 3D printed motor mounts did not mesh the worm gear and wheel for each driven axle. For the purposes of testing initial operation, thin spacers made from cardboard were used to remedy this issue. This allowed for proper operation to be observed.

The initial tests of the full system locomotion, supported the expected operation of the initial drive system. This meant that the polycarbonate links appropriately transferred motion and power to the driven axles and that no binding or other interference occurred. Further, the worm gears did not disengage and the worm wheel did not turn the worm gear, as was expected. However, given that the printed mounts did not supply adequate enough support, the worm wheel still transferred force to the worm wheel, causing the entire motor and mount to shift. As the links were manually raised and lowered, the 3D-printed motor mount was seen to deflect up to a quarter inch in either direction, and print layers were

observed separating from those around them. Eventually the entire mount cracked from this stress. This issue necessitated a design change in the entire mechanism, as detailed in the methodology chapter.

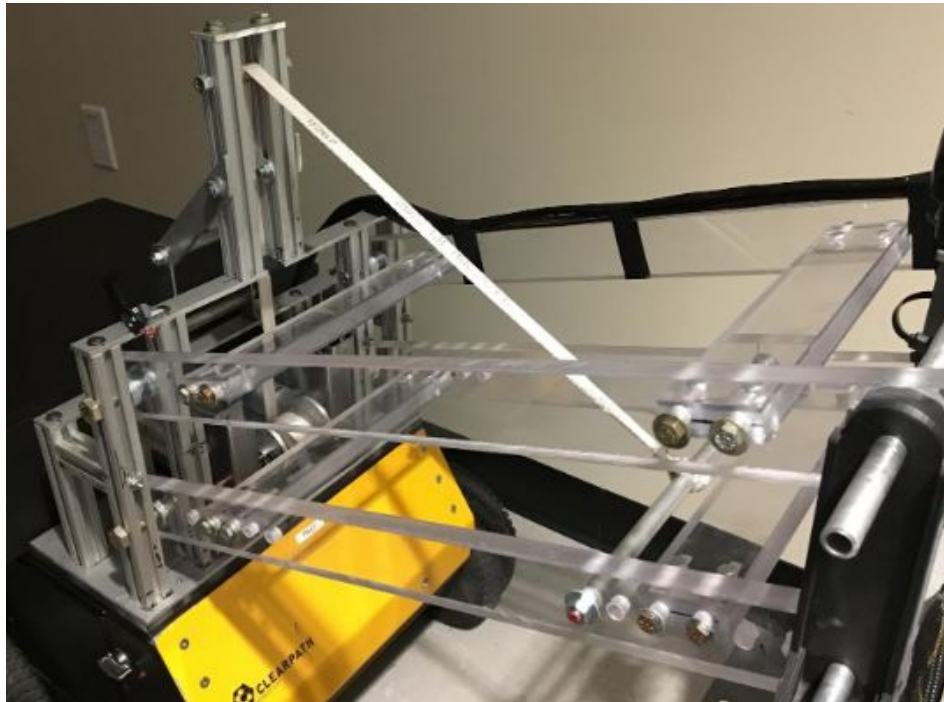


Figure 4.4: Assembled four-bar mechanism with redesigned drive system

4.4.2 Component Redesign

As previously described, a MATLAB script was used to calculate the static loads on the joints of the four-bar as well as the torque required from the motors as the mechanism raised the slider and sensor arm from a horizontal position to the maximum rise angle of approximately forty degrees. Over 300 cases were generated for winch diameters of two or three inches, support heights between six and fourteen inches above the top of the ten-inch superstructure, and support members from halfway along to the end of the driven link. From each, a graph similar to the two seen below was generated. These graphs described the torque required from the motors to hold the system at the designated angle.

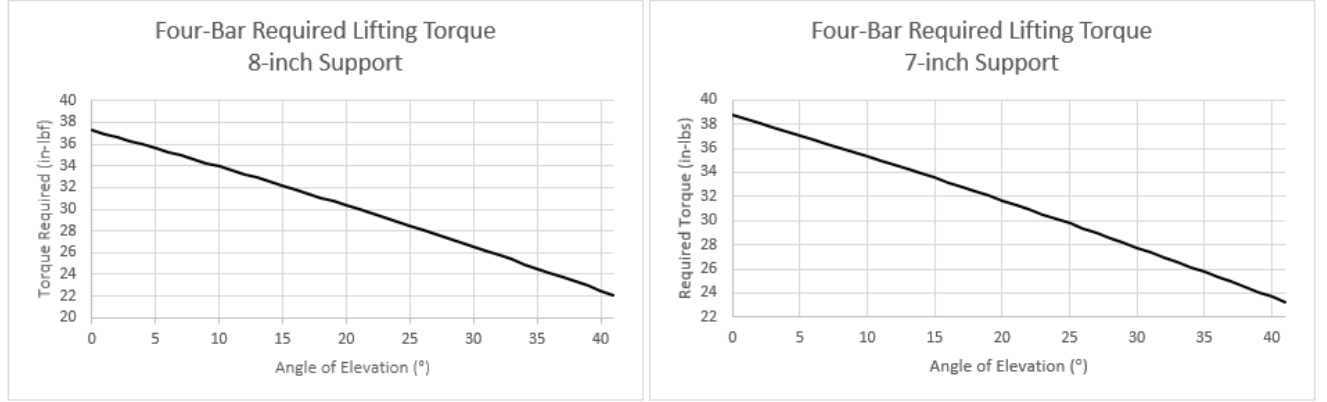


Figure 4.5: Torque required for drive system with support height of (a) 7 inches and (b) 8 inches

The above two graphs were the two cases consulted for the final system design. The two cases deal with a 2-inch diameter winch and a cross member distance of 17 inches, or approximately $\frac{2}{3}$ the length of the driven link. It can be seen from these two scenarios that the maximum torque required from the motors is approximately 37 to 39 inch-pounds force. This value was well within the operable range of the redesigned drive system. As seen in the following calculation, the redesigned drive system is projected to be able to produce over 850 inch-pounds of torque at stall.

$$T = 2 * (T_{Stall}) * (\eta) * (\mu) * (u) \quad (4.1)$$

$$T = 2 * (700 \text{ ozin}) * 0.70 * 0.70 * 20 \quad (4.2)$$

$$T = 2 * (43.75 \text{ inlb}) * 0.7 * 0.7 * 20 \quad (4.3)$$

$$T = 87.5 \text{ inlb} * 0.49 * 20 \quad (4.4)$$

$$T = 87.5 \text{ inlb} * 9.8 \quad (4.5)$$

$$T = 857.5 \text{ inlb} \quad (4.6)$$

In the above equations, T_{stall} is the stall torque specified for the Tetrax Torque-NADO motors, η is the efficiency of the worm gear system, μ is the operating torque set by the team, and u is the gear reduction from the Tetrax worm gear.

Further, the redesigned drive system decreased overall deflection of the metal detector. The following equations illuminate how the inclusion of the cross member support at 17 inches decreased overall deflection by 97%. The first equation calculates the deflection at the end of the links, past the cross member. This region behaves similar to a cantilever beam. Since the cantilever arm of the beam is significantly reduced - now only 5 inches - the initial deflection is also greatly reduced.

$$\delta_1 = \frac{F L^3}{3 E I} \quad (4.7)$$

$$\delta_1 = \frac{16lb(5in)^3}{3(0.377E6psi)(0.5in^4)} \quad (4.8)$$

$$\delta_1 = 0.00353in \quad (4.9)$$

This initial deflection is influenced by the deflection that occurs between the half-inch axle and the cross member support. To find these, the following two equations are solved for the two intermediate angles of deflection at the cross member. These propagate a deflection at the end of the beam as can be seen.

$$\theta_2 = \frac{W L_w (L_s^2 - L_w^2)}{6 L_s E I} \quad (4.10)$$

$$\theta_2 = \frac{3.6lbs * 11in * ((17in)^2 - (11in)^2)}{6(17in)(0.377E6psi)(0.5in^4)} \quad (4.11)$$

$$\theta_2 = 0.000475rad \quad (4.12)$$

$$\delta_2 = \theta_2 * L \quad (4.13)$$

$$\delta_2 = (0.00475rad)(5in) \quad (4.14)$$

$$\delta_2 = 0.00237in \quad (4.15)$$

$$\theta_3 = \frac{M L_s}{3 E I} \quad (4.16)$$

$$\theta_3 = \frac{(16lb * 5in)(17in)}{3(0.377E6psi)(0.5in^4)} \quad (4.17)$$

$$\theta_3 = 0.02404rad \quad (4.18)$$

$$\delta_3 = \theta_3 * L \quad (4.19)$$

$$\delta_3 = (0.02404rad)(5in) \quad (4.20)$$

$$\delta_3 = 0.012022in \quad (4.21)$$

With the two intermediate deflections solved for, the overall deflection can be calculated according to the following equation. In solving for the complete deflection, it should be noted that a "positive" deflection is a change towards the ground while a "negative" is just the opposite.

$$\delta = \delta_1 + \delta_3 - \delta_2 \quad (4.22)$$

$$\delta = 0.00353in + 0.12022in - 0.00237in \quad (4.23)$$

$$\delta = 0.01318in \quad (4.24)$$

4.4.3 Result of Redesigned System

After implementing the discussed redesign, it was clear to see that the combined load was lifted with greater ease than the original design. However, there were still some observed issues.

First, a great deal of screeching was heard upon initial activation of the motors. Careful inspection revealed that the gears were the source of this noise, as the bare metal of each was grinding against not only the other but also the wall of the steel channel. A liberal application of greased quieted this noise.

Further noise also came from the interaction between the polycarbonate links and the 3D printed L-brackets attaching the coupler plate to the four-bar. Given that cross-braces were added during the redesign, the distance between the outside faces of the links was greater than the original design. This meant that the the plastic from the brackets was rubbing against the polycarbonate, causing unnecessary friction. The brackets were reprinted such that the new outside dimension of 13 inches was respected.

Finally, it was observed that the four-bar had more trouble lifting the load at lower angles. This was to be expected. As they approached horizontal, the torque to turn the links about the driven axle increases to the maximum. However, given that the new design relies on the winch to pull the system up, as the links rotate away from the maximum rise angle, a toggle orientation is approached. At this orientation, the system is attempting to pull the mechanism and attached load into itself and the half-inch bolts acting as axles. This orientation occurs far below the horizontal, though, and was not experience during full system tests. Further, the mechanisms rarely even approached horizontal during full system tests. This was due to the sensor arm being taller than the 16 inches to horizontal and that the rover's difficulty navigating slopes - discussed elsewhere - dissuaded tests while descending a hill.

4.5 Sensor Platform Testing

During testing, most of the functions of the sensor platform were successful individually but when combined the ability to keep the metal detector parallel with the ground caused problems with other more critical abilities of the sensor platform. These results are

discussed in detail in the following sections. An image of the constructed sensor platform can be found in Figure 4.6.



Figure 4.6: Constructed Sensor Platform

4.5.1 Individual Component Testing Results

Testing of the metal detector code on the Arduino Uno was successful in its entirety. The metal detector coil was able to detect the same pieces of metal detected in previous tests and at the same depth. The Arduino Uno was able to translate the PWM wave output from the metal detector PCB and create a binary signal to be sent to the Raspberry Pi. The auto-ground calibration was also successful in that it will report an error if calibration was unsuccessful and operate as normal if it was successful.

The ultrasonic sensors were able to accurately measure the ground distance when parallel to a flat, hard surface like the floor but showed decreasing accuracy when at angles above 10 degrees. This mainly effected the two ultrasonic sensors in the front when raised farther above the table. When the readings were accurate, the software responsible for calculating the appropriate angle to rotate was also accurate along with the signal to raise or lower the four-bar.

The motors were tested not connected to any drive train to verify the motor direction in the software. Once this was figured out along with some other errors in the motor control code, the yaw and roll motors were rotating appropriate amounts and responding to limit switch presses as intended. The interrupt driven stopping worked well which was tested by giving the motor a movement command, then making the software enter an infinite loop. The motors would still stop after the set rotation has been completed. The software controlling the pitch rotation motor operated correctly, stopping the motor once the limit switch was pressed. The servo was able to actuate the spray can after values for the duty cycle were found that would move the lever arm appropriately.

4.5.2 Combined Component Testing Results

The individual components of the sensor platform were tested together, initially with the ultrasonic angle calculation and motor control running. The metal detector coil was able to be moved parallel to the floor and table when the sensor platform was moved at angles about 10 degrees or less consistently. Once the angle increases beyond this, the readings became very large, appearing that the sound waves were not bouncing directly back but off of other surfaces first. Not all five ultrasonic sensors showed this behavior at the same time but rather only one or two would have a very large reading, usually being the sensors that are rotated to be further from the table. These large readings caused the angle calculation to be in the opposite direction needed, moving the metal detector coil further from parallel with the flat surface. This was also tested using a large cardboard sheet. The readings were accurate when the metal detector coil was parallel with the cardboard but were inconsistent even at small angles. The calculation for the roll angle was correct more often than the yaw calculation, likely caused by the way the cardboard was bending from holding it by hand. The rotation was tested again with the spray can inserted into the holder. Running the motors at full power allowed the rotation about the yaw access to be successful, although the movement was slow. The motor controlling Roll was unable to turn the joint, resulting in stalling and the reset-able fuse to trip. Despite the torque calculations done previously, the inefficiencies in the worm gear drive as well as the large amount of friction between the 3D printed parts seemed to decrease torque to make the feature unusable with the spray can inserted.

The auto-orientation functions with the spray can removed were tested again with the metal detector operating and it was discovered that the magnetic fields generated from the motors were creating noise that the metal detector picked up. The motor controlling

pitch was too far to interfere but the servo, yaw and roll motors all caused interference when operating. Interference from the servo would not be a problem because it only runs while marking a landmine. To try and remedy this issue, the metal detector code was modified so that the Arduino Uno filters out this noise by reporting a longer duration of metal being detected, however this comes at the cost of slowing down the time it takes to report metal is detected and reduces the sensitivity of the metal detector, as small bits of metal can produce similar readings to this motor noise.

4.5.3 Full System Testing Results

Testing the communication between the Raspberry Pi and Arduino boards, while also commanding the infrared camera showed the system to work as intended. A live video feed from the infrared was retrieved and commands were sent to the Arduinos. Each Arduino was able to receive each command and execute them. Using the IICSem ROS node, the IIC bus was confirmed to only be accessed by either the infrared camera node or the sensor platform node at any time. If the IIC bus was in use and the other node needed access, it was stalled until the other task completed. Given how infrequent the infrared camera and sensor platform use the IIC bus and that only a couple bytes are transferred per command, testing the case where both nodes needed the IIC line at the same time had to be forced in order for it to be observed without waiting a while for it to occur naturally.

Testing the ability for the sensor platform to adjust to changing ground heights while the Slider mechanism was moving it was not done because the previous testing showed the sensor readings to not be accurate enough and the roll motor provided insufficient torque to rotate the roll joint with the paint can attached.

4.5.4 Sensor Platform "Slim" Redesign

Due to the problems discovered with the automatic coil adjustment function of the metal detector, the mechanical components of this feature were removed resulting in an additional redesign of the sensor platform excluding this feature. This is referred to as the "slim" model because it is lighter in weight and has significantly less wiring. This lowers the required torque to lift the sensor platform with the four-bar and would reduce any twisting or bending of components of the slider. The Solidworks model of this updated assembly can be found in figure 4.7. An image of the constructed assembly on the Slider mechanism can be found in figure 4.8. This assembly also includes the new configuration for the control boards and mounting plate to address problems found with the slider which will be discussed in later sections.

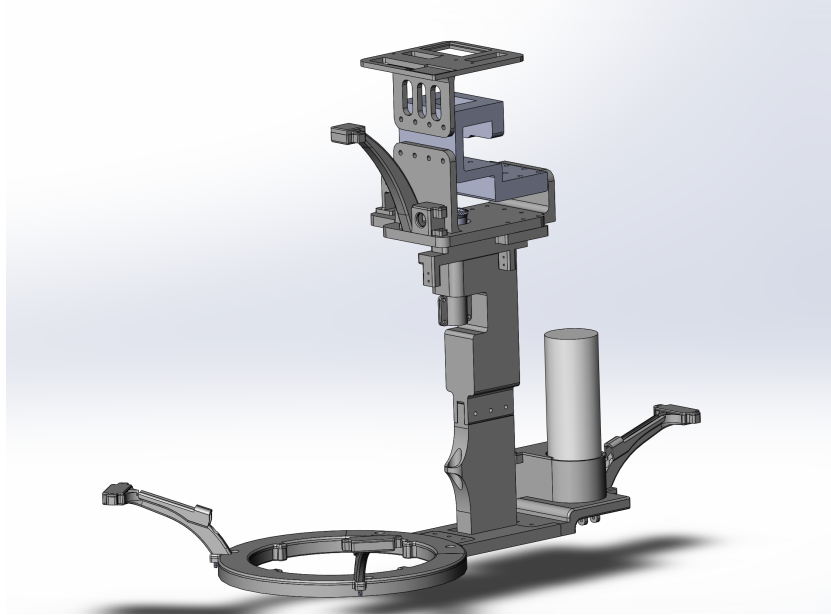


Figure 4.7: Solidworks model of sensor platform "Slim"

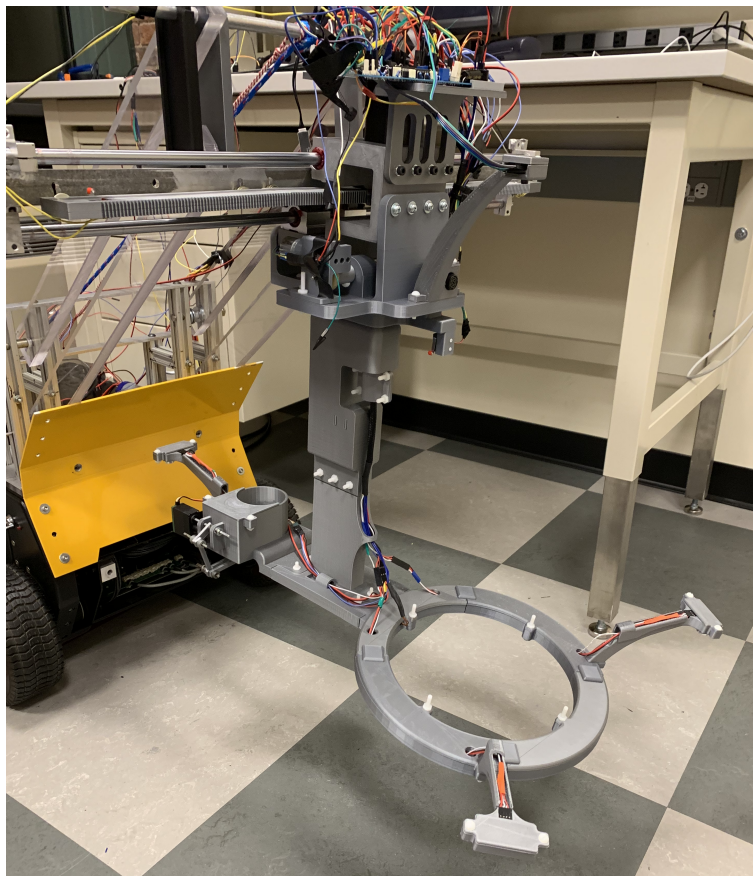


Figure 4.8: Sensor platform "Slim" attached to slider mechanism

The main design change is that the yaw and roll joints and all associated components were removed. A side effect of this change is that the spray can for the marking system is no longer positioned diagonally and off to the side of the sensor platform. In the previous design, orienting the spray can vertically and centered would cause it to hit the sensor arm at certain orientations. The change to the slim design shifts the center of mass of the sensor platform to be centered along the width of the slider rather than offset to the left. This resulted in less bending of the 3D printed components and thus less friction in the pitch joint, allowing for smoother rotation than before. Second, without the need to adjust the orientation of the metal detector coil, the five ultrasonic sensors were reduced to three and used solely as feedback to control the height of the coil via the four-bar, ensuring that it remained one inch above the ground. The two sensors in the front will provide an average reading for the height in front of the sensor platform, while the one in the rear will ensure the sensor platform is not lowered into anything without proper clearance. For example, should the rover scan as it descends a slope, the rear sensor would still detect the ground, while the foremost two would declare an all clear. Finally, a lesser issue with the previous design was the prevalence of inadequate wire-ways. Therefore, the slim design added additional paths for wires while simultaneously reducing the overall amount of wiring needed. To present a cleaner and more organized wiring configuration, wire connectors were crimped on where necessary.

However, despite the benefits, the slim design did introduce some new obstacles. Given that this new slim design does not have the same level of mobility, the process to create the painted square mark identifying a landmine was altered. Instead of rotating the platform, the mark will be created while the robot backs up as part of the planned navigation code. This required new commands so that the sensor platform will rotate the spray can out, hold it there for the duration of the back up movement, and then rotate the coil back into the original orientation. Therefore, the code on the Arduino Mega was rewritten to accept an "Extend Paint" and a "Retract paint" command. The "Extend paint" command rotates the paint can and actuates the servo to spray the paint. Meanwhile, the "Retract paint" command releases the lever arm depressing the paint can and rotates the metal detector coil back out. The sensor platform ROS control node also had these commands added so that the system could be tested. Testing was a success with all new functionality working as intended.

The ultrasonic sensor now only needed to be used to keep the Sensor Platform from hitting the ground so it was simplified. Now, Five sensor readings from each of the three ultrasonic sensors are taken and averaged to reduce noise in the readings every time the sensor arm checks the ground distance. Now that the software no longer needs to monitor the yaw and roll motors, whenever a command from the Raspberry Pi is not being processed, the ground distance is being measured. Readings could take as long as 70ms each if there is a cliff or other large drop but under normal circumstances, a readings should only take about 0.9ms each. 15 readings are taken before making any calculations, which would take 13ms to execute. This allows the Sensor platform to take about 76 readings of the ground per second allowing for fast reaction times to changing slopes. This is a significant improvement over the old implementation that could only take two ground readings per second. The software will

use this data to ensure that each ultrasonic sensor is at least the minimum desired distance to keep the metal detector coil raised 1in above the ground. If any of the sensors are too close to the ground, the four-bar will be told to raise. If all of the sensors are too far from the ground it will be told to lower. Otherwise the four-bar is told to sit idle. This functionality was tested with the completed four-bar and it proved to work well on surfaces that are flat to properly redirect the sound back at the sensor. The responsiveness to changing ground heights had no observable delay and the system would not overshoot the intended height.

4.5.5 Integration with Slider Results

With the Sensor Platform operating on its own, the next step was to integrate it with the slider and determine algorithms to move the metal detector to best detect metal. When the Slider's stepper motor is powered off and all other metal is removed from the area, the metal detector was as sensitive as the previous sensor testing results had shown, receiving no interference from the metal structures on the rover. With the stepper motor powered, however, a significant magnetic field was generated that caused noise in the metal detector readings. As a temporary fix, the Arduino Uno filtered these readings through software by timing the duration of the noisy signals. A real detection of metal would cause a reading to last much longer than the noise from the stepper motor. The threshold was determined through trial and error, increasing the time duration until the noise was no longer detected while still detecting real metal. This comes at the cost of detection depth though. Currently, the metal detector can detect 4 grams of metal at a distance of 2 inches which is about half the distance it can with the Slider stepper motor turned off. The interference from the stepper motor would have to be removed to regain full detection depth.

The next step was to test metal detection with the Slider moving back and forth. As the metal detector passed over a piece of metal, it would read it while the coil was directly over the metal but not when the metal was in the center of the coil, resulting in two distinct readings. This behavior influenced the algorithm for detecting landmines to search for two readings per landmine instead of one.

4.5.6 Subsystem Controller ROS Node Creation

To properly integrate the Sensor Platform, Slider and Navigation algorithm, the Subsystem Controller ROS node was created. The main feature is that it coordinates the Slider and Sensor Platform to scan for two readings per landmine as observed in testing. After the Subsystem controller has received data from the Slider, signifying the Slider has homed itself, it will wait for the Navigation Algorithm for a sweep command. The Subsystem Controller will then signal the Slider to freely sweep while the controller monitors both the landmine detection node and the slider position. When metal is detected, the Slider position when this metal is detected gets logged and the Subsystem controller then waits for the detection signal to stop. The stop location is also logged. The Subsystem controller will

then let the Slider move the distance of two motor rotations as it waits for the second reading of the metal. After the second reading of metal is received and ceases, the average Slider location is calculated and sent to the Slider to move to. If after 400 steps have passed and the second reading has not begun, the calculated position will be in-between where the metal was first and last detected for the first reading. This scenario occurs when the metal only passes under the metal detector coil near the front or bottom of the coil. Once the Slider has reached the calculated position, the Subsystem controllers signals the Sensor Platform Controller to extend the paint can and begin spraying. The Navigation algorithm is also told that a landmine has been found at the current Slider location. Once the Navigation algorithm has moved the robot back to make the square mark, it tells the subsystem controller to continue sweeping. The paint can is retracted and the Slider will then continue its sweep across the front of the robot. Once the Slider reaches the other end of the rack gears, the Subsystem Controller will command it to stop moving and notify the Navigation algorithm that the sweep is completed. This node simplifies the communication between the navigation and the rest of the system to only five total ROS messages, two of which the Navigation algorithm sends and three that it receives.

In testing, this node performed as intended, however it showed that the ROS system was having trouble keeping up with all of the processes. The Slider posts its current position once per step increment, so in a system with fast processing and communication, the Subsystem controller should be able to know exactly when the Slider reaches the end of the rack gears. Due to the execution time of all the processes on the Raspberry Pi, the subsystem controller would often miss that the Slider reached the end of the rack gears, causing the Slider to continue to sweep back and forth until the Subsystem controller caught it at the appropriate bound. To prevent this, the subsystem controller was given a 10 step threshold so that if the slider is within 10 steps of the end of the rack gear boundaries, it would command the slider to stop sweeping. Communication with the Navigation Algorithm wasn't observed to have any problems, working as intended.

4.5.7 Outdoor Testing Results

The full system was tested on a flat portion of grass in Institute Park and in the parking lot of 85 Prescott Street. The ultrasonic sensors were found to work well when on the pavement of the parking lot but their reliability decreased dramatically on the grass in Institute Park. Over areas where there was little grass, the sensors were able to get an accurate reading but over soft or dead grass they were unable to read the ground distance at all, returning a distance of 0cm which is the behavior when an echo is not received. Not every sensor returned this value at the same time, depending on the type of terrain they were over, but the system requires accurate readings from all of the ultrasonic sensors to work properly.

The metal detector performance in the parking lot was inconclusive due to there being railroad tracks underneath the parking lot causing constant metal detection readings

during movement. When testing the system in Institute Park, the interference from the stepper motor seemed to have completely subsided, however the ultrasonic sensor ping and echo wires were creating interference when signals were sent across them. After disabling the ultrasonic sensor following the discovery that they do not work well consistently on grass, all magnetic interference was gone even when the Slider was operating. The four Bar was raised about 5 inches above the ground since the automatic adjustment was disabled. Small samples of vex steel were laid on the ground and the metal detector was able to detect them, better than expected with the interference no longer present.

4.6 Navigation Protocol

The navigation protocol was adapted from simulation to work on the physical rover, however during this process a few changes needed to be made. The initial tests were done before the sensor arm was attached to the rover. This allowed the team to minimize potential damage to the sensor arm while also allowing rover to start being tested before the sensor arm was completed.

4.6.1 Straight Line Test

The first experiment that was run was to test the rover traversing a single row of a minefield. In order to do this the rover had to move in a straight line stopping every 20cm to simulate the scanning protocol. There were some issues with the Husky drivetrain which are discussed in Section 5.1.1, but after those issues were resolved the test was a success. The encoders on the Husky proved to be accurate to within one centimeter during this testing which is adequate for our application.

4.6.2 Back and Forth Test

The second test was to have the rover traverse an area of the minefield slightly wider than the rover. This would cause the rover to scan the first row of the minefield, turn, move forward until it was in the unscanned area of the minefield, turn again and return in the direction it originally came from. This test ran into some issues because the Husky did not come equipped with an IMU (Inertial Measurement Unit) as originally thought. This device calculates the change in angular position of the rover and is important for being able to accurately turn to the right position. Therefore, a supplementary BNO055 IMU was added to the rover. This supplementary IMU proved to be very accurate. It was able to detect changes in orientation of less than a degree and was able to function for an excess of ten minutes without drifting. The IMU was also the first node that was being run on the Raspberry Pi instead of the Husky's onboard computer. As a result, at this point the communication

between the Raspberry Pi and the Husky computer had to be configured. The first issue that was encountered was that the rover's onboard power supply did not provide enough power to run the Raspberry Pi. An external battery was added to the system in order to properly power the Raspberry Pi. This battery provided power for a longer period of time than the battery's for the rest of rover so the team felt that this solution was sufficient. After the Raspberry Pi was properly powered, it had to be configured to be able to send and receive messages from the onboard Husky computer. Before the onboard computer and Raspberry Pi could communicate, an onboard router had to be installed and connected to both of these computers. The router did have some issues as described in Section 5.2.1, but it performed satisfactorily for small scale tests. Upon the successful communication between the Husky onboard computer and the Raspberry Pi, the rover was able to perform the previously described test without any additional hiccups.

4.6.3 Simple Minefield Test

The next test was to have the rover traverse a larger minefield (approximately 3m x 3m). During this test the team also simulated the rover detecting a mine and had the rover navigate around it. Due to the sensor arm still not being attached at this point, the team simulated the mine with a keyboard input. The robot was able to perform this task with minimal complications.

4.6.4 dGPS Integration Test

Afterwards, the rover's integration with the dGPS system had to be tested. To accomplish this task, the team took the rover's current dGPS location at the beginning of the test and constructed a map for the rover to traverse based off this data. The rover during this time would be continuously polling its current dGPS position and correcting its position on its local map in order to match this data. A few problems arose when testing this functionality. First, the base station for the dGPS system takes a significant amount of time in order to be able to self-survey and find its position. Even with giving the base station a significant amount of time to locate itself, it would only get to an accuracy of about a meter. Additionally, once the base station is set up, the rover dGPS unit then takes a significant amount of time communicating with the base station to achieve the desired level of accuracy. Overall, this process takes over half an hour every time you set the rover up in a new location. Additionally, the dGPS units are not meant for urban environments. As many minefields in the world are not in urban environments, the team felt the dGPS systems would still be a useful addition the project. However, since the team is located within a city it was difficult to find areas to test the system to its full capacity. Once a satisfactory location had been decided upon, testing could begin. The IMU added to the robot had a compass but it was not used and the robot orientation was assumed at startup. When the rover was in the correct starting orientation, the rover could successfully navigate a minefield with dGPS

feedback. The compass on the IMU should be implemented in later work to avoid making assumptions.

4.6.5 Google Maps API Integration Test

The penultimate test performed was integrating the sensor arm with the navigation code. This test was performed inside to ensure that the system worked in a controlled environment before moving it to an environment with more potential to cause damage to the rover. During this test, the metal detector was turned off due to the copious amounts of metal in the floor, however the ROS message that the metal detector sends upon detecting metal could be manually sent from the base station. The rover did have some complications when the sensor arm was attached due to the fragile nature of PLA plastic. If the rover moves too fast, especially during turns, a significant amount of stress is put on the sensor arm. This caused a few pieces to shear during testing. In order to combat this, the pieces were redesigned to be more rugged and the rover was command to turn at a slower pace. Additionally, due to the slow pace at which the rover turned, the navigation algorithm was modified in order to prefer straight lines to excessive turns. This causes the rover to navigate minefields in a spiral pattern as opposed to a sweeping pattern. At this slower pace, the rover was able to navigate the minefield successfully, scanning every square and avoiding mines that were detected by a message from the base station.

4.6.6 Full System Test

Lastly, the team performed a full system test outside. During this test, the rover was able to navigate the minefield, stopping to scan every 20cm, and scan for metal. If it detected metal it would immediately start the marking procedure: center on the metal, turn the sensor arm so that the paint can is over the metal, simulate painting the mine and return to default position to finish the scan. Then it would turn and avoid the piece of metal while finishing its mapping of the minefield. Due to the difficulty that the rover has turning at low speeds on difficult terrain, the motor controllers sometimes act a little erratically. Otherwise, the rover performed its task adequately.

4.7 Drivetrain Results

4.7.1 Failed Analysis

Analysis of the tires was attempted in ANSYS to no avail. Hours were spent learning the ANSYS program with Dr. Adriana Hera. ANSYS was used to test the flexibility of the tire designs. The tire designs were modeled separately to ease simulation strain on the

computer. After setting up the simulation, it was run, accurately showing deformations in the tire walls at a small scale. As soon as the simulation was completed (the full load was placed on the tire), the tire walls started intersecting each other. This was an issue because contacts between each of the intersecting walls were set and present. The contacts were effectively being ignored which is not representative of real material properties.

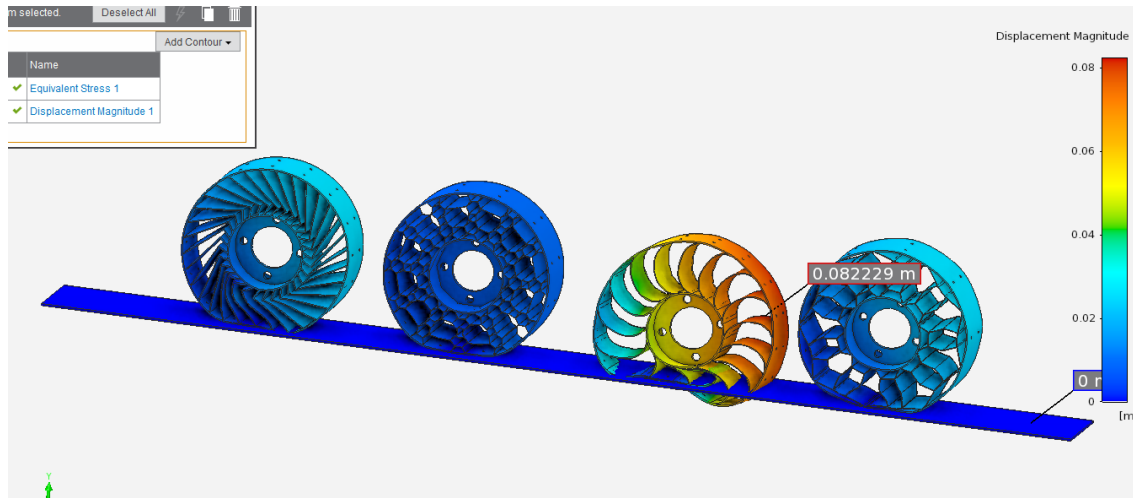


Figure 4.9: Simulation Results

To attempt to mitigate and fix this intersection issue, the tires were all put in an assembly with a ground piece. The force was also switched to being applied to the axle hub instead of the outer wall pushing inward. This worked better overall, but the walls were still intersecting with each other. Another setting was still wrong that the team was unaware of.

A 2D simplification was sought after because the point of the ANSYS simulation was to model deformation patterns. This would demonstrate the directions that the tire's inner walls would bend in to see if the tire would offer a suspension of sorts, or if it would just crumple under a load. The 2D simplification is not necessary to run the simulation, it just reduces the amount of meshing time before the simulation is run along with the time it takes for the solution to be generated.

One issue the team had with ANSYS was that WPI has many different versions of ANSYS installed. The version that the team needed to use was titled "ANSYS Workbench" whereas the team was using "ANSYS AIM". The program ANSYS AIM was misleading because it seemed like it had the appropriate tools for completing the study. There were different modes to select such as "Static" and "Thermal" analysis options. There may be a way to simulate the tire in this mode, however, the team was informed that this was not the correct version of the program to use after struggling for weeks to get the simulation working.

Switching over to ANSYS Workbench didn't fix the simulation entirely. It made the process of setting up the simulation easier, but the simulation was still failing. Further time was spent adjusting various resolution settings among other simulation parameters along

with redefining the contacts and forces within the simulation. This proved to not be as helpful for the amount of time put into changing the settings.

The last setting that was explored was a “Large Deformation” setting. The team didn’t look for this setting because it was assumed that if the simulation ran, the setting didn’t need to be selected. In the team’s eyes, the deformation being measured wasn’t large either, ruling out the large deformation setting. Alas, a test was run on the tire shape informally with the “Large Deformation” setting selected. Even though some of the contacts were set incorrectly, the contacts that were correctly set allowed the tire to deform correctly. This setting needs to be selected for any deformation that is large, relative to the size of the part.

This was done very quickly and on a tight schedule, so screenshots were not able to be grabbed with the tire deforming correctly.

4.7.2 Materials Testing

The team tested materials even though the simulation was not completed. This was to see if the tire could even be produced with the campus resources available.

Nylon was tested by the team. It was found to be too stiff for the thickness of the spokes that could be printed. Due to printer restrictions, the nylon is not able to be 3D printed at a thickness that would yield the correct bending properties. A test bench with various thicknesses was printed to test wall thicknesses that could the printer could handle. This piece stepped through the thicknesses, increasing them from 0.025 inches to 0.15 inches by 0.0125 inches between each step. This demonstrated the flexibility of the nylon. It was determined that the cyclic loading of the wheel would deform it, not allowing it to return to its original shape.

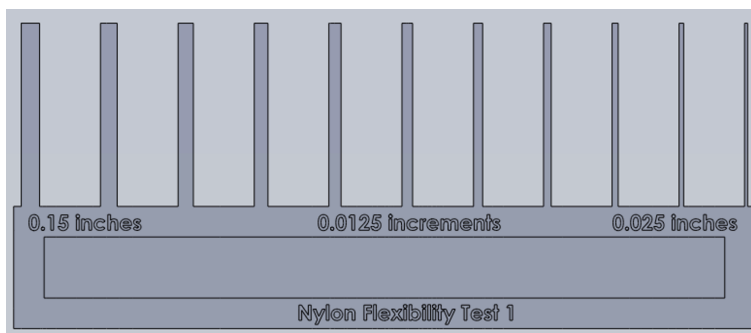


Figure 4.10: Simulation Results

The piece was bent back and forth in slow succession to simulate cyclic loading. Further testing to validate this claim needs to be done, but the team believes that nylon is not the best material to use.

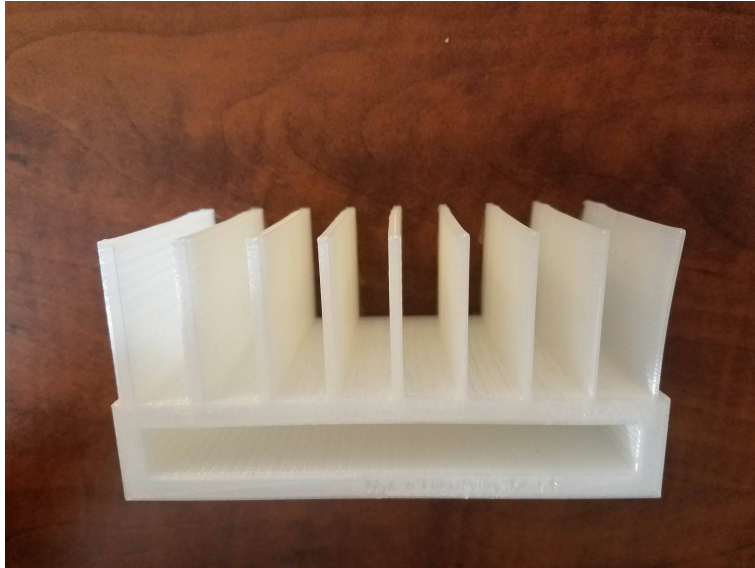


Figure 4.11: Simulation Results

TPU is a type of flexible material that was explored for a time but proven to be too difficult to 3D print using the available on-campus resources. This seems like it would be an ideal material that yields the correct elasticity for the task at hand. However, the manufacturability of the piece using this material proves to be out of the realm of possibility for the WPI campus.

Chapter 5

Discussion

5.1 Problems with Equipment

A number of unanticipated problems were encountered when getting the Husky rover - previously dubbed Polly - operational that delayed progress in some areas. This section serves to cover these issues such that future iterations of the project will have additional information on how to solve similar problems.

5.1.1 Husky Drivetrain

A major problem encountered with Polly was that the drive motors were unable to be properly controlled. The keyboard control program provided by Clearpath should have resulted in the rover driving forward at 10 centimeters per second. However, when this command was sent, drive motors instead would spin at full speed and quickly attempt to change directions. The sudden power draw from this would often cause the on-board computer to lose power. Further, once a command was sent, the motors would not stop until the control program was terminated via the keyboard. The control program also allows the user to stop the robot by pressing the backspace key, but this had no effect. Upon inspecting the motors, motor drivers and control board, it was found that the PWM signal cable powering the right side motor had an unknown substance melted onto the connector. This cable was removed and cleaned. With the right side motor disabled in this way, the left side was tested. The left motor was able to be controlled successfully. However, the motor was still moving faster than the commanded speed but not quite at full speed as before. Further, and most importantly, the control program was able to appropriately stop the motor once started. With the knowledge that the left side motor and cable worked correctly, the right side motor was connected to the motor driver via the left PWM cable. Despite receiving the opposite desired signal, the right motor operated the same as the left had previously. This meant that the problem originated with the right PWM cable.

Following this, the right cable was appropriately cleaned off and inserted into the left side PWM connector. With the cables swapped as thus, the motors were able to be controlled as desired. The exact cause of the motor troubles is not known. However, it is obvious that the previous configuration combined with the melted substance on the PWM cable was somehow interfering with appropriate signal generation for both motors. In the interest of time, this issue was not investigated further, given that the rover was able to be driven at a sufficient enough speed for the task at hand.

5.1.2 Operating System Upgrade

The operating system on Polly's installed computer was Ubuntu 12.04 LTS with the Desktop GUI installed. This was sufficient for keyboard control. However, many of the ROS packages provided by Clearpath for navigation required Boost Chrono - a C++ library which was unavailable for this operating system. Given that the project planned on using the official ROS packages, an upgrade to the operating system was required. Clearpath does provide images for the A100 Husky rover on a variety of Ubuntu versions, although the specific repository was difficult to find since the link was broken on their website, at the time of this report. With that, Ubuntu 14.04 server was installed onto Polly's computer which required the hard-drive to be formatted such that important files were backed up prior to the installation process. The server version of Ubuntu does not include a desktop interface so the Ubuntu desktop was also installed. The desktop installation, however, caused a program called Modem manager to crash, resulting in the Husky being unable to connect to the internet. After a number of re-installations and trial and error, the only desktop gui installation available that did not cause this issue was the Lubuntu minimal desktop installation. After confirming proper operation of the Modem manager, other needed applications were able to be reinstalled. The specific cause of the modem manager crash is not known. Internet forums concerning this crash pointed to a variety of different problems, with no single solution being presented. This suggests that the on-board computer is in need of an upgrade to support newer software. Clearpath provides installations of Bionic Beaver, the latest version of Linux, for the Husky and it may be beneficial to upgrade the computer to this version. That way, it's possible to take advantage of whatever further upgrades clearpath has made to the Husky software in the ROS melodic version. For the purposes of this project, though, any further upgrades were not required.

5.1.3 Computer Damage

Polly's on-board computer contained four holes in the casing that allow it to be placed on four bolt heads inside the robot enclosure to fix it in a particular orientation. During a re-installation of the computer, it was noticed that one of the bolts was missing. This was dismissed, as it was assumed that it was simply lost at some point. It was quickly found out that the reason for the missing bolt was to prevent damage to the computer. Unaware of this, the computer was placed on the screws in a more obvious and practical

orientation which resulted in stick of ram being destroyed. The computer would not boot until the damaged ram was removed. Fortunately, the computer had dual channel RAM so it was still operational. Unfortunately, the computer was only left with only two gigabytes of RAM instead of the original four. To prevent further damage in the future, the computer was insulated from the bolt heads by covering each in electrical tape and then adding a layer of cardboard over the four.

5.1.4 Raspberry Pi Hardware Issues

As previously mentioned, a Raspberry Pi was used to off load some processes from the main computer. To do this the Pi's GPIO pins were utilized. A number of hardware problems were discovered during testing that are important to note in future iterations if this board is to be used.

The first issue is that there is a known hardware bug between the I²C hardware of the Adafruit BNO055 IMU and the Raspberry Pi that prevents proper communication between the two devices (DiCola). This required the IMU to communicate over the RX and TX pins of the Raspberry Pi. This may have caused the timeout error between the IMU and Raspberry Pi that required the IMU program to be restarted or, in some cases, a full power cycle of the Pi and the IMU.

The Raspberry Pi also had a number of pins that caused unintended behavior when used. For example, When trying to use Pin 29 as a control signal for the four-bar mechanism, the Raspberry Pi would display the shut down screen and disable mouse clicks. Once the four-bar program was killed, normal control returned. After changing the pin to another nearby pin, this behavior stopped. Other pins nearby (they were not noted) also were unable to be controlled by the GPIO library despite the Raspberry Pi Documentation saying otherwise. There may be more pins that exhibit abnormal behavior that were never used. All of the pins currently in use are known to function properly. These problems decrease the amount of GPIO pins available and limit future expansion of peripherals. It is unknown if the Raspberry Pi has been damaged for this behavior to occur or if it is a flaw in the hardware.

5.2 Improved System Architecture

Although the system was shown to function as intended, there are a number of improvements and optimizations that should be implemented into the rover before deployment into a real minefield.

5.2.1 Network Communication to the Base Station

The rover currently has a standard Linksys router connecting the Husky computer, Raspberry Pi and Base station but this router would not be suitable for rover exploration of a large minefield or one that is far from the base station. To avoid losing wireless communication over WIFI between the rover and the base station, a high power outdoor WIFI system should be used instead. Many options exist that could enable communication over many kilometers with a clear line of sight. An example of a system that may be sufficient for long range communication is shown in Figure 5.1. These are antennas that communicate directly with each other with a range of up to 5km (Newegg.com). Deployment of the rover to a minefield that is 5km away is not recommended since the batteries will likely be drained before reaching it. Antennas with this amount of power may make communication between the base station and rover at shorter distances with many obstructions possible. Testing would need to be done to verify this but it is known that the current Wifi router has a short range and is easily obstructed by walls and floors. It is important to maintain communication between the base station and rover so that detected mine locations can be transferred in real time. In the event the rover detonates a landmine accidentally, the locations would be backed up to the base-station so their position would not be lost.



Figure 5.1: Example of long range WIFI Antenna (Newegg.com)

5.2.2 Computer Architecture

The rover has four on-board computation units, which are the Arduino Uno, Arduino Mega, Raspberry Pi and Husky Computer. The main reason for offloading so many processes to the Raspberry Pi and Arduinos was because the team was familiar with using them and their GPIO pins were needed to communicate with all of the peripherals. They proved to be sufficient in completing all of the needed tasks but added complexity to the robot with large amounts of wiring to connect it all. The ROS system was operating slower than expected due to the large amount of ROS nodes running in parallel and exchanging

messages between the Raspberry Pi and Husky Computer. This is especially noticeable when the slider is trying to move to an estimated landmine location as the speed of travel is inconsistent and severely slower than usual. The slider speed is set constant for all movements in the program but due to the software architecture and limited amount of threads able to run in parallel on the Raspberry Pi, the time between step pulses slows down significantly when the slider is being commanded and monitored by other nodes in the system. The slider speed is also slowed when the Raspberry Pi is also trying to raise or lower the four bar mechanism. All exchanged ROS messages between nodes in the Raspberry Pi have to be sent to the ROS master on the Husky computer, adding additional latency and slow down. Optimizations in code could be made to reduce the number of messages and optimize thread execution but it is recommended that future iterations of the project upgrade the Husky computer and use GPIO expansion boards to so that all processes can be run off of one system, removing the need to have multiple computers communicate over IIC and Ethernet as in the current system. The upgraded system should have a powerful GPU, plenty of RAM and many CPU cores with a faster clock rate than currently on the Husky. This would enable many more ROS nodes to be run in parallel and allow future teams to process data on-board from powerful sensors such as 3D LIDAR for obstacle detection with little latency. This would also allow for the installation of more modern releases of Ubuntu as discussed before. The new computer could be a pre-built system but if a future team has the time, it may be more beneficial to build the computer from scratch to meet the needs of the robot and fit the components into a custom case that fits inside the rover more efficiently and neatly. Building a computer from scratch may be cheaper in cost.

5.3 Slider Discussion

5.3.1 End Result

The end result after assembling, testing, and revising the system, was that the slider became fully functional. The slider moves the metal detector back and forth in between set bounds after being initialized. It is now ready to be interfaced with the rest of the robot.

5.3.2 Success

The slider successfully coupled the sensor arm to the four-bar via the linear slide system. The code was designed in a way to catch any issues or errors with slider control. It will make sure the slider reverses direction if at any point the slider goes over the bounds. The motor direction pin was successfully switched at each of the slider bounds that were set with the code. The slider never over-shot any of the limits. Even if it did, the limit switches properly switched the motor direction pin.

The linear bearings slid with no resistance to the stepper motor movement. Bending

only occurred when the robot was moving and even then, the flex didn't interfere with the slider operation.

The stepper motor was a good decision for the design challenges. This way, if the motor controller ends up breaking, the motor will stop moving because supplying the stepper motor with an unregulated DC voltage source moves the motor only one step in any direction instead of continually turning an armature. The steps and location of the stepper motor at any one instant was successfully tracked. The slider also properly centered itself when it detected metal.

5.3.3 Improvements

To improve this part of the robot, many steps can be taken. First, the 3D printed parts should be replaced with machined aluminum. This makes the parts stronger and less flimsy. Next, with more funds available, a better bearing system should be employed. This will reduce the bending within the rods. Lastly, the L-bracket should be re-manufactured so that the holes line up as intended.

5.4 Four-Bar Discussion

The four-bar mechanism ultimately worked to the specification of lifting the sensor platform in response to changes in the terrain. The use of polycarbonate links mitigated vertical deflection, preventing the slider from coming in contact with the ground. The redesigned drive system mitigated torque requirements from the TorqueNADO motors, which in turn reduced torsional strain and shear stress on the drive shaft. Further, the addition of cross members reduced angular deflection of the mechanism from over an inch to less than a half of an inch.

5.4.1 Improvements

While the overall system worked to the specification, several improvements could be made to improve the overall performance. These improvements include the overall material quality, machining standards, and assembly.

Foremost, the improvement of the overall quality of materials would further aid in mitigating deflection of the overall system. The primary links of the mechanism were composed of $\frac{3}{4}$ -inch thick polycarbonate links. The inclusion of these links was primarily due to the fact that they were donated to the project. As mentioned in Section 3.3.2, these links produced a 0.3 inch deflection when unsupported by the mule tape. This deflection could be greatly reduced by implementing a stronger material. For example, aluminum links

with the same dimensions would deflect approximately 0.01 inches. While aluminum is more dense than polycarbonate - aluminum has a density of $0.098 \frac{lb}{in^3}$ while polycarbonate is $0.0434 \frac{lb}{in^3}$ - the strength and durability of aluminum allow for smaller overall link dimensions.

Material	Young's Modulus (psi)	Density (lb./in ³)	Deflection at 24 in (in)	Weight (lbs.)	Cost (USD)
Aluminum	1.02E+07	0.098	0.173	1.171	\$\$
Steel	2.76E+07	0.278	0.064	3.338	\$\$\$
Maple	1.83E+06	0.023	0.967	0.271	\$
Oak	1.60E+06	0.021	1.109	0.257	\$
Polycarbonate	1.35E+07	0.043	0.131	0.520	\$\$

Figure 5.2: Weighted decision matrix for material selection

Next, the motor mounts should be substituted. The final mounts comprised of welded steel channels and attached Tetrax DC mounts. Tolerances in machining led to the gears not meshing properly and the entire drive system being slightly crooked. These mounts should instead be machined out of a single piece of aluminum or steel in order to fix these issues. Finally, the ABS printed angle brackets attaching the four-bar to the coupler plate should be replaced with machined aluminium with bushings between them, the links, and the half-inch alignment tubes.

Next, the issue of manufacturability should be addressed. Along the same lines as the polycarbonate links, the 1-inch square 80/20 extruded aluminum channels that comprised the superstructure for the mechanism was used primarily because it was donated to the project. While this was sufficient for the task at hand, it presented several issues when attempting to drill the holes for the axles. Since it was not a flat surface, attempting to accurately and repeatedly mark out the locations to drill and cut caused misalignment issues when first assembling the superstructure. During initial movement tests, there was significant binding in the drive and driven axles. To improve this, sheet metal should be properly machined using CNC mills and welded together to replace the current structure. This same process should be applied to the entire mechanism. As mentioned above, the thinner aluminum links replacing the polycarbonate should feature welded cross members. Further, the links could also be directly attached to the output gear or shaft from the gearbox. This would allow for the primary drive design detailed in the four-bar methodology to be implemented with more success. Finally, the implementation of CNC machining would ensure that the gears comprising the mechanism transmission would mesh more securely.

Finally, the overall assembly of the mechanism and superstructure could be altered to increase ease of access and attractiveness. At the writing of this report, the majority of components were attached using threaded rods or bolts. The exposed threads and bolt heads presented an overall rough and unfinished image. Manufacturing the components as mentioned above would aid in cleaning up this image. Next, the components should be

assembled in order to enable a user to more easily swap out components. For example, at the writing of this report, the primary winch was situated directly below the aluminum flat bars supporting the head pulley and in-line with the 80/20 channels comprising the original superstructure. This can be seen in figure 5.3. The intent was to directly replace the original drive system. In implementing this, the main axle became inaccessible, thus requiring a user to completely disassemble the entire robot in order to replace the axle. This was also the case with the base plate for the super structure. Should a component need to be replaced, the entire robot would need to be disassembled to remove the bolts secured through the underside of the plate. This requirement interfered and delayed final testing several times.

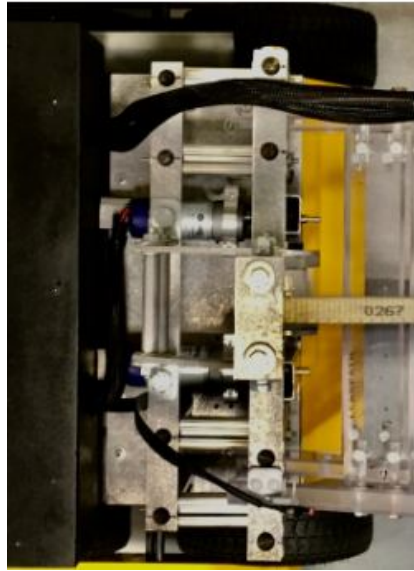


Figure 5.3: Overhead view of winch and superstructure

The topics touched upon in this section are first and foremost suggestions for future implementations for this project. Should they be implemented, the system would be moderately more apt to complete the task at hand. Overall, the four-bar is a passable solution to supporting the sensor rig. The system should, instead, be replaced entirely by a multi-degree of freedom linkage or arm. This would enable scanning to take place in a larger area around the rover as well as supporting more weight and being more able to handle uneven ground.

5.5 Sensor Platform Discussion

The sensor platform was successful in the features of metal detection and marking a landmine. However, the automatic adjust of the coil orientation and the infrared camera were not successful in this implementation. This sections discusses the significance of these results and suggests improvements in future iterations of the project.

5.5.1 Results Discussion

The metal detector was successful and in testing showed to have capabilities greater than what our goal was as it could detect 0.5g of steel at 3 inches and the goal was to detect 5g of steel at up to 3 inches. The drawback of this is that it makes the metal detector more sensitive to interference, leading to the design of the sensor platform needing to ensure all metal is outside of the detectable zone or not moving relative to the coil.

After the third redesign of the sensor platform that excluded the yaw and roll joints, the marking system was able to rotate out and back with less friction due to the lower and evenly distributed weight. Nearly all of the parts for the sensor platform were 3D printed at 0.2mm layer height. Although the printer used generally produced great quality prints, the tolerances were not as close as required for the axles of each joint. All of the joints can be rotated a few degrees before the worm gear prevents further movement. The plastic also had a rough texture and is designed to stick to itself, creating a lot of friction at the joints. Lastly, the plastic used was PLA which is what was easily available but not a good material to make load bearing parts for long term use. PLA tends to deform slowly over weeks to months when a force is exerted on it and also has a low melting point. The sensor platform was stored for about two months on its side, with a large force constantly exerted on the yaw axle. By the end of the 2 months, some of the layers pulled apart and the axle split in half demonstrating the potential for failure over long periods. If the sensor platform was stored in a hot environment, such as a car on a hot day, it would likely soften or melt, compromising the entire structure.

The inaccurate readings from the ultrasonic sensors suggest that using ultrasonic sensors is not ideal for detecting all types of terrain. A single sensor had worked on grass in testing but trying to obtain accurate readings from five sensors all at potentially different heights proved to be difficult. The purpose of this feature is to orient the metal detector coil so that the magnetic field generated is pointed directly at metal in the ground for better readings, however if the sensors fail to orient it correctly then the feature only reduces sensitivity.

The required torque to rotate the metal detector coil was higher than expected due to inefficiencies in the drive train so the motors would need to be upgraded, although this may lead to more interference problems with the metal detector. Even if the coil was consistently kept parallel with the ground, the interference from the motors lowered the sensitivity of the metal detector, negating the benefits of keeping the coil parallel in the first place and could cause many more false positive readings. This feature was successfully demonstrated with the floor as the surface and without the spray can attached, acting as a proof of concept but will need significant improvement to make it a feature worth adding to the sensor platform.

The ultrasonic sensor's poor performance on grass during the full system testing in Institute park confirms that a different method of detecting the ground is necessary for the metal detector to be successfully adjusted over obstacles on a variety of terrain. Distance

readings were accurate on dirt and pavement, allowing the four-bar to properly adjust the metal detector height to be a constant height above the ground. Although, the method of sensing the ground needs to be improved, this does act as a proof of concept. The software is proven to work so better sensors should make this system fully functional.

The infrared camera functionally worked but the testing shows that the thermal differences caused by the landmine underground may be insignificant to detect reliably. Some basic computer vision software was implemented to detect circles in the images. Since the camera is only able to detect landmines that are either under a thin layer of soil or at least partially visible from the surface, the computer vision may be better implemented to look for any area that is much different in temperature than other areas regardless of shape. Potential false positives could occur if other objects like rocks are present since they may have different thermal properties than the surrounding soil. The infrared camera was ultimately removed due to its lack of ability to detect landmines and that the Raspberry Pi was unable to sync with the device properly over the long (1.5M) SPI cables. This made images malformed and put the camera in an error state. Implementation of this camera would require the master device to be placed close to the camera in order for data to be transferred properly.

5.5.2 Final Cosmetic Improvements

To improve the appearance of the sensor platform and ready it for testing outside, the wires were made into cables with nylon braided sleeves, and an enclosure for the Arduino and metal detector boards were printed. The enclosure and braided cables can be seen in Figure 5.4.



Figure 5.4: Cosmetic Improvements to Sensor Platform

5.5.3 Future Improvements

The sensor platform was a new addition to the husky in this iteration of the project and having built and tested two versions has given the team many ideas of improvement of the system.

The first improvement would be to acquire a ground penetrating radar to act as the secondary landmine detection sensor alongside the metal detector. The team had wanted to test a ground penetrating radar but did not have the budget for one and settled on the infrared camera in an attempt for some sensor fusion.

The ultrasonic sensors were tested to be insufficient for accurately and consistently

detecting the ground distance and also are another component that will cause interference with the metal detector if mounted too close to it. While the sensors worked well in the implementation on the Slim Sensor Platform, any obstacles such as small mounds the size of the metal detector may not be detected since none of the ultrasonic sensors will be over it, causing a crash with the ground. The current implementation assumes the terrain is flat and free of any mounds tall enough to collide with the metal detector when raised 1in above the ground. One potential improvement would be to use a 3D LIDAR mounted to the husky below the 4-bar to scan the ground underneath the metal detector for more accurate and detailed data. This sensor could also be used to scan the surroundings of the robot for obstacle detection. Implementing one on each side of the robot may be better than the spinning LIDAR we have used and give the robot a clearer image of its surroundings. A drawback would be that the on-board computer may need to be upgraded to process all of the data.

The motors for rotating the yaw and roll axles were not powerful enough since the worm gear drive selected seemed to be inefficient and the drive-train had a lot of friction. The motors also interfered with the metal detector readings. To solve these issues, a new system involving pulleys to actuate roll and yaw may be better by moving the motors further away from the metal detector coil to avoid interference and may be a more efficient way of increasing torque output.

Lastly, the marking system should be replaced by some sort of paint pump, with the pump located on the Husky with a long tube to spray paint extended to the center of the metal detector coil. The spray can being a large aluminum can creates significant interference with the metal detector and can overwhelm all other readings if brought too close. This is the single factor in the decision to make the sensor platform spin the metal detector coil and marking system. Replacing the spray can with a pump and tube, would lighten the sensor platform significantly, remove a major source of interference, remove the need to rotate the pitch of the sensor platform and remove the servo actuating the spray can. The spray can was used because the team believed running a tube from the Husky to the metal detector would be difficult since the four-bar and slider would be moving it around frequently, and purchasing an adequate pump would be out of our budget at the time. The problems associated with the spray can were not foreseen either.

The last three discussed improvements may be sufficient to fully implement the feature allowing for the metal detector coil to be kept parallel with the ground as the torque required to rotate the metal detector coil would be significantly reduced and better data of the ground terrain would be available. If the weight of the sensor arm can be reduced enough by these changes, then it may be possible to simplify the four-bar and slider design, possibly swapping them out for a robotic arm made out of plastic and actuated by pulleys so the motors can remain far from the metal detector. This may also solve the interference issue from the stepper motor on the Slider since this component would no longer exist.

If future iterations of the project pursue these improvements, it is also recommended to make the final version of the sensor platform out of material that is not PLA. If the parts

are still 3D printed, something like PETG would be better since it will not melt in conditions found in a hot car and is much stronger than PLA. The joint axles should be constructed from materials with tighter tolerances and lower coefficients of friction to improve sturdiness and ease of movement of the joints.

5.6 Navigation Discussion

5.6.1 End Results

In the end the rover was able to receive a map generated on the base station, navigate throughout that map, stop every 20 cm in order to scan the ground in front of it, record mine locations and avoid previously detected mines.

5.6.2 Improvements

There are many improvements that can be made to assist the navigation of the robot. The team's first recommendation is to use the BNO055 IMU compass function so that the robot no longer assumes its position. The IMU on the rover allows the rover to know its orientation relative to its starting orientation, but when working with GPS signals the rover needs to know what its global orientation is. Secondly, in its current state the rover will only move in the four cardinal directions: north, south, east and west. Adding the ability to move in diagonals will allow it to cover ground more efficiently when traversing large areas that have already been scanned and allow it to spend less time in the minefield. The team's third recommendation is to rework the API with a new system that allows for more accurate specification of GPS coordinates. Google Maps is a good choice because of the public's familiarity with the GUI, but it does not provide the level of accuracy needed for this project. Fourth, gearing down the drivetrain will let the rover move at a slower speed and give the rover a higher torque output. This is desirable because operating the motors efficiently results in a large moment on the Sensor Platform that can break it and running the motors at slow speeds draws more current than the motor controllers can handle. Fifth, adding PID control to rover, especially during turns, may help reduce the stress on the sensor arm. This cannot really be accomplished until the issue with the motors moving too slowly is addressed, however it would be useful. Additionally, adding in a mechanical braking mechanism would be helpful for allowing the rover to traverse inclines. Another improvement that could be made is the integration of obstacle avoidance procedures for non-mine obstacles. Currently the rover is equipped with sensors such as a LIDAR and a front-facing ultrasonic sensor that can inform it of above-ground obstacles. However, this sensor data is not integrated into the navigation code yet. Overall these changes will make the rover be able to traverse a minefield more effectively.

Chapter 6

Conclusion

Unmarked landmines are responsible for about 6000 deaths per year (Landmine Monitor, 2016). This project had the goal to create a comprehensive, low-cost and autonomous rover capable of detecting and marking landmines and autonomously navigate and search a minefield. This system is intended to replace human and animal operators that normally would be putting their lives in danger by searching for the landmines themselves.

This project was able to build subsystems for detecting landmines on-top of a commercially available ground rover from Clearpath Robotics. A metal detector was successfully implemented, capable of detection small amounts of steel, comparable to the amount found in PMN landmines, at a depth that the landmines are expected to be at. A system capable of marking the ground with spray paint was built for marking landmine positions on the terrain. A slider mechanism was built and implemented, responsible for moving the metal detector and marking system across the front of the rover to scan for landmines. A four-bar mechanism, driven with a winch, lifts these systems in response to feedback from ultrasonic sensors around the metal detector to keep the metal detector one inch above the ground. Additionally, software was written to allow the rover to receive a minefield defined with GPS coordinates and then navigate and search the defined area. The minefield is defined using a web application that utilizes Google Maps that allows the user to draw out the minefield over an area of their choosing.

The project was successful in finding and marking sudo-landmines placed in grass and was able to search a given minefield to completion. However, the team has made several suggestions for how to improve this project in future iterations to improve the reliability, functionality and durability of the system. Overall, the rover in its current state provides a solid starting point for other teams interested in the project in the future.

References

AIS. (2015). *Seismic Underground Security Sensors*.

Retrieved from <https://applied-infrared.com.au/products/24-seismic-security-solutions>

Alibaba. (2018). *GD-4.5 geophone & seismic Sensor & detector & vibration sensor*.

Retrieved from https://wholesaler.alibaba.com/product-detail/GD-4-5-geophone-seismic-Sensor_1039116678.html

Bonsor, K. (2018).

How Landmines Work. Retrieved from <https://science.howstuffworks.com/landmine4.htm>

Casey, B. & Rocks, T.. (2018).

Landmine Detection Rover. Worcester, Massachusetts. Worcester Polytechnic Institute

Convention on the Prohibition of the use, Stockpiling, Production and Transfer of Anti-Personnel Mines and on their Destruction. (1997, September 18). Retrieved from http://www.un.org/Depts/mine/UNDocs/ban_trty.htm

David-Barret, L. , Plestina, D. , Lampe, J. R. , & Bracewell, C.W.

(2018, August 28). *Croatia* Retrieved from <https://www.britannica.com/place/Croatia>

DiCola, T., (N.D) *BNO055 Absolute Orientation Sensor with Raspberry Pi and*

BeagleBoneBlack Retrieved from <https://learn.adafruit.com/bno055-absolute-orientation-sensor-with-raspberry-pi-and-beaglebone-black/hardware>

Doswald-Beck, L. , Herby, P. & Dorais-Slakmon,

J. (1995, January 1). *Basic Facts: the human cost of landmines*. Retrieved from <https://www.icrc.org/eng/resources/documents/misc/57jmcy.htm>

Facts About

Landmines. (2018). Retrieved from <http://www.landminefree.org/facts-about-landmines/>

GeoModel. (2014). *Ground Penetrating Radar Services*

Nationwide. Retrieved from <https://geomodel.com/methods/ground-penetrating-radar/>

Grainger. (2018).

Infrared Camera, -4 to 482 Temp. Range, Focus Range: 0.40m to Infinity. Retrieved from <https://www.grainger.com/product/FLIR-Infrared-Camera-22UL83>

Gooneratne, C.P., Mukhopahyay, S.C., & Sen Gupta, G. (2004, December 15). *A review of Sensing Technologies for Landmine Detection: Unmanned Vehicle Based Approach*.

Retrieved from

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.6844&rep=rep1&type=pdf>

Harris, W. (2005, May 11). How Car Suspensions Work.

Retrieved December 8, 2018, from <https://auto.howstuffworks.com/car-suspension.htm>

Henkler,

D. (2015). Spray Paint FAQ's. Retrieved from <https://inmyownstyle.com/spray-paint-faqs>

Irin.

(2013, November 8). *Concern over unexploded ordnance in eastern DRC*. Retrieved from

<http://www.irinnews.org/report/99087/concern-over-unexploded-ordnance-eastern-drc>

Kasban, H., Elaraby, S. M., Zahran, O., Abd El-Samie, F. E., & El-Kordy, M. (2010).

A Comparative Study of Landmine Detection Techniques. *Sensing and Imaging*, 11(3), 89-112. doi:10.1007/s11220-010-0054-x

Krylon. (n.d.). Can I spray paint grass.

Retrieved from <https://www.krylon.com/how-to/faqs/product/can-i-spray-paint-grass.jsp>

Lampe, J. R., Pickering, P., Malcolm, N. R. (2018, March 28). *Bosnia and*

Herzegovina. Retrieved from <https://www.britannica.com/place/Bosnia-and-Herzegovina>.

Land Mine. (N.D). Retrieved from https://en.wikipedia.org/wiki/Land_mine

Landmines. (N.D). Retrieved from <http://www.reachingcriticalwill.org/resources/fact-sheets/critical-issues/5439-landmines>

Landmines, Explosive Remnants of War and IED Safety Handbook [PDF File]. Retrieved from http://www.mineaction.org/sites/default/files/publications/Handbook_English.pdf

Land Mines:

Hidden Killers. (1995). Retrieved from <https://www.unicef.org/sowc96pk/hidekill.htm>.

Landmine Monitor 2016. (2016, December 22). Retrieved from <https://reliefweb.int/report/world/landmine-monitor-2016>

Lockman, A., Tighe, G., Haydon, T.. (N.D). *De-Mining with UAVs*. Worcester, Massachusetts. Worcester Polytechnic Institute.

MacDonald, J., & Lockwood, J. (2003). *Alternatives for Landmine Detection*. Santa Monica, CA: RAND Corporation.

Metaldetectorlist.com. (N.D). *How Different Types of Metal Detector Work*. Retrieved from <https://metaldetectorlist.com/how-different-types-metal-detector-work/>

Mottl, D.,A. (2015, January 1). *File: Cambodia's rice fields.jpg*. Retrieved from https://commons.wikimedia.org/wiki/File:Cambodia%27s_rice_fields.jpg

Newegg.com. (N.D). *File: LongRangeWifiAntennas.png*. Retrieved from <https://www.newegg.com/Product/Product.aspx?Item=9SIA91S6M>

Olmstead, T., & Fischer, E. (2009). *Estimating Vertical Stress on Soil Subjected to Vehicular Loading*. doi:10.21236/ada496790

Ottawa Treaty. (N.D). Retrieved from https://en.wikipedia.org/wiki/Ottawa_Treaty

Overton, L. C & Chandler, D. P. (2018, July 31). *Cambodia*. Retrieved from <https://www.britannica.com/place/Cambodia>.

PMN Mine. (N.D). Retrieved from https://en.wikipedia.org/wiki/PMN_mine

Radiodetection.com. (2018). *RD1000+ Ground Penetrating Radar (GPR)*. Retrieved from

<https://www.radiodetection.com/en/products/discontinued-products/rd1000-ground-penetrating-radar-gpr>

Sergei UpstateNY. (2018, May 31). *Reviews for Pulse Induction (PI) Metal Detectors*. Retrieved from https://www.metaldetectingworld.com/reviews_pulse_metal_detector.shtml.

Tyson, J. (N.D). *How Metal Detectors Work*. Retrieved from <https://electronics.howstuffworks.com/gadgets/other-gadgets/metal-detector2.html>

Tyson, J. (N.D). *How Night Vision Works*. Retrieved from <https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/nightvision2.html>

Yadav, N., Bhardwaj, B., & Bhardwaj, S. (2015). Design analysis of Rocker Bogie Suspension System and Access the possibility to implement in Front Loading Vehicles. IOSR Journal of Mechanical and Civil Engineering, 12(3), iii, 64-67.
doi:10.9790/1684-12336467

Appendices

Appendix A

Final ROS Node and Topic Graph

This is the ROS graph generated with the all of the ROS nodes run during the final testing of the system. The graph is too large to fit on a single page so it has been broken up into three sections. The first section is the navigation stack that Clearpath provides. The rest are nodes and topics the team designed. Refer to Section 3.6 for a description of the ROS Nodes.

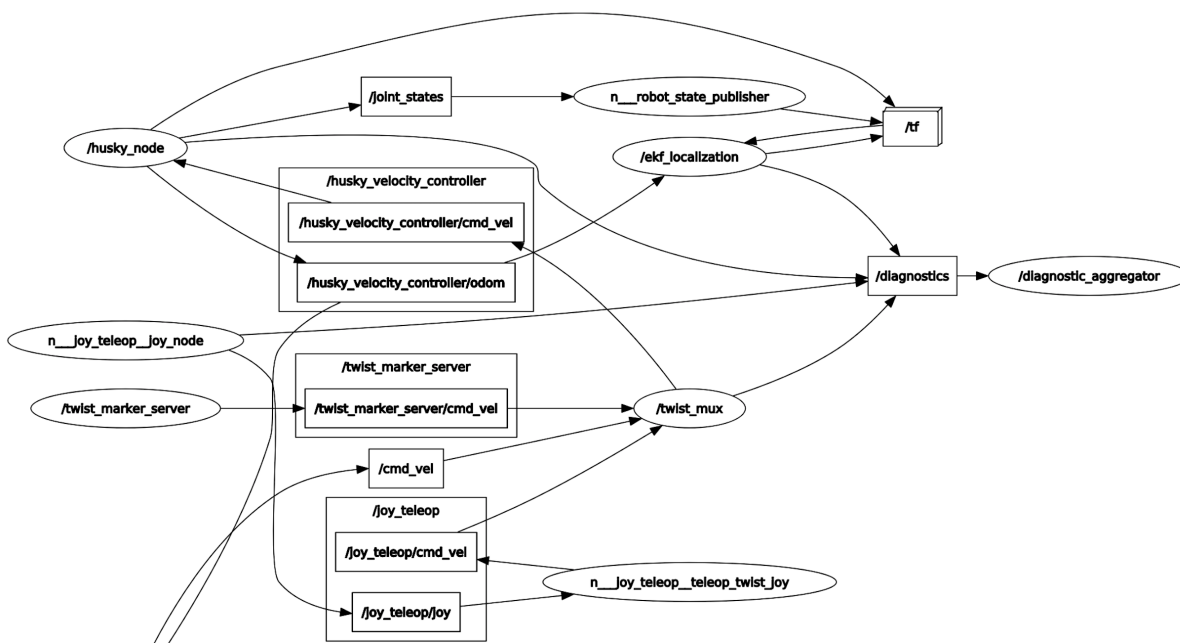


Figure A.1: Section one

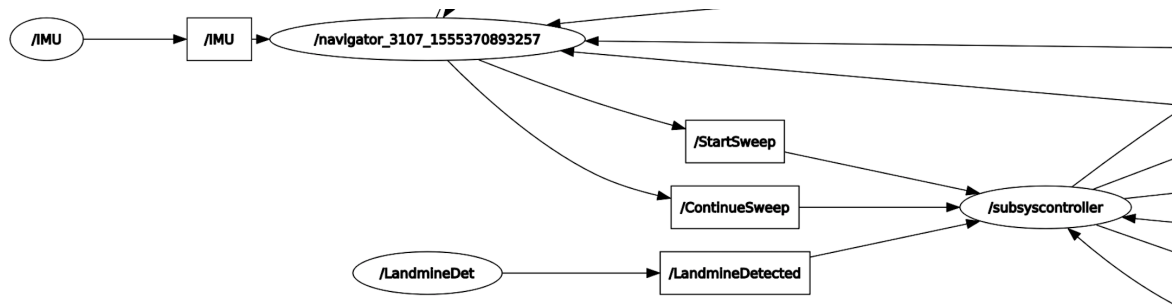


Figure A.2: Section two

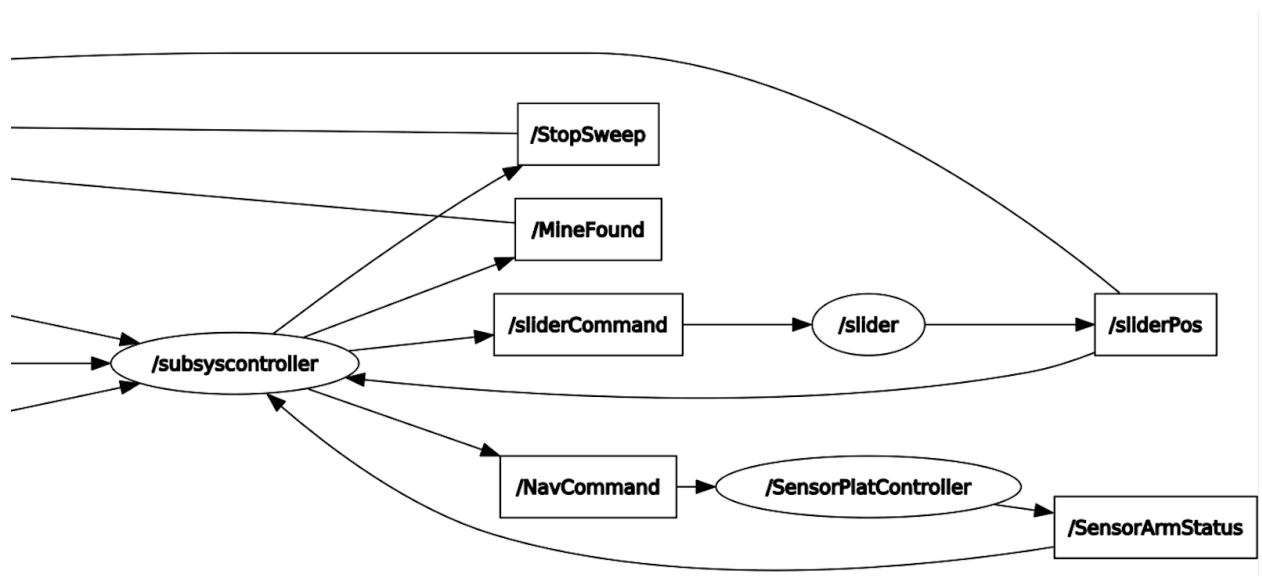


Figure A.3: Section three

Appendix B

Slider Calculations

In the following appendix, the script that proves that the system is mechanically sound is detailed. This MATLAB script combined a set of equations obtained from the free body diagrams of the system along with the dimensions of the system.

Contents

- Influential force on pinion at 30 degree incline
- Stepper Motor Torque
- Torque required
- Stepper Motor Max Angular Velocity
- Bending Strength
- Results
- Free body diagram calculations
- Front view at middle distance
- Side view
- Deflection

Influential force on pinion at 30 degree incline

This section details the tangential force of the metal detector acting in the pinion gear when the robot is at a 30 degree incline changing the robot's roll angle to 30 degrees.

```
F_w = 10; %weight of metal detector (lbf)
theta = 30; %max incline angle (deg)
f_inf = F_w*sind(theta); %influential force (lbf)
```

Stepper Motor Torque

This calculates the maximum torque that the stepper motor can put out at any given instant.

```
T_h = 85; %holding torque (oz. in)
r_p = 0.75; %pitch radius (in)
d_p = r_p*2; %pitch diameter (in)
F_t_oz = T_h/r_p; %tangential load (oz.)
F_t = F_t_oz/16; %tangential load (lbs)
```

Torque required

This is the torque required to be able to counteract and overcome the torque produced from the influential force above.

```
T_r = f_inf*r_p*16; %torque required (oz. in)
```

Stepper Motor Max Angular Velocity

This calculates the maximum angular velocity of the stepper motor to be used to find the linear tangential velocity later on.

```
Imax = 0.64; %max current (Amps)
L = 13.2; %inductance (mH)
Ve = 12; %applied voltage (V)
spr = 200; %steps per revolution
omega = Ve/((L/1000)*2*Imax*spr); %angular velocity (revolutions
%                                per second)
```

Bending Strength

This section finds the maximum bending stress allowed for the pinion gear teeth accounting for dynamic loading.

```
W_t = F_t; %tangential load (lbs)
P_d = 24; %diametral pitch (in^-1)
w_f = 0.5; %face width (in)
y = 0.377; %lewis form factor for 36 involute teeth 20 degrees
```

```

v = (omega*r_p); %tangential velocity
s_ut_MPa = 37; %ultimate tensile strength of pla plastic at .2mm
%          layer height 100% teeth infill (MPa)
s_ut = s_ut_MPa*145.038; %converted to psi
a_bs = s_ut/3; %allowed bending stress (psi)
k_v = (1200 + v)/1200; %Barth velocity factor v is velocity at pitch
%          diameter (ft/min)
m_bs = (W_t*P_d)/(w_f*y); %maximum bending stress (psi)
dynamic_m_bs = m_bs*k_v; %accounted for velocity (psi)

```

Results

This section contains the results showing that the specifications are met.

```

allowed = a_bs %(psi) Allowed bending stress
actual = dynamic_m_bs %(psi) Actual bending stress

weight = f_inf %(lbf) Weight having an influence on shearing
applied = F_t %(lbf) Applied force tangential to teeth counteracting
%          the weight

required_Torque = T_r %(oz. in) torque required
supplied_Torque = T_h %(oz. in) torque holding

```

Bending Stress of Pinion Gear Teeth

```

allowed = 1.7888e+03 psi
actual = 903.8584 psi

```

Force of Weight Acting Against Pinion Gear Movement

```

weight = 5.0000 lbf
applied = 7.0833 lbf

```

Torque Requirement for the Stepper Motor

```

required_Torque = 60.0000 oz. in
supplied_Torque = 85 oz. in

```

Free body diagram calculations

This details the calculations done for static analysis along with deflection calculations.

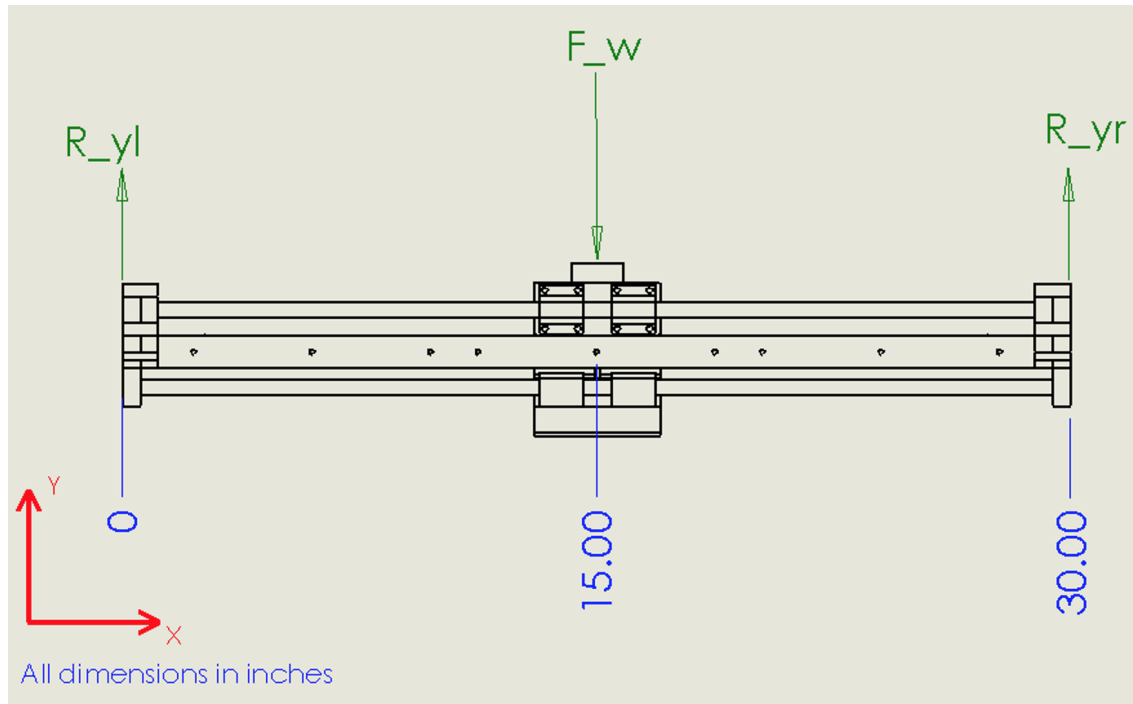


Figure B.1: Free body diagram of slider (Front-View)

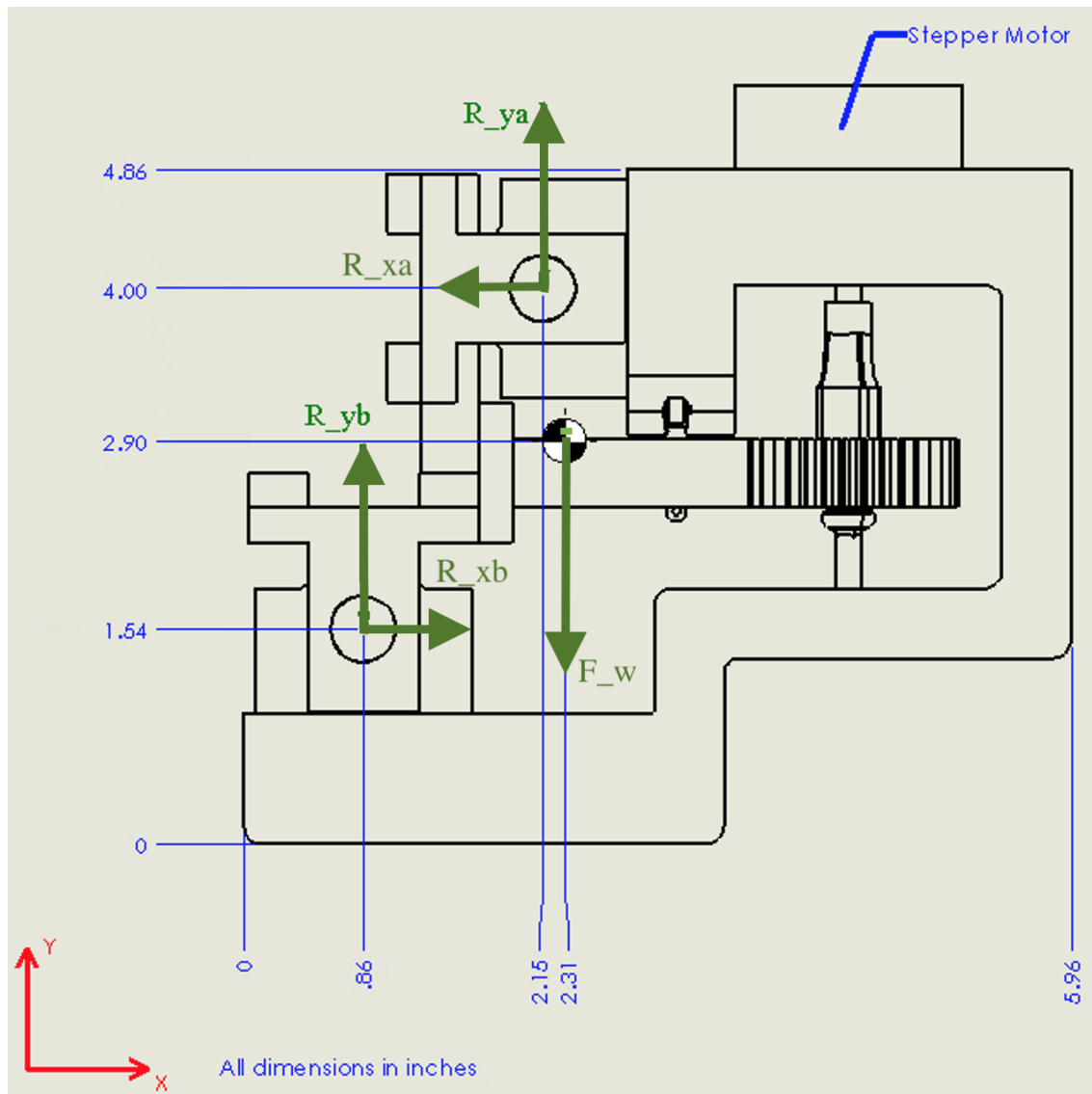


Figure B.2: Free body diagram of slider (Side-View)

% Front-View Constants

L = 30; %(inches) width of bar

dmm = 12; %(mm) diameter of rod

din = 0.472441; %(inches) diameter of rod

F_w = 10; %(lbs) weight of slider and sensor arm

% Side-View Constants

Ls = 5.96; %(in) width of the coupler at a side view perspective

COMX_ref = .162; %(in) math behind finding the Center Of

% Mass X location

distXrods = 1.29; %(in) distance between linear rods x-direction

distYrods = 2.45; %(in) distance between linear rods y-direction


```

COMX = (Ls/2) - 2.5; %(in) Center of Mass location x-direction
COMY = 8.5; %(in) Center of Mass location y-direction
COMloc = [COMX_ref, 8.5]; %(in) Center of Mass location

```

Front view at middle distance

Static Analysis From the Front-View With the Slider Centered.

```

Sum of Moments around the left brackets
Sum of M_l = (RFRy * L) - (Fw * L/2) = 0

Sum of Moments around the right brackets
Sum of M_r = -(RFLy * L) + (Fw * L/2)
R_yr = (F_w * L/2)/L %(lbf) Reaction force in the y-direction
%              coming from the right brackets combined
R_yl = (F_w * L/2)/L %(lbf) Reaction force in the y-direction
%              coming from the left brackets combined
SumFx = 0 %(lbf) Sum of the forces in the x-direction
SumFy = R_yl + R_yr - F_w %(lbf) Sum of the forces in the y-direction

```

Forces of the system

```

F_w = 10 lbf
R_yr = 5 lbf
R_yl = 5 lbf

```

Sum of the forces in the x-direction

```
SumFx = 0 lbf
```

Sum of the forces in the y-direction

```
SumFy = R_yr + R_yl - F_w = 0 lbf
```

Side view

Static Analysis From the Side-View With the Slider Centered.

```

Sum of Moments around the "a" bracket
Sum of M_a = (R_yb * dXrods) - (Fw * COMX_ref)
Sum of Moments around the "b" bracket

```

```

Sum of M_b = -(R_ya * dXrods) + (Fw * (COMX_ref + dXrods))

Forces of the system
F_w = 10 lbf
R_yb = ((F_w * COMX_ref) / distXrods) / 2 %(lbf) Reaction force
%                                     in the y-direction coming
%                                     from the "b" bracket on
%                                     each side
R_ya = ((F_w * (COMX_ref + distXrods)) / distXrods) / 2 %(lbf) Reaction
%                                     force in the
%                                     y-direction coming
%                                     from the "a"
%                                     bracket on each
%                                     side
SumFx2 = 0 %(lbf) Sum of the forces in the x-direction
SumFy2 = R_ya - R_yb - F_w/2 %(lbf) Sum of the forces in the x-direction

Reaction Forces of the system
R_yb = 0.6279 lbf
R_ya = 5.6279 lbf
R_xb = R_xa    lbf

Sum of Forces of the system
SumFx2 = R_xb - R_xa = 0 lbf
SumFy2 = -8.8818e-16 lbf

```

Deflection

This shows the calculations determining how much the slider will deflect downward.

```

Constants
E = 30000000; %(psi) Modulus of Elasticity
F_w_simplified = 5; %(lbf) Pounds acting downward per rod
L = 30; %(in) Length of rod
I = (pi*(din^4))/64 %(in^4) area moment of inertia
Def_max = (F_w_simplified * L^3) / (48 * E * I) %(inches) max deflection
%                                     downward per rod

```

Maximum deflection downward per rod

Def_max = 0.0383 in

Appendix C

Four-Bar Free Body Diagrams

This appendix shows the free-body diagrams that were used to analyze the operation of the four-bar lifting mechanism as specified in Section 3.3 Four-Bar Lifting Subsystem. In analyzing the forces on the system, six diagrams were generated, resulting in seventeen static equations. Each presented diagram will be accompanied by the static equations associated with it. Reaction forces are represented by capital letters. Finally, special variables will be explained where necessary. The resulting equations were used in the MATLAB script which can be seen in Appendix D.

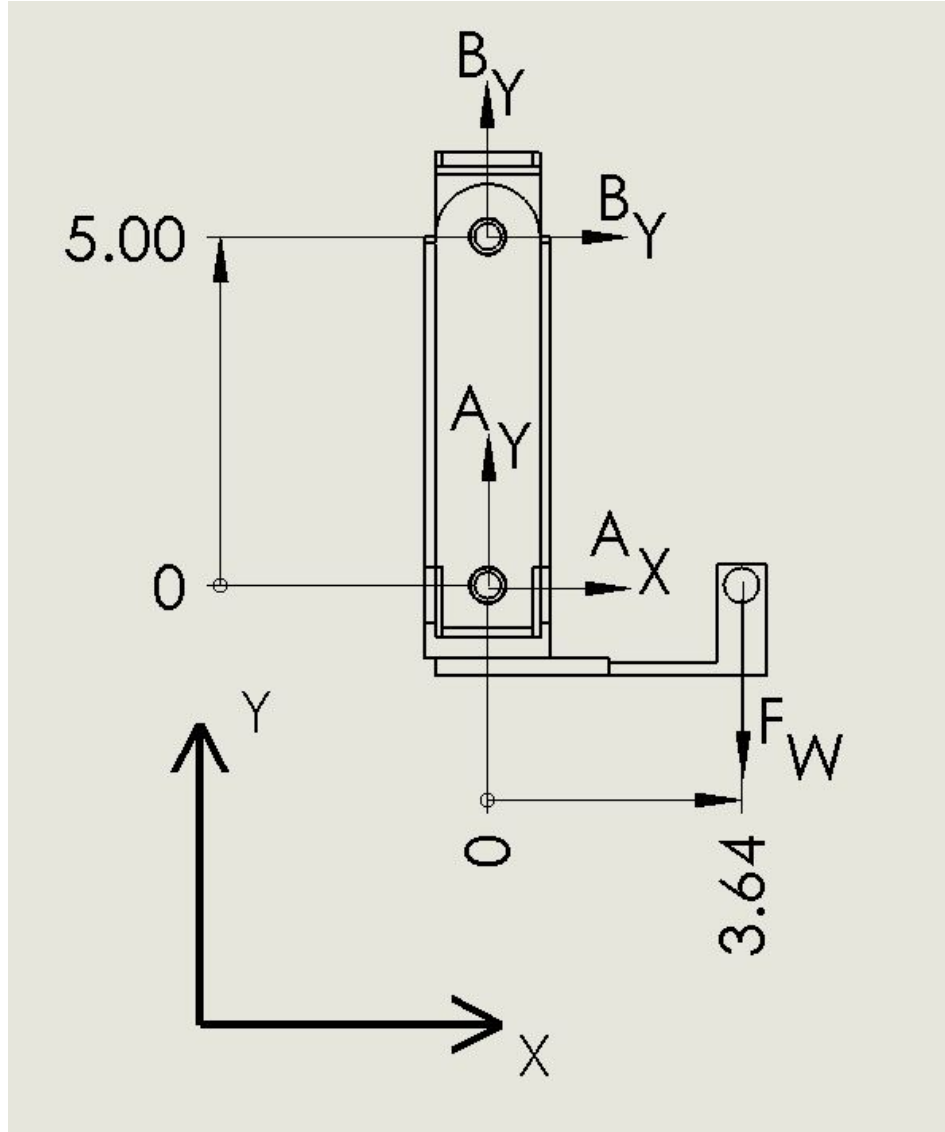


Figure C.1: FBD for coupler plate

$$\Sigma F_x = A_x + B_x = 0 \quad (C.1)$$

$$\Sigma F_y = A_y + B_y - F_w = 0 \quad (C.2)$$

$$A = -B_x * 5 - F_w * 3.64 = 0 \quad (C.3)$$

In the above equations, F_W is the estimated combined weight of the slider. For the purposes of initial analysis, it was assumed to be 16 pounds. The actual value after system redesigns and manufacturing was approximately 10 pounds, enabling a small factor of safety. Further

the distance 3.64 inches was obtained using Solidworks and may not be representative of the real distance between joint A and the center of the combined mass. This value could increase in accuracy if all Solidworks models contained appropriate material properties.

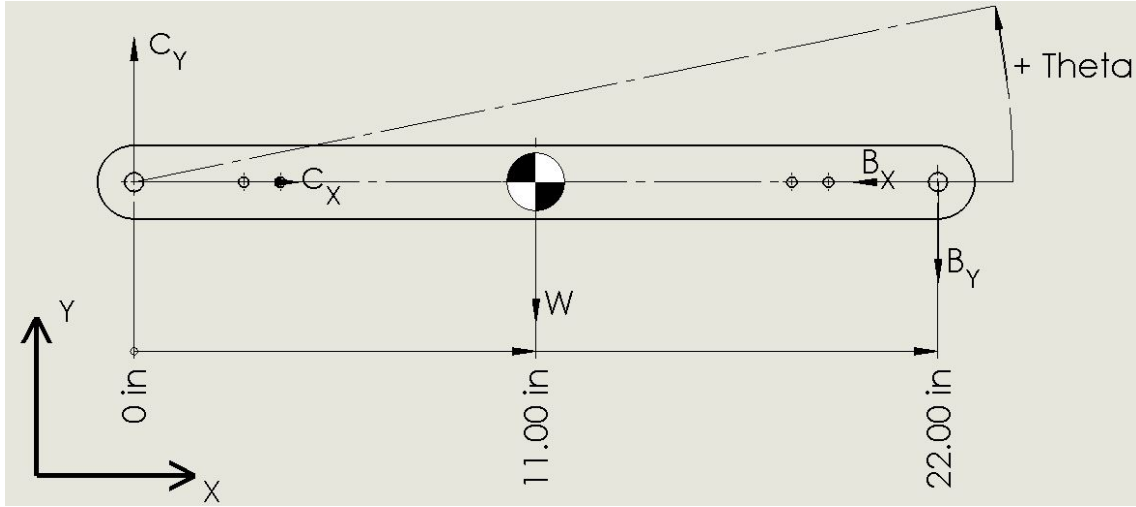


Figure C.2: FBD for follower (upper) link

$$\Sigma F_x = C_x - B_x = 0 \quad (C.4)$$

$$\Sigma F_y = C_y - B_y - W = 0 \quad (C.5)$$

$$M_C = len * B_x * \sin \theta - len * B_y * \cos \theta - \frac{len}{2} * W * \cos \theta = 0 \quad (C.6)$$

In the above equations, the variable len is the length of the link from joint C to joint B. This distance, as can be seen in figure C.2, is approximately 22 inches. Further, the variable W is the combined weight of the links and cross members. Together, these components weigh a combined 2.4 pounds.

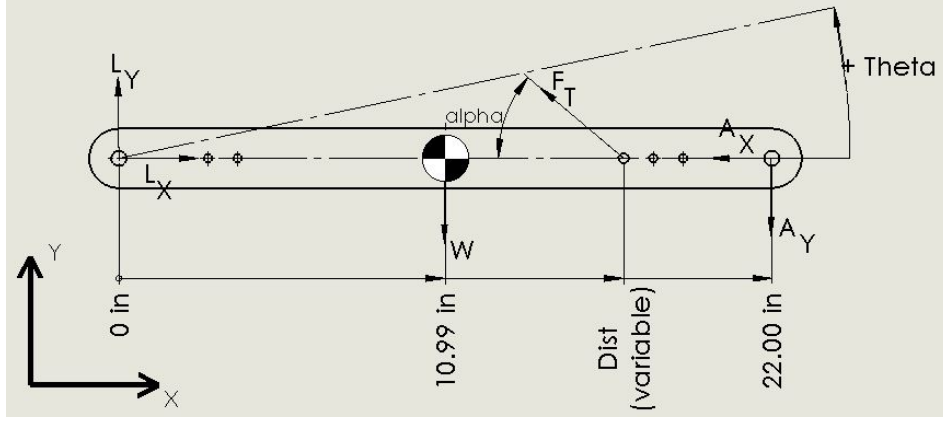


Figure C.3: FBD for driven (lower) link

$$\Sigma F_x = L_x - A_X - F_T * \cos \alpha = 0 \quad (C.7)$$

$$\Sigma F_y = L_y - A_y + F_T * \sin \alpha - W = 0 \quad (C.8)$$

$$M_L = d * \cos \theta * F_T * \sin \alpha + d * \sin \theta * F_T * \cos \alpha - len * A_Y * \cos \theta + len * A_X * \sin \theta - \frac{len}{2} * W * \cos \theta = 0 \quad (C.9)$$

As with the equations for the follower link, the values of len and W are 22 inches and 2.4 pounds, respectively. The variable d is the distance from joint L to the attachment point for the mule tape. This distance was varied from 11 inches to 22 inches in the MATLAB script. This value, along with the variable h discussed below and the value of θ , dictate the value of the variable α

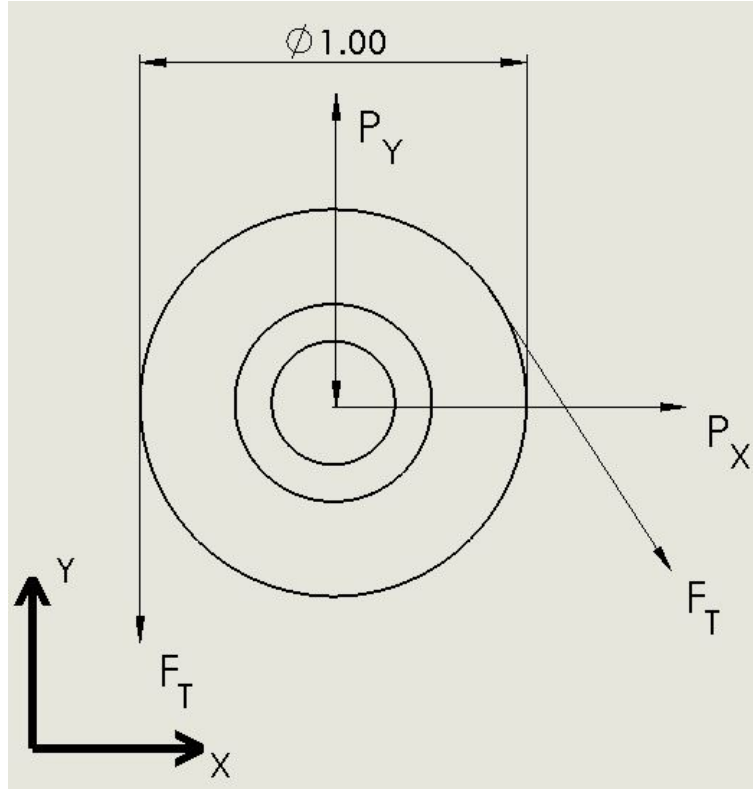


Figure C.4: FBD for Head Pulley

$$\Sigma F_x = P_X + F_T * \cos \alpha - F_T * \cos \beta = 0 \quad (C.10)$$

$$\Sigma F_y = P_Y - F_T * \sin \alpha - F_T * \sin \beta = 0 \quad (C.11)$$

In the above equations, β is the angle created by the difference in radii between the driven winch and the head pulley. This value was recalculated for each new winch size in the MATLAB script. Further, for the purposes of this analysis, it was assumed that the head pulley would rotate freely and without friction around point P, negating the need for moment calculations.

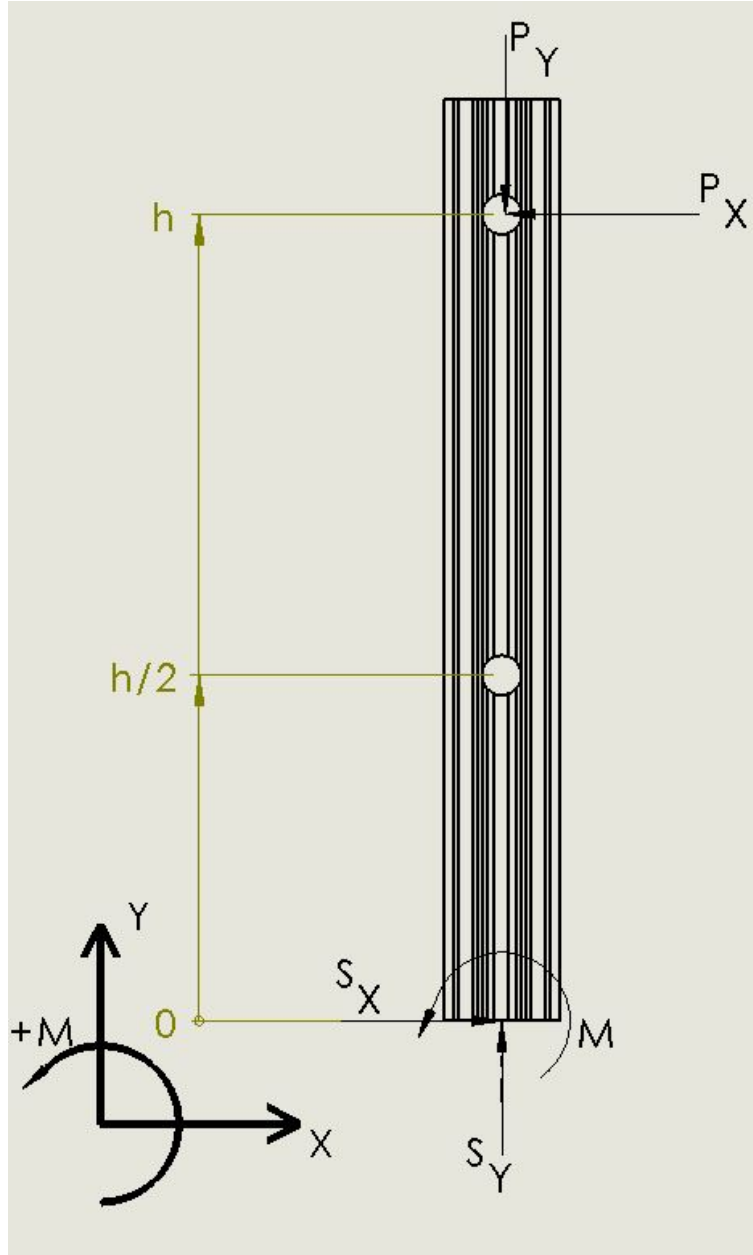


Figure C.5: FBD for Head Pulley Support

$$\Sigma F_x = S_x - P_x = 0 \quad (\text{C.12})$$

$$\Sigma F_y = S_y - P_y = 0 \quad (\text{C.13})$$

$$M_s = M + h * P_x = 0 \quad (\text{C.14})$$

In the above equation, h is the height from the top of the existing superstructure to joint P. This distance will be varied in the associated MATLAB code to reduce strain on the entire system. Further, M is the moment generated about the base of the support. This was of particular interest as it would be used in torsion stress and shear calculations on the flat-bar top of the superstructure and the bolts securing the support, respectively. Ultimately, however, this was not taken into account after observing minimal deflection during live tests.

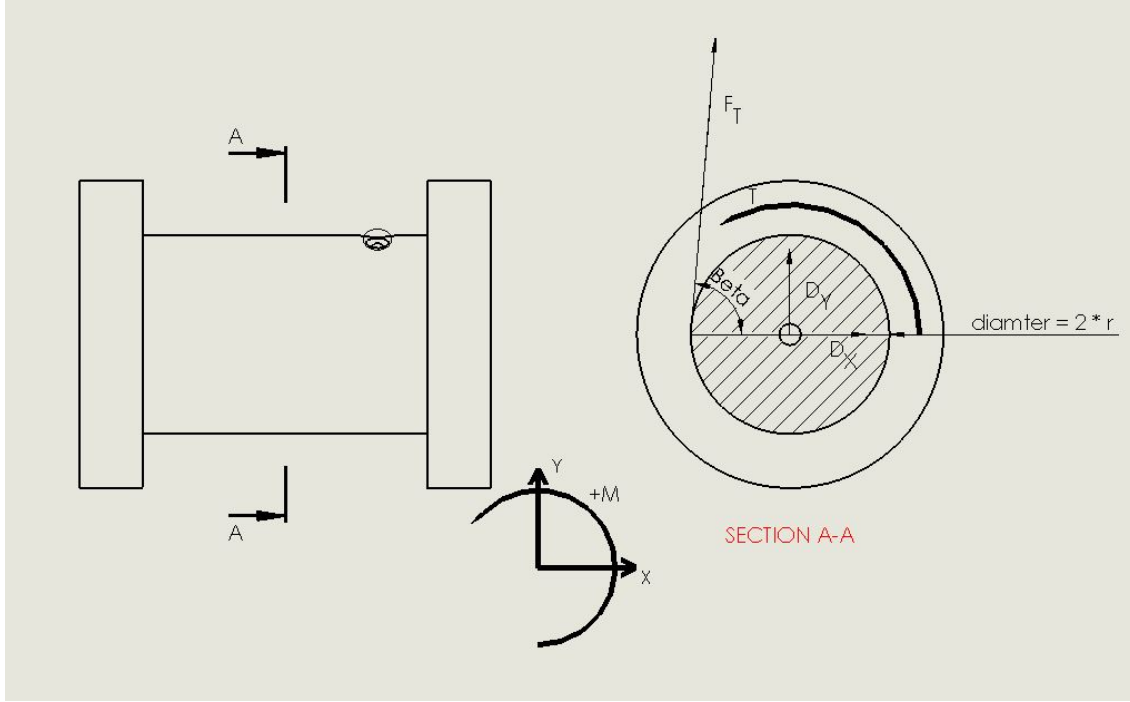


Figure C.6: FBD for main winch

$$\Sigma F_x = D_x + F_T * \cos \beta = 0 \quad (C.15)$$

$$\Sigma F_y = F_y + F_T * \sin \beta = 0 \quad (C.16)$$

$$M_D = T - F_T * R = 0 \quad (C.17)$$

In the above equations, R represents the radius of the winch, while T represents the required torque to hold the system at the current orientation. The radius R was varied in the associated MATLAB script.

Appendix D

Four-Bar Winch Drive System Force Calculations

In the following appendix, the script used in selecting the appropriate dimensions for the winch drive system is detailed. This MATLAB script combined a set of equations obtained from the free body diagrams of the system (see Appendix C) with values for the distance to the head pulley, distance to the cross-member, and the radius of the winch. From the tabulated results, it was seen that this drive system was far more effective at supporting the entire system than the prior drive system (see Section 4.4 Four-Bar Results).

Contents

- Initialization
- Constants
- Free Body Diagram Equations
- Output Structures
- Main Loop

Initialization

The following commands clear the previous data and set up the system for symbolic operation

```
clear all
```

```
syms Ay Ax By Bx Cx Cy Dx Dy Lx Ly Px Py Sx Sy M Ten T d h phi alpha theta beta r
```

Constants

```
% Estimated weight on coupler plate
F = 16;
% Combined weight of two 22-inch polycarbonate links and two 11-inch cross members
W = 3*1.6;
% Link length
L = 22;
% Final support height for head pulley
height = 6.75;
% Final winch radius
radius = 0.75;
```

Free Body Diagram Equations

The following equations were generated from the free body diagrams associated with the four-bar

```
eqn1 = Ax + Bx == 0;
eqn2 = Ay + By - F == 0;
eqn3 = -5*Bx-3.64*F == 0;

eqn4 = Lx - Ax - Ten*cosd(abs(alpha)) == 0;
eqn5 = Ly - Ay + Ten*sind(abs(alpha)) - W == 0;
eqn6 = d*cosd(theta)*Ten*sind(abs(alpha))+d*sind(theta)*Ten*cosd(abs(alpha))...
      -L*Ay*cosd(theta)+L*Ax*sind(theta)-L*W/2*cosd(theta) == 0;

eqn7 = Px + Ten*cosd(abs(90-beta)) + Ten*cosd(abs(alpha)) == 0;
eqn8 = Py - Ten*sind(abs(90-beta)) - Ten*sind(abs(alpha)) == 0;

eqn9 = -Px + Sx == 0;
eqn10 = Sy - Py == 0;
eqn11 = h*Px+M == 0;

eqn12 = Dx + Ten*cosd(beta) == 0;
eqn13 = Dy + Ten*sind(beta) == 0;
eqn14 = T - Ten*r == 0;

eqn15 = Cx - Bx == 0;
eqn16 = Cy - By - W == 0;
eqn17 = L*sind(theta)*Bx - L*cosd(theta)*By - L/2*W*cosd(theta) == 0;
% The following matrices store the equations, variables, and sysm variables
```

```

% for later use.
eqnMat0 = [eqn1 eqn2 eqn3 eqn4 eqn5 eqn6 eqn7 eqn8 eqn9 eqn10 eqn11 eqn12 ...
           eqn13 eqn14 eqn15 eqn16 eqn17];
varVect = [Ay Ax By Bx Cx Cy Dx Dy Lx Ly Px Py Sx Sy M Ten T];
calculables = [r h d beta theta alpha phi];

```

Output Structures

The following variables are used to track and store the output data of the script

```

count = 1;
titles = {'Drum Radius (in)', 'Support Height (in)', 'Distance Along Links (in)'...
          'Angle of elevation (degrees)', 'Tension (lbf)', 'Torque Out (lbf-in)'...
          'Ax (lbf)', 'Ay (lbf)', 'Bx (lbf)', 'By (lbf)', 'Cx (lbf)', 'Cy (lbf)',...
          'Dx (lbf)', 'Dy (lbf)', 'Lx (lbf)', 'Ly (lbf)', 'Px (lbf)', 'Py (lbf)',...
          'Sx (lbf)', 'Sy (lbf)', 'Moment at Support Base (in-lbf)'};
titleRange = 'A1:U1';

```

Main Algorithm

The following loop calculates the forces necessary to hold the estimated force at a given angle in the range of zero to forty-one degrees.

```

for radius = 0.75:2
    rFdlr = strcat('Radius_', num2str(radius), '/');

    for height = 6:14
        hFdlr = strcat('Height_', num2str(height));
        b = atand((7+height)/(radius - 0.25));

        for distance = L/2:L
            % Specify output file location
            % NOTE: All folders must be manually created prior to executing
            % this script

            fileName = strcat('Pulley_Calc_results/Run_2/', rFdlr, hFdlr, '.xlsm');
            worksheet = strcat('D_', num2str(distance));
            % Apply column titles to the new worksheet
            xlswrite(fileName, titles, worksheet, titleRange);
        end
    end
end

```

```

for th = 0:41

    % Specify the range to write to in the worksheet
    thisRange = strcat('A',num2str(th+2),':U',num2str(th+2));
    fprintf("testing case %d, appending to file %s, and worksheet %s\n",...
count,fileName, worksheet);

    % Calculating alpha and phi
    alp = atand(((7+height)-(distance*sind(th)))/...
        (0.25-distance*cosd(th)));
    ph = 180 + alp;
    calculated = [radius height distance b th alp ph];

    % substitute all symbolic values for the calculated values
    % for this run
    sol = solve(subs(eqnMat0,calculables,calculated),varVect);
    thisSolution = [radius, height, distance, th, ...
        eval(sol.T),eval(sol.Ten),...
        eval(sol.Ax),eval(sol.Ay),...
        eval(sol.Bx),eval(sol.By),...
        eval(sol.Cx),eval(sol.Cy),...
        eval(sol.Dx),eval(sol.Dy),...
        eval(sol.Lx),eval(sol.Ly),...
        eval(sol.Px),eval(sol.Py),...
        eval(sol.Sx),eval(sol.Sy),...
        eval(sol.M)];

    % Amend to worksheet at specified location
    xlswrite(fileName,thisSolution,worksheet,thisRange);
    count = count + 1;
end
end
end
end
end

```


Appendix E

Landmine and Environment Sensor Testing

This is the full sensor test report that the Sensor testing sections in the Methodology and Results refer to. This contains all materials, procedures and data collected.

Landmine and Environment Sensor Testing

Demining MQP

Written by: Nicholas Lanotte
Testing Conducted by: Nicholas Lanotte

Edited by:
Benjamin Wagner
Dillon Arnold
Irina Lavryonova

November 12th, 2018

Table of Contents

<i>Table of Figures.....</i>	<i>iii</i>
<i>Table of Tables.....</i>	<i>iv</i>
1.0 Introduction	1
2.0 Materials	2
3.0 Procedure.....	3
3.1 Inductor Coil Preparation and Measurement	3
3.2 Optimal Coil Selection.....	3
3.3 Measure Depth of Detection Depending on Steel Mass.....	4
3.4 Measure the Effect Surface Area Has on Detectability.....	4
3.5 Measure the Ability of the Metal Detector to Remove Constant Interference	5
3.6 Measure the Effect Soil has on Detectability	5
3.7 Infrared Camera – Test Thermal properties of Dummy Landmine	6
3.8 Ultrasonic Sensor HC-SR04 Grass Testing	6
3.9 Maxbotix Ultrasonic Range Finder Testing	7
3.10 Outdoor LIDAR Testing.....	7
4.0 Results.....	9
4.1 Inductor Coil Measurements.....	9
4.2 Optimal Coil Selection Results.....	9
4.3 Depth Detection with Varying Steel Mass	9
4.4 Depth Detection with Varying Surface Area	11
4.5 Effects of Interference on Detectability.....	11
4.6 Soil Effects on Detectability	12
4.7 Infrared Camera Test Results	13
4.8 Ultrasonic Sensor HC-SR04 Test Results	16
4.9 MaxBotix Ultrasonic Range Finder Test Results.....	16
4.10 Outdoor LIDAR testing	17
5.0 Discussion and Conclusions	18
5.1 Metal Detector Discussion and Conclusions	18
5.2 Infrared Camera Discussion and Conclusions	19
5.3 Environment Sensors Discussion and Conclusions.....	20
6.0 References	21

Table of Figures

Figure 1: Coil Winding and tape placement.....	3
Figure 2: Coil Wire Length	3
Figure 3: Metal Detector Coil Set Up.....	4
Figure 4: Soil Test Area	6
Figure 5: Graph of Steel Mass vs. Detectable Height	10
Figure 6: Set up of Heat lamp and IR Camera	13
Figure 7: Example of soil with exposed landmine	13
Figure 8: Exposed Landmine Images	14
Figure 9: Landmine heating 0.25in-1in below soil	15
Figure 10: Landmine in sand cooling.....	15
Figure 11: Examples of Tested Objects with the MaxBotix Ultrasonic Range Finder	16
Figure 12: LIDAR point cloud and actual objects.....	17

Table of Tables

Table 1: Materials	2
Table 2: Coil Measurements	9
Table 3: Coil Depth Detection	9
Table 4: Steel Mass vs. Detectable Height	10
Table 5: Detectability with Varying Surface Area	11
Table 6: Surface Area Measurements	11
Table 7: Interference Results	12
Table 8: Soil Effects on Detection Results	12

1.0 Introduction

The purpose of this test procedure is to understand the capability of the Surf Pulse Induction 1.2 DIY Metal detector kit and the FLIR Lepton 500-0690-00 Infrared Camera to detect the PMN family of landmines for use in the Demining Autonomous System Major Qualifying Project. Additionally, the HC-SR04 Ultrasonic range finder, the MaxBotix Ultrasonic Range Finder EZ1 and the Scanse LIDAR will be tested for use as environment detection sensors for the obstacle avoidance protocol.

The Surf Pulse Induction Metal Detector kit will be tested for future implementation as the primary detection sensor for a demining robot. The metal detector will be tested with a range of induction coils to determine which inductance range is optimal for detection. The best coil will be used to test sensing capability in a variety of simulated environments, with different shapes of metal and with varying levels of interference. Many tests will be conducted in ideal detecting conditions followed by a fewer number of tests in non-ideal conditions to scale the ideal detection data. This way an estimation of performance in a variety of conditions can be made without the need to have extensively test each condition. Performance will be measured in terms of detection depth and amount of metal detected. It will also provide insight into how the demining robot should be designed to reduce interference.

Alongside the metal detector, the FLIR Lepton Infrared Camera will be implemented as a secondary sensor on the demining robot. The infrared camera was selected for testing because it may be able to detect thermal differences in topsoil caused by the buried landmines during transitional times of the day when the ground is being heated or cooled by the sun. Landmines have different thermal properties than the surrounding soil and this difference theoretically can be detected. The infrared camera will first be tested by placing dummy or modelled landmine in a controlled environment made of compacted dry soil, wet soil and sand contained in a large bucket. A separate test will be done in ground soil with grass outside. After heating the soil samples with a heat lamp to simulate the sun and the infrared camera will be pointed at the soil and the resulting data will be analyzed for detection ability. Additionally, this will be done at varying angles to determine which angle, if any is most effective at detecting a thermal difference.

The HC-SR04 ultrasonic sensors will be tested for their ability to accurately detect the ground for use in orienting the metal detector parallel with the ground. The MaxBotix Ultrasonic sensor will be tested for its area of detection and ability to detect trees and bushes. The LIDAR will be tested for its ability to detect objects outside including trees and bushes.

2.0 Materials

Table 1: Materials

Item Name / Model # if applicable	Manufacturer	Supplier	Supplier P/N	Quantity
18AWG Enameled Copper Wire	Remington Industries	Amazon	18H200P	200ft
Surf PI 1.2 Metal Detector	Silverdog UK	Silverdog UK	N/A	1
Multimeter	Tektronix	N/A	N/A	1
Oscilloscope	Tektronix	N/A	N/A	1
Function Generator	Tektronix	N/A	N/A	1
12V Battery Pack / FBA BH383	Mosuch	Amazon	FBA_BH383	1
AA Batteries 10ct	Up&Up	Target	008-08-0077	1
Many Steel Pieces Varying Sizes from 0.5g to 100g	N/A	N/A	N/A	20+
Brushless DC Motor	DFRobot	Digikey	1738-1157-ND	2
Marking Paint	Rust-Oleum	The Home Depot	463487	1
Aluminum Foil	Up&UP	Target	253-01-0888	As Needed
Yard Stick (wood)	N/A	N/A	N/A	1
9.5in Diameter Cooking Pot	N/A	N/A	N/A	1
IR Camera	FLIR	DigiKey	500-0690-00-ND	1
Lepton Breakout Board PCB	FLIR	Digikey	250-0587-00-ND	1
250W Heat Lamp Bulb	Philips LED	Amazon	415836	1
250W Lamp with Bulb Gaurd	Woods	Amazon	0165	1
Clear Plastic Bin	Sterilite	The Home Depot	896791	4
Garden Soil	Vigoro	The Home Depot	1002151875	1
50lb bag Play Sand	Quikrete	The Home Depot	1001709466	1
Dummy Landmine	N/A	N/A	N/A	1
Arduino Uno R3/ A000066	Arduino	Amazon	B008GRTSV6	1
Ultrasonic Range Finder EZ1	Maxbotix	Amazon	LV-EZ1	1
Ultrasonic Sensor HC-SR04	Smraza	Amazon	B01JG09DCK	1
Outdoor LIDAR	N/A	N/A	N/A	1
Breadboard wires	N/A	N/A	N/A	As Needed

N/A means “not available”

3.0 Procedure

The following section details the procedure for testing the metal detector, infrared camera, lidar and ultrasonic range finder. The purpose of this section is to provide detailed instructions for each test so that work by future teams may replicate the tests or perform similar tests on other sensing technologies.

3.1 Inductor Coil Preparation and Measurement

To prepare the metal detection coils used for testing, acquire approximately 200 feet of 18AWG enameled copper wire and a cylindrical object, such as a cooking pot, with an outside diameter of 9 to 10 inches. Create a few coils with varying number of coil windings in the range of 6 to 30 coils. At least one coil should be in the range of 12-15 windings. This will elicit an inductance comparable to that of professional metal detectors. After winding, tightly bunch the coil windings together and secure them in place by wrapping 4 pieces of clear tape around the coil. An example can be found in Figure 1. Ensure that there is at least 4 feet of wire extending from the coil in order to remove interference from the board as seen in Figure 2. Attach a Molex connector for use with the PI 1.2 metal detector kit to the end. Measure the resistance of the coil with a multimeter and the inductance of the coil with an oscilloscope. To measure inductance, connect the coil to a function generator with a 50Ω internal resistance in series. Connect an oscilloscope to the coil in parallel. Set the function generator to a 2kHz sine wave with a 0V minimum and a 1V maximum. Adjust the frequency until the waveform on the oscilloscope has a peak-to-peak value of 0.5V. The inductance is 4.57 divided by the frequency in Henrys (Dekker). The explanation behind this can be found by going to the website referenced.



Figure 1: Coil Winding and tape placement

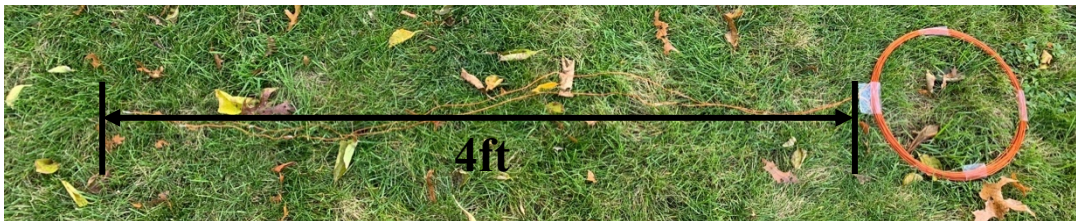


Figure 2: Coil Wire Length

3.2 Optimal Coil Selection

This procedure will determine which of the created coils is most effective at detecting metal in near ideal conditions and will be used for the remainder of the testing procedure. An environment with low amounts of nearby metal and electrical interference, such as a room with no electronics or an outside area far from buildings or electrical equipment, is necessary for this. Begin by bringing the metal detector, a 12V battery, the coils, a piece of steel of at least 50 grams and a yard stick into such an environment.

Connect the first coil to the metal detector and turn the metal detector on. Adjust the threshold until the buzzer is heard just above silence. Wave the piece of steel over the coil and

adjust the delay potentiometer with a flat head screw driver until the metal is detected at the maximum distance. Adjusting this potentiometer will change the phase delay expected by the induced magnetic field. Different metals cause different phase delays, so this should only need to be adjusted once for steel. The metal detector is now calibrated once completed. Record the highest vertical distance between the coil and the steel. Repeat the measurement portion of this experiment for every coil.

If a significant trend in detectability, such as a bell curve, can be found from this data, it may be beneficial to construct a new coil with the number of loops that appears to be optimal and repeat this procedure to verify. The coil with the largest detection distance should be used in future tests. If they yield similar results, use the coil with the most windings because it will require less power from the batteries.

3.3 Measure Depth of Detection Depending on Steel Mass

Acquire at least 20 pieces of steel with varying masses ranging from 0.5g to 200 grams that are as flat as possible. Set up the metal detector and calibrate it if not already as described in procedure 3.2. Starting with the 200g piece of steel, wave it over the coil in an orientation that would expose the face with the largest surface area to the magnetic field and record the maximum detectable range. Repeat this process for the remainder of the steel pieces in order of decreasing mass either until every piece has been tested or a piece is not detectable at any range. The data should represent the metal detector's optimal sensitivity by steel mass in ideal conditions (nearly no interference, large surface area of metal to detect, and detecting ferrous metal). An example of the test area can be found in Figure 3.



Figure 3: Metal Detector Coil Set Up

3.4 Measure the Effect Surface Area Has on Detectability

Choose 3 of the steel samples used in procedure 3.3 and acquire another 3 pieces of steel, one with the same mass as each of the previous samples. Each new sample should have a face with about half the surface area as larger face from the corresponding sample from the previous procedure. Measure the surface area of the selected face along with the smallest and largest face on the previous samples.

Return to testing in the same low interference environment used before with the calibrated metal detector and the selected steel samples. For each of the selected samples from 3.3, wave it over the metal detector so that the face with the smallest measured area is perpendicular to the magnetic field and record the maximum detectable range. Repeat this process for the measured faces of the 3 new samples. This data will provide three data points for three different steel masses to scale the data from Section 3.3 to estimate the detectability of a sample of steel by weight and surface area.

3.5 Measure the Ability of the Metal Detector to Remove Constant Interference

The design of the Demining Robot currently has 2 motors and an aerosol can that lie inside the magnetic field created by the coil, all of which add interference to readings. This test is to determine how well the metal detector will be able to operate with these items present and whether edits to the robot design are required.

Acquire an aluminum spray can and two brushless DC motors and bring the metal detector along with the samples that were detectable from procedure 3.3 of the experiment to the same low interference, controlled environment as before. Hold the largest sample of steel over the metal detector and wave it back and forth with one of the motors in the center of the coil. Record the distance the metal detector can detect the sample if at all. Decrease the mass of the sample until the metal is no longer detectable. If all samples are detectable, record the average decrease in detectable range. Repeat the process starting with the largest sample, with only the two motors without the can being used as interference, only the can and no motors, and finally both the two motors and the can together.

Wrapping components in aluminum foil may be helpful in shielding the metal detector from the interference from the motor. Wrap the motors in aluminum foil and repeat this testing procedure with one motor and then both motors being used as interference.

Repeat this process but place the interference objects outside the coil and move them away from the coil until the steel samples are detectable at their ideal conditions distance. Lastly, repeat the process holding the interfering objects above the metal intended on being detected and record move the interference vertically upward until the steel samples can be detected at their greatest distance. Every test here does not need to be done, but enough scenarios should be conducted so that the behavior of the metal detector can be predicted. For example, if the aluminum can makes detecting metal impossible, it is not necessary to test the can with the two motors.

3.6 Measure the Effect Soil has on Detectability

The Demining Autonomous System Project is targeting landmines that are buried 3 inches or less under the surface. This portion of the experiment will determine the metal detector's capability to detect steel in a variety of environment conditions. Rather than test 20 or more pieces for each environment, this experiment will test 3 (or more if desired) and the results will scale the ideal detection curve measured in the previous section.

Before testing, prepare a testing area for burying the steel. This area should contain dry dirt, wet dirt, sand, and soil with grass in a space that is far from large amounts of metal or electrical interference. Select 3 of the detectable pieces of steel used to test in Section 3.3 that vary greatly in mass to test in all four burying scenarios. For each piece of steel, bury it 3 inches below the surface. Wave the metal detection coil over the area as close to the ground as possible. If the steel can be detected, bury the steel deeper in increments of 0.5 inches, testing for detectability at each depth until it is no longer detected. If the steel cannot be detected initially, uncover the steel in increments of 0.5 inches and test detectability until it is detected. Record the deepest detectable depth for each selected piece. These results will scale the curve found in the previous section to estimate the other pieces of steel. An example of the test area can be found in Figure 4.



Figure 4: Soil Test Area

3.7 Infrared Camera – Test Thermal properties of Dummy Landmine

The infrared camera is being introduced to the sensing suite for the demining robot as it may be able to detect landmines based on thermal properties rather than metal content. This may increase efficiency with respect to detecting minimal metal mines in lieu of using a metal detector. This test uses a dummy landmine that was created by previous work at WPI to model a PMN1 landmine but any plastic object similar in shape to a PMN-1 landmine should suffice for the purposes of this test.

Begin by setting up a test environment of dry dirt, wet dirt, sand and soil with grass and acquire a 250W heat lamp. This test will attempt to simulate a few possible environments where the landmines may be buried with the heat lamp simulating the sun warming the soil. Bury the landmine in the dry soil so that the top surface of the dummy landmine is exposed. Orient the infrared camera so that it is facing downward towards the surface of the soil and turn on the heat lamp. Observe the video feed from the infrared camera and wait to see if the camera is able to detect a difference in temperature of the soil above the mine compared to the soil around it. Eventually the heat lamp will stop increasing the soil temperature as it will have reached equilibrium. If no thermal difference has been detected by this time, then the test has failed. Turn off the heat lamp and observe the infrared camera video as the soil cools to ambient temperature. If no thermal difference was detected by when the soils fully cools, the test has failed. Repeat this process for each soil condition.

Repeat the procedure but bury the dummy landmines beginning at 0.5 inches below the soil in each case. If a thermal difference between the soil above the landmine and the surrounding soil is detected, keep burying the dummy landmine deeper by 0.5 inches until a thermal difference is no longer detected. This will provide an understanding of the maximum depth the IR Camera can sense a difference in soil temperature.

3.8 Ultrasonic Sensor HC-SR04 Grass Testing

Ultrasonic sensors were selected for use with the metal detector arm of the robot to keep the metal detector coil parallel with the ground. These sensors are known to work well on hard

flat surfaces, but the robot is expected to traverse soft soil and grass, so this test is meant to measure the ability of the sensor to detect loose dirt and grass. This procedure requires a method of reading in sensor data and powering the device which can be achieved in a number of ways. The recommended method is to acquire an Arduino Uno and connect the sensor to it using the breadboard wires. Program the Arduino to read in sensor data using the NewPing Library designed for this sensor and output to serial on a connected computer. Test code will be provided in the Demining Autonomous System MQP files so use that if available. Basic programming knowledge is needed to write the code from scratch.

Once the sensor, Arduino Uno, and computer are connected and sensor readings are being read, bring the sensor outside where there is some grass and loose soil. A measuring device such as a tape measure or ruler will be useful here. Face the sensor toward the ground where there is a hard surface such as pavement and verify that the sensor readings are consistent and reasonable using the measuring device to acquire the actual distance from the sensor to the ground. The sensor will likely be an inch or so off the actual reading, which is acceptable if the readings are consistently off by this amount. Verify that the distance can be measured from 12 inches to 1 inch. Repeat this process on grassy terrain. The readings are expected to be noisy but if the readings are accurate more often than not then this will be acceptable since the noise can be filtered out in software. Repeat once again on loose soil and record all observations for all tests.

3.9 Maxbotix Ultrasonic Range Finder Testing

The Maxbotix EZ1 Ultrasonic Range Finder is different from the HC-SR04 ultrasonic sensors in that it is able to detect objects in a wide area rather than directly in front of it, making it better for detecting obstacles of varying height that are in front of a robot. This sensor has been thoroughly tested by the manufacturer, so this test is to verify some of their findings and verify that the sensor does not have any defects.

Maxbotix has some code for the Arduino Uno that make getting started with the sensor easy. This code can be found [here](#) or at the link provided in the references. Once the Arduino, computer and sensor are configured as described in the link, go into an environment with some bushes, trees, and a hard wall close by. Verify that the sensor is able to detect the tree and bushes at a distance of at least 3 feet from the sensor. Next go to an area with a hard wall and face the sensor parallel to the wall but off to the side of it into an area of open space so that the sensor reads the maximum distance of about 253 inches. Move the sensor in the direction parallel to the wall until the sensor is able to detect it. Do this with the sensor in a range of 1 foot to 3 feet away from the wall face in the perpendicular direction to observe where the sensing area is. This area is documented in the datasheet so this test is just a verification.

3.10 Outdoor LIDAR Testing

A LIDAR was selected for detecting obstacle around the robot in the event it needs to back up or turn and avoid obstacles. The purpose of this test is to verify that the LIDAR is able to detect objects the robot is expected to encounter like trees and bushes. The LIDAR being used in this experiment is the Scanse Sweep LIDAR but this company has gone out of business and the LIDAR modules are known to degrade quickly so it is recommend that an outdoor LIDAR manufactured by a reputable company is used here. This project is using the Scanse Sweep because it was donated after the previously selected LIDAR from EAS was found to be defective.

Most LIDARs are “plug and play” meaning they have a USB connection and manufacturer supplied programs that allow for the point cloud to be viewed. Install this software on a laptop and connect the chosen LIDAR. Take the LIDAR outside and verify that the point cloud is able to detect objects like trees, bushes, walls, etc. The robot is going to be used outdoors so it is important that the chosen LIDAR is capable of detecting these types of objects at a distance of at least 5 feet.

4.0 Results

This section reports on all of the sensor findings collected for all of the experiments. The metal detector findings are reported first, followed by the infrared camera and finally the environment sensors.

4.1 Inductor Coil Measurements

Four coils were constructed and measured for resistance and Inductance. The results from this section can be found in Table 2. It is obvious that more coil windings increase inductance and resistance.

Table 2: Coil Measurements

# Coil Windings	Resistance (Ω)	Inductance (μ H)
6	0.195	54
12	0.313	218
15	0.339	286
30	0.650	1305

4.2 Optimal Coil Selection Results

The results from testing the maximum range of each coil are summarized in Table 3. Each coil had the same detectable range but the coil with 6 windings was not tested. When powering the board with this coil, there was an audible ring coming from the power supply capacitors. This is not normal behavior and could be a sign of damage occurring, so the power was immediately turned off. The resistance of this coil is very small compared to the other coils, likely causing the capacitors to discharge at a rate faster than intended, causing the ringing. Since the other 3 coils had the same range, the coil with 30 windings was selected for use in future tests since it demands the lowest power from the power supply given its larger resistance.

Table 3: Coil Depth Detection

# Coil Windings	Detectable Range (in)
6	Board Malfunction
12	11
15	11
30	11

4.3 Depth Detection with Varying Steel Mass

Twenty samples of steel ranging from 0.5g to 212g were tested for detectability. The metal detector was successful in detecting all samples with larger masses being detectable at farther ranges. The general trend of this data is that the detectable range increases rapidly when moving from 0g of steel to about 8 g. After this point, the detectable range begins to flatten out with the data trend suggesting the maximum range for this metal detector is about 18-19 inches.

Masses of steel that are below 0.5 grams also will likely be very difficult to detect. Table 4 and Figure 5 show the data trend with a logarithmic trend line.

Table 4: Steel Mass vs. Detectable Height

Mass (g)	Height (in)
212	17
152	16
112	15
78	13
62	13
50	12
42	12
34	12
28	10
26	10
22	9
20	9
14	8.5
10	7.5
8	7
6	7.5
4	5
2	4.5
1	4
0.5	3.5

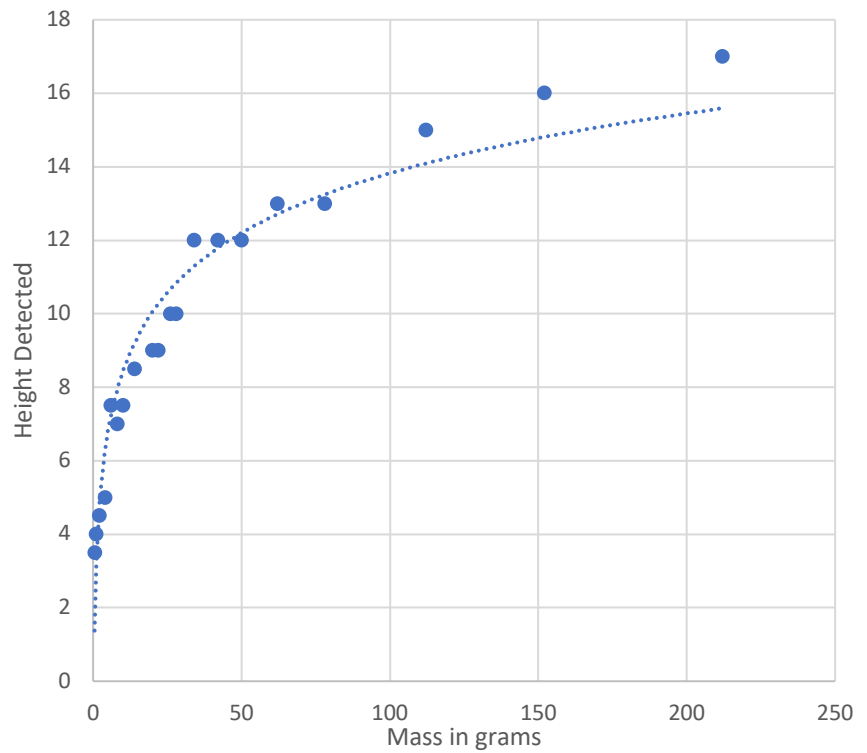


Figure 5: Graph of Steel Mass vs. Detectable Height

4.4 Depth Detection with Varying Surface Area

Three steel masses were tested to try to determine the effect surface area perpendicular to the magnetic field has on detectability. Table 5 shows this data with the “Low”, “Mid” and “High” values referring to the relative areas measured in Table 6. The test results show that detectability tends to decrease when surface area decreases but the sample remains thin and flat. The thin flat samples used for high surface area were rotated 90 degrees so that the steel thickness was the only surface area perpendicular to the magnetic field. The results from this test were surprising as detectability increased dramatically. There was a large surface area parallel with the magnetic field in these cases, which may have led to this result. The large 212g steel piece was turned and tested but the range for this sample decreased. This sample is relatively long so in the “low” surface area orientation much of the metal was over 19 inches above the coil, likely not returning an induced magnetic field. The data suggests that steel samples that are similar in proportion and orientation to the “low” surface area samples will have detectability increased by about 2 inches as long as the entirety of the sample is with the 19-inch detectability limit of the metal detector. The scaling factor in the “mid” category is less clear. The metal sample used for the “mid” measurement of 38g was much thicker than the other “mid” samples, which may be why the detectability is increased in that case.

Table 5: Detectability with Varying Surface Area

	Surface Area		
	Low	Mid	High
Mass (g)	Height Detectable (in)		
6	9.5	6	8
38	13.5	12	11
50	14	10	11.5
212	14	No Sample	17

Table 6: Surface Area Measurements

Mass (g)	6	38	54	212
Surface Area (in ²)	0.09	0.12	0.13	0.21
	0.88	3.75	3.75	-
	2.63	7.97	8.75	30.32

4.5 Effects of Interference on Detectability

The metal detector was tested for detectability of three samples of metal with DC motors and an aluminum spray can being used as interference. The Demining Robot is going to have motors and an aluminum spray can near the metal detector, so this test was designed to determine placement of these objects in the final design. The interference was placed in the middle of the coil to represent a worst-case interference scenario. When tested with one motor inside the coil, the detector was able to detect the steel samples at a decreased range. Using two

motors or the aluminum can as interference made the sensor completely unresponsive to the steel samples. Wrapping one motor in aluminum foil also made the sensor unresponsive to any other metal. An audible sound was heard coming from the aluminum can when placed directly in the center of the coil. Table 7 summarizes these findings.

Table 7: Interference Results

	1 Motor	2 Motors	Aluminum Can	All Three Items	1 Motor wrapped in Aluminum
Mass (g)	Height detectable (in)				
212	14	0	0	0	0
6	5	0	0	0	0
0.5	2	0	0	0	0

Further testing was conducted with the interference being above the steel samples and outside the coil. These tests determined that the interference from metal placed outside the coil at 3 inches or more is negligible. Interfering metal that is farther away from the coil than the sample intended on being detected reduces detectable range by about one inch at maximum.

4.6 Soil Effects on Detectability

The 6-gram steel sample from Section 4.3 was used to test detectability through various soil compositions. The initial intent was to bury the metal in the material and wave the metal detector over the ground to detect it. Unfortunately, since the coil is made from 18awg enameled wire which is very stiff, moving the coil was difficult to do without bending the wire leading to the metal detector. This bending caused the coil to warp slightly and resulted in a false reading. To combat this, the coil was placed underneath the plastic bin and the metal was waved over the surface of the ground material. This should not change results because the coil would still be detecting metal through the same amount of medium. This did mean that testing at various heights was not possible without changing the amount of dirt in the bin. The 6-gram sample was chosen because it was the smallest sample of metal still detectable through five inches of sand with the additional one inch of plastic at the bottom of the bin. In all four soil samples, the 6-gram sample was detectable at just above the surface of the soil, making the detectable range six inches. This is 1.5 inches less than the test in ideal condition so detecting through dirt and sand does decrease detectability but only by about 20% in this case. This data is summarized in Table 8.

Table 8: Soil Effects on Detection Results

Ground Material	Total Detectable Range (Air and ground material)
5in sand	6 in
4in dry dirt	6in
5in wet dirt	6in
5in dirt with grass	6in

4.7 Infrared Camera Test Results

Testing of the infrared camera has shown that the camera is very good at detecting landmines that are at least partially exposed but the ability to detect landmines underground is significantly less than expected. Testing of each landmine during cooling conditions was not possible due to the heat lamp being unable to heat the soil past about 0.25 inches in depth. This is likely due to testing being done in a cold garage, the heat lamp not being high enough power, and the time constraints to get testing done. The test set up of the heat lamp and the camera can be seen in Figure 6 with an example of a soil sample with the landmine in it in Figure 7.

In each test case with the landmine exposed, the heat lamp caused the temperature of the landmine to increase dramatically, making a clear distinction between the soil and the landmine. The images from each scenario can be found in Figure 8. It is visible in the grass test that the grass also heats very quickly along with the landmine.

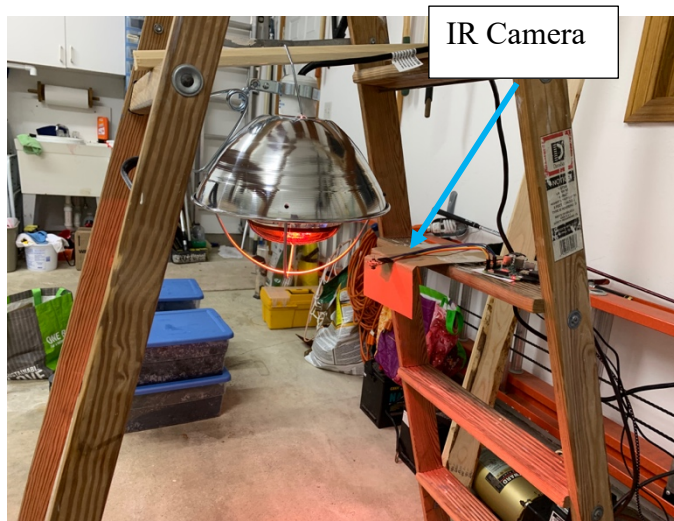


Figure 6: Set up of Heat lamp and IR Camera



Figure 7: Example of soil with exposed landmine

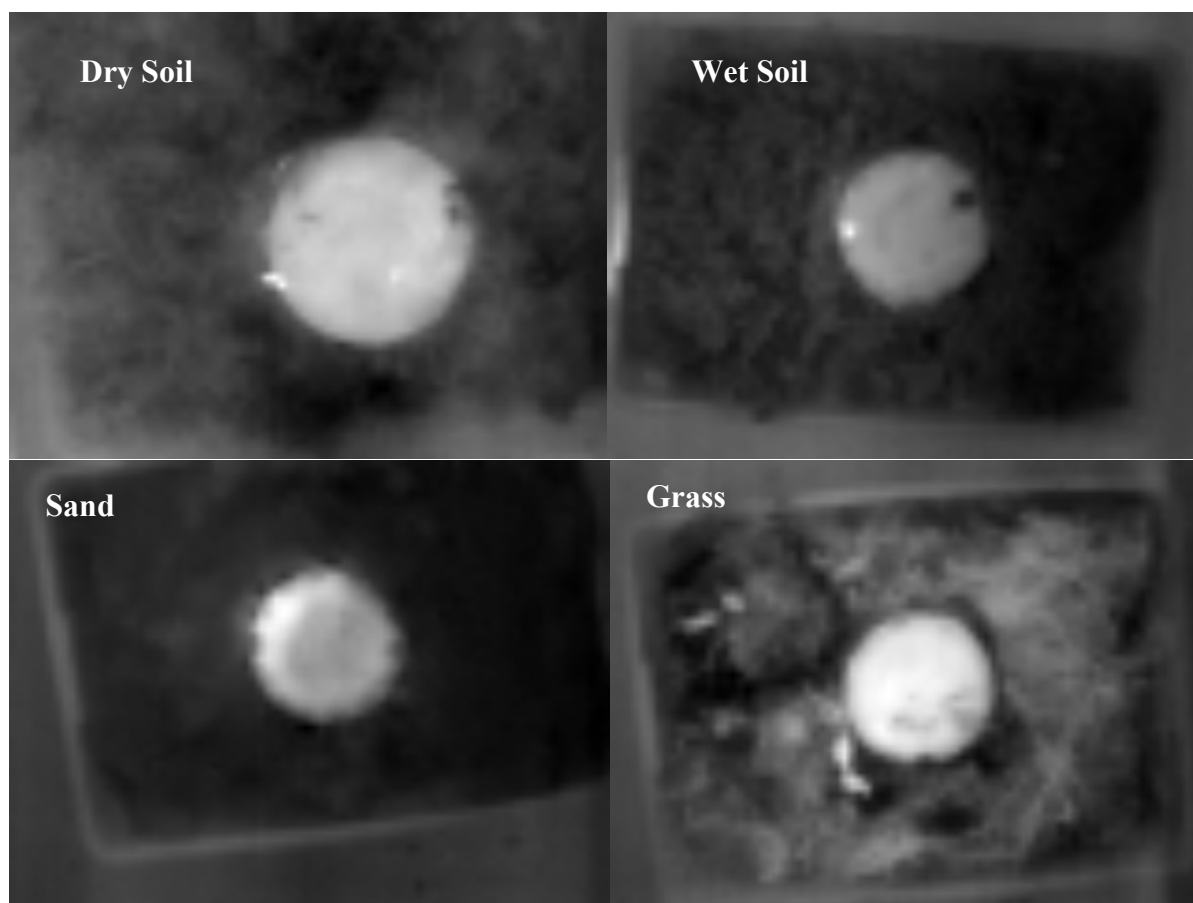


Figure 8: Exposed Landmine Images

Procedure 3.7 asks for the landmine to be buried in increments of 0.5 inches with the results of the previous test only a covering of soil between 0.25 and 0.5 inches was used as the heating source was not sufficient to heat more soil than that. In the case of the grass there was an inch of soil due to the roots of the grass holding the soil together. The results of this test show that the heating of the soil in each case increased mostly evenly leaving no sign of a landmine below. Overall, the tests showed that the soil in each case heated about evenly with only the sand having a faint outline. These images can be found in Figure 9.

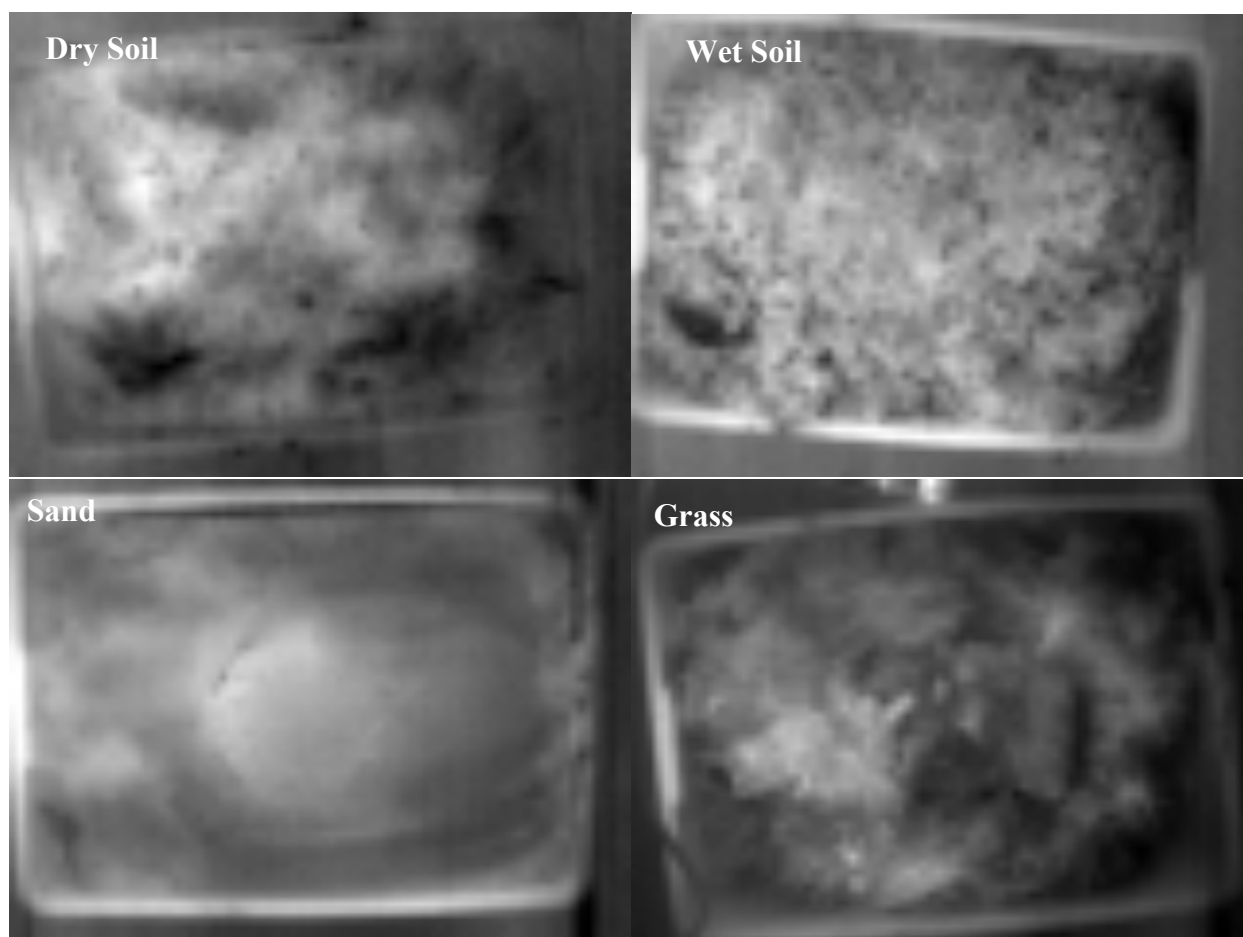


Figure 9: Landmine heating 0.25in-1in below soil

During cooling of the sand, it was interesting to find that the outline of the landmine became clearer, shown in Figure 10. This seems to be because the sand around the landmine was losing its heat much more quickly to the colder sand below it than the sand over the landmine. This did not occur in any of the other soil conditions as the heat distribution appeared to happen at a slower rate and distributed the heat more uniformly.

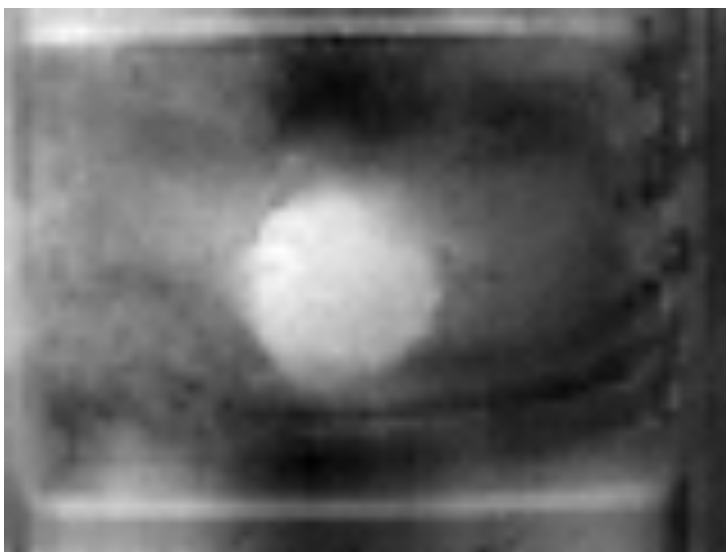


Figure 10: Landmine in sand cooling

4.8 Ultrasonic Sensor HC-SR04 Test Results

The results of this ultrasonic sensor on pavement were consistent for distances of 12 inches to 1 inch but the readings were consistently 1 inch greater than the actual. When tested on soft dirt, the measurements were consistent for 12 inches to 1 inch but again 1 inch greater. The measurements over grass had a lot of noise but most of the readings were within 1 inch of the actual amount for distance of 12 inches to 3 inches.

4.9 MaxBotix Ultrasonic Range Finder Test Results

The MaxBotix Ultrasonic Range finder was successfully able to identify trees and bushes from at least six feet away and overall was very successful at detecting large objects it was pointed at. Some examples of objects detected at this range are as shown in Figure 11. When tested against the wall, the radius of detection was eight inches around the sensor for objects 18 inches away and 12 inches for objects 30 inches away, verifying that there is a cone area that the sensor can detect obstacles within.



Figure 11: Examples of Tested Objects with the MaxBotix Ultrasonic Range Finder

4.10 Outdoor LIDAR testing

The Scanse LIDAR was successful in detecting objects such as trees, buildings and cars outside in the sun at distances of at least five feet. An example of the point cloud can be found in Figure 12.



Figure 12: LIDAR point cloud and actual objects

The Scanse LIDAR did show the defects the LIDAR has been known for. After a period of minutes to seconds, the spinning LIDAR would start slipping which caused the orientation of the laser to be lost to the logic circuit inside the LIDAR. This caused the LIDAR to enter an error state. This was solved by slowing the LIDAR down to a rotation speed of 1Hz and tightening screws on the bottom of the enclosure.

5.0 Discussion and Conclusions

This section analyzes the results from each section and discusses the potential errors that may have occurred during testing. The metal detector is discussed first, followed by the infrared camera and environment sensors.

5.1 Metal Detector Discussion and Conclusions

Overall, the metal detector testing was successful in verifying that the selected metal detector will be useful for detecting metal in ranges that landmines contain.

The tested coils all had the same detectable range, which was surprising but makes sense given the physics of inductors and magnetic fields. The inductor creates a magnetic field proportional to the amount of current running through it. The coils with a lower number of windings have less resistance resulting in more current. Higher current leads to a larger magnetic field. Lower coil windings also mean less inductance so sensitivity to the induced magnetic field decreases. This increase in strength of the magnetic field seems to have been compensated by the decreased ability to detect it. The 30-winding coil was selected because it will draw less current and therefore less power from the battery while providing the same detection capabilities. The inductance measurements were different than expected. Online coil inductance calculators suggested that the 30-winding coil would be about 1mH, while it is 1.3mH. The 12 and 15-winding coils were expected to be closer to 400 μ H but were in the 200 μ H range and the 6 winding coil was also half its predicted value. This does not seem to be a problem with detectability since all coils (except the smallest one) had the same detectable range, but it does show that there may have been error in the measurement equipment or with the coils. The 30-winding coil was more tightly wound than the others which may have resulted in the smaller coils having a lower than expected inductance.

The depth detection of the steel in ideal conditions shows that the metal detector should be able to detect landmines with 1 gram of metal at a 3 in depth, assuming there is no other metal interfering but if the metal in the landmine is any deeper the detector will likely miss it. Detecting 4-6 grams of metal had nearly twice the range so detecting landmines with this much metal or more will likely have a much higher rate of being successfully identified. One source of error in the ideal conditions testing was that a human arm was waving the metal over the coil probably causing some slight changes in height throughout the motion, although much effort was made to keep the arm steady. This is why measurements of height could only be made to the closest half inch even though the yard stick had 1/8-inch markings. Another source of error was that not all samples were perfectly flat and all contained holes. Heavily used vex metal was used and flattened to an extent but this test would have been better if the shape of the steel samples were more similar and contained no holes.

The effects on perpendicular surface area to the magnetic field are largely unclear and would require more careful testing to fully understand. It became apparent that the parallel surface area and thickness of the metal also played a role in detectability. In general, the more surface area of the object there is, the greater detectability but the object volume, size and shape effect detectability as well, making prediction of various shapes and sizes of metal difficult with the current data. For the purposes of the project, this data is sufficient in that this information suggests that objects with larger overall surface area are easier to detect than smaller objects. The obvious source of error in this experiment was that too many variables were different between samples making the correlation between perpendicular surface area and detectable range inconclusive.

The results of the expected interference were very insightful in how the design of the robot should be modified. Trying to detect metal closer than the interfering metal appears to be possible with only a slight decrease in detectability and metal over 3 inches outside of the coil seems to pose no interference. The magnetic field likely doesn't extend out farther than this, so the husky and four-bar should pose no problem in detecting landmines. The aluminum spray can in the center of the current design needs to be moved in order for the metal detector to work at all. Wrapping one motor, which previously still enabled good detectability, also stopped the metal detector from working. The aluminum can was emitting a sound when placed inside the coil, so it appears that aluminum is very effective at neutralizing a magnetic field. This suggests aluminum foil will be effective shielding for the motors actuating the metal detector. Although the motors will not inhibit detection of metal much based on the results of this test, they would set off the metal detector constantly as the coil is adjusted to be parallel with the ground. The robot may be able to detect the landmine but reliably distinguishing it from the motors will be impossible. Placing a sheet of aluminum foil between the motors and the coil so that the sheet remains fixed in reference to the coil seems to be a promising solution and will be an area of future testing when redesign of the metal detector sensing arm is being conducted. Overall, creating a design that allows the metal detector to operate reliably while still meeting the other mechanical goals of the project seems highly possible.

Testing the detectability through various soils also yielded promising results that the metal detector will largely be unaffected by the soil. It was unideal to have the coil underneath the various soils and the metal above, but the magnetic field still had to travel through the same medium. The detectability of the 6-gram metal sample decreased by 1.5 inches in every test. This suggests that as long as there are not large amounts of other metal in the soil, the exact composition will not matter in predicting detectability. The metal detector was successful at detecting the metal through 4-5 inches of soil and 1-2 inch of plastic and air. The metal detector will be operating through a plastic casing, a small amount of air and finally through inches of soil, so it is reasonable to assume that the robot will display the same capability once it is fully assembled and being tested in similar conditions. This further supports the expectation that the robot will reliably be able to detect landmines, whose metal is around 4-6 grams and now up to 5 inches in depth.

5.2 Infrared Camera Discussion and Conclusions

The only conclusion that can be made from this test is that the infrared camera should be able to identify landmines that are exposed on the surface as the plastic heats up much more quickly than any of the soil samples tested. The test equipment proved to be insufficient in thoroughly testing the sensor with landmines buried. All of the soil samples had moisture, adding unintended thermal mass to the sand and dry soil. This was due to rain dampening the bags of soil and sand from The Home Depot during the time of purchase. The 250W heat lamp was insufficient in heating the roughly 900 cubic inches of soil in each bin. The soil had been outside prior to testing with the temperature being about 40 degrees, and the garage was about 50 degrees during testing so much of the heat given off by the heat lamp was quickly lost to the environment. The results may have been better if the heat lamp heated the soil over a longer period of time, but the dry soil sample was heated for about 20 minutes with only about 0.25 inches of soil on the surface being warm. The following tests were done with the heat lamp being held close against the soil and moved around to speed up the process but heating each soil bin evenly throughout would have taken hours each if possible at all. It was promising to observe that the landmine under the sand became more visible with time after the heat lamp was

removed. This suggests that if the heat lamp were more powerful, the temperature difference between the surrounding sand and the sand above the landmine would have been much clearer during heating. It is also a notable source of error that the landmine used to test was a dummy landmine from a previous MQP that does not represent the actual thermal properties of the PMN family of mines.

The testing does emphasize how difficult it is to achieve a noticeable thermal difference in the soil. The wet soil had the most thermal mass making heating it take the most time out of all four soils. To the touch, the wet soil was still relatively cold compared to the other three samples having been exposed to the heat lamp for similar amounts of time and seemed to heat more evenly than the dry soil and grass. The sand heated more evenly than the wet soil but at a faster rate and eventually displayed a large temperature difference between the sand above the landmine and the surrounding sand. This suggests that soil that can be heated quickly and transfer heat to cooling soil below it at a fast rate will allow for the clearest images of the underlying landmine but much more testing needs to be conducted to be able to confirm this finding.

Further testing for this sensor is required to fully determine its effectiveness but these tests should be conducted either outside over the course of a day in an environment similar to where these landmines are located or in a controlled environment capable of simulating those same conditions. Being able to retest this sensor in one of these environments is not going to be possible given the time and resources available. However, it seems like the sensor should be able to easily sense landmines that are exposed on the surface. Future testing of the robot could include testing the metal detector and the infrared camera on landmines that are at least partially visible at the surface as a proof of concept that the automated detection abilities of the robot are operational assuming the landmine is possible to detect. There is not enough budget left to investigate any other sensors, but it is recommended that future work do so in addition to organizing a more rigorous and realistic set of tests for the infrared camera.

5.3 Environment Sensors Discussion and Conclusions.

The HC-SR04 Ultrasonic sensors were successful in detecting the ground accurately at close distances, especially over hard terrain. There was a considerable amount of noise when the sensor was over grass but a software filter to average out the readings should be sufficient to remove the problem. There will be 4-5 sensors on board the metal detector arm, so the combined feedback should provide enough data to accurately estimate the distance and angle of the ground.

The MaxBotix Ultrasonic Range finder was successful in detecting the desired objects and the cone area of detection had a large enough radius to enable the robot to detect objects between 3 and 27 inches high at a distance of 2.5 feet or further if the sensor is located 15 inches above the ground of the robot as currently designed. This sensor will be sufficient for sweeping the front of the robot for obstacle avoidance.

The LIDAR was effective at detecting large objects around itself as desired for use with the obstacle avoidance software to be implemented on the robot. The Scanse LIDAR tested did exhibit degradation of the mechanism that tracks the rotation of the laser, but at its current capacity it should fulfil the needs of the project. If it degrades further, this test is sufficient to prove that a similar LIDAR will be able to meet project requirements.

6.0 References

- Dekker, R. (n.d.) *A Simple Method to Measure Unknown Inductors*. Retrieved from <http://www.dos4ever.com/inductor/inductor.html>
- MaxBotix. (2017). *How to Use an Ultrasonic Sensor with Arduino [With Code Examples]*. Retrieved from https://www.maxbotix.com/Arduino-Ultrasonic-Sensors-085/?_vsrefdom=adwords&gclid=Cj0KCQiA2o_fBRC8ARIsAIOyQ-mPdfj5upq-IH-DRLJj0cAkWIyWI5kL72UM95R6czg-D8qTILCmHcEaAiQqEALw_wcB