Worcester Polytechnic Institute
Robotic Engineering Program

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:
Brendan Casey
Trevor Rocks

Submitted To:
Craig Putnam

Abstract

The goal of this project is to create an adaptable landmine detection platform to allow for autonomous marking and detonation of PMN-1 anti-personnel landmines without the need of endangering personnel and to make this landmine disposal operation economically feasible for poor regions of the world. This project is intended to be an adaptable prototype to be built upon by future teams. This project produced a prototype landmine detection and marking robot that utilized GPS localization, autonomous navigation and mapping, a prototype metal detection system and novel landmine marking system.

Table of Contents

List of Figures

Introduction

The Demining project is designed to create a safe, reliable, and inexpensive method of detection and disposal of PMN-1 and similar class landmines for civilian application with minimal training. This is the second phase of the de-mining project it is intended to be expanded on by future groups and focuses on detection and marking for disposal. This project uses an Unmanned Ground Vehicle (UGV) to search for, detect, and mark landmines – taking humans out of the most dangerous aspect of the demining. The final goal of detection and marking is to have a reliable method that provides a low cost solution to small towns and de-mining companies that works in concert with a compatible disposal system.

Landmines pose a serious threat beyond their impact on the battlefield. Many people across the globe suffer from landmines that cause crippling injuries or death. The populations who face the greatest risk from landmines are noncombatants in countries such as Iraq, Afghanistan, Sudan, Syria and Cambodia, which still have active cold war era landmines that have long outlived their purpose but still lurk just beneath the surface waiting to take innocent lives. This unintended effect on the people who come to live near them and the landmine's inability to discriminate against noncombatants is what resulted in the nearly global ban on landmines in 1977. Unfortunately, despite this ban, landmines continue to show up in conflict zones across the globe. In order to better prevent further mutilation and death of innocent people, accessible solutions are needed for both the detection and disposal of landmines in areas affected by conflict around the globe.

This project focuses on the detection and marking of cold war era PMN-1 antipersonnel land mines as they are easier to detect and are more commonly found than the newer versions. The PMN series of anti-personnel land mines is one of the most widely used landmines in the world. It was designed in the Soviet Union to be inexpensive and simple to manufacture, which led to it being one of the most commonly found landmines during de-mining operations. The PMN-1 landmine has an abnormally high content of explosives at around 240 grams of explosives compared to similar class antipersonnel landmines that average around 50 grams. This makes the PMN-1 a deadly threat to anyone unfortunate enough to accidently set one off. An effective detection and disposal system for this class of anti-personnel landmine could help save thousands of lives around the globe and make an appreciable difference in the de-mining industry.

Current autonomous technology is now well positioned to begin addressing this dangerous task. Rovers have been getting cheaper and more capable while the controller hardware has been getting faster and smaller. An autonomous rover equipped with landmine detection equipment, navigational sensors, and a marking system could potentially search an entire minefield without the need to endanger human lives.

Background

This section details the background research completed for this project. This includes the history and use of landmines, a summary of the current need for de-mining tools, an analysis of current technologies available for detection, an overview of the previous de-mining MQP and a description of the proposed solution for this project.

History and Use

Development

The idea of burying an explosive charge in order to cause a surprising explosion beneath an enemy has been around since the invention of black power. The Chinese military employed the precursor to the modern landmine against the Mongol Hordes of Kublai Khan in 1277. These mines were delicate, and used a trigger similar to that of a flintlock to detonate black powder, an unstable explosive. Figure 1 shows an ancient Chinese schematic for a black powder landmine, one of the earliest examples of such a device.

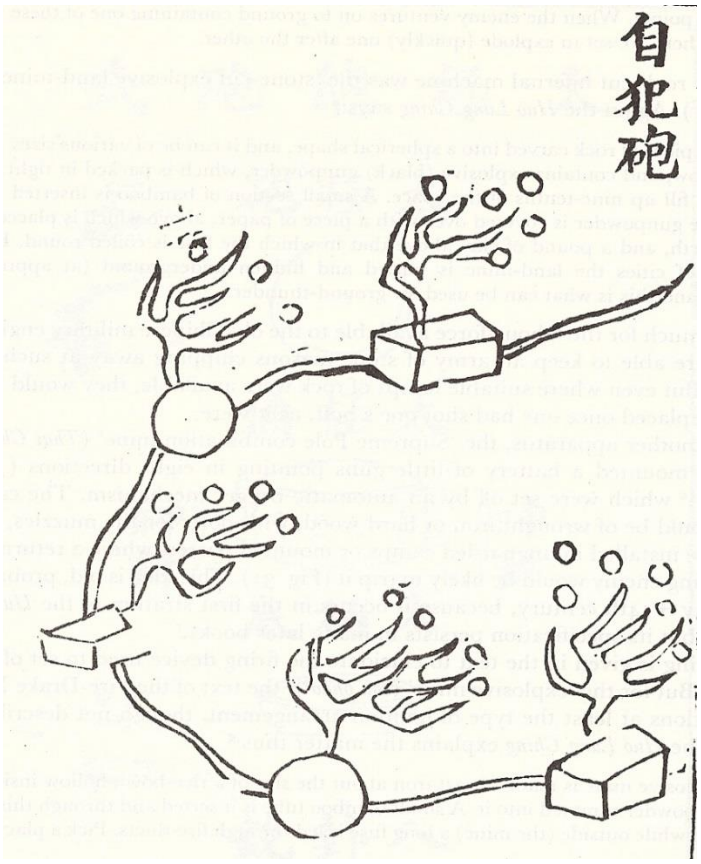

Figure 1 Ancient Chinese self-tripped trespass landmine[1]

[1] Yu, Jiao and Liu Ji. *Huolongjing (Fire Dragon Manual)*.

Because of their unreliability and instability, landmines were largely unused through the middle ages, although the concept remained. It was not until The American Civil War that explosive technology had advanced to a point where landmines could become widespread. Soldiers in the Confederate Army modified explosive artillery shells for use defending entrenched positions from Union soldiers. Confederate generals found the use of "land torpedoes" was effective in defending cities and military positions when soldiers could not be spared in certain areas (see Figure 2).

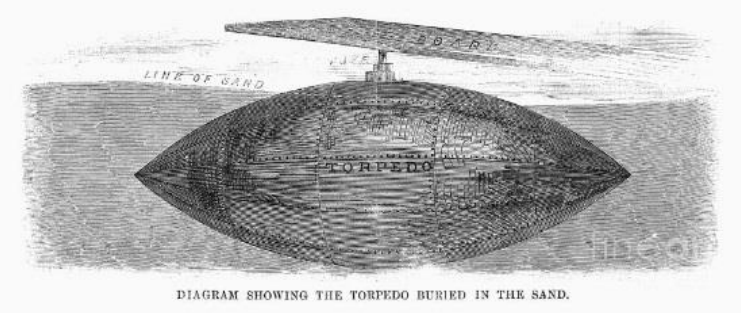

DIAGRAM SHOWING THE TORPEDO BURIED IN THE SAND.

Figure 2 American Civil War era landmine[2]

As a result, landmines became widespread throughout the war in the South. These Civil War era mines began to become similar to landmines used today. They consisted of a sealed, buried container, which used a pressure trigger to create an explosion causing metal fragments to injure or kill whoever steps on the landmine.

Due to the nature of trench warfare in World War I, traditional landmines could not be used extensively. More often, armies used the older style of tunnel mines, tunneling under enemy defense before placing a large explosive charge in the tunnel to destroy enemy fortifications.[3] Despite only minor use, landmine technology advanced during the First World War to become a highly effective weapon in World War II and every major conflict since. Both the Axis and Allied in the Second World War used landmines heavily in order to deny locations to enemy troops and slow the advance of enemy armies As such, mines were deployed throughout much of Europe, North Africa, and the Pacific islands. Figure 3 shows an example of a German landmine, showing the compact and simple design of mines of this era.

---

[2] "Mines." Weaponry in the Civil War., accessed 3/25/17, , https://sites.google.com/site/weaponryinthecivilwar/mines.

[3] Schneck, William C. 1998. "The Origins of Military Mines: Part 1." *Engineer Bullentin*. https://fas.org/man/dod-101/sys/land/docs/980700-schneck.htm.

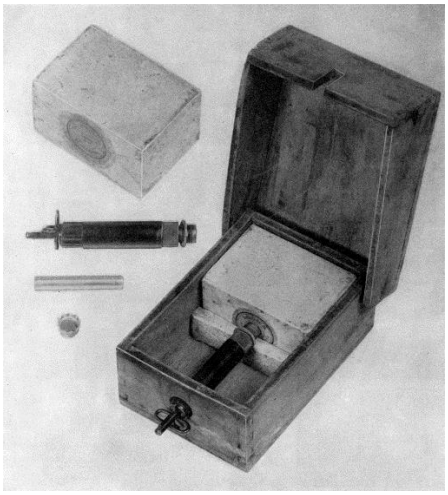

Figure 3 World War 2 Era German Anti-personnel landmine[4]

During the Cold War, landmines were used in many smaller conflicts around the globe such as in Asia, Central America and South America. This has resulted in deadly minefields needing to be cleared in every continent except Antarctica.

Use

Landmines are used primarily as defensive weapons in conventional warfare. They are used to slow enemy advances, deny locations to enemy troops, and focus enemy forces. Landmines are also used as harassment and demoralizing weapons in random attacks. While many minefields are marked, the location of individual mines is rarely recorded and tend to shift. Some modern landmines are designed to become inoperable or self-destruct after a certain period of time but this feature is missing in the more common landmines. Landmines are also used in guerrilla warfare where they are often placed singly and almost never marked making them especially dangerous and unpredictable.

[4] United States. War Dept. 1971. *Handbook on German Military Forces*. US Army Manual. US Government. http://hdl.handle.net/2027/ien.35556026179069.

Impact

Landmines are a serious threat facing many civilians in countries across the globe. Despite being designed for military use with guidelines mandating the clear marking of minefields, and a requirement for removal post-conflict, many landmines remain buried Often non-state parties such as terrorist groups place mines indiscriminately and without recording locations, without regard for international law. Their indiscriminate nature and long service life means that they remain a potent threat to anything or anyone unfortunate enough to detonate the mine. Every year thousands of people die or are permanently disabled by landmines.[5] In addition to the severe injury or death that may face the person who sets off the landmine, survivors place additional burdens on their communities as they are often left crippled, requiring lifelong assistance from their community.

Given their serious risk to human life and livelihood, landmine detection, marking, and cleanup has long been an important subject for human aid efforts. This has led to the development of safe and effective methods but most use tools that are expensive, costly to operate, and require skilled technicians. However, more affordable time-tested methods are what often get employed in the areas most effected where funds are limited and skilled personnel are short on hand. These methods tend to place humans at risk and require extensive training and caution to be effective.

The largest impact of landmine use in the last century has not been as a turning point in any major battles, but in the effect of minefields left behind when the war ends. While an armistice may cause an end to the battles some minefields are left behind. These minefields can remain active for decades. In 2015 landmines injured or killed 6,461 people in 61 different countries according to the United Nations. 79% of these causalities came from civilians.[5] Survivors of landmine detonation often lose limbs, which can ruin the quality of life for the victim permanently. Crippled survivors are often unable to work, facing a bleak future. In impoverished areas where proper medical care is often unavailable people who have lost limbs face many further physical and mental health issues resulting from their injuries. Even immediate survivors of a direct explosion from a landmine may end up dying as a result of their injuries.

In 1997, the *Convention on the Prohibition of the Use, Stockpiling, Production and Transfer of Anti-Personnel Landmine and on their Destruction* was drafted and presented to the United Nations. Also known as the Ottawa Treaty, this accord forbids any signatory nation from the use of Antipersonnel Landmines in any circumstance, citing the hundreds or deaths and injuries worldwide. The Treaty also forbids nations to research or further produce antipersonnel mines, and obligates nations to destroy their stocks of landmines, only keeping mines for the purpose of

---

[5] "Landmine and Cluster Munition Monitor." the-monitor.org., accessed 1/13/17, , http://the-monitor.org/en-gb/reports/2016/landmine-monitor-2016/casualties-and-victim-assistance.aspx#ftn2.

research and training in landmine removal. While the treaty seeks to solve the issue of antipersonnel mines, anti-vehicle mines, which are designed to only be detonated under the larger weight of a vehicle, as opposed to the weight of a person, are not prohibited by the treaty.

To date, 162 states have signed the treaty, with 35 states refusing to sign[6]. Figure 4 shows the nations in blue, which have signed the treaty since its creation. Noticeably among those 35 non-signatory states are China, Russia and the United States, all permanent members of the United Nations Security Council, and major global military presences.

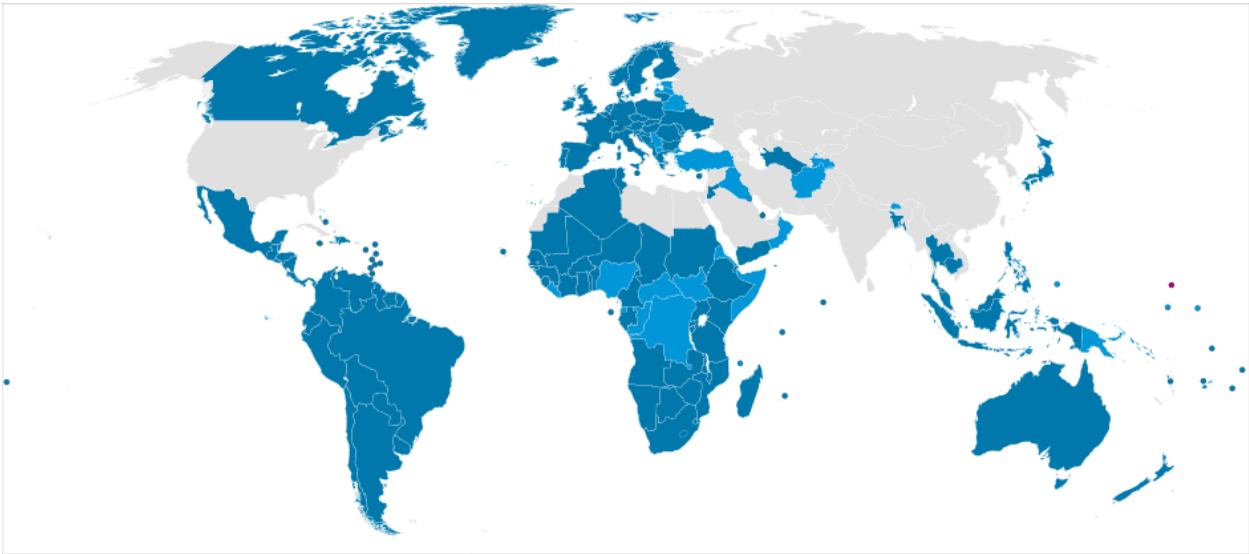

Figure 4 Map showing the states that have signed the Ottawa Treaty[7]

In 2014, the United States announced that it would be changing its policies regarding Anti-personnel landmines, and agreed to align its policies with that of the Ottawa Treaty, except with regard to the Korean Peninsula.[8] This is because while the United States agrees that the dangers created by using landmines are inherently dangerous to civilians, the military advantage, especially in the heavily militarized border between North and South Korea, can outweigh the

[6] "Treaty Status." The International Campaign to Ban Landmines., accessed 2/15/17, , http://www.icbl.org/en-gb/the-treaty/treaty-status.aspx#.
[7] By odder (talk) - Based on File:BlankMap-World6, compact.svg by Canuckguy (talk · contribs) et al. & Ottawa Treaty members.png by Gabbe (talk · contribs)., Public Domain, https://commons.wikimedia.org/w/index.php?curid=8563581
[8] "US Landmine Policy." US Department of State., last modified Sep 1, accessed 3/15/17, , https://www.state.gov/t/pm/wra/c11735.htm.

risk. Despite the non-signatory nations, the Ottawa Treaty is considered a success, and has helped to reduce the use of Anti-personnel landmines in numerous conflicts.

PMN1

The project focuses on the PMN-1 landmine, originally developed in the Soviet Union in the 1960s, and built in massive quantities. The design was distributed to other soviet nations, and the PMN-1 design was used extensively in China, Iraq and Hungary to produce near identical mines.[9] The PMN-1 has one of the largest explosive charges utilized in any anti-personnel landmine, with 240 grams of TNT packed into the Bakelite casing. The blast from the landmine would turn the Bakelite and metal casing into a burst of shrapnel spreading several meters. The lethality of the landmine, along with the simplicity of the design made the PMN-1 a cost effective and deadly weapon. As a result it became the most widely deployed antipersonnel landmine in the world, being used by many belligerents throughout the Cold War.[10] Figure 5 shows a cut away of the design of the PMN-1 mine.

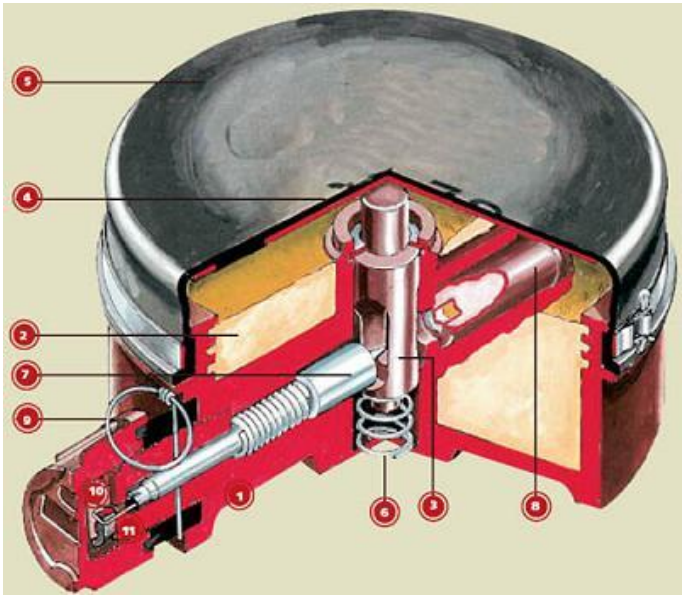

Figure 5 PMN-1 Cutaway[11]

The PMN-1 has a simple reliable design, encased in Bakelite, an early plastic, with a rubber top secured by steel bands. Once the safety pin has been removed, there is a several minute delay before the landmine is armed. Under the rubber top, there is a pressure plate covering the entire

---

[9] . *Munitions Reference Guide*. James Madison University.
[10] Swinton, R., & Bergeron, D. (2004). Evaluation of a silent killer, the pmn anti-personnel blast mine.
[11] "Soviet / Russia PMN-1 Bakelite Landmine." BuyMilSurp.com., accessed 11/15/16, , https://www.buymilsurp.com/soviet-russia-pmn1-bakelite-landmine-p-5098.html.

surface of the landmine. Any pressure on the surface of the landmine is transferred to the central plunger aligning the firing pin to release the striker pressed by a Belleville spring setting off the landmine. Only 10 Newtons of force need be applied to set off the explosive charge if it has not deteriorated due to age, overtime the mine can become more sensitive, due to deterioration and failure of the internal spring.

Detection

Detection is the most dangerous part of the de-mining operation as it places the personnel who must detect the landmines into unknown harm. Landmines are designed to avoid detection and some have anti-tampering sensors, which makes any attempts to detect and disarm the mine more dangerous. Traditionally, metal detectors have been used to detect the metal content in landmines. Because of this, landmine designers have continued to minimize the signature of the mines to the point where modern landmines are almost invisible to metal detectors. This development has led to the common use of prodding for mines, which place personnel at ever-greater danger and necessitated the physical interaction with the landmines. However, other techniques have been developed that utilize new technology for detection such as Infrared detectors/cameras and ground penetrating radar.[12] This section goes into further detail about appropriate systems for our application.

Prodding

The most available, and inexpensive method for detecting mines is by physically probing the ground to identify mines and unexploded ordinance (UXO). This is done by using prodders, which are typically 25cm long rigid sticks of metal often with a blast resistant guards to protect the hand of the deminer. Landmine detection using the prodding method is dangerous and slow. This is because the de-miner has no appreciable standoff distance and the detection area is very small for each probing action and each action must be done as if encountering a landmine. [13] Figure 6 shows an example of prodding being used to clear minefields in Bosnia, following the Bosnian War.

[12] Bruschini, Claudio and Bertrand Gros. 1998. "A Survey of Research on Sensor Technology for Landmine Detection." . http://www.jmu.edu/cisr/journal/2.1/bruschini.htm.

[13] Smith, Andy. 2014. "Understanding the use of Prodders in Mine Detection." *The Journal of ERA and Mine Action* (18.1). http://www.jmu.edu/cisr/journal/18.1/notes/smith.shtml.

Figure 6 Demining in Bosnia via prodding technique[14]

Metal Detectors

Metal detectors are also commonly used for landmine detection. They offer a wide search area, handheld versions offer more standoff distance than prodders, and they search more land, faster. Metal detectors are not without problems; with mines being made to have minimal metal content the metal detector's sensitivity must be such that many objects not of interest are also detected. This false positive rate can exceed 1000 per landmine detection slowing down the removal rate.[15] There are less common forms of metal detectors that have a lower false positive rate and to a limited extend even detect the type of landmine. Metal detectors are usually used to speed up prodding by detecting search sites rather than prodding everything.

---

[14] By Werner Anderson of Norsk Folkehjelp Norwegian People's Aid from Norway [CC BY 2.0 (http://creativecommons.org/licenses/by/2.0)], via Wikimedia Commons https://commons.wikimedia.org/wiki/File:Demining_in_Bosnia.jpg

[15]Takahashi, Kazunori and Dieter Gulle. "ITEP Evaluation of Metal Detectors and Dual-Sensor Detectors." *The Journal of ERW and Mine Action*. https://www.jmu.edu/cisr/journal/14.3/r_d/takahashi/takahashi.shtml.

IR

Infrared landmine detection technology uses the IR radiation difference caused by mines buried just below the surface to detect them. This is possible as the landmine has a different thermal mass than the surrounding earth. This makes its radiation stand out especially during peak times of day where the difference in temperature is change is most evident. This technology is more dependent on ground cover than other methods, but offers standoff distance and a lower false positive rate at the expense of computational and imaging power needed to process the data. With additional knowledge of the soil, IR systems can even differentiate different types of landmines. Figure 7 shows an example of IR feedback of a minefield, with landmines appearing in black as they maintain their heat more efficiently than the surrounding soil.

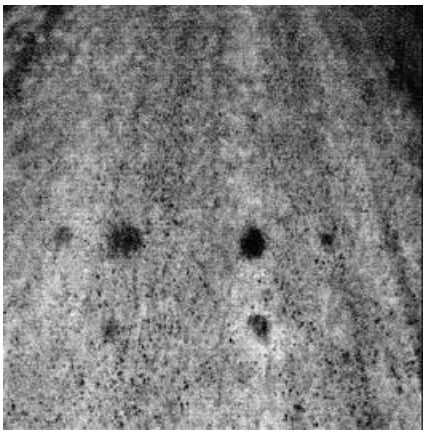

Figure 7 Nighttime IR image 1996[16]

GPR

Ground penetrating radar (GPR) is an interesting development in landmine detection as it uses radar pulses to map subsurface zones. As radar waves travel through the ground, different materials effect the waves in different ways. By measuring the strength of the returning waves, the depth and composition of different materials can be determined. This active detection method has the benefit of potentially finding out the position and type of landmine. GPR allows for highly accurate and reliable landmine detection, but comes with several drawbacks. The data that is returned from the sensor is difficult to interpret and therefore usually requires an experienced operator to be effective. There has been progress made by computer processing of the data but this requires a lot of computing power to be used in a firm real-time environment, typically in the form of an offsite server or a vehicle mounted computer rack. Additionally GPR generally

[16] Bruschini, Claudio and Bertrand Gros. 1998. "A Survey of Research on Sensor Technology for Landmine Detection." . http://www.jmu.edu/cisr/journal/2.1/bruschini.htm.

requires a relatively large amount of power for its detection reducing its operational time. The power draw is of addition concern where power is at a premium such as a on a robot, and in rural, recovering or under developed areas where landmines contamination is more of an issue. GPR is also highly affected by the type of surface being surveyed, needing to be calibrated to different soil types. The feedback is particular sensitive to soil with higher levels of conductivity, which limits the performance. [17]

Olfactory Detection

TNT and other explosives can be detected by scent. While these scents are too mild to be detected at any significant distance by the human nose, several animals have the ability to detect explosive scents in the range of parts per trillions.[18] The most common bomb-sniffing animals are dogs. Found worldwide and easily trained, dogs can smell tens of thousands of times better than humans. [19] A dog and its handler will head into a search area, and upon smelling a landmine, the dog will signal to its handler to mark the area for removal or destruction. Figure 8 shows a handler leading a German Shepard, a popular breed in landmine detection due to their powerful olfactory senses.

---

[17] Takahashi, Kazunori, Holger Preetz, and Jan Igel. 2011. "Soil Properties and Performance of Landmine Detection by Metal Detector and Ground-Penetrating Radar — Soil Characterization and its Verification by a Field Test." *Journal of Applied Geophysics* 73 (4): 368-377. doi:10.1016/j.jappgeo.2011.02.008. http://www.sciencedirect.com/science/article/pii/S0926985111000450.

[19] Tyson, Peter. 2012. "Dogs' Dazzling Sense of Smell." *Inquiry: An Occasional Column*, Oct, 4. http://www.pbs.org/wgbh/nova/nature/dogs-sense-of-smell.html.

Figure 8 Landmine Detection dog and Handler clearing farmland of mines[20]

This system has disadvantages as dogs will only work with their specific handler, and are heavy enough to detonate a landmine, meaning that a false negative can result in the death of the dog and trainer. Dogs are also relatively expensive, requiring training, dog food, shelter and often adapting poorly to foreign regions. Other programs have utilized rats for scent detection. Rats are cheaper to acquire, raise and breed than dogs, and can be trained faster. Rats also adapt to new environments easily and are cheap to transport, while being light enough to not set off landmines.[21] As you can see in Figure 9, the rat is working in close proximity to a mine as it conducts a search grid, without a risk of explosion.

[20] Hirsch, Jesse. 2014. "The Dogs that Sniff Out Landmines." *Modern Farmer*, Jun 11,.
[21] Sullivan, Michael. 2015. "In Cambodia, Rats are being Trained to Sniff Out Land Mines and Save Lives" *Morning Addition*, July 31,. http://www.npr.org/sections/parallels/2015/07/31/427112786/in-cambodia-rats-are-being-trained-to-sniff-out-land-mines-and-save-lives.

Figure 9 Rat searching along a line to detect landmines[22]

Olfactory detection has advantages over other types of detection, in that the animals are smelling explosives, not searching for metal or other materials underground. Metal detectors or GPR systems can have high rates of false positives due to other debris, and certain mines are designed to be undetectable by metal detectors, and are made almost entirely out of plastic or ceramic components. Dogs and rats can detect explosives of all types, no matter the composition, with a low the rate of false negatives. Any debris that does not contain explosives will be ignored, saving time and resources from being devoted to false positives. Unfortunately, there are a number of disadvantages to using animals to detect landmines. Often the price of training the animals can be high, several thousand dollars per animal, and the training process takes several months.[23] This makes it difficult to deploy landmine detection animals in large numbers, and limits production. Additionally, for the purpose of this project, taking several months to train an animal to detect landmines would be out of the field of robotics.

The Demining Project

The de-mining project is a series of projects that seeks to cover all aspects of the de-mining operation for civilian application. The de-mining operation is not only detection and disposal but also making a difference in affected areas. The first MQP to focus on the de-mining project addressed the issue of landmine disposal.

---

[22] "Hero Rats Sniff (and Snuff) Out Landmines and TB." 2014.*CNN Wire*, Sep 26,.

[23]Sullivan, Michael. 2015. "In Cambodia, Rats are being Trained to Sniff Out Land Mines and Save Lives" *Morning Addition*, July 31,. http://www.npr.org/sections/parallels/2015/07/31/427112786/in-cambodia-rats-are-being-trained-to-sniff-out-land-mines-and-save-lives.

De-Mining with UAVs

The MQP focused on using an unmanned aerial vehicle (UAV) to provide an affordable and safe solution for disposing of landmines show in Figure 10. They identified the PMN-1 as the target for the de-mining project citing it as it is a wildly produced, copied, commonly found and deadly antipersonnel landmine. The project did not deal with detection methods, assuming that the landmines would be clearly marked. An approximate GPS location of the mine is also required, and by searching for the visible marker, in the given GPS location, the UAV is able to target the landmine. Therefore, the rover must also be able to record the GPS location of the mine. The project developed a payload and software for an octocopter airframe that was made available through the robotics program.

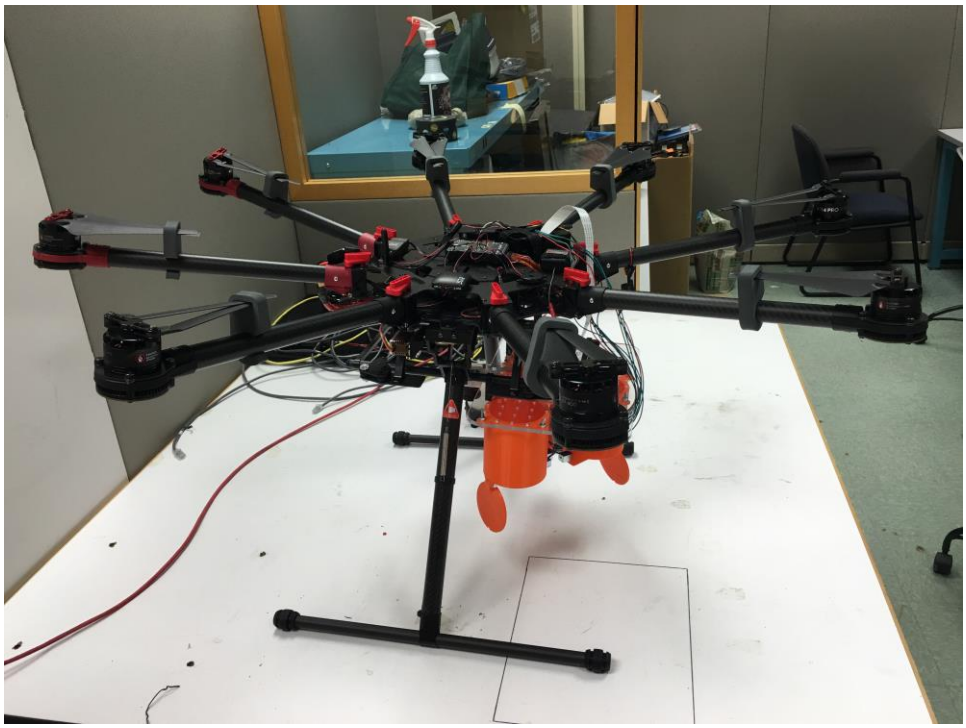

Figure 10 Demining UAV from 2016 MQP

The octocopter payload consisted of a payload bay, onboard computer, and camera system. The payload bay went through several cycles of design trying to make the system autonomously reloadable but ultimately was made to be four single use payload bays to hold the detonation method they designed and tested. This method used available soil and plastic wrap to make an inexpensive weight, of about 0.33 kg, that would impact the landmine once dropped from the UAV; this would detonate the landmine using its own trigger. They tested their method by creating a simulated landmine that can sense the impact force on its face plate using a calibrated strain gauge and relayed that information to an Arduino where it was recorded and for future analysis. The octocopter that was used had an onboard pixhawk computer for flight controls but this could not be modified so the team also installed a Raspberry Pi. The Raspberry Pi's job is to

interpret GPS data, accept external commands, evaluate raw image data, control the payload, and to send flight controls to the pixhawk. The Raspberry Pi Camera was selected as the camera to be used as it offered reasonable resolution and was designed to be compatible with the Raspberry Pi. The team calculated the fragmentation pattern of a PMN-1 landmine to create the optimal path away from harm, the end result was to have the UAV move horizontally at a minimal acceleration of 9.72 m/s^2 from an altitude of at least 20m hover over the landmine. Image processing utilized openCV square target identification algorithm. Due to the minimum altitude requirement, the algorithm had better results with larger targets but this was as much a restriction placed upon the vision system by the camera and lens as the altitude. In theory, any target can be used if an appropriate algorithm can be devised to detect it. They went on to outline several paths for future work to expand on their MQP. One development path of interest for our project is the recommendation for a de-mining rover. They suggested the creation of a detection robot that is recommended to be a rover equipped with a marking mechanism, Wi-Fi to communicate with the UAV or a controller, and a GPS receiver for recording the location of mines.

Proposed Solution

The proposed solution is implement phase two of the de-mining project by modifying a UGV platform to autonomous traverse a minefield, and mark detected landmines. The system would also need to ensure marked landmines have GPS coordinates that must be saved in a safe location and made available for the UAV from the previous de-mining project.

There were many different platforms to consider but ultimately we decided to use a UGV. For the UGV platform there were a great number of different kits, prebuilt UGV's and existing platforms to choose from. After looking at what was available within WPI, we decided that buying or assembling our own UGV would be unnecessary and wasteful. There were several UGV's available that were fully assembled and potentially available within WPI; they included the Husky A100, Walrus, and several custom-built platforms that professors had made available. After contacting the departments and owners of the various UGV's we acquired two Husky A100 rovers owned by Professor Michelson for the project. This rover was selected as it met and in many ways exceeded our initial design considerations. The specifics of the criteria and selected UGV are discussed further in the Methodology and Discussion section Unmanned Ground Vehicle Base.

An important decision that extended the proposed solution was whether to include detection as part of the proposal. Initially it was assumed that the landmine would be "detected" for us so that we did not need to be concerned with the specifics of how to detect a landmine. However, during discussions, it was decided that detection was feasible for the project so long as it was a stretch goal. The ramification of this decision is elaborated on in the Methodology and Discussion section Detection.

The proposed solution would be semi-autonomous - requiring some operator training. The system would autonomously search a given minefield and mark any detected mines. The marking system would deploy markers that should comply with the previous de-mining project's requirements for their marker and be able to mark multiple landmines. Upon detection, the system would transfer the GPS coordinates back to the base station and avoid running over the landmine.

Methodology and Discussion

Unmanned Ground Vehicle Base

The project intended to use an unmanned ground vehicle as the platform of the project rather than an aerial vehicle like the previous project as they tend to be more stable, have longer operating times and provide more space for attaching components. There were many options for selecting a ground vehicle particularly within the robotics department. To choose an appropriate base that would best suit our requirements we began the selection process.

Selection

The rover had to remain operational in terrain where a minefield might be located. This means that the rover needed to be capable of traversing moderately rough terrain easily such as hills, deserts and grasslands. Operation time was a big requirement as it directly impacted the ability to detect and mark landmines. We speculated that in order for the de-mining rover to be efficient it had to have an operation time of at least one and a half hours to traverse and operate over significant portion of a minefield. We considered a weight restraint to avoid setting off the landmines but after further research particularly on the deterioration of antipersonnel landmines we found that the rover would need to apply less than one kilogram of force to the surface of the landmine in order to ensure that there would be no detonation.[24] However it was expected that the weight of the plausible detection equipment, marking equipment, onboard electronics and batteries alone would cause the rover to exceed the one kilogram limit. So the weight constraint was changed to limit the weight of the rover such that the operator would not need to deploy the rover with any additional equipment that could not be easily obtained in the expected regions of operation. Additional considerations but not requirements for the UGV included the presence of an onboard payload bay or mounting surface to house and attach electronics, existing software that could be easily modified, and additional onboard electronics such as a GPS sensors or wireless receivers and transmitters.

After considering several options including the walrus rover and a stripped down drive train given to us by Professor Putnam we were informed of the Clearpath Husky A100 rover that was available through Professor Michelson, which immediately stuck out as an obvious choice.

Clearpath Husky A100

The Husky A100 was known to operate in rough terrain and its previous work under Professor Michelson showed that it was more than able to meet this requirement. It had an onboard power supply capable of powering the rover for more than the time requirement assuming that there is no additional draw on the batteries. Although the Husky A100 is a relatively heavy rover with batteries installed it is a manageable weight, we were able to lift and move the rover with two

[24] "Soviet / Russia PMN-1 Bakelite Landmine." BuyMilSurp.com.,

operators. It also had some nice features including a protected payload bay, a top mounting plate, a serial communication port, and power ports. Michelson also gave us the computer previously used on the rover by a different project. The computer is a single board machine that runs Linux, has USB, VGA, HDMI and Ethernet ports, and has its own mounting point inside the payload bay. The Husky utilizes this onboard computer to talk to the MCU, which controls the motor driver, houses the IMU sensors, and connects to the motor encoders. The MCU is directly connected to the E-stop controls mounted on the side of the rover. The rover uses 6 wheels connected by belt in a tank drive configuration to 2 CIM 12 volt DC motors giving it great off-road characteristics. The internals of the rover are shown in Figure 11. The rover has been known to drive through walls if improperly operated. The Husky A100 is not equipped with any suspension so any undulation in the terrain is transferred directly to the rest of the rover. Despite this it is a very stable base for our possible detection equipment due to the low operation speeds demanded by the detection methods. The Husky A100 does come with some limitation as it should not be placed in certain extremes such as traversing slopes of greater than 40 degrees on soft ground, solid ground obstacles higher than 3 inches, or converging slopes of greater than 20 degrees as this would potentially result in the rover becoming stuck. In testing operation, this would not be much of an issue as it could be retrieved but if operating in a minefield becoming stuck would likely be a death sentence for the rover.

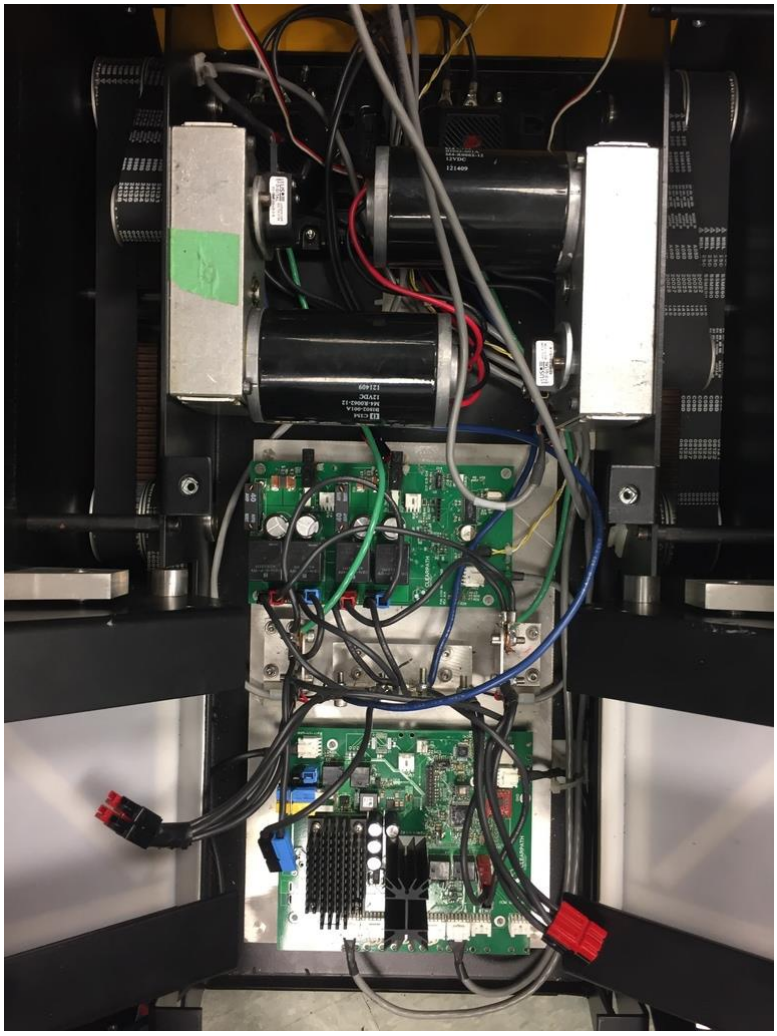

Figure 11 Polly Internal

Setup

Initial set up of the rover faced several challenges. We were given access to a pair of rovers, nicknamed Polly and Wolfgang, neither of which were in proper operating condition. The rovers had been unused for several years since a previous project that worked on creating a new navigation system. In order for us to use the rover, we had to address and repair these issues. Hampering our efforts was a lack of technical specifications on the rover. Clearpath Robotics has not produced the A100 rover in several years, having moved onto the production A200 model. Despite reaching out to Clearpath as well as several other universities that have published projects using the A100 we were unable to find proper documentation on operation and maintenance of the A100. Luckily, software support was available through the Robot Operating System (ROS) wiki as the A100 is compatible with the same software as the newer A200, but no data sheet or user manual was available, so we were left to guesswork. After locating missing batteries and chargers, it became clear that the internal systems of the Wolfgang rover were not

operational. To get at the internals the protective internal plate had to be taken out but as there was no documentation on how to remove the plate it took more time than we had anticipated. After removing the plate, it was found that the previous project to use the rover had used their own control system and modified the rover's control circuitry to work with it. This resulted in the rover's internal circuitry being disconnected including the safety measures, and the encoder wiring being cut. As we could not be sure if there were any other missing parts and did not know what was required to have the rover be fully operational due to the lack of documentation, we could not safely use and operate Wolfgang. The rest of the Wolfgang rover was functioning, with the onboard computer running properly and both batteries properly maintaining a charge. Upon inspection of the second rover, Polly, it appeared to be in better shape. While most of the power and communication connections within the rover were unplugged and disorganized we were able to reconnect the proper terminals and power up the MCU and motor control system within the robot properly. Unfortunately, the onboard computer for the Polly rover was not operational, so the working computer from the Wolfgang rover was transplanted. The setup process was not done as when we tried to communicate using the previous projects code to test tethered operation the connection was refused and we did not understand why, after digging into the code and documentation we found that the wrong port had been used in several configuration files. Once all of this had been worked out we were able to run the diagnostic report program and test tethered teleoperation by using keyboard control to ensure the rover was working correctly. Before we got to work on the rover, we were told by Professor Michelson of an issue the previous team ran into when running the rover. The previous team had found out that when running the rover the motors could draw enough power that it would cause a brown out for the external power supply connector meaning the computer would be reset. To resolve this we acquired new connectors and setup an additional battery that was isolated from the rest of the power supply to avoid brown outs.

Communication

While tethered operation was fine for testing the rover it was not sufficient for use in the project, we required wireless control. As the rover would be mostly self-contained and only had to report the location of the landmines we considered a hobby radio solution. Professor Putnam has worked with ZigBee radio systems in the past and recommended that we use their system for wireless control. After evaluating the system, it was decided to not go forward with the use of a hobby radio solution. When we reevaluated the needs of the wireless communication system it was decided that future projects might need the rover to communicate more than just the landmine location back to the base station such as navigation commands which would require us to create our own custom messaging protocol which was beyond the scope for this project. However, we still needed wireless communication for the rover to be usable. We looked into the previous projects on the Husky and they had used a wireless router, which Professor Michalson was able to supply us with, along with the proper connectors to allow the router to operate from a

12-volt battery. This enables the rover to handle substantially more information than the simple implementation with the ZigBee would have allowed.

Husky Launch

The Husky A100 was designed with ROS in mind and so should integrate nicely with the move base package with minimal configuration allowing the rover to accept simple navigation goals. However when we attempted use this functionality by running the husky launch file that should have started up the husky ROS node it failed to run the node. After tracking down the file it became known that it and it is depend files had been modified to an unknown configuration. Unfortunately, the unmodified file was nowhere to be found and there was no known launch configuration that worked. To overcome this problem we decided to use a simulation of the rover for testing. To model the rover and its onboard sensors including the GPS we customized a gazebo model of the husky A200, the newer model of the Clearpath rover that has many similar characteristics to the A100, as seen in Figure 12.

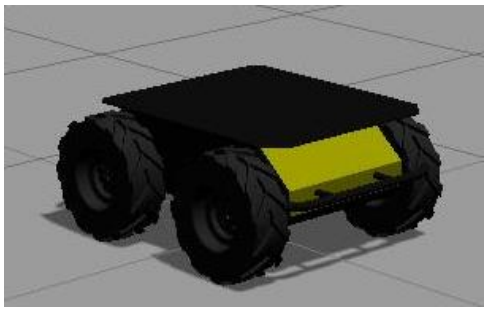

Figure 12 Gazebo Model of A200

Navigation

While the rover is able to accept simple move commands using only the move_base package to drive around it is not enough, for the project it needed to traverse an entire minefield. Initially we intended to create a lawnmower search pattern algorithm to efficiently traverse the minefield. The algorithm would take in the map as a grid that represented the locations the rover could move to. This grid would have a resolution small enough that we would not miss any section of the minefield. We would then traverse the grid representation of the minefield using a heuristic search algorithm that would try to avoid researching an area and turning where possible but would not miss any section of the minefield. Despite its promising start the search algorithm proved to be more difficult to implement and after we looked into how the minefield could be represented and its possible complexity, we decided that we could not make our own algorithm that would ensure the entire minefield would be traversed. We instead turned to using the frontier_exploration package.

One of the nice things about this package is that it has a way of artificially bounding the search area in any shape by using its exploration_client. This client listens to the clicked_point topic typically used by RVIS, an interface program for ROS, and is documented as a way of

demonstrating the capabilities of the package by allowing the user define a custom shape to act as a boundary for the program to search within.[25] The frontier_exploration package does this by looking across the boundary regions, where known regions and unknown regions connect, in the ROS map topic that are not blocked and creating a list of frontiers that represents these boundary regions. The package then chooses a frontier and publishes a navigation goal that the move_base package uses to move the rover to that frontier. While frontier exploration is a feasible solution to the problem of navigating a minefield and not missing anything it typically uses the gmapping package. Gmapping is a bulky program that uses LIDAR sensor data published as laser_scans messages or camera data published as point_cloud messages to do mapping, localization, and to create a cost map that represents obstacles. The rover does not have this sensor nor did we plan to install the sensor as it would increase the cost of the de-mining system. Initially we decided to do away with gmapping and to create our own program that would replace it and work with our detection system but after looking deeper into what gmapping was actual doing it was decided that such a program was out of the reach of our two-man team to stay within the desired timeframe. None the less by using the frontier_exploration package we could define and completely traverse the minefield. Our solution to the gmapping problem can be found in the Detection section Fake Sensor.

Detection

Selection

Several options were investigated for our landmine detections sensor, including ground penetrating radar (GPR), infrared scanners and metal detectors. Various factors were considered in our selection of a landmine-detecting sensor. The detector must be low cost and require minimal computing power to operate, to meet our goal of keeping the rover low cost. Furthermore the sensor should be able to work in a large variety of environments, as landmine contamination is persistent in countries around the world. Ground penetrating radar provides a clear image of what is buried beneath it, with many commercially available models scanning 6 feet through the ground. While this could be a reliable landmine detection system, there are several drawbacks to using GPR. Our research shows that most landmines are buried under less than a foot of dirt, meaning that the ability to search deep into the ground is unnecessary, and the data from a GPR is complex, and would require a large amount of computer power onboard the rover to analyze. Additionally GPR systems can cost thousands of dollars, which would drastically increase the price of our rover. Infrared scanners can detect landmines by distinguishing the contrasting surface temperature of soil and the small surface of soil on top of a landmine. While this is a viable method, infrared scanners are greatly affected by environmental

[25] "ROS.org." *Robot Operating System*. wiki.ros.org.

factors, such as soil type and weather conditions. Additionally the data returned from an infrared scanner would require high amounts of computer analysis.

Metal detectors are reliable throughout varying weather conditions, and can be made at a low cost. The majority of common landmines contain high enough metal contents to be detected by a metal detector, including the PMN-1 landmine that we chose to focus on. Metal detectors are widely used to detect landmines throughout the world, and currently the United Nations utilizes personnel equipped with metal detectors to clear landmines for humanitarian purposes. During military operations more aggressive clearance options are used such as carpet-bombing potential minefields, or plowing through the mines with heavily armored vehicles. While some mines can be designed with minimal amounts of metal to avoid detection by a metal detector, these models are rare. For these reasons, we chose to use a metal detector as our landmine detection sensor.

In order to increase the area our robot can scan, we chose to use a system utilizing multiple metal detector antennas, working with the same circuitry. While it would be possible to search using a single antenna head on a pivoting arm, this would slow the movement speed of the robot, and add complexity to the system, increasing the price of the rover, therefore we chose to build the system with several separate detection antennas. This allows us to scan the entire front section of the robot, which ensures that the robot will not run over any landmines. The Arduino board used in the analysis of the metal detector data is the most expensive component at 25 dollars, so by using a single set of circuitry, switching between multiple detector antennas, the price remains low, and the search area is increased.

Design

For our detector we built a metal detector based on the circuitry shown in Figure 13. By using a voltage controlled switch to shift between several metal detector antennas, we are able to use one set of circuitry and a single Arduino Uno board to control and analyze the output of our metal detector. This helps to increase our search area and accuracy, while using only one Arduino Board.

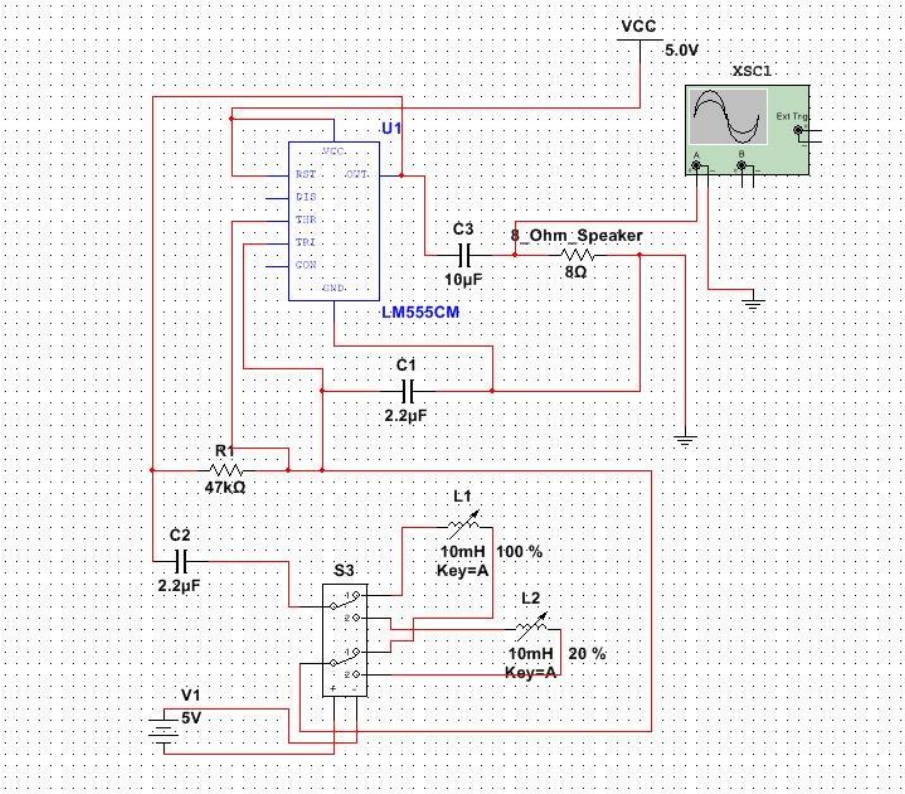

Figure 13 Circuit diagram utilizing a voltage controlled switch

The coil on the metal detector (represented by a variable inductor in the circuit diagram) modifies the wave passing through the RLC circuit as the inductance value of the coil is modified by the presence of nearby metal. This in turn can be read, traditionally by the tone of a speaker, but in our case by reading the analog input signal to the Arduino board. In this simulation, the Arduino board reading the analog wave is represented by an oscilloscope. Because the Arduino cannot accept negative voltage signals, a half wave rectifier circuit was utilized to remove the negative half of the wave before being read by the Arduino.

Construction

The metal detector was built using various electrical components from the WPI Electrical and Computer Engineering Shop. In the place of the variable inductor in the simulation of the circuit, we built a search coil using 30 feet of enameled copper wire. Enameled wire has an extremely thin insulation, with a thickness of roughly 7 μm. This allows for a tight coil, necessary for an inductor such as the search coil, and ensures that there will be no shorts in the coil. The search coil prototype can be seen in Figure 14.

Figure 14 Metal Detector Search Coil

The remaining circuitry required was constructed on a breadboard, and attached to both the Arduino and the search coil. The metal detector was designed to use a 5-volt power supply, which is provided by the USB power supply to the Arduino. The final construction can be seen in Figure 15.

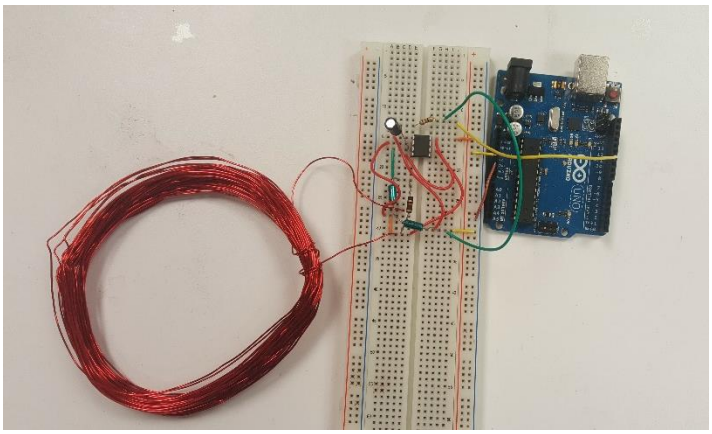

Figure 15 completed metal detector including circuitry and Arduino reading board

The code controlling the system works by sending a pulse through the circuit, and reading the resultant pulse back into the Arduino. This code can be seen in Appendix C: Arduino Detection and Marking Code.

Testing

In testing our metal detector worked reasonably well. A clear variance in the wave was detected on an oscilloscope in lab conditions when steel was present within 1 inch of the detection coil. This proved to be a proof of concept that our simple and low cost metal detector could be developed to be an operational landmine detector. Our testing showed that if the length of the

pulse is return at greater than 920 microseconds that the coil is detecting nearby metal. In order to increase the sensitivity of the sensor the values of the RLC circuit can be modified and calibrated, and a larger detection coil could be created. Increasing the size of the coil increases the inductance and makes the search coil more sensitive to the presence of metallic objects. We did not pursue the perfection of our metal detector, as it was a stretch goal of our project.

Fake Sensor

In order for the robot to work correctly, the navigation system needs to know where the detector has scanned. Initially we created a cost map that represented the detector head, which is documented in the Appendix section Cost map sensor. This program was positioned correctly in front for the rover but was not compatible with the gmapping package and so could not be used. When we looked into the sensor data that gmapping uses we found examples for creating both the point could and laser scan data types. We chose to use the laser scan message as it best represented what we wanted to do being a 2D sensor sweep in front of the rover that could project an obstacle if we changed the range data. We created a program, documented in the Appendix section Fake laser scan that would publish a laser_scan message at 50 Hz whose range was determined by subscribing to the Arduino ROS topic. The program is able to switch between being far enough away to represent the detector sweep and not detecting the mine and close enough to place an obstacle in front of the rover that the navigation system would then avoid as seen in Figure 16. The range data was set to expire just before the detector head by modifying the gmapping configuration file parameter maxUrange, which sets the usable range for the sensor data. This program could also be triggered if the user published on the Arduino topic that a mine had been found allowing for simulation testing without the rest of the detection system.

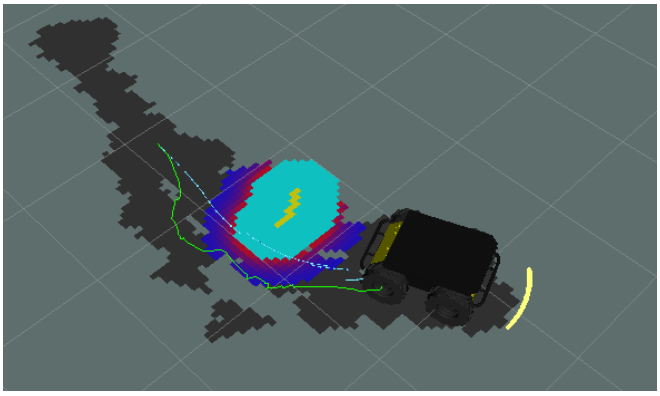

Figure 16 landmine detection and avoidance

Arm

Design and Construction

In order to allow the robot to properly detect a mine, and maintain a safe distance from the mine the detection suite is placed on a 2-foot long arm. Each arm consists of a parallel four bar system. In order to allow the detection suite to maintain the optimal distance from the ground through

whatever contours may be present in the terrain, each side of the arm is independently driven and controlled. The arm was designed using Dassault Systémes SolidWorks Computer Aided Design Software. This allowed us to simulate the arms motion, and place the arm on a facsimile of the Clearpath Husky A100 to ensure proper functionality of the system when created in the real world. The model is shown in Figure 17.

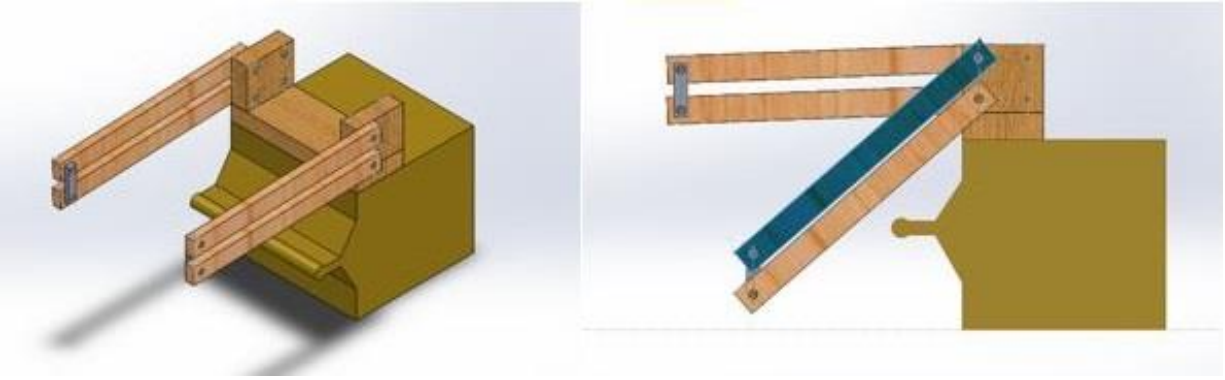

Figure 17 Left: Arm mounted to Husky Facsimile, Right: example of arm reaching the ground without interfering with rover

We chose the GB37Y3530-50EN DC motor to drive our arm, and utilize a 1:39.27 speed reduction gearing system. By using a worm gear stress on the motor is reduced, as the worm gear cannot be back driven, and well as generating a large amount of torque for the motion of the arm. The DC motor produces 4.4 Newton Meters of Torque, which is multiplied to 172.8 Newton Meters or 127.45 Foot Pounds by the gearing ratio. This amount of force is significantly higher than any we expected our arm to encounter, but is more than adequate for our needs. In order to reduce the price of production of the arm, it was designed and built from Oak and Cedar wood. While a steel arm would be stronger, the wooden components are lighter, lower cost, and readily accessible in rural areas. Additionally the wooden components allow for flexing of the arm, so that the arms can maintain different angles. The arm was designed to hold a sensor suite weighing a maximum of 10 lbs. On a two-foot arm, that creates a downward torque of 20 ft. lbs. Therefore, our arm must be able to tolerate that downward force. We calculated the deflection caused by this level of force on each of the arms, given that the four bar give 2 beams worth of support per side. The calculations below showed that the deflection caused to the arm by this force would be 1.34 inches, which we considered negligible for our uses.

$$\theta = \frac{Pl^3}{3EI} = \frac{5\ lb * 60.9^2}{3 * \frac{1}{6}^4 * .93 * 10^7} = 3.21^\circ\ \sin^{-1}(3.21) * 24\ in = 1.34in$$

The SolidWorks simulations verifies that a downward force of 5lbs per arm would not cause any damage to the arm, nor would it displace the arm an unreasonable amount. The analysis showed less displacement than calculated, a maximum of .804 in. This analysis can be shown in Figure 18

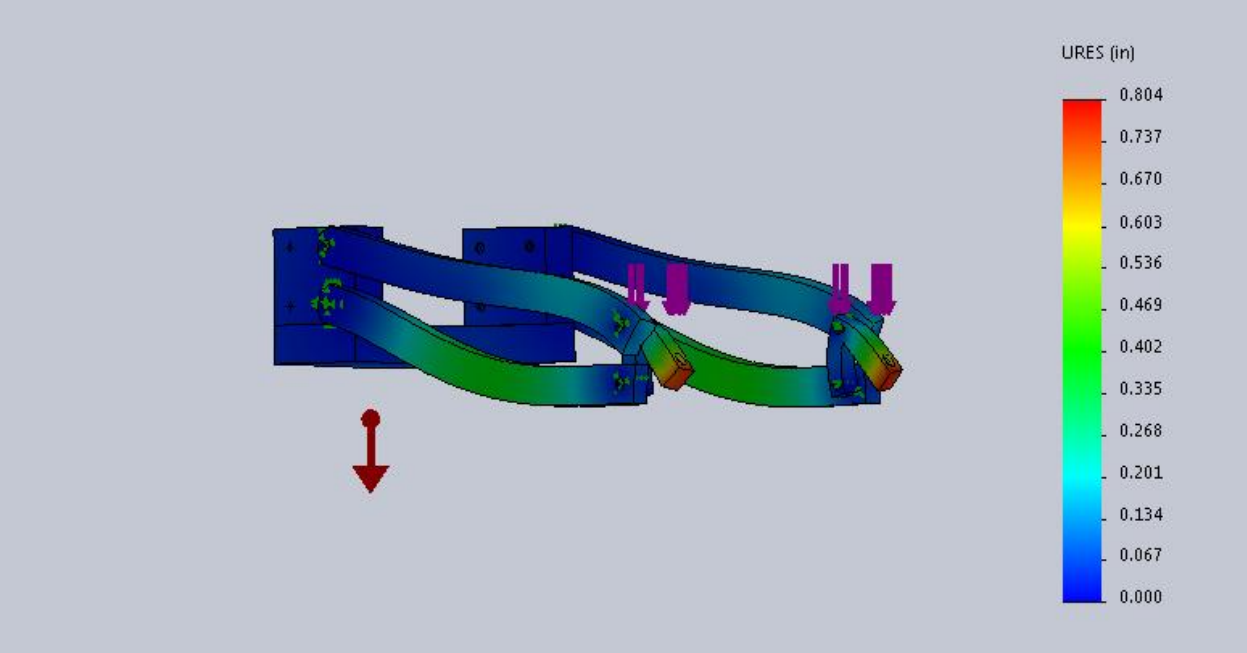

Figure 18 Downward force resulting in only .8 inches deflection

While the arms are both driven independently, allowing the difference in angle of the arms to be too high could damage the arm system. Therefore, we decided on the constraint of a difference in angle of 10 degrees. Working with the constraint that the arms must be within 10 degrees of each other, out calculations show that the change in distance between the ends of the arms will not exceed .359 inches. This means that each arm would have to flex at least .179 inches over the length of 2 feet. Figure 19 shows a plot of the distance between the ends of the arms as a function of the angular difference between the two arms.

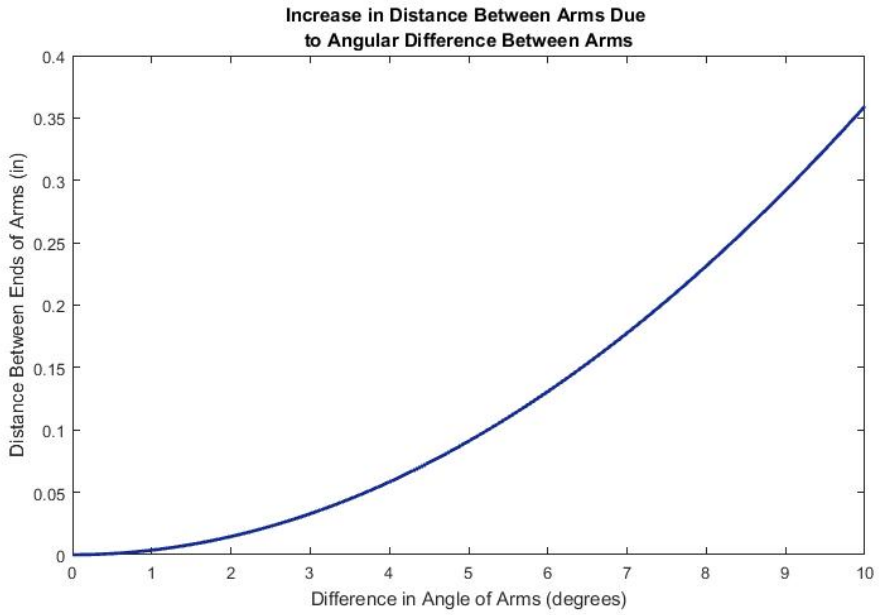

Figure 19 Plot of the distance between the ends of the arms

With this data, the next step was to determine what materials could flex this distance without damaging itself or requiring an unattainable amount of force. In order to determine the force required to apply this type of deflection on various materials we used the equation shown below. We used inches for our calculations.

$$P = {3 * E * I * \delta}/{L^3} \quad Where\ E = Modulus\ of\ Elasticity, I = area\ moment\ of\ Inertia,$$
$$\delta = maximum\ deflection\ distance, and\ L = length\ of\ the\ beam$$

By looking through the modulus of elasticity of several different types of wood, we settled on Cedar Wood, which has a modulus of elasticity of $.93 * 10^6$. Cedar allows for a proper deflection of the beam, using only 6.05 newtons without failing. Cedar is also an inexpensive material available worldwide, which helps to keep the price of the arm low. The modulus of elasticity of Atlantic White Cedar is .93* 10^6.[26] The Area Moment of Inertia is calculated as shown:

$$E = .93, I = \frac{1}{12} * 1 * 2 = \frac{1}{6} inches^4, \delta = .18\ inches$$

[26] "Wood Strengths." WoodWorkWeb.com., http://www.woodworkweb.com/woodwork-topics/wood/146-wood-strengths.html.

$$P = \frac{3 * (.93 * 10^6) * \frac{1}{6} inches^4 * .18\ inches}{24\ inches^3} = 6.05\ Newtons$$

This tells us that the force required to cause the inward deflection is only 6.05 Newtons total or 3.025 Newtons per arm. To verify that this force would cause the proper deflection the SolidWorks model was analyzed. SolidWorks has the ability to simulate the effects of various forces on a simulated object, and produce the effect those forces will have on the materials used to create the object. By applying the proper inward force of 3.025 Newtons to the end of each arm the robot the software produced a displacement plot showing the effect of that force on the arm. The results of this analysis can be seen in Figure 20. As predicted, the force caused at least the required deflection, without damaging any part of the arm construction.

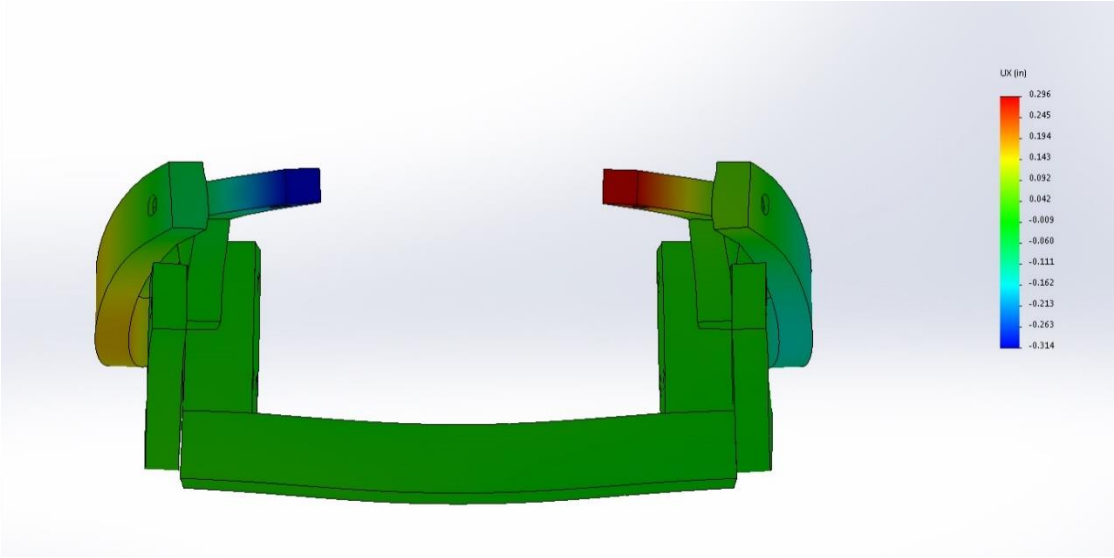

Figure 20 Displacement plot of inward deflection of the arm

As neither the downward or inward force showed a destructive potential to the arm, the analysis of the combined forces similarly showed reasonable strain and displacement on the arm, as shown in Figure 21.

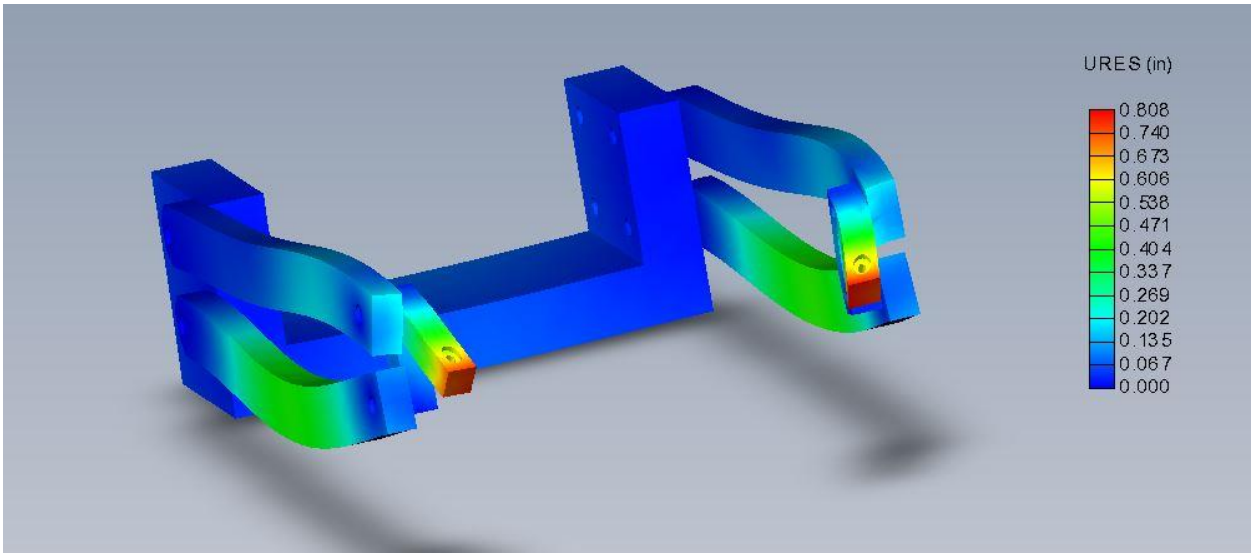

Figure 21 Combination of forces working on the arm, causing reasonable displacement

With this verification of the functionality of the arm, we were able to move forwards with construction. The base of the arm was constructed from 2 by 6 inch Oak board, and the arm itself was constructed from beams of Cedar wood, as our design called for. Using materials purchased from the Home Depot and Stock Drive Products and Sterling Instruments, the arm was constructed as shown below in Figure 22.

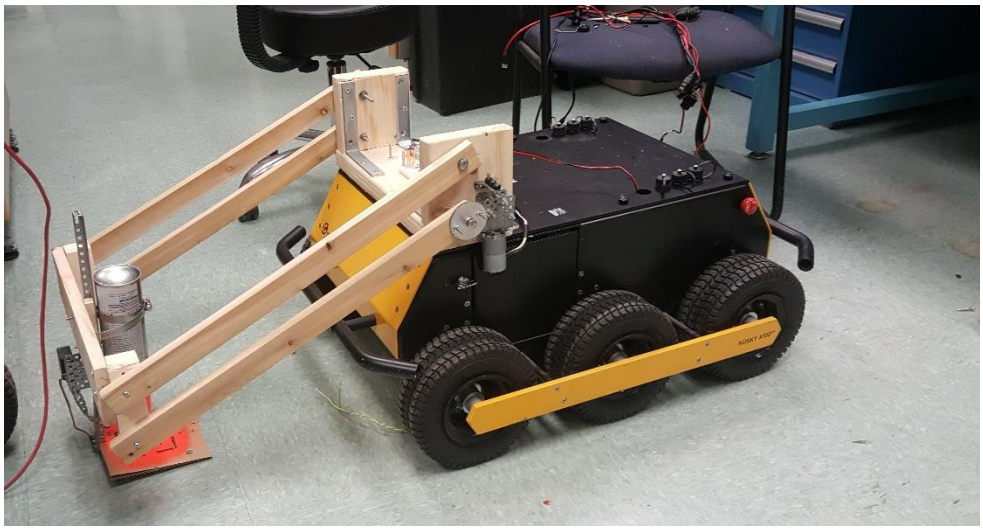

Figure 22 Final Robot complete with Arm

Control System

The control system for the arm is designed to operate with ultrasonic distance sensors on each arm. By using simple trigonometry, we know that a change of 1 degree in the slope of the arm moves the end of the arm by .42 inches as shown below.

$$24 * \sin(1°) = .42$$

Knowing this and using encoders on the each motor, the relation between one motor rotation and the distance from the ground of the end of the arm can be determined. One full rotation of the motor is reported as 16 ticks of the encoder.[27] Using the known values of the motors internal gearbox, a 1:90 reduction, and the gearbox we created, 1:39.27 the movement of the motor can be related to the movement of the end of the arm. The calculations to determine this relationship are shown here.

$$\frac{1\ motor\ rotation}{(90 * 39.27)\ gear\ reduction} = \frac{1\ motor\ rotation}{3534.3\ gear\ reduction} = \ 2.83 * 10^{-4}$$

Therefore, for every 16 counts of the encoder, the arm moves $2.83*10^{-4}$ rotations or .102 degrees. Knowing this and knowing the distance from the end of the arm to the ground as reported by an ultrasonic rangefinder, the distance the arm needs to travel could be represented in terms of rotations of the motor using the calculations shown here.

$$.41\frac{inches}{degree} * .102\frac{degrees}{rotation} = \ .042\frac{inches}{rotation}$$

Using this the Arduino code shown in the Appendix D: Arduino Arm Control Code was developed to control the height of the arm in terms of the movement of the motor. The motors were then controlled using a single Arduino board and an H bridge circuit to supply 12-volt power and allow the motors to be driven in 2 directions. This circuitry is shown here in Figure 23.

[27] "12V DC Motor 251rpm W/Encoder (SKU: FIT0186)." DF Robot., accessed 3/10/17
, https://www.dfrobot.com/wiki/index.php/12V_DC_Motor_251rpm_w/Encoder_(SKU:_FIT0186).

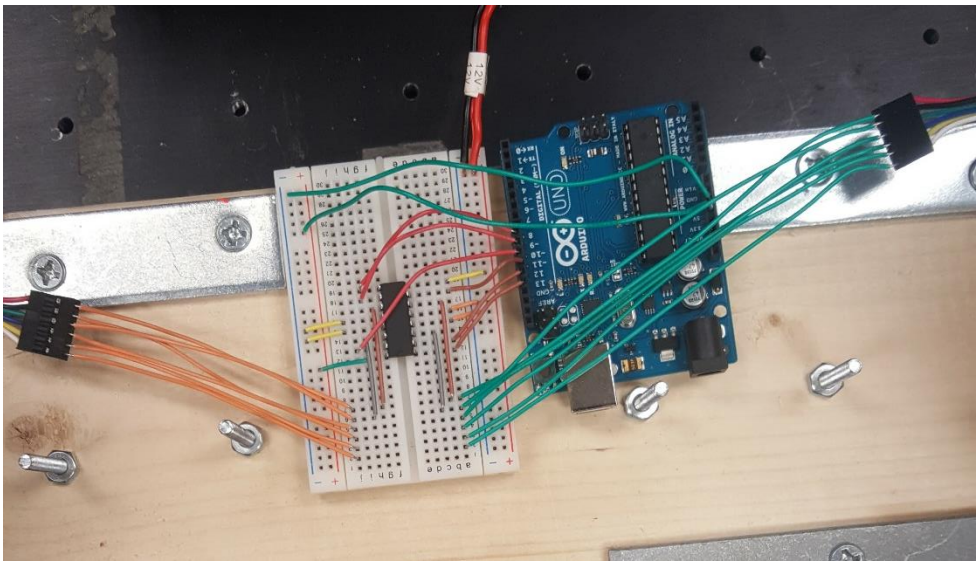

Figure 23 Control circuitry for sensor arm

Marking

Selection

The previous MQP utilized a camera running openCV in order to detect the marker denoting a landmine. As previously mentioned this system searched for a square target on the ground, although any shape could be searched for with a modified algorithm. The stipulation for our marking system is that it must create a square target on the ground, similar to the target used in the previous MQP, which was a black square on white paper, as shown in Figure 24.

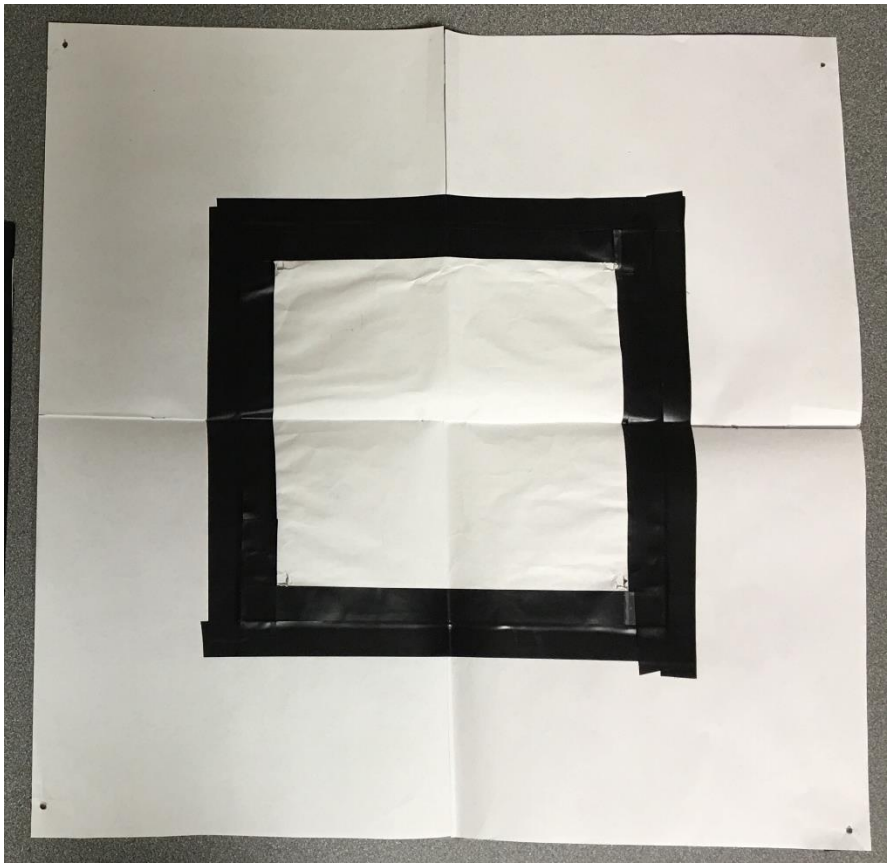

Figure 24 Marker Used by Previous MQP

While we could have used this marker, the paper presented several disadvantages. Firstly, this marker would have to be applied to the ground in such a way that it would stay put for a long period of time, but also deployed without exerting 10 newtons of force to the landmine, in order to avoid detonation. Additionally the marker should be resistant to weather, in the event that the detection by the rover and detonation by the UAV has a large delay. Because of these reasons, we chose not to use the marker system used by the previous MQP, but to design our own compatible system. After consideration of several methods of placing a marker, we found no low cost, durable marker that can be applied gently enough to the surface of the landmines to guarantee no detonation. Therefore, we decided to mark the landmines using spray paint. Spray paint creates a highly visible marker, on any type of terrain, and the mark will not be destroyed by rain, snow or wind. By spraying the paint through a mask, the shape of the mark can be controlled, and made to be a square similar to the one used in the previous project and detectable by the openCV camera system with only minor modifications.

Design

We designed our sprayer using a can of field grade spray paint to a metal bar, aiming towards the sprayer mask. In order to produce a shape detectable by the camera system, we created a mask, which would spray a large square onto the mine and an additional outer square to increase the

area the camera system can detect. The mask was carved out of cardboard, and the spray paint is controlled by a servo mounted on the arm. The sprayed mark would be within 2 inches of the center point of the mine, meaning that the center of the mark is on top of the physical mine, and assuming the payload from the drone lands in the center of the mark it is still within the area to detonate the landmine. Spray paint is readily available worldwide, and the design allows the can to be easily removed and a new can inserted. This is vital because this is the only section of the robot that requires refilling. Spray paint makes for a cheap system that can be sourced globally.

Construction

For our prototype, the spray can was mounted to the arm by a pair of hose clamps, which allows for size adjustments on the spray can. The mask was mounted 5 inches away from the end of the spray paint can in order to allow the paint to properly spread out before hitting the mask. A small arm mounted to a servo motor controlled the spraying by depressing the nozzle of the spray can. This assembly was then mounted to the end of the arm, in order to allow the robot to mark a landmine when located, and continue to avoid it. The entire construction can be seen in Figure 25.

Figure 25 Sprayer and mask system

Testing

In order to determine the best system to spray a clear shape on the ground, we had to test several criteria. The distance from the can to the sprayer mask, the distance from the mask to the ground, and the amount of time spraying. Several masks with different shapes were cut in order to test the clarity of the shape created. Initial tests showed that a small mark could be made very clearly, when the mask was within one inch of the ground and the can is mounted 5 inches away from the mask, as seen in Figure 26 .

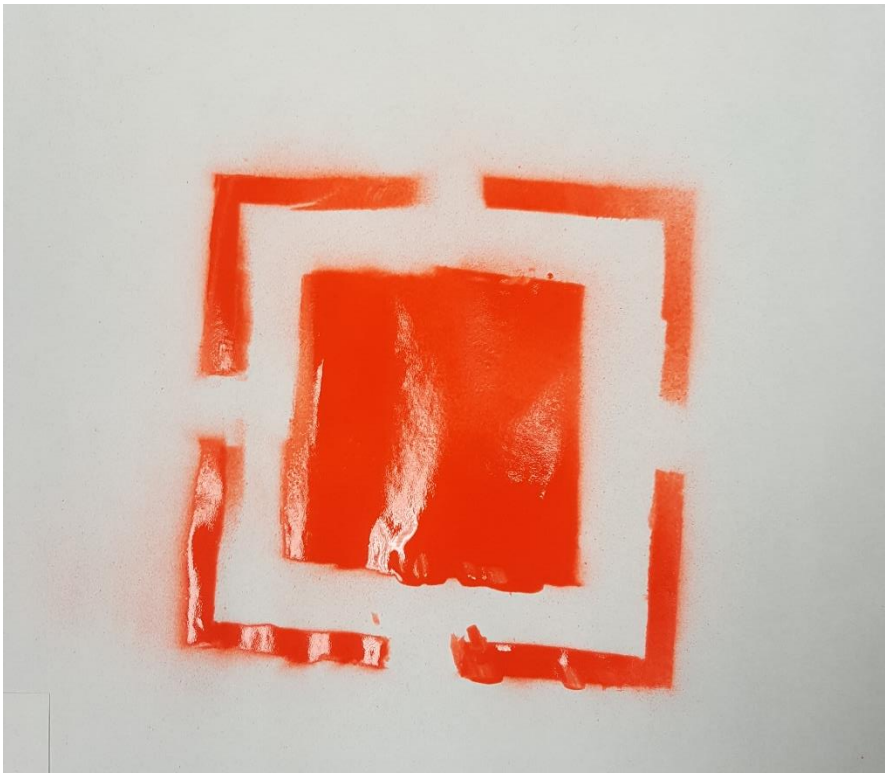

Figure 26 Mark made by our marking system

While this mark looks significantly different from the mark used by the previous project, show in Figure 24, the main functionality of the openCV system they used is to search for a certain shape, and therefore the system could be modified to search for the smaller mark. The most important quality is the clear shape with sharp edges, with contrast to the background. Our mark fulfills these necessary constraints to remain compatible with the previous MQP, and meets our constraint of applying less than 10 newtons of force to the surface of the landmine and being usable for multiple landmines.

GPS

GPS plays a critical role in the project from localizing the rover and marking the mines, to interfacing with the operator. The following sections detail the use of GPS in the project.

M8P Differential GPS system

For this project, we were given a differential GPS system made up of two u-blox C94-M8P chips with centimeter level accuracy relative to each other from Professor Putnam. This system uses a "rover" and "bases station" setup to accurately report the real location of the rover unit. This is accomplished by "fixing" the location of the base station by correcting for interference and difference between satellites, then using radio communication to the rover chip to correct its location relative to the location of the base station. Initial testing via the evaluation program can be seen in Figure 27 demonstrating the user interface and drift when not given enough time to fix its location.

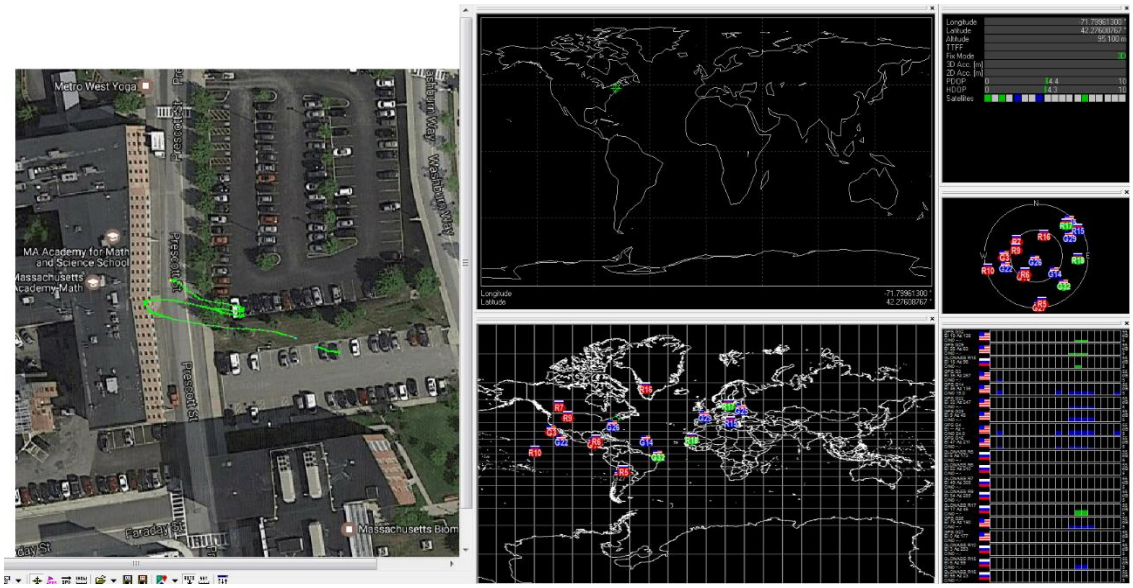

Figure 27 U-Blox Control Interface

Localization

Localization is the process by which the robot understands its position relative to the world and is necessary to navigate the rover autonomously with any accuracy over distance. Localization is often achieved by comparing a known map that is either generated or given to sensor data such as a LIDAR. The husky A100 rover uses dead reckoning for its normal localization as the rover does not have adequate sensors to generate its own map and localize based off of this data. The rover's internal sensors (imu) however are not sufficiently accurate to avoid having its perceived location differ from its actual location causing the rover to "drift" over time. Thankfully, we were given the two u-blox C94-M8P differential GPS chips that allow for centimeter accuracy. Using the u-blox ROS driver the M8P chip is able to communicate to the rover and publish the GPS coordinates on the NavSatFix ROS topic. This data is then used by the ROS navsat_transform node to enable the rover to accurately perceive its real location in the world by integrating the GPS data into the localization process. The process as shown by Figure 28

corrects for the map drift that would build up from the inaccuracy over time of the internal sensors.

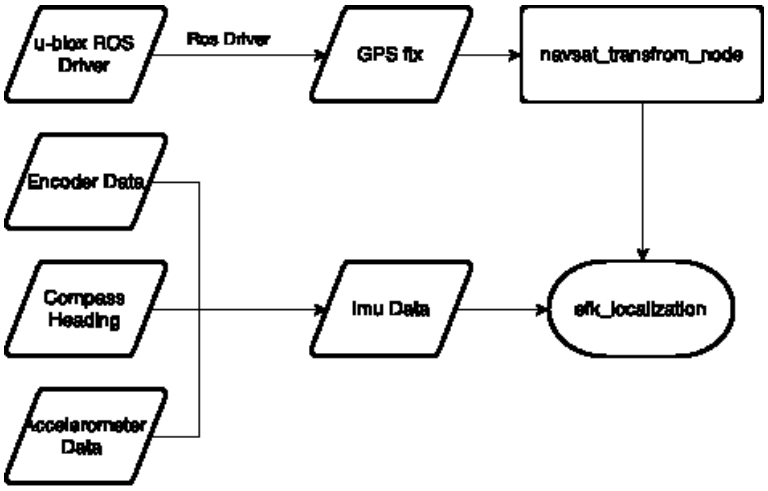

Figure 28 GPS Localization

Google Maps

This project requires the rover to know where it is supposed to be working in. In other applications the map can be defined by hand going out and recording the locations that bound the operation area. However going out and recording the boundaries is not an option in a minefield. To allow the user to define minefield we turned to web based mapping, utilizing the recent emergence of high definition maps available online to provide a convenient, free and powerful tool that only requires an internet connection. We were first directed to use Google Earth but found it had been removed due to security concerns. After searching for an alternative Google Maps was found to provide the best support for our project. It has good documentation, a strong API suit, free developer keys, and works with JS fiddle, a development platform. Neither of us had ever worked on a web application or worked with google maps so being easy to use was very important in the selection. Even still, there was considerable documentation to read up on before we could begin work on the application.

After going over this documentation we found a demo program that showed how to place pre-defined shapes. While this is great for displaying a possible minefield, it was useless unless the user could edit the web page as the shape was defined in the code. We were able to modify the program in java script using JS fiddle by listening to mouse events to allow the vertices of the polygon to move based off of drag commands and their locations to be reported by clicking in the polygon and printing all the vertices GPS location to a Google Map infoWindow by traversing the polygon. This had the unexpected benefit of the user being able to make any shape and move it around, as between every vertex, another could be added shown in Figure 29.

Figure 29 Example of marked off minefield in Google Maps program

While this could meet our requirements, it was not a good user experience. The program would start at a set location with the polygon in the middle so the user would need to drag the polygon to their desired location in the world and if they messed up the shape the only way to remove it was to reload the page and start over from the beginning. To address this we used the geo-location functionality of the navigator library to bring the user to their current location and moving the shape there but editing the shape was still a hassle. While searching for a solution we found the drawing tools library this allows the users to draw their own shape and make more than one shape at a time. Despite the significant improvement to the user experience this offered it broke the preceding code. Before we could refer to the shape explicitly now we would need to identify the shape with every request. Never the less we decided that it was worth it and rewrote the code to work in the drawing library. This required us to rewrite the event listener to use the drawing manager tool from the drawing library so that we could identify each shape. While working on the event listeners we found a demo to delete a vertex from an explicitly defined polygon and after some work we were able to get it to remove a vertex from the new user defined polygon too. This enabled the user to do everything we wanted, define any shape modify it and record its location, the flow chart of this application is shown in Figure 30 and the documentation of the final code can be found in Appendix B: Google Maps Interface Code.

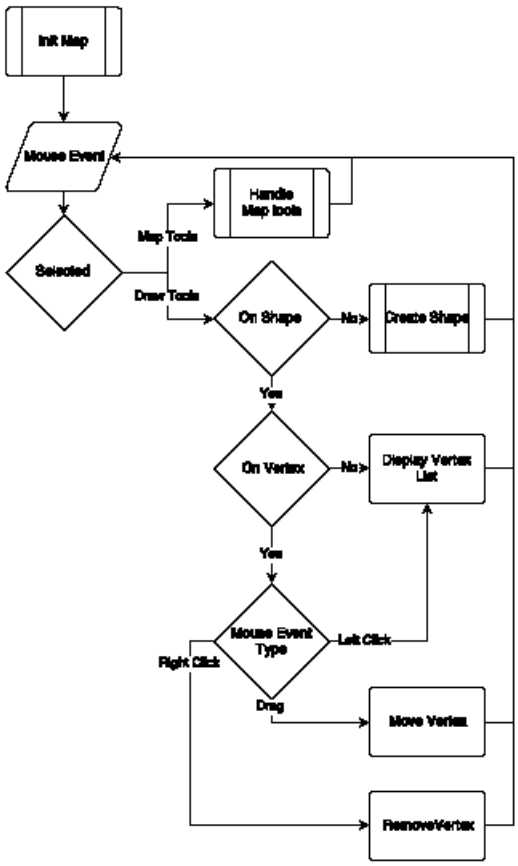

Figure 30 Google Map Flow Overview

Despite achieving the user experience we wanted we were unable to have the web application automatically download the list of vertices to be used directly or be launched on terminal start.

Mine Recording

Once the detector system on the rover reports that it has found a landmine it needs to be recorded. The previous MQP required the GPS coordinates of the target landmine within 3.5 meters for their aerial search to be effective. The detector head is within this distance requirement to the u-blox M8P GPS chip mounted on the center of the robot. The robot coordinates are given by the NavSatFix topic published by the u-blox ROS driver. So in order to record the location of the landmine we chose to use the current coordinates of the robot.

It is important for the record of the landmines to be kept out of danger. Should the record be lost the entire mission would be wasted. For this reason the program that records the landmine location is kept on the base station where the log file is stored. This ensures that even if the rover is lost the locations of the landmines will not be. The program that accomplishes mine recording is in Appendix section A: Rover System Code subsection GPS Recording.

Rover Control Interface

The majority of the rover operation is hands off but the initial set up requires user input. As discussed previously the user defines a minefield in google maps and has access to the vertex list of GPS coordinates. To make use of this the user must enter this list by copying and pasting it into a terminal interface. This interface takes care of marking out the minefield for the rover by publishing on the clicked point topic in ROS. The operation of this interface is show in Figure 31.Then the user must select a starting location for the rover in as shown in Figure 32.We initially desired to have the interface take care of this as well but as the map can have any shape and this last placement starts the rovers operation in the field we left it to the user.

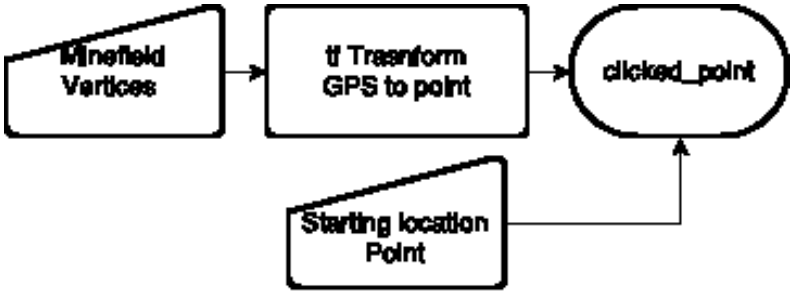

Figure 31 Mine Field Interface

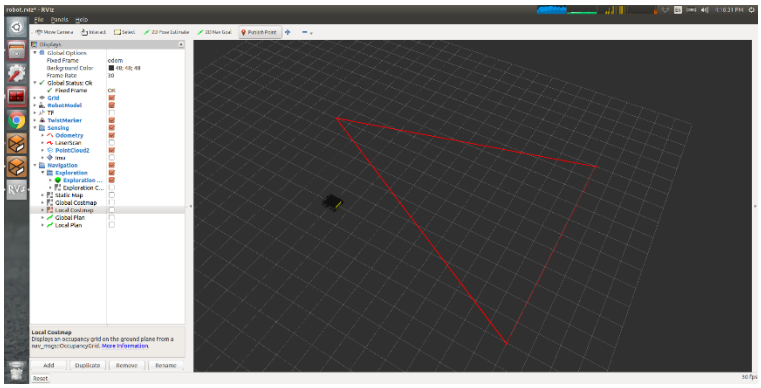

Figure 32 RVIZ waiting for starting location

Results

All of the system of the project were proven to work individually, though they were never tested all at once. The final project was able to:

1. Let the user define a minefield
2. Autonomously travers a minefield
3. Avoid detected landmines
4. Visually mark found landmines
5. Report landmine locations
6. Detect metallic content
7. Use GPS data to correct for drift

The full integration of these systems was never tested due to the driver issue for the rover never being resolved. This left the rover unable to be effectively operated with complete autonomy. Despite this the system should be able to operate given a functioning rover platform that is ROS compatible.

Marking System

Our marking system is capable of creating a highly visible mark on top of a landmine, without applying enough pressure to cause the mine to detonate. The reliability of the shape is not perfect, with environmental factors such as wind effecting the ability to mark the landmine. The system is capable of recording a GPS location of the landmine for the UAV to search. While we were unable to access the camera system used by the previous MQP for testing, our marker is compatible with the previous project requirements, and the GPS coordinates are within the search area of 3.5 meter specified by the previous project.

Detection

Our detection system is capable of reliably detecting metal at short ranges. Creating a highly effective landmine detector was a stretch goal for our project, and the metal detector we created is a prototype designed to be a proof of concept and requires further development to be reliably used to detect landmines for our rover system. In order to safely traverse the minefield we would ideally like to use a system which has a near zero-rate of false negatives, which our current sensor could not attain. Despite this, our system shows that a similar detector such as a more developed metal detector, or Ground Penetrating Radar can be used with the rover system for landmine detection and marking.

The representation of the detector as a laser scan messages worked better than expected. The system was unable to distinguish it from a normal LIDAR sensor and it was able to place obstacles that were registered by path planning.

Navigation

The navigation system of the rover while meeting the requirements of the project was lacking. Particularly in the use of the frontier_exploration package. The algorithm used to search the minefield was far from efficient and would need to re-traverse areas it had explored to reach unexplored regions. The other issue with the system is that the rover would on occasion drive directly over the marked landmine despite recognizing it as an obstacle and planning a path around it but this is most likely an issue with a configuration file for the move base.

The navsat_transform node worked as advertised and was able to correct for map drift when given an accurate GPS location for the rover. The U-blox M8P chips turned out to be more difficult to work with. The system takes a few days' worth of time to accurately fix itself to achieve its advertised centimeter level precision. This was an unacceptable time frame to for testing so the M8P was never able to give its most accurate location possible in testing.

Minefield Definition

The Google Maps interface turned out better than expected. It was almost able to meet all of our considerations being able to define any minefield the user could think of with a pleasant user experience. The only issue with the system was that the coordinates of the minefield needed to be copied over to the robot interface and were not able to be done automatically.

Conclusion

Landmine contamination remains a major problem worldwide. Thousands of injuries and death result from landmines worldwide, despite the fact that landmines have been banned in warfare for 20 years. Landmines are not a problem that will disappear anytime soon, and current methods of clearing minefields are too expensive, or unacceptably risky. Mine clearance is exhausting, dangerous, repetitive work, poorly suited for humans, but perfect for a robotic solution.

The purpose of this MQP was to find and mark a landmine for detonation by the UAV demining project. Our rover system can autonomously scan a given minefield, support any compatible landmine detection suite, and to report to the base station when a landmine has been found. Upon finding a landmine, the GPS location is recorded and the location is marked with a highly visible marker. Meeting our requirements that the UAV from the previous project could operate successfully, working in concert with our rover. Our system shows that this method of landmine detection and marking is viable, and our project created a base for further development of this system. This project shows that a robotic solution to the problem of landmine detection and removal is a practical alternative to current solutions.

Future Work

This project can be built upon further in several ways. First the sensor suite can be expanded upon massively. A metal detector is capable of detecting landmines, but general has a high rate of false positives, and newer minimal metal landmines are undetectable by most metal detectors. There is an opportunity to build a custom landmine detector using GPR technology or infrared systems to identify what is a landmine and what might be other remnants of war scattered throughout the minefield. Either technology has the benefit of being able to detect nonmetallic objects, but require greater data analysis to translate into a landmine detector. Another path would be for the rover to be overhauled. The Clearpath Husky was a good base for our prototyping, but it has some unnecessary features adding to its cost. In order to minimize the price of the rover a custom-built system could be used with minimal features and a lighter structure. The system needs to remain all terrain, and a certain amount of onboard computing is necessary, but a simpler, cheaper system could be used just as successfully as the Husky. Additionally the UGV system could be changed to a UAV system with detections technologies such as infrared mounted to a UAV, and research could be done into using an aerial landmine marking system. An interesting opportunity that our project did not look into was for thee system to be entered into landmine detection and disposal competitions around the world, such as the "Minesweepers: Towards a Landmine-Free Egypt."[28]

[28]Khamis, Dr Alaa. 2013. Minesweepers: Towards a Landmine-Free Egypt, an Outdoor Humanitarian Demining Robotic Competition." The Journal of ERW and Mine Action. http://www.jmu.edu/cisr/journal/17.1/pdfs/Khamis.pdf.

References

"12V DC Motor 251rpm W/Encoder (SKU: FIT0186)." DF Robot., accessed 3/10/17, , https://www.dfrobot.com/wiki/index.php/12V_DC_Motor_251rpm_w/Encoder_(SKU:_FIT0186).

C94-M8P
U-Blox RTK Application Board Package
User Guide 2016. ublox.

Contamination by Anti Personnel Mines 1999-2025 2014a. The International Campaign to Ban Landmines.

"Hero Rats Sniff (and Snuff) Out Landmines and TB." 2014b.*CNN Wire*, Sep 26,.

"Landmine and Cluster Munition Monitor." the-monitor.org., accessed 1/13/17, , http://the-monitor.org/en-gb/reports/2016/landmine-monitor-2016/casualties-and-victim-assistance.aspx#ftn2.

"Mines." Weaponry in the Civil War., accessed 3/25/17, , https://sites.google.com/site/weaponryinthecivilwar/mines.

Munitions Reference Guide. d. James Madison University.

Ottawa Landmine Treaty 2006. Vol. 2.

"Soviet / Russia PMN-1 Bakelite Landmine." BuyMilSurp.com., accessed 11/15/16, , https://www.buymilsurp.com/soviet-russia-pmn1-bakelite-landmine-p-5098.html.

"Treaty Status." The International Campaign to Ban Landmines., accessed 2/15/17, , http://www.icbl.org/en-gb/the-treaty/treaty-status.aspx#.

"U.S. Policy regarding Landmines." 2008.*The American Journal of International Law* 102 (1): 190-191. http://www.jstor.org/stable/40007795.

"US Landmine Policy." US Department of State., last modified Sep 1, accessed 3/15/17, , https://www.state.gov/t/pm/wra/c11735.htm.

"Wood Strengths." WoodWorkWeb.com., http://www.woodworkweb.com/woodwork-topics/wood/146-wood-strengths.html.

A Major Qualifying Project. *De-Mining with UAVs*.

Bruschini, Claudio and Bertrand Gros. 1998. "A Survey of Research on Sensor Technology for Landmine Detection." . http://www.jmu.edu/cisr/journal/2.1/bruschini.htm.

Hirsch, Jesse. 2014. "The Dogs that Sniff Out Landmines." *Modern Farmer*, Jun 11,.

Khamis, Dr Alaa. 2013. "Minesweepers: Towards a Landmine-Free Egypt, an Outdoor Humanitarian Demining Robotic Competition." *The Journal of ERW and Mine Action*. http://www.jmu.edu/cisr/journal/17.1/pdfs/Khamis.pdf.

"ROS.org." *Robot Operating System*. wiki.ros.org.

Schneck, William C. 1998. "The Origins of Military Mines: Part 1." *Engineer Bullentin*. https://fas.org/man/dod-101/sys/land/docs/980700-schneck.htm.

Smith, Andy. 2014. "Understanding the use of Prodders in Mine Detection." *The Journal of ERA and Mine Action* (18.1). http://www.jmu.edu/cisr/journal/18.1/notes/smith.shtml.

Sullivan, Michael. 2015. "In Cambodia, Rats are being Trained to Sniff Out Land Mines and Save Lives." *Morning Addition*, July 31,. http://www.npr.org/sections/parallels/2015/07/31/427112786/in-cambodia-rats-are-being-trained-to-sniff-out-land-mines-and-save-lives.

Takahashi, Kazunori and Dieter Gulle. "ITEP Evaluation of Metal Detectors and Dual-Sensor Detectors." *The Journal of ERW and Mine Action*. https://www.jmu.edu/cisr/journal/14.3/r_d/takahashi/takahashi.shtml.

Takahashi, Kazunori, Holger Preetz, and Jan Igel. 2011. "Soil Properties and Performance of Landmine Detection by Metal Detector and Ground-Penetrating Radar — Soil Characterisation and its Verification by a Field Test." *Journal of Applied Geophysics* 73 (4): 368-377. doi:10.1016/j.jappgeo.2011.02.008. http://www.sciencedirect.com/science/article/pii/S0926985111000450.

Tyson, Peter. 2012. "Dogs' Dazzling Sense of Smell." *Inquiry: An Occasional Column*, Oct, 4. http://www.pbs.org/wgbh/nova/nature/dogs-sense-of-smell.html.

United States. War Dept. 1971. *Handbook on German Military Forces*. US Army Manual. US Government. http://hdl.handle.net/2027/ien.35556026179069.

Yu, Jiao and Liu Ji. *Huolongjing (Fire Dragon Manual)*.

Appendix

A: Rover System Code

Clicked Point Boundary

```
#!/usr/bin/env python
#Clicked point demonstration implementation to create boundary from vertex
#bdcasey@wpi.edu
import rospy
from std_msgs.msg import String
from geometry_msgs.msg import PointStamped, Point

##create a PointStamped message
def create_point_stamped(point, frame_id = 'global'):
    p = PointStamped()
    p.header.frame_id = frame_id
    p.header.stamp = rospy.Time()
    p.header.stamp = rospy.get_rostime()
    p.point = point
    return p
# create a point
def create_point(x,y,z = 0):
    p = Point()
    p.x = x
    p.y = y
    p.z = z
    return p

def talker():
    pub = rospy.Publisher('clicked_point', PointStamped, queue_size=10)
    rospy.init_node('talker', anonymous=True)
    rate = rospy.Rate(1) # 1hz
    x = 0
    while not rospy.is_shutdown():
            x += 1
        p_point = create_point_stamped(create_point(x,0,0),'map')
#! "hello world %s" % rospy.get_time()
        rospy.loginfo(p_point)
        pub.publish(p_point)
        rate.sleep()

if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException:
        pass
```

Fake laser scan

```
#!/usr/bin/env python
# fake laser scan sensor to publish scan data to be used by
frontier_exploration via gmapping to explore the map
# fake laser scan to publish obstacles found by the detector head
# bdcasey@wpi.edu
# TODO will need a tf topic to do transformation despite being identical to
the base link in this application
import math
```

```
import rospy
import numpy
from std_msgs.msg import String
from std_msgs.msg import Header
from sensor_msgs.msg import LaserScan
#create a laserscan that represents the sensor head (should extend a bit
short the sensor to ensure complete coverage during exploration) Must be in
numpy.float32 format to work!

def create_laser_scan
(angle_min,angle_max,angle_increment,time_increment,scan_time,range_max,range
s,frame_id = 'base_link'):
      scan = LaserScan()
      scan.header.frame_id = frame_id
      scan.header.stamp = rospy.Time()
      scan.header.stamp = rospy.get_rostime()
      scan.angle_min = angle_min
      scan.angle_max = angle_max
      scan.angle_increment = angle_increment
      scan.time_increment = time_increment
      scan.scan_time = scan_time
      scan.range_min = numpy.float32(0)#0m min range
      scan.range_max = range_max
      scan.ranges = ranges
      scan.intensities = numpy.array([],dtype=numpy.float32) #empty
      return scan
#TODO
#Changes snsor readng baseedon locatin of mine
def detectorListener():
      return False

def talker1():
      pub = rospy.Publisher('scan', LaserScan, queue_size=20)
      rospy.init_node('talker1', anonymous=True)

      #define the mine sensor relitive to the base link, assumes symetric
sensor from center of robot
      # Units: meters and radians
      width_half = 0.3048 #2ft across
      distance_center= 0.762 # extends 2ft from the front and mounted 11in
infront of center == 35 in, set to 30 in
      scan_rate = 50 # 50hz use for time between scans and publishing
      rate = rospy.Rate(scan_rate)
      #end definition
      #define the scan definition of the sensor
here==================================
      # Units: meters and radians
      angle = math.atan(width_half/distance_center)
      angle_min = numpy.float32(0-angle)
      angle_max = numpy.float32(angle)
      time_increment = numpy.float32(0)
      scan_time = numpy.float32(0) #1/scan_rate) #change to Hz of publisher,
optional
      range_max = numpy.float32(distance_center) #in m should be less than
the distance to the sensor for full coverage
```

```
	count = math.floor(width_half * 200) #1 count per cm
	# angular distance between measurements [rad]
	angle_increment = (abs(angle_max) + abs(angle_min))/count #! not tested
	#depreciated #count = (angle / math.pi) * 360 #2 counts per degree for
resolution
	#end definition ================================================
	noWall = []
	value = numpy.float32(range_max-0.001) #offset so that the data is used
0.761
	for i in range(numpy.int(count)+1):
			noWall.append(value)
	wall = []
	value = numpy.float32(0+.5) #offset so that the data is used 0.5
	for i in range(numpy.int(count)+1):
			wall.append(value)

	mine = False
	while not rospy.is_shutdown():
		if mine: #swich on comand
			ranges = wall
		else:
			ranges = noWall

		scan =
create_laser_scan(angle_min,angle_max,angle_increment,time_increment,scan_tim
e,range_max,ranges,'base_link')
		#hello_str = "ima fireing my lazor %s" % rospy.get_time()
		#rospy.loginfo(hello_str)
		pub.publish(scan)
		mine = detectorListener()
		rate.sleep()

if __name__ == '__main__':
	try:
		talker1()
	except rospy.ROSInterruptException:
		pass
```

Cost map sensor

```
#!/usr/bin/env python
#cost map implementation of fake sensor
#bdcasey@wpi.edu
import rospy
import numpy
import tf
from std_msgs.msg import String
from nav_msgs.msg import OccupancyGrid,MapMetaData
from geometry_msgs.msg import Pose,Point,Quaternion

#create a OccupancyGrid message with header
def create_occupancy_grid(data,mapmetadata,frame_id = 'map'):
	og = OccupancyGrid()
	og.header.frame_id = frame_id
	og.header.stamp = rospy.Time()
```

```
      og.header.stamp = rospy.get_rostime()
      og.info = mapmetadata
      og.data = data
      return og
##MapMetaData begin--------------------
#create a MapMetaData
Message(resolution=[m/cell],width=cells,height=cells,origin of the map [m, m,
rad]); default assumes 3 sensors: resolution = 0.08128, width = 5, height = 1
def create_mapmetadata(resolution, width, height,pose):
      mm = MapMetaData()
      mm.map_load_time = rospy.Time()
      mm.map_load_time = rospy.get_rostime()
      mm.resolution = resolution
      mm.width = width
      mm.height = height
      mm.origin = pose
      return mm
#create Pose Message This is the real-world pose of the cell (0,0) in the
map.
def create_pose(point,quaternion):
      p = Pose()
      p.position = point
      p.orientation = quaternion
      return p
#create a point
def create_point(x,y,z = 0):
    p = Point()
    p.x = x
    p.y = y
    p.z = z
    return p
#create a Quaternion
def create_quaternion(x,y,z,w):
      q = Quaternion()
      q.x = x
      q.y = y
      q.z = z
      q.w = w
      return q
##MapMetaData end ----------------------------
#create int8[] data to define OccupancyGrid
#where along the sensor is the mine set as 100
#bad implementation (need to edit with any changes)
def create_data(a = 0,b = 0,c = 0,d = 0,e = 0):
      d = numpy.array([a,b,c,d,e],dtype=numpy.int8)
      return d

def talker():
      pub = rospy.Publisher('map2', OccupancyGrid, queue_size=10)
      rospy.init_node('talker', anonymous=True)
      rate = rospy.Rate(10) # 10hz

      #define the grid definition of the sensor
here==================================
      data = create_data(100)
```

```
	point = create_point(0.5969,0.2032,0)#23.5in in front of rover, 8in
offset
	ql = tf.transformations.quaternion_from_euler(0, 0, -1.5708) #in
radians
	quaternion = create_quaternion(ql[0],ql[1],ql[2],ql[3])#should use the
tf transform euler to quat (no idea what i am doing yet 4/8/2017)
	pose = create_pose(point,quaternion)
	mapmetadata = create_mapmetadata(0.08128, 5, 1,pose)
	grid = create_occupancy_grid(data,mapmetadata,'base_link')
	#end definition ================================================

	while not rospy.is_shutdown():
		hello_str = "hello grid %s" % rospy.get_time()
		rospy.loginfo(hello_str)
		pub.publish(grid)
		rate.sleep()

if __name__ == '__main__':
	try:
		talker()
	except rospy.ROSInterruptException:
		pass
```

GPS Recording

```
#!/usr/bin/env python
# Listen to detector and log GPS location fron nav fix mesage
# bdcasey@wpi.edu
import rospy
from std_msgs.msg import String
from sensor_msgs.msg import NavSatFix

lastLandMineID = -1
#print and write the GPS lat and long with the  Id# to record the landmine
pos
def callback(GPSdata,landMineID):
	global lastLandMineID
	# make sure there is only 1 log for the landmine
	if landMineID != lastLandMineID:
		lastLandMineID = landMineID
		rospy.loginfo("Id %i latitude %f longitude %f", landMineID,
GPSdata.latitude, GPSdata.longitude)
		txt =
str(landMineID)+","+str(GPSdata.latitude)+","+str(GPSdata.longitude)+"\n"
		target = open("GPS_Landmine_Log", 'a')
		target.write(txt)
		target.close()

def listener():
	landMineID = 1
	rospy.init_node('listener', anonymous=True)
	rospy.Subscriber("/navsat/fix", NavSatFix, callback,(landMineID))
	rospy.spin()

if __name__ == '__main__':
```

```
    try:
        listener()
    except rospy.ROSInterruptException:
        pass
```

B: Google Maps Interface Code

Http

```
<div id="map"></div>
<!-- Replace the value of the key parameter with your own API key. -->
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCkUOdZ5y7hMm0yrcCQoCvLwzdM6M8s5qk&libraries=drawing&callback=initMap" async defer></script>
```

JavaScript

```
// This requires the Drawing library. Include the libraries=drawing
// parameter when you first load the API. For example:
// <script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&libraries=drawing">
// bdcasey@wpi.edu
var map;
var infoWindow;

function initMap() {
  map = new google.maps.Map(document.getElementById('map'), {
    center: {
      lat: -34.397,
      lng: 150.644
    },
    zoom: 15,
    mapTypeId: 'terrain'
  });
    // Try HTML5 geolocation.
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function(position) {
      var pos = {
        lat: position.coords.latitude,
        lng: position.coords.longitude
      };
      map.setCenter(pos);
    });
  } else {
    // Browser doesn't support Geolocation
  }

  var drawingManager = new google.maps.drawing.DrawingManager({
    drawingMode: null,
    drawingControl: true,
    drawingControlOptions: {
      position: google.maps.ControlPosition.TOP_CENTER,
      drawingModes: ['polygon']
    },
```

```
    polygonOptions: {
      fillColor: '#ffff00',
      fillOpacity: 0.3,
      strokeWeight: 5,
      clickable: true,
      editable: true,
      zIndex: 1
    }
  });

  drawingManager.setMap(map);

  // this assumes `my_poly` is an normal google.maps.Polygon or Polyline

  google.maps.event.addListener(drawingManager, 'polygoncomplete',
function(polygon) {
    polygon.addListener('click', showArrays);
    polygon.addListener('rightclick', deleteNode);
  });

  infoWindow = new google.maps.InfoWindow;
}

function deleteNode(event) {
  if (event.vertex == undefined) {
    //this.setMap(null);
    return;
  }
  var path = this.getPath();
  path.removeAt(event.vertex);
  this.setPath(path);
};

/** @this {google.maps.Polygon} */
function showArrays(event) {

  // Since this polygon has only one path, we can call getPath() to return
the
  // MVCArray of LatLngs.
  var vertices = this.getPath();

  var contentString;
  if(vertices.getLength() == 1){
  contentString =
    '<b>Base Station</b><br>'
  }else{
  contentString =
    '<b>MineField</b><br>'
  };

  // Iterate over the vertices.
  for (var i = 0; i < vertices.getLength(); i++) {
    var xy = vertices.getAt(i);
    contentString += '<br>' + 'Coordinate ' + i + ':<br>' + xy.lat() + ',' +
      xy.lng();
```

```
  }

  // Replace the info window's content and position.
  infoWindow.setContent(contentString);
  infoWindow.setPosition(event.latLng);

  infoWindow.open(map);
}
```

CSS

```
/* Always set the map height explicitly to define the size of the div
 * element that contains the map. */

#map {
  height: 100%;
}


/* Optional: Makes the sample page fill the window. */

html,
body {
  height: 100%;
  margin: 0;
  padding: 0;
}
```

C: Arduino Detection and Marking Code

```
//Trevor Rocks
//tarocks@wpi.edu
//code for the control of the metal detector
//and for spraying spray paint on the landmine
#include <Servo.h>
#include <ros.h>
#include <std_msgs/String.h>

Servo servo;

ros::NodeHandle  nh;
//set up ros publisher
std_msgs::String str_msg;
ros::Publisher mineMessage("mineMessage", &str_msg);
//message to be sent by ros
char mine[4] = "mine";

double pulse;// meassure of the metal detector pulse
bool isMine;//boolean for denoting a landmine

void setup() {
  // put your setup code here, to run once:
  servo.attach(3);
  servo.write(0);
```

```
  delay(500);
  //set up for ros service message
  nh.initNode();
  nh.advertise(mine);

  pinMode(13, OUTPUT); //power for induction coil
  pinMode(11, INPUT); //reads wave changes to detect metal

  Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
  isMine = detect();
  if (isMine = true) {
    void mine();
  } else {
    delay(250);
  }
}
void mine()
{
  str_msg.data = mine;
  mineMessage.publish( &str_msg );
  nh.spinOnce();
  delay(1000);
}
bool detect() {
  digitalWrite(13, HIGH);
  delay(5);
  digitalWrite(13, LOW);
  delayMicroseconds(100);
  pulse = pulseIn(11, HIGH, 5000);
  if (pulse > 920) { //920 can be changed to calibrate detector
    return true;
  } else {
    return false;
  }
}

void spray() {
  servo.write(10);
  delay(1500);
  servo.write(1);
  delay(250);
}
```

D: Arduino Arm Control Code

```
//Trevor Rocks
//tarocks@wpi.edu
//code to control the arm of the demining MQP rover
//control system using ultrasonic sensors and encoders
#include "math.h"
```

```
int encoder0PinA = 2;
int encoder0PinB = 4;
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int n = LOW;

int armLength = 24;//inches length of robot arm
int pingPin = 5;//pin for distance sensor
//using sn754410NE h bridge
int Lmotor1Pin = 11;//motor pins for H bridge left
int Lmotor2Pin = 12;
int LenablePin = 13;

//motor pins for h bridge right
int Rmotor1Pin = 10;
int Rmotor2Pin = 9;
int RenablePin = 8;

//ideal distance of sensor to ground
int ideal = 8;


void setup() {

  // set up for left motor
  pinMode(Lmotor1Pin, OUTPUT);
  pinMode(Lmotor2Pin, OUTPUT);
  pinMode(LenablePin, OUTPUT);
  //setup for right motor
  pinMode(Rmotor1Pin, OUTPUT);
  pinMode(Rmotor2Pin, OUTPUT);
  pinMode(RenablePin, OUTPUT);

  pinMode (encoder0PinA, INPUT);
  pinMode (encoder0PinB, INPUT);
  // put your setup code here, to run once:

  Serial.begin(9600);
}

void loop() {
  int  Lerror = distanceFromIdeal(ideal, 1);
  int  Rerror = distanceFromIdeal(ideal, 2);
  if (abs(Lerror) > .5) {
    moveToDist(Lerror, 1);
  } else if (abs(Rerror) > .5) {
    moveToDist(Rerror, 2);
  }

}
double distanceFromIdeal(int ideal, int side) {
  //int side determines which arms sensor
  int  error = ideal - range();
  return error;
}
```

```
void moveToDist(int distance, int motor) {
  int goal;
  goal = determineRotation(distance);
  if (goal > 0) {
    toEncoderGoal(goal, 1, motor);
  }
  else if (goal < 0) {
    toEncoderGoal(goal, 2, motor);
  } else if (goal = 0) {
  }

}

//determine how many encoder ticks will move the end of the arm to desired
height from ground
//height input from ultrasonic sensors
int determineRotation(double distance) {
  //inverse sin of distance/armLength
  //opposite over hypotenus
  double theta = asin(distance / armLength);
  //convert theta to rotation of gears at base of arm
  //theta/360, convert from degree to rotations
  // *(72/22)  22:72 gear ration
  // * 12 1:12 worm gearing
  // * 90 motor internal gearbox
  // 1/360 * (72/22) * 12 * 90 = 9 rotations per degree
  // 16 counts per rotations from encoder data sheet
  // 9.81 * 16 = 157.0909

  int ticks = theta * 157.0909;
  ticks = round(ticks);
  return ticks;
}

void runLeftMotor(int dir) { //start the motor input 0 for stop, 1 for
forward, 2 backward
  switch (dir) {
    case 1:
      //do something when var equals 1
      digitalWrite(Lmotor1Pin, LOW);   // set leg 1 of the H-bridge low
      digitalWrite(Lmotor2Pin, HIGH);  // set leg 2 of the H-bridge high
      digitalWrite(LenablePin, HIGH);
      break;
    case 2:
      //do something when var equals 2
      digitalWrite(Lmotor1Pin, HIGH);   // set leg 1 of the H-bridge low
      digitalWrite(Lmotor2Pin, LOW);  // set leg 2 of the H-bridge high
      digitalWrite(LenablePin, HIGH);
      break;
    case 3:
      digitalWrite(LenablePin, LOW);
      break;
    default:
```

```
      digitalWrite(LenablePin, LOW);
      break;
  }
}

void runRightMotor(int dir) { //start the motor input 0 for stop, 1 for
forward, 2 backward
  switch (dir) {
    case 1:
      //do something when var equals 1
      digitalWrite(Rmotor1Pin, LOW);   // set leg 1 of the H-bridge low
      digitalWrite(Rmotor2Pin, HIGH);  // set leg 2 of the H-bridge high
      digitalWrite(RenablePin, HIGH);
      break;
    case 2:
      //do something when var equals 2
      digitalWrite(Rmotor1Pin, HIGH);   // set leg 1 of the H-bridge low
      digitalWrite(Rmotor2Pin, LOW);  // set leg 2 of the H-bridge high
      digitalWrite(RenablePin, HIGH);
      break;
    case 3:
      digitalWrite(RenablePin, LOW);
      break;
    default:
      digitalWrite(RenablePin, LOW);
      break;
  }
}

double range() {
  // variables for duration and distance in inches
  long duration, inches;

  // The PING is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is u-sed to read the signal from the PING))): a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  //speed of sound 1130 feet per second
  // 74.64 microseconds per inch roughly 75 (google)
  // travels to target and back therefore/2
  inches = duration / 75 / 2;
  return inches;
}
```

```
//goal number of counts, 16 per rotation
void toEncoderGoal(int goal, int dir, int motor) {
  if (motor = 1) {
    runLeftMotor(dir);
  } else if (motor = 2) {
    runRightMotor(dir);
  }
  n = digitalRead(encoder0PinA);
  if ((encoder0PinALast == LOW) && (n == HIGH)) {
    if (digitalRead(encoder0PinB) == LOW) {
      encoder0Pos--;
    } else {
      encoder0Pos++;
    }
    if (goal >= abs(encoder0Pos)) {
      if (motor = 1) {
        runLeftMotor(3);
      } else if (motor = 2) {
        runRightMotor(3);
      }
    }

    Serial.print (encoder0Pos);
    Serial.println ("/");
  }
  encoder0PinALast = n;
}
```