Dig Safe Autonomous Cable Detection Robot

A Major Qualifying Project
submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfilment of the requirements for the
degree of Bachelor of Science

Submitted by:
Brigid Auclair – RBE
Casey Gosselin – RBE, ME
Amber Lindberg – RBE
Kyle Mitchell – RBE, ME

April 28, 2022

Report Submitted to:

Greg Lewin
Assistant Teaching Professor – Robotics Engineering

Jing Xiao
Professor and Department Head – Robotics Engineering

Worcester Polytechnic Institute

Dig Safe Autonomous Cable Detection System

Brigid Auclair (RBE), Casey Gosselin (RBE, ME), Amber Lindberg (RBE), Kyle Mitchell (RBE, ME)

4/28/2022

***Abstract*—Dig Safe regulations require utility companies to mark the location of buried utilities prior to the start of a construction project. Detecting and marking cables is time consuming, monotonous, and can be dangerous to utility workers due to the working environment. The goal of this project is to develop a semi-autonomous robotic system that detects, and marks buried electrical cables safely and efficiently. The robot manipulates an industry-standard utility locator to find buried electrical cables. Standard utility markings are spray painted on the ground above these electrical lines using a two degree of freedom arm. GPS and LiDAR sensors are used to localize the robot within its environment. The robotic prototype is capable of detecting and following a cable in a straight line, marking the cable using spray paint, and navigating a controlled environment containing static obstacles such as curbs and streetlights.**

I. Introduction

The goal of this project is to develop a prototype robotic platform that will efficiently and accurately detect, follow, and mark the location of buried electrical cables to assist utility companies. Utility companies can partner with Dig Safe, an organization that helps manage the marking of underground utilities before construction projects begin. As a partner with Dig Safe, technicians must process requests to locate and mark underground cables for customers within 72 hours of the request being made. Technicians currently drive to the site, consult resources that show the location of the cables, verify the location of the cables with a hand-held cable locator, make notes if the cable locations deviate from the records, and mark the locations of buried cables with marking paint.

Having a robot complete the cable marking tasks that technicians do has several benefits. Locating and marking cables can be dangerous for the technicians. In April 2020, an Eversource worker was hit by a car as they were repairing utilities in the middle of the road [5]. In August 2020, a utility worker in North Carolina was killed after being hit by a car while marking underground utilities, despite wearing a reflective shirt [6]. Robots and automated systems can improve safety for utility workers when working in high-risk areas by allowing workers to stay out of harm's way while the robot completes the cable marking process. Had a robot been performing the utility work, both workers could have stayed out of harm’s way and would not have been hit.

Cable marking can be mundane and repetitive, which can lead to human error. In 2021, 26% of all Dig Safe violations in Massachusetts “concerned a utility company’s mismark or failure to mark an excavation site” [7]. In 2017, mismarking errors accounted for 48% of all filed Dig Safe violation reports. Human error can result in the mismarking of a cable, putting the people digging at risk of damaging equipment and getting injured. In 2015, The Common Ground Alliance reported that over the last two decades, 421 people have died and another 1,906 people have been injured from striking underground pipes, wires, or cables [2]. In addition, accidents caused by unsafe digging resulted in over $1.7 billion in property damage [2]. These damages present a clear need for accurate and efficient marking of buried cables. Having a robot detect and mark cables can reduce the effects of human error and increase safety for utility workers. A robot can enhance the quality of work in cable detection by repeating the same task continuously without getting distracted or fatigued, thus reducing a utility company’s error, and keeping people safe while digging.

This project is a continuation of previous work at WPI. Last year, an MQP team began to create an autonomous robot to increase the efficiency of the cable marking process and a technical manual of how their system worked. They ran into various challenges and this year’s team has learned from the shortcomings of their design. Last year’s team used a Clearpath Husky A100 chassis as the base of their robotic system. Testing revealed that this chassis was unable to complete crucial tasks such as traverse over a curb or handle off-road terrain. The Husky had also reached its product end-of-life, making software support nearly impossible. Since the chassis problems made it difficult for the team last year to test other functionalities of the robot, they suggested replacing the chassis. Last year’s MQP team also developed code for implementing a LiDAR sensor for navigation, a filtering circuit for cable detection signal processing, and a two-degree of freedom (2 DoF) arm for marking buried cables. The robot is also integrated with GIS (Geographic Information System) software to update databases referencing cable locations. Their final system is shown in Figure 1.

Figure 1: The previous MQP team's robot

The goal of this project is to create a prototype robotic platform that will autonomously detect, follow, and mark the location of buried electrical cables according to Dig Safe standards. This robot will help utility companies transition from a manual marking process to an automated one. The following list of functionalities was created to describe the robot's operation.

- *Locomotion*: Traverse through both smooth and off-road terrain and travel up and down curbs
- *Detection*: Locate and follow a buried electrical cable
- *Marking*: Mark a cable with the appropriate markings in accordance with Dig Safe standards
- *Navigation*: Autonomously and safely navigate through an unknown and dynamic environment

By the end of the project, the robot will achieve the tasks listed above to demonstrate a working prototype for technicians.

II. Background

A. *Dig Safe*

Dig Safe is a non-profit organization that notifies participating utility companies of a person's or organization's dig plans [1]. State law requires that people and organizations contact Dig Safe before starting on projects that involve digging to reduce the risk of striking buried utilities. When a person wants to dig, they must first mark out their plans with white spray paint or white flags. Once the customer has clearly marked out their dig plans, they must contact Dig Safe at least 72 hours before they plan to start digging. Dig Safe then contacts utility companies to mark off their utility lines. Dig Safe is the middleman between the customer and each utility company. Once all nearby lines have been marked by the utility company, the customer is free to proceed with their digging project.

B. *Automation in Utility Industry*

Automation in the utility industry is a growing field as robots become more widespread and cheaper to manufacture. For work that can be repetitive or located in hazardous areas, robots can help to automate these processes and improve safety. For example, Ameren Corporation, a utility company in Missouri, recently bought a robotic platform to simplify and improve safety for their boiler inspection process [3]. This robotic platform collects visual and sensor data to assess the state of the boiler. It can operate in confined spaces that would be difficult and dangerous for a human to navigate. Ameren Corporation estimates that the robotic system reduced the time required to complete the inspection by about sixteen hours. Using a robot in these inspections was more efficient and safer than when they were completed by a human. Another example of robots in the utility industry involves a robot at a nuclear power plant. GE Hitachi Nuclear Energy uses an ultrasonic robot to complete inspections of a buried pipe and provides live data to a control station [4]. This robot is a much cheaper and efficient solution for inspecting the pipes compared to other methods of inspection because the use of the robot did not require any modification to the piping system. The inspection took a total of eight hours. Manual inspection processes could cost hundreds of thousands of dollars for excavation and could take weeks to complete. These examples show how robots can be a useful tool for utility companies to perform difficult and tedious operations more efficiently and safely.

III. System Development

A. *Locomotion*

In order to operate effectively, the robot chassis needs to be able to handle rough terrain, traverse curbs and hills, maneuver around obstacles, support the cable detection and marking subsystems, and have an adequate battery life. The robot's battery life should be able to complete the cable locating and marking for a jobsite in a timely manner. The chassis should be able to travel with a technician who will set up and initialize the robot. It may need to maneuver through unknown environments. A pre-manufactured chassis will also aid in speeding up the process of creating a working prototype.

The previous year's MQP team recommended that the current team replace their Husky A100 with a custom chassis. The Husky A100 had an ineffective and underpowered drivetrain, and it was difficult to mount the cable detection and marking systems on to it. To determine the best replacement for the Husky A100, the current team considered multiple chassis options, weighed the task of designing a custom chassis, and analyzed the pros and cons (see Appendix A) to determine the best option. The best option was to order a six-wheel-drive robot from SuperDroid Robots, shown in Figure 2. A custom chassis would be too time-intensive and other chassis options were either too expensive or not as robust as the one from SuperDroid Robots. The HK1500 comes fully assembled, wired, and configured with a ROS control package.

ROS (Robot Operating System) is an open-source framework for robotic applications that allows for programming in both C++ and Python. It allows multiple pieces of code, called nodes, to run concurrently by publishing data and subscribing to various topics. The ROS framework is beneficial to this project because it allows multiple sensors and motors on the robot to operate and communicate simultaneously. The robot can then complete multiple functions at the same time such as following underground power lines and checking for obstacles to avoid.

Figure 2: The HK1500 chassis from SuperDroid Robots

The HK1500 can handle rough terrain, has a high ground clearance of 5.4 inches, and can drive over curbs up to six inches. Since battery life is an important component in this project, a 42 amp-hour battery package was purchased instead of the default 35 amp-hour battery. The robot is expected to last for at least one jobsite. The chassis also has a payload capacity of 250 pounds. The chassis is easily modifiable with the bars on the front and back as a place to mount the different custom subsystems and room inside for any extra electronics. The robot is set up with encoders to track wheel rotation. The HK1500 was the most cost-effective option that met the application requirements.

1) *System Architecture*

The system is centered around an Nvidia Jetson Tx2 Xavier that came pre-installed with the SuperDroid Robots chassis. Rosserial is used to communicate between the Jetson and an Arduino Mega. The Arduino Mega controls components on the robot such as stepper motors and servo motor drivers for the detection and marking subsystems. The Jetson also processes data from onboard sensors that communicate the robot's position and nearby obstacles, such as the IMU, LiDAR, drivetrain encoders, and GPS. More detailed descriptions of each component can be found in their respective sections. An overview of the architecture can be seen in Appendix B.

B. *Detection*

At its core, the robot relies on the ability to detect buried electrical cables. Continuing the work of the previous year's MQP team, the robot uses an electromagnetic utility locator for cable detection. The locator used is a Vivax-Metrotech vLoc3 RTK-Pro utility locator, supplied by Eversource. The goal of the detection subsystem is to find cables underground and determine their orientation. Since the locator is designed for human use, the robot needs to operate it similar to how a human would use it. The design and integration of this subsystem is detailed in the following sections.

1) *Locator Settings*

The Vivax-Metrotech vLoc 3 RTK-Pro utility locator, as seen in Figure 3, was supplied to the project by Eversource. The locator detects buried cables based on electromagnetic fields created by alternating current. The locator can operate in either active mode or passive mode. In active mode, the locator requires a transmitter that is connected to an external fixture, such as a fire hydrant, to transmit a frequency chosen by the technician underground. In passive mode, the locator detects the electromagnetic frequencies emitted by buried cables themselves. Passive mode has two locating modes to read and measure these signals. The first setting, peak mode, provides a maximum signal over the buried line and changes with the orientation of the locator. When operating in peak mode, the locator has the capability to detect the orientation of a buried cable through the strength of the signal received. If it is parallel to the cable, it reads the full strength. The second setting, omni peak, is similar to peak mode but ignores the orientation of a cable, reading full strength regardless of the locator's orientation to the cable.

Figure 3: The Vivax vLoc3 RTK-Pro utility locator with its central axis shown in red

When the locator detects a cable, it emits a sound through a speaker, and displays a number representing the signal strength on screen and a bar corresponding to the strength. A picture of this screen can be seen in Figure 4. The locator also has a built-in 4G cellular connection, global navigation satellite systems (GNSS) tracking, and real-time kinematic (RTK) positioning functionalities for tracking the location of a cable and uploading it to a database or web portal. GNSS and RTK are used to determine the location of the device.

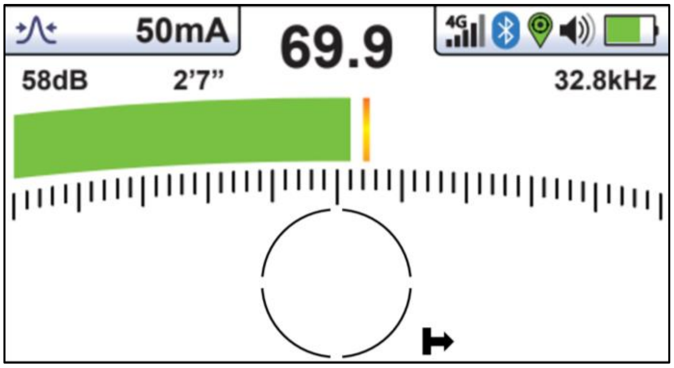


Figure 4: Vivax vLoc3 RTK-Pro utility locator screen

2) Interfacing

The locator is useful for detecting underground cables, but it has a few flaws for the needs of this particular project. The locator is designed with a human operator in mind and interfacing it with the robot is not a simple task. During human operation, signal strength is communicated through a signal strength indicator on the locator's screen or through a buzzer speaker. The first attempt to interface the locator with the robot was a microphone circuit that listened to the volume of the buzzer, as shown in Figure 5.

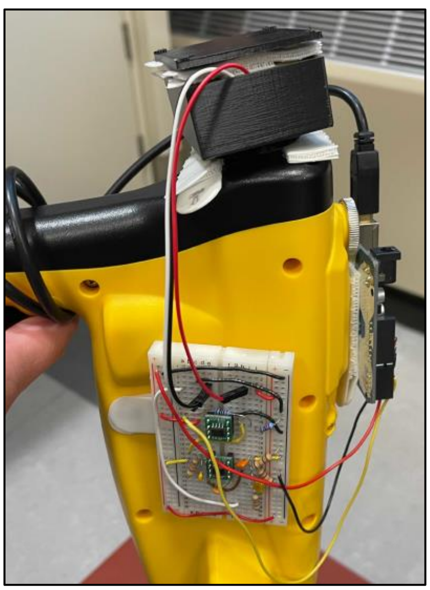


Figure 5: Previous MQP team's external microphone setup

This design introduced several problems, one being that ambient noise affected the circuit, such as vehicles passing by or people. The signal being sent to the speaker was not a standard signal waveform, making any internal signal processing difficult. Using camera vision to read the signal strength bar proved to be the best option for interfacing because it was least prone to noise, did not require complex filtering circuits, and could process data faster than the locator's screen refresh rate. The image processing algorithm applies a color mask to the screen, isolating the colored signal strength bar. The presence of colored pixels corresponds to a higher signal strength and therefore the presence of a cable.

3) Motion Requirements

When the locator is used in peak mode by a human operator, it is moved back and forth and rotated in order to determine the location and direction of the cable. To mimic the motion of a human operator, the robot must also move the locator laterally as well as rotate in place while remaining vertical, similar to how a metal detector is used. The translational movement helps to find a cable within the area in front of the robot. The direction of the cable can be determined by rotating the locator while it is above the center of the cable.

The previous MQP team designed a pair of wood panels that would move the locator in an arc shape, with the locator set to omni peak mode as shown in Figure 6. The linkage for the locator had problems with durability, reliability, and build quality. While this provided a start for cable tracking, it was not ideal for how the locator operated. Some shortcomings of this design were that it provided no control over the orientation of the locator and that the omni peak mode made it impossible to determine the direction of the cable. These issues would make following a cable difficult and inefficient.

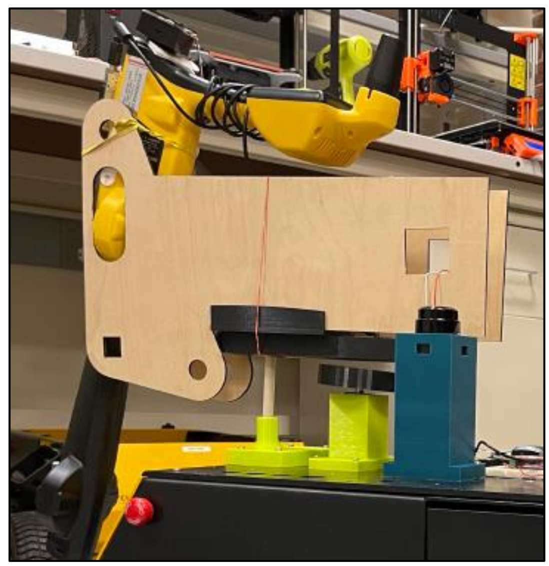


Figure 6: Previous MQP team's locator support

4) Rotational Subsystem

The goal of the rotational motion mechanism (RMM) is to orient the locator in line with a buried cable. As detailed in the vLoc 3 user manual, once the user locates a peak, "rotate the vLoc3 RTK-Pro on its axis to obtain the maximum signal. The [locator] is now directly over the line and with the blade across the line" [8]. Due to the operating procedures outlined by the manufacturer, the locator needs to be mounted such that it can rotate about its central axis, as shown in Figure 3.

Initially, a casing that clamps around the locator and sits inside a housing was considered, as seen in Figure 7. The outer diameter of the clamp was a gear or pulley driven by a DC motor. This design was discarded due to the inability to find bearings large enough to fit around the locator (about four inches). Instead, a joint had to be inserted in between the upper and lower halves of the locator.

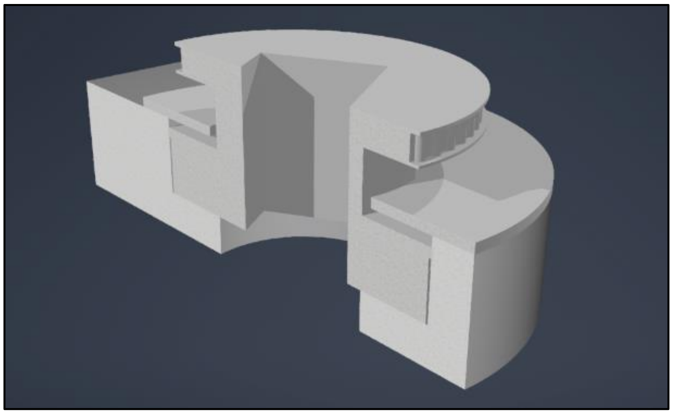


Figure 7: First iteration of the RMM, clamps around the locator

A small ribbon cable connects the upper and lower electronics inside the locator. With the much smaller size of a ribbon cable, smaller bearings could be used to support each half of the locator. Face-mounted bearings, as shown in Figure 8, were considered due to their low profile, easy mounting options, and commercial availability. A support plate could be mounted to the outer ring while the locator could be mounted to the inner ring and rotate freely. To overcome the high cost of face-mounted bearings, a custom solution had to be designed using more affordable components.

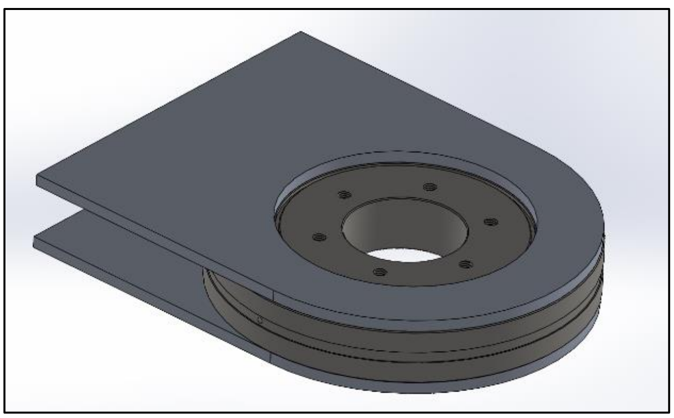


Figure 8: Second iteration of the RMM, face-mounted bearings

A helpful resource for the final RMM were joints from an open-source SCARA robot [12], as shown in Figure 9. This design utilized a series of two thrust and two radial bearings to create a rotating assembly sandwiched around a center mounting plate. The locator would be fastened to the top and bottom faces of the rotating assembly with the ribbon cable passing through the center of the bearings. The top and bottom components to the rotating assembly were pulled together using several bolts.

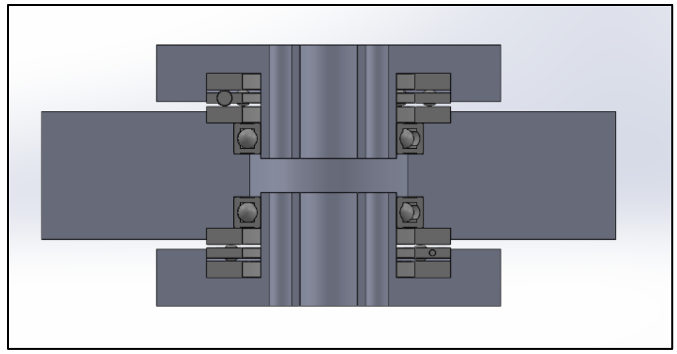


Figure 9: Cross section of the SCARA style joint

3D printers were used for rapid prototyping of locator mounting components. The team chose spur gears to rotate the system because a belt and pulley would require a belt tensioning device and extra hardware. A gear could also be easily incorporated into the top half of the joint.
Based on human use of the locator, 10 RPM was chosen as a starting output angular velocity for the locator. The locator's approximate mass moment of inertia was calculated using SolidWorks. A motion profile was generated using a quintic trajectory, as shown in Figure 10. The maximum angular acceleration was determined to be two rad/sec^2. Based on a dynamic torque analysis, as shown in Appendix C, 0.765 N/m of torque was required to rotate the locator. When choosing a motor, a stepper motor was the ideal type of motor to use in this application for several reasons, including a stepper motor's ability to produce a holding torque when not in motion and operation with open and closed loop feedback using an encoder. A 1.26 Nm NEMA 23 stepper motor with a 12:80 gear reduction was chosen due to the motor's output torque and power capabilities.

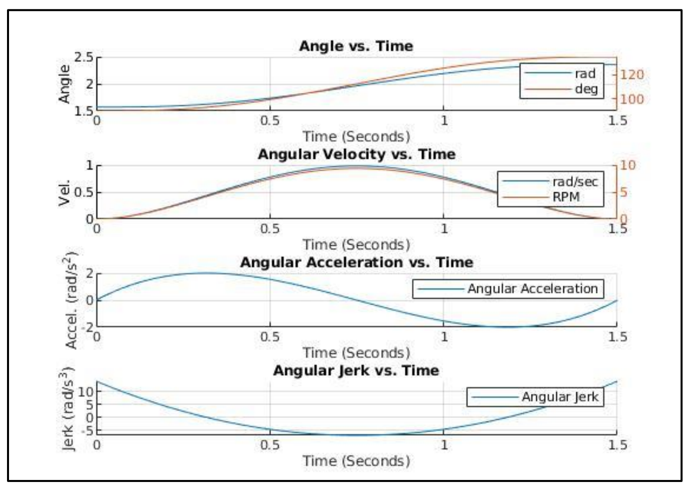


Figure 10: The motion profile for the RMM

The mounting bracket for the RMM was designed to be 3D printed based on the assumption that the entire locator and RMM assembly would weigh less than 20 pounds. The locator was placed closer to the robot to decrease the bending moment caused by its weight. The RMM mounting bracket was designed to be mounted to a subsystem that would translate the locator.

After the first test with the rotation system, several problems were discovered. Primarily, the locator's portion on top of the RMM was unstable and wobbled when the robot would move. This was caused by a high center mass from the battery and screen. The section of the locator containing the screen and battery were moved inside the robot's chassis to fix this issue, as seen in Figure 11. The two induction coils from inside the locator were mounted externally to maintain their intended orientation and location. The locator connects to the coils using a ribbon cable with intermediate cable extensions. Moving the screen and battery for the locator inside the chassis reduced the amount of weight on the detection system and helped to solve the flexing issue.

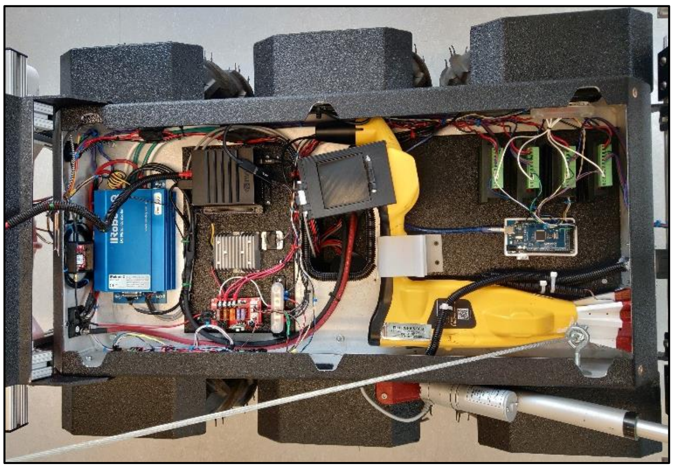


Figure 11: Upper half of the locator mounted inside the chassis

5) *Translational Subsystem*

The goal of the translational motion mechanism (TMM) is to move the locator and the RMM back and forth, simulating the movements a human operator would make. The TMM was mounted on the front face of the robot chassis. Based on human testing and operation, the locator should move back and forth at least six inches to establish a peak signal strength. The 2020 Demining Autonomous System [9] MQP project served as inspiration for a linear movement system. This robot had a translational mechanism that moved a metal detector for landmines back and forth using linear slides and a lead screw, as shown in Figure 12.

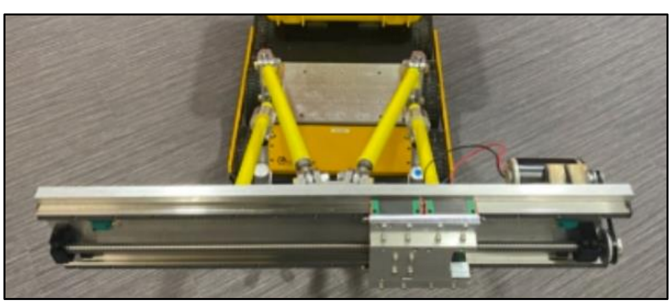


Figure 12: The demining project's translational motion mechanism

The possibility of using similar linear slides was explored due to their high load and moment capacity, tight tolerances, and mountability which is useful for this project. Initial design considerations for the TMM included the use of 80/20 extruded aluminum for easy mounting to the chassis and an RMM mounting solution that allows for quick iteration and improvement without having to disassemble both systems.

An early design for the TMM used two rails as a support for translational motion. Using two linear rails would reduce deflection and provide better support for the RMM. However, any type of linear rails presented high costs. Despite finding slightly lower cost options, such as slides that used PTFE strips instead of ball bearings, rails would be cost-ineffective and an over-engineered solution. An early design mockup with the entire locator outside of the chassis is shown in Figure 13.

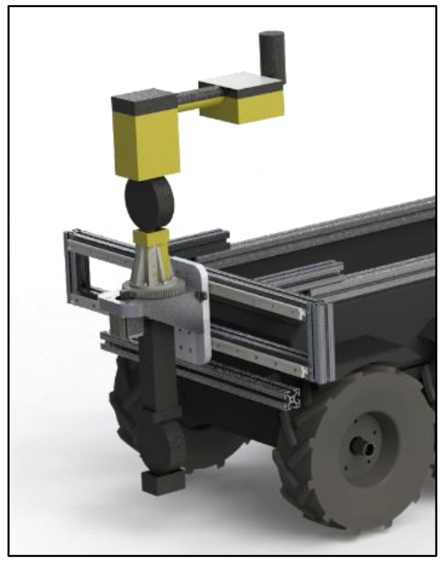


Figure 13: Preliminary mockup of linear rail system

Similar to their use in 3D printer or CNC mill construction, roller wheels can sit in 80/20 slots, as seen in Figure 14. Roller wheels present a cost-effective solution that meets the project's requirements. A carriage mounted with roller wheels was the first prototype of the TMM. The RMM mounting bracket would be fixed to this sliding carriage.

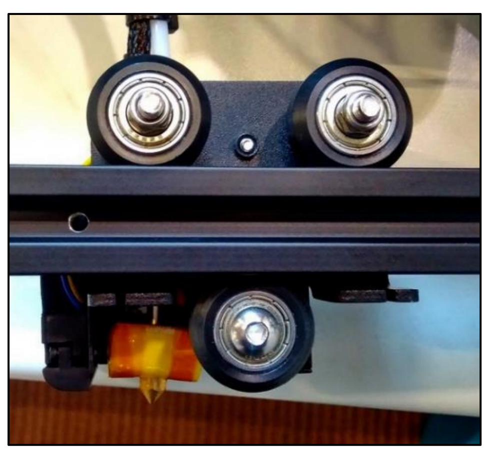


Figure 14: A roller wheel carriage from an Ender-3 3D printer

There were a few different methods that could provide linear actuation along with the roller wheels, such as ball end lead screws and belt drives. Benefits of lead screws include higher accuracy when rotating and quick response time. However, lead screws are limited in their linear speed capabilities, lengths, and are costly. If a lead screw was used, a belt and pulley assembly would be required to connect the motor's output shaft to the lead screw. Belt driven systems however have no limitations to their length and have higher linear speed capabilities [11]. Therefore, a belt driven system was the best choice for the first prototype.

Similar to the RMM, using a stepper motor was beneficial due to its ability to operate in discrete steps and apply a holding torque. If the translational system were to be retrofitted for closed loop feedback in the future, an encoder could be used on the motor to track the carriage's position. Based on an estimate of how fast a human would use the locator, the team decided that a starting linear velocity of 15 inches/sec was sufficient for the translational system. In the end, a 2.83 Nm NEMA 23 stepper met the requirements for this system. The motor directly drives the pulleys. An Arduino Mega controls the motors using the AccelStepper and MultiStepper library [15]. The Arduino communicates with the Nvidia Jetson using ROS Serial, a communication system to connect low-cost controllers with ROS over serial connection, to receive messages that state the positions the steppers need to move to.

The steppers motors drive a carriage assembly back and forth. The carriage consists of a mounting plate machined from 1060 Aluminum with holes for four roller wheels. The RMM bolts to the mounting plate, allowing for the RMM to be easily removable without affecting the TMM. The subsystem was constructed from 1.5 inch 80/20 extruded aluminum, with 3D printed support brackets and gussets.

Testing revealed that the locator is subject to interference from the drivetrain motors when mounted close to the chassis. To help mitigate the effects of EMI, the TMM was extended a foot out from the chassis. The finalized TMM with the RMM mounted can be seen in Figure 15.

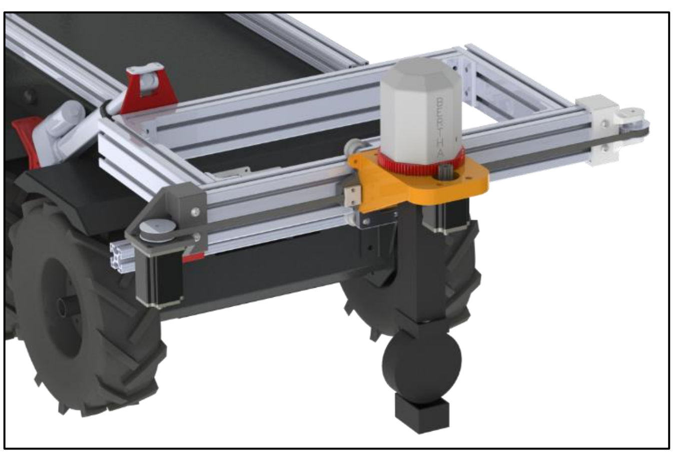


Figure 15: Render of both finalized detection subsystem.

During standard operation, the tip of the locator is five inches off the ground, making it impossible for the robot to travel over uneven terrain or over obstacles such as curbs without a mechanism to move the locator out of the way. Even though a typical curb height is six inches [14], curb heights can range anywhere from three inches to nine inches depending on when the curb was constructed or how frost heaves have affected the area. To establish a baseline functionality, a mechanism to raise the locator was designed for a six-inch standard curb. If the robot were to encounter a taller curb, a technician would have to step in to manually drive the robot over the curb while monitoring the locator's clearance.

To increase ground clearance and eliminate the risk of breaking the locator, the translational subsystem was mounted on custom hinges, allowing the system to tilt up and down. The tilting function is actuated by a Pololu linear actuator, as seen in Figure 16.

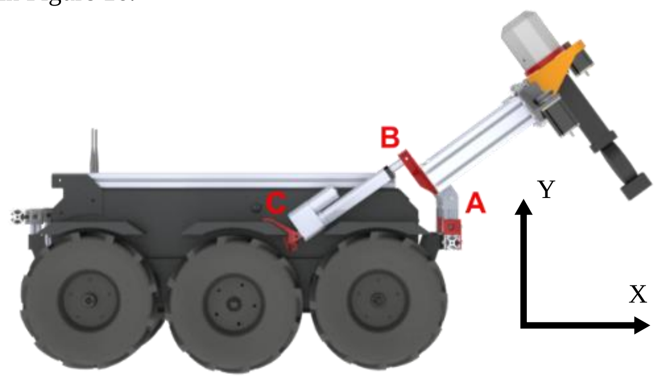


Figure 16: Side view of the robot with the locator tilted back for obstacles and joints

The linear actuator is controlled by a Pololu JRK21 v3 motor controller. The JRK incorporates feedback from the linear actuator's built-in potentiometer to monitor the current position with a defined setpoint. The JRK receives setpoints from the on-board Arduino Mega using I2C commands. The actuator has a maximum push capacity of 110 pounds and stroke length of six inches. When the linear actuator retracts, the locator moves to approximately 20 inches above the ground. The linear actuator takes about 10 seconds to retract or extend. To verify that one linear actuator was sufficient for supporting the detection system, a static force analysis was conducted, as shown in Appendix D. The results of the analysis showed that the linear actuator is subjected to less than 50 pounds, as shown in Figure 17.

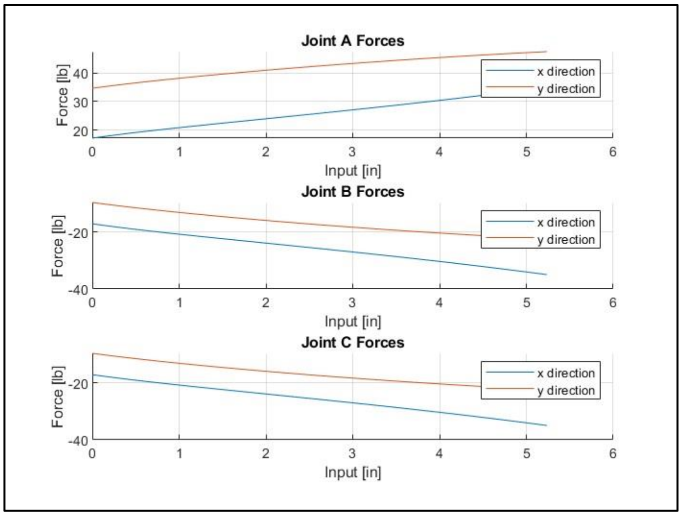


Figure 17: The forces acting on each joint as the linear actuator extends.

6) *Control Structure and Implementation*

For the detection subsystem, information from the locator screen needed to be sent to the path planning node and the motion of the locator needed to be controlled by the stepper drivers. A brief overview of the communication can be seen in Figure 18. The locator's motion is controlled in a Python file run on the Jetson. The locator sweeps back and forth keeping track of where the peak signal strength occurs through a separate Python file. In order to sweep the locator back and forth, a message is published to the appropriate topic to be processed by the Arduino Mega. The Arduino Mega uses the AccelStepper and MultiStepper libraries to control the motion of the steppers. These libraries allow the Arduino Mega to control multiple steppers in parallel using multiple digital stepper drivers. The Arduino Mega takes in the number of steps, velocity, and acceleration, sends that information to the drivers as pulses, and publishes the current position of each motor to the /curr_pos topic as they move. Throughout the motion of the locator, images of the locator screen are processed by the Nvidia Jetson on the robot. This reads the signal strength and publishes it to the /signal topic. This information is available for a separate path planning module to access and adjust the robot's motion accordingly.

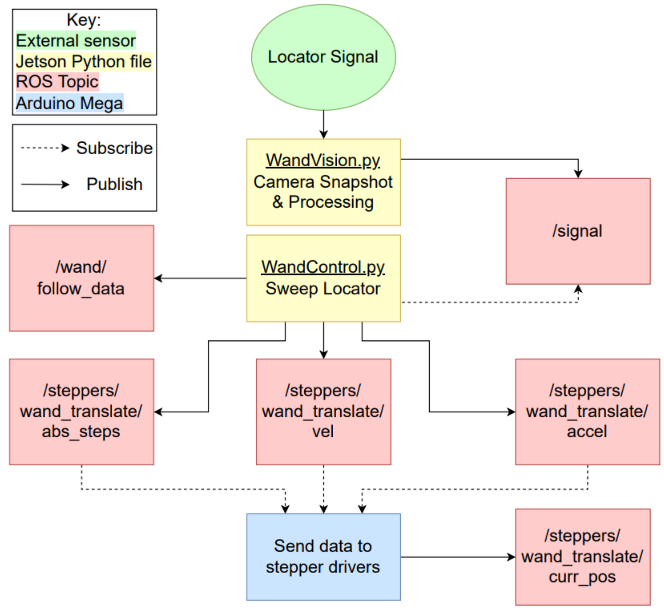


Figure 18: Overview of ROS structure for detection

7) *Testing Results*

When testing the detection subsystem, several problems were encountered with the locator's operation. With the locator coils mounted at the front of the robot, signal data did not present itself as a distinct peak, which made determining the location of a cable difficult. The signal strength displayed on the locator's screen would perform inconsistently, with the signal jumping between 20% and 99%. According to testing results, the stepper motors on the TMM and RMM played a major factor in affecting signal strength problems. If the stepper motor on the rotation mechanism was energized and sent any motion command, the signal strength on the locator would drop to zero and not recover. Following a sudden drop in signal strength, any attempt to re-calibrate the locator would result in a failure, unless the entire robot was powered off. EMI reducing materials, such as a nickel-iron ferromagnetic alloy called MuMetal, were used to try and reduce any interference caused by the stepper motors. However, such a material either amplified the signal fluctuations or absorbed all cable signals, resulting in the appearance of a 0% signal strength regardless of a buried cable's location.

Testing also revealed that the locator is subject to vibrations when used in conjunction with the robot. As the locator swept back and forth, the tip of it vibrated slightly. These vibrations magnified the signal fluctuations caused by stepper motor interference.

Lastly, the vision webcam-locator display interface had problems with timing and synchronization. When the locator would detect a cable signal, the slight delay caused by the screen's refresh rate offset the cable signal strength and location data. This caused a cable's location to be inaccurate. Timing delays were implemented in software to help sync data; however, the speed at which signal strength data and locator position data was received was not consistent.

C. Marking

The robot must be capable of producing clear markings that comply with Dig Safe standards. Each type of utility has a specific color that the technicians must mark their lines with, as shown in Figure 19. Each individual company has different cable markings/styles, but they will always mark different utilities in the appropriate industry-standard color. The team worked closely with local utility companies to determine what marking the prototype robot should make.

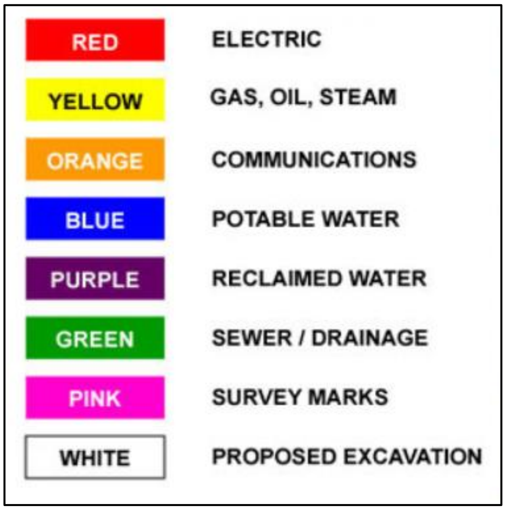


Figure 19: Dig Safe color marking chart

Eversource marks their buried cables with a red circle with two lines on either side. The circle has an eight-inch diameter, and each line is up to 18 inches on either side of the circle, marking a buffer zone. An example of this marking is shown in Figure 20. This buffer zone is to ensure that the cable will not be hit while people are digging near it [13].

Figure 20: Example electrical marking

The 2 DoF linkage from the previous MQP team included the can of spray paint pointing towards the ground at the end effector location, shown in Figure 21. The old linkage was 3D printed, actuated using two Pololu 37D motors, and used a servo motor to depress the nozzle of the can to start spraying paint. This solution had the ability to cover a large enough area of ground to spray the needed marking for the cables. However, there were flaws with the system. For example, the joints of the linkage were not rigid enough to support the weight of the system and the can of paint, causing the arm to droop and cause the motion of the linkage to become less precise over time.

After reviewing the system and analyzing other solutions, including a Cartesian frame system similar to what is found on a 3D printer, iterating the 2 DoF linkage system was the best option for the project. The rationale for this decision was that the 2 DoF linkage was the most compact system for the robot, which would ensure the robot is as light and small as possible in order to make transportation easier. 2 DoF linkages can be found on commercial robots, such as SCARA manipulators used in industrial settings. In addition, continuing the previous work of this project provides a foundation to build on rather than starting a new design from scratch.

Figure 21: Marking solution from previous MQP team

To improve the next iteration of the 2 DoF linkage, it was necessary to design the joints to be more rigid by using better mounting methods as well as increasing the number of bearings supporting each joint. Another improvement to make the system more rigid was to make the linkage as light as possible to limit the loads that the joints would need to support. The easiest way to remove weight from the previous design was to not have the can of marking paint attached to the arm. To reduce the load the spray arm must support, the spray paint can was fixed to the top of the robot and a valve was added at the end effector of the spray arm to control the spray. The can and the valve were connected with a tube to transport the paint through the system. In addition to moving the spray can, mounting the motors in a manner that kept them separate from the links further reduced the weight of the links, and thus decreased the loads the joints need to support.

One consideration for this solution is that the tube and valve cannot become clogged in between jobs or workdays. In addition, the can needs to be in a position that is easy to access for technicians to change when needed. Finally, the changing process needed to be simple, clean, and quick.

1) Spray Can Holder

In order to keep the can of spray paint on the chassis, a spray can holder was designed. The spray can holder secures the can in place on top of the robot, making it easy for a technician to remove the can. A tube is attached to the end of the paint can nozzle and runs through the arm to the release valve. Only a small amount of force applied to the can's nozzle is necessary in order to release paint. Therefore, a micro servo motor is attached to the end of a linkage which pushes the nozzle of the can with enough force for it to spray paint into the tube. Originally, the can holder held the paint can sideways. After testing the system, it was determined that this configuration released most of the air pressure, spraying a very little amount of paint out of the tube. This caused the paint cans to not have their full product life, leaving paint in an unpressurized can. This issue changed the can's configuration to be vertical, similar to how a person would hold the can when marking the ground. The final design is shown in Figure 22.

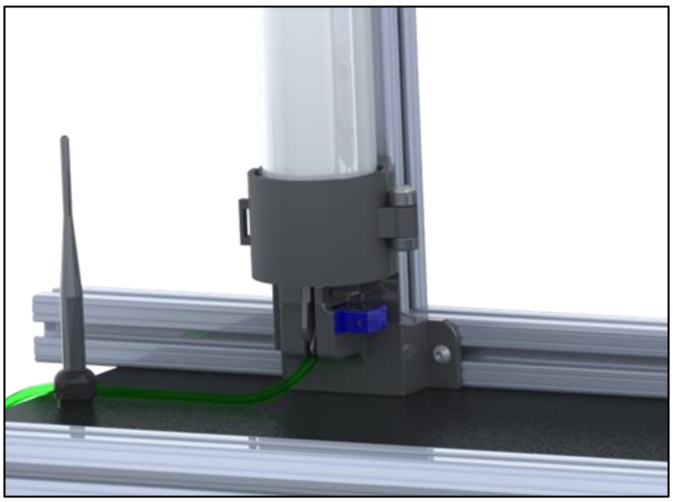


Figure 22: Spray paint holder used to attach the paint can onto the robot

2) Spray Arm

As mentioned previously, the spraying arm was a remodeled 2 DoF linkage based on the lessons of the previous version. The updated model is shown in Figure 23.

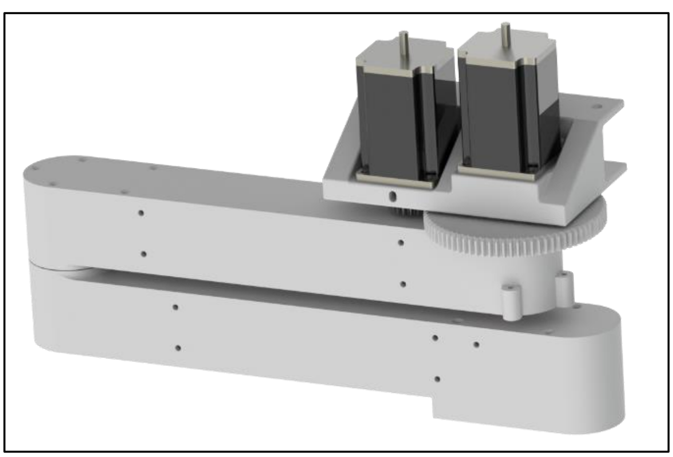


Figure 23: SolidWorks model of the whole spray arm assembly

After the model was completed, torque requirements were calculated to determine what type of motor should be used to power the spray arm. Two NEMA 23 stepper motors were chosen to drive the links. One motor drives the first link with a 1:4 gear ratio. The other motor drives a 1:1 belt and pulley system that moves the second link. The Arduino Mega board controls the motors through their individual stepper drivers. The motors connect to a bracket that is secured to the 80/20 rail that is situated on the back of the robot. This configuration can be seen in Figure 24.

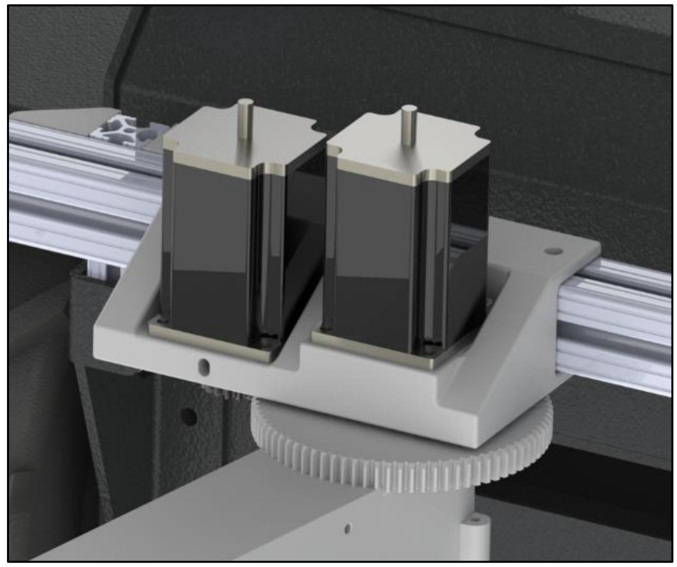


Figure 24: SolidWorks model of the 80/20 bracket and stepper motors

An important part of designing this system was how to make the joints strong, rigid, and as frictionless as possible. One of the most valuable resources to determine how to accomplish this was a homemade SCARA robot project [12]. From this robot, the team was able to inspect the CAD model of the joints to learn from what others have done. The homemade SCARA joint is shown in Figure 25.

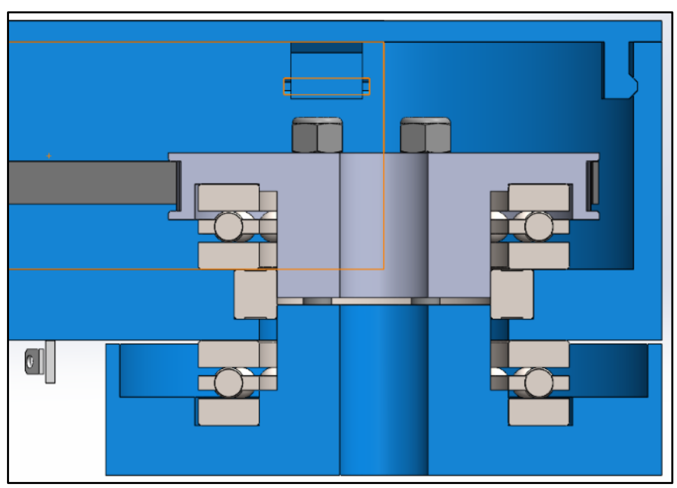


Figure 25: SolidWorks model of homemade SCARA robot joint

To hold the link securely, there was an internal ‘cap’ piece that was bolted to the bottom link and pinched the top link between a series of bearings. There is a thrust bearing supporting the bottom of the link and a radial and thrust bearing supporting the top of the link by being compressed downwards by the cap piece. The first joint of the spray arm was designed based on what was learned from the SCARA robot model. The first joint is more complex than joint two because the first link needs to pivot about the same central axis as the shaft that drives the belt and pulley system which independently drives the second link. To make this work, a part similar to the ‘cap’ piece of the SCARA robot was designed to be used for joint one to have the functionality of pinching the link between bearings. Two thrust bearings and one radial bearing are used between the first link and the ‘cap’ piece to reduce friction. The thrust bearings provide a smooth surface to rotate about for the link. The radial bearing is sandwiched in between the two thrust bearings to absorb most of the axial load. Together, this system of bearings allow smooth and almost frictionless rotation about the central axis.

The pulley shaft required some special components to be machined to work well. Because the stepper motor shaft is only 23 mm long, the shaft needed to be extended. The pulley has a 0.375 inch inner diameter for a shaft, so a coupler was designed to combine the 0.375 inch shaft extension and the 0.25 inch motor shaft. These components were machined out of 0.75 inch diameter steel round stock. The pulley itself is sandwiched between a thrust bearing and a radial bearing. The thrust bearing provides a smooth surface for the bottom of the pulley to rotate with, separating the pulley from the cap piece that is screwed into the bracket to hold the first link in place. The top of the pulley is encompassed by a radial bearing that has the same inner diameter as the outer diameter of the collar section of the pulley. This supports the pulley and shaft at the top, thus making this system fully supported. The joint that connects the first link to the bracket is shown below in Figure 26.

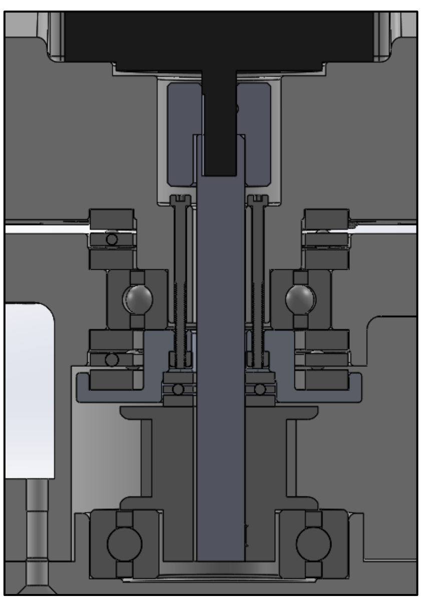


Figure 26: SolidWorks model of joint one (combination of 80/20 bracket and link one)

The links could not be printed as one piece because the 3D printers the team used have build plates that are 220 mm by 220 mm, and the lengths of the links are much greater than 220 mm. Because of this, the links were designed to print in three sections that get bolted together on the sides, as shown in Figure 27.

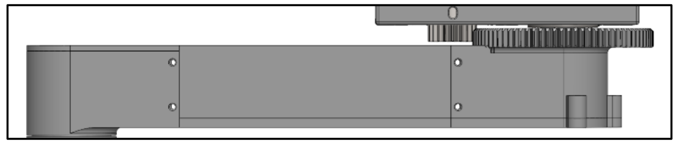


Figure 27: SolidWorks model of the first link separated into three pieces

Joint two was modeled using the same insight gained from the homemade SCARA robot joints. This joint is based around a 0.375 inch outer diameter shaft with a through hole 0.175 inches in diameter. This through hole is for the servo motor wires and paint tube to pass through the joint. Both ends of the shaft are threaded so that a lock nut can be tightened onto the ends, taking most of the tension forces off the collars holding components to the shaft. The pulley is secured with the nut on one end and a radial bearing on the other to hold it coaxial to the center of rotation. Link one and link two are separated by two thrust bearings, where one is inside of the other to help distribute the load of the second link more. Attached to link two, there is a collar that has a flange that allows for it to be bolted to a surface normal to the axis of the shaft. This collar was bolted to link two so that when the pulley rotates, it will spin the shaft which is hard connected by this clamping shaft collar with the flange bolted to the link, thus pivoting the link as the pulley moves. There is a nut tightened on this end of the shaft as well to help support the load acting on the flanged clamping collar. This setup is shown in Figure 28.

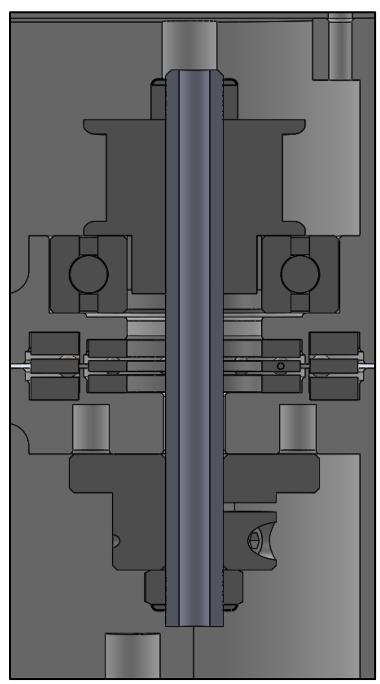


Figure 28: SolidWorks model of joint two (joint between the two links).

3) Valve

The end of link two has a small blind hole that allows for a valve to be pointed down at the ground that a plastic nozzle sets into. The servo motor used to actuate the valve is secured in the end of link two, and the method of securing the servo also secures the nozzle in place so that it is always facing the ground.

There is a PVC valve at the end of the end effector which controls the output of paint from the arm. The inlet to the valve is connected to the outlet of the paint can tube. The valve's handle is controlled by a servo motor whose output is rigidly attached to the handle. When the paint needs to be sprayed, the servo articulates the handle to the 'open' position. When the paint spray needs to be stopped, the servo articulates the handle to the 'closed' position. This part of the system is contained within the end piece of link two, where there is a hole for the nozzle to protrude from the arm for the paint to be sprayed. A cross section of this system is shown below in Figure 29.

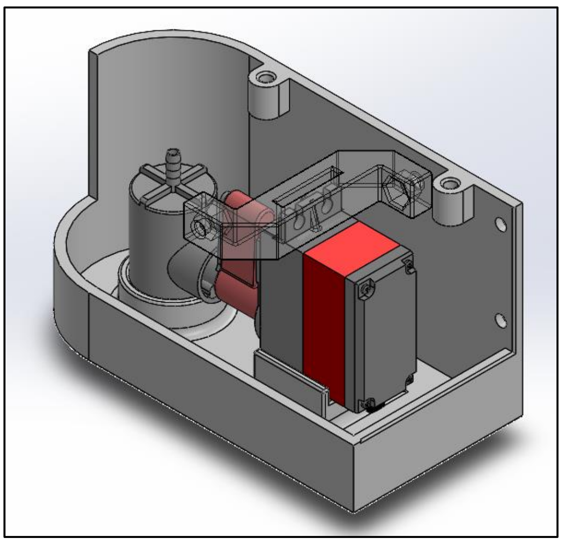


Figure 29: Cross section of servo and valve system inside link two

4) Marketing System Testing

Initial testing revealed that there were a few problems with the system that needed to be fixed. The spray arm was mechanically responsive and the inverse kinematic calculations allowed it to function as intended, but there were issues with the painting system. The first test was completed without a valve at the end effector and instead only had a micro servo press down on the spray can's nozzle. The first major issue with this design was the large delay between pressing down on the spray can's nozzle to release paint versus the paint actually spraying out of the tube. This problem occurred because of the time required to pressurize and depressurize the tube. To rectify this issue, more precise control over the spraying of paint was needed. Two possible solutions included using an electromagnetic solenoid or a PVC valve. With testing, it was found that the solenoid clogged after use whereas the PVC valve did not. For this reason, the PVC valve and servo motor system was added to the end effector to control the outlet of paint.

Another issue identified during the first system test was that the paint marks made by the spray arm were not clear. This is due to the pressurization issue as well as the nozzle being too high above the ground. For this test, the spray arm nozzle was 16 inches above the ground. To try to make the marks clearer, the spray arm was lowered so that the nozzle is 7.5 inches above the ground. With the end effector closer to the ground, the paint is concentrated into a narrower stream, making the markings clearer.

5) Control Structure and Implementation

The overview of the ROS messages for the Spray Control is shown in Figure 30. First, there is the drawing module that is executed on the Nvidia Jetson. The spray arm's trajectory is generated to determine 2D Cartesian coordinates for the arm to move through as it completes its motion to create clear lines. Coordinates are converted to the joint angles for both link one and link two. Then, the inverse kinematics for the spray arm are calculated in order to determine how to control the location of the end effector of the spray arm. Lastly, the number of motor steps required to move the links to the proper angle is published as a ROS message as well as when the spray should be toggled on and off. This information is sent to the Arduino Mega which commands the stepper motors through two digital stepper drivers and publishes the current position of the stepper motors to the appropriate topic.

After the first test, some updates were made to the software. Initially, only stepper positions were sent as ROS messages to the Arduino. This caused the spray arm to stop at each point instead of continuously moving between them. To fix this, velocity control was implemented so that the spray arm could move smoothly between points. There were also improvements to ensure that the robot would not attempt to move outside of its range.

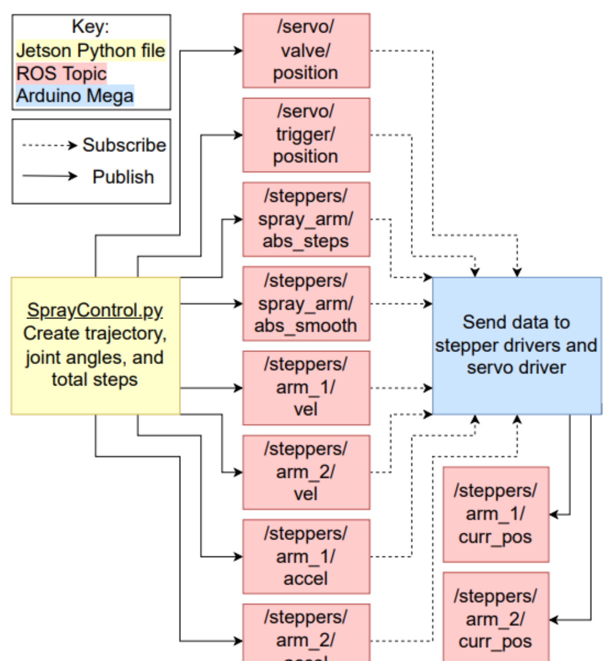


Figure 30: Overview of ROS structure for marking

D. Navigation

The robot needs to know its location in the environment to determine where the expected and actual cable location is relative to the robot. Additionally, the robot will need to monitor its environment to avoid both stationary and moving obstacles, as shown in Figure 31. To maintain a record of its current position and to avoid obstacles, the robot uses an Inertial Measurement Unit (IMU), rotary encoders, RTK (Real-Time Kinematic) connections, and a 2D LiDAR. The RTK connection uses a GPS signal and a network connection to more accurately determine the robot's global position. Using the collected data, an Unscented Kalman Filter is used to fuse the position data, shown in Figure 32.

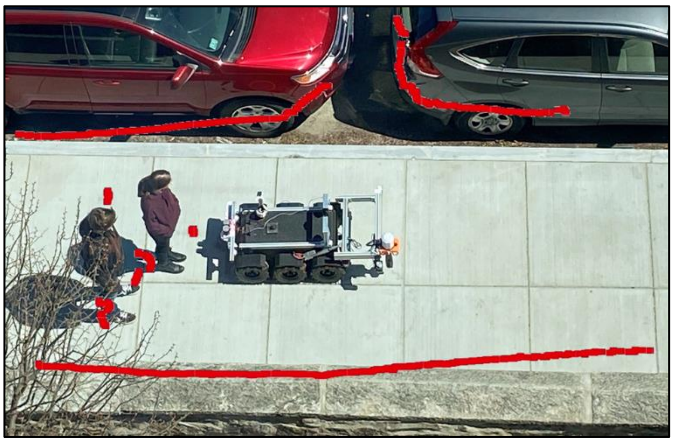


Figure 31: Robot using LiDAR to see surrounding obstacles

The path following module on the Nvidia Jetson interprets data from locator readings and adjusts the robot's position to stay centered over the cable. During this process, the robot keeps track of and processes information from its various sensors. It uses the sensor readings to determine the position of the cable and uses encoder values, IMU data, and GPS data to keep track of its location as it moves.

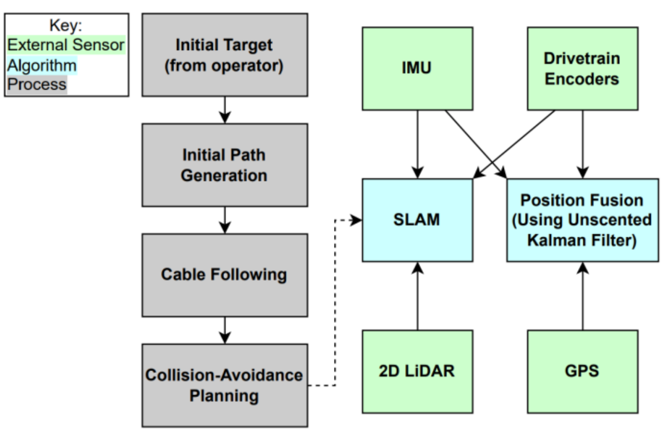


Figure 32: Navigation hierarchy

1) Control Structure and Implementation

The overview of ROS messages is shown in Figure 33. For controlling the motion of the robot, the HK1500 came with a control package pre-configured by SuperDroid Robots. The SDR controller uses the /cmd_vel topic in order to publish the left and right chassis speeds to the motors. There is also a topic for an emergency stop that, when activated, stops all motion of the motors.

The SDR controller publishes ROS messages containing information from the left and right drive train encoders. This information is then processed by a localization module along with information from sensors onboard the robot. These three pieces of information are sent through a Kalman filter to determine the robot's relative and global locations. The localization module then publishes the robot's pose.

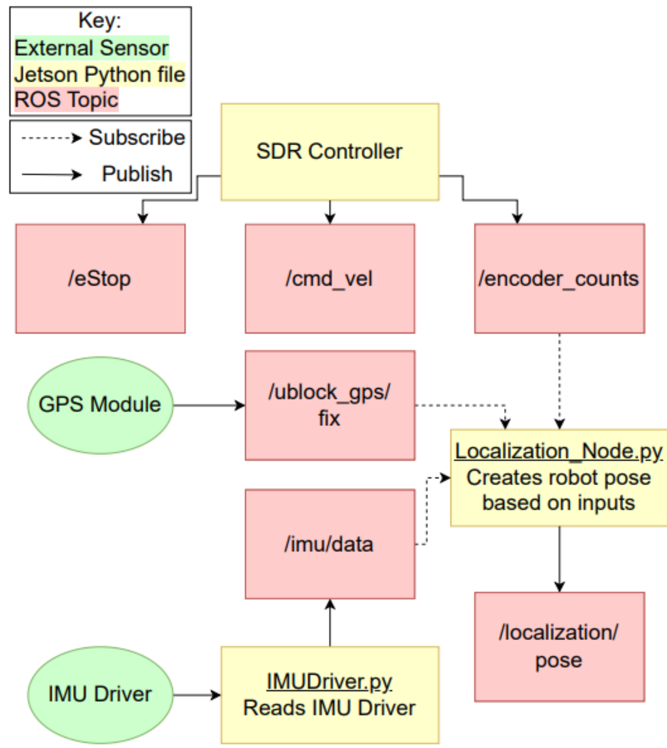


Figure 33: Overview of ROS structure for localization

2) GPS and Path Planning

Initially for cable following, a simple proportional controller was used to correct the robot's position. This allowed the robot to be successful in following the cable, but it caused a lot of unnecessary zig-zag motion, which was an inefficient method of following the cable. A new algorithm was developed to better follow a cable.

The new cable following algorithm, shown in Figure 34, works by using past data from the locator to calculate a best fit line and predict the future direction of the cable. A Gaussian curve is fit to the raw signal strength data of one sweep of the locator in order to find the peak of the signal strength per pass of the locator over time. In the future, this algorithm can be modified to use maps of known cable locations to make predictions as well as the recorded signal data. Simulations were created to test the algorithm before testing it on the robot. In the simulation, noise was added to the signal strength readings so that the response of the algorithm would more closely resemble the real world.

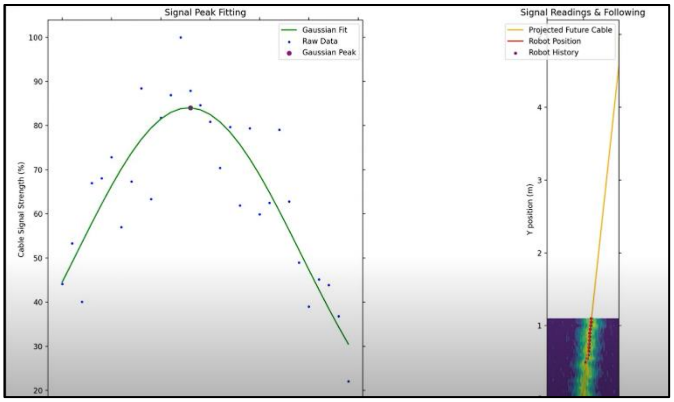


Figure 34: Cable following algorithm simulation.

Additionally, obstacle avoidance using the 2D LiDAR was developed. The robot is able to detect objects in front of it and plan a path around the obstacle. This is done by navigating around the obstacle in such a way that the robot keeps a buffer zone between itself and the obstacles while doing its best to navigate back to the projected path of the cable as fast as possible. This process has been tested in simulation and is shown in Figure 35.

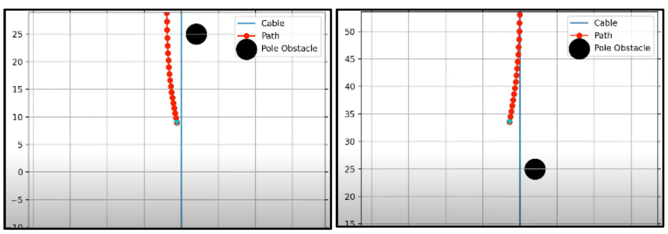


Figure 35: Obstacle avoidance algorithm simulation

IV. System Integration

A. Demonstration One: Basic Cable Detection

The focus of demonstration one was testing the integration of the locator with the robot in order to follow underground power lines. The demonstration was held outside the Fall semester lab space, 85 Prescott Street. During this demonstration, the locator was able to sweep back and forth and process signal strengths using the vision camera. The robot was tasked with following a cable along a sidewalk, across a street, and at a 90-degree cable turn. The robot also used object detection to stop when the robot got too close to anything.

Figure 36: The robot at the testing location for the basic cable detection demonstration

This demonstration revealed flaws in the first iteration and allowed for design improvements to be made. With the whole locator attached to the front of the robot, the linear motion was jerky and there was a lot of slop due to the weight of the locator. This resulted in inaccurate data being sent to the cable following algorithm. The movement of the locator also caused the bolts supporting the upper half of the locator to bend and caused several 80/20 brackets to break. Another problem was that the front 36-inch rail was too wide to fit through a doorway, making it difficult to bring the robot outdoors.

Resolutions to these issues were quickly implemented. The weight issues caused by the upper half of the locator led to it being relocated inside the chassis, leaving only the bottom half outside of the chassis. To increase the strength of 80/20 brackets that failed, they were re-printed using higher infill densities and wall thicknesses. Finally, the translational sub-mechanism was cut down to be 28 inches long.

When demonstrating the autonomous cable detection functionality, it was difficult to quickly switch between remote control mode and cable following mode. To switch, various command line interface messages had to be sent to the robot to launch and stop various processes, which would be difficult for a technician to do on the job. The team planned to make the robot switch between operator mode and autonomous mode using the remote controller for the next demonstration. To do this, a button on the remote controller would toggle between the two modes.

The demonstration also showed issues with the cable following algorithm. The robot was unable to handle crossing the street or handling a 90 degree turn. This could have been attributed to the inaccurate data received from the locator. The lack of consistent cable following resulted in a new, updated algorithm. GIS data could have also helped the robot navigate the 90 degree turn and other complex cable layouts, as the robot would have information to reference to better guide its cable location prediction. However, it was noted that the location used for this demonstration was not an ideal location, as it was difficult to find buried cables using the locator by hand, let alone with the robot. Future demonstrations would be held in locations that have cables that are easily detectable by hand and with the robot.

B. Demonstration Two: Full System Proof of Concept

The goal of demonstration two was to test all subsystems on the final robot prototype, as seen in Figure 37. This could be demonstrated by having the robot follow and mark a cable as it drove down a straight sidewalk. This demonstration was held outside the Spring semester lab space, 27 Boynton Street. This location proved to be better suited as a testing environment because a known cable was found outside the building using the locator. WPI facilities schematics also helped to confirm the location of the cable.

Figure 37: Render of robot for the full system demonstration.

Leading up to the demonstration, the goal was that each functionality would be implemented; however, several problems were encountered. The majority of these problems arose from the detection subsystem. While the translation and rotation subsystem mechanically worked as expected, the locator showed signs of severe interference and inconsistency. Interference caused cable signal peaks to be flattened and the locator's calibration process to fail. Since the signal peaks were flattened, the cable following algorithm could not identify the location of a cable. If the locator failed its calibration process, the signal strength would fall to zero and read inaccurately. During the test, the locator was not able to detect a cable, despite the robot being placed directly over one, as seen in Figure 38. Multiple tests were conducted by trying different configurations of electronics being powered and unpowered and observing the locator's response. It was found that the stepper motors being on and being commanded to move in any manner caused the locator to not work properly. EMI reducing material was placed between each stepper motor and the locator; however, the accuracy of the signal strength values did not improve.

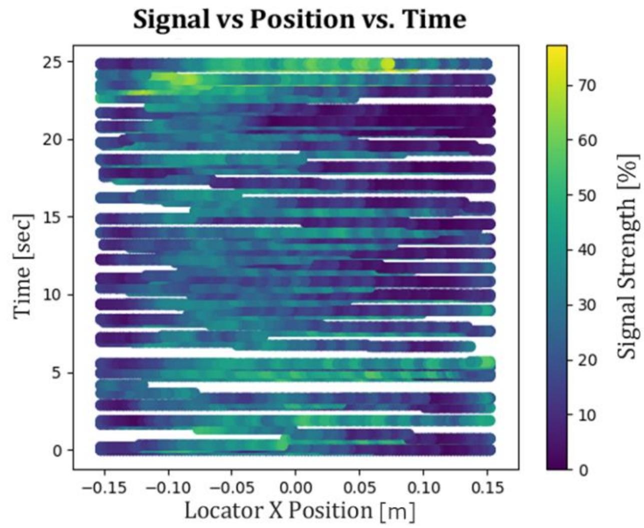


Figure 38: Data from a test when the locator was experiencing interference.

For the detection subsystem, the linear actuator was able to be mechanically implemented but not fully implemented in software. Due to communication issues with the Arduino Mega, the linear actuator could not be implemented with ROS. If the linear actuator was sent a command to move to a particular setpoint and then the locator was swept back and forth, the Arduino Mega would freeze. However, the linear actuator can be manually controlled using the Pololu JRK configuration utility when connected to a computer. This problem could be caused by a software bug in the JRK code. Due to time constraints, this issue could not be solved.

Despite the problems encountered with detection and navigation, the marking subsystem performed as expected. When sent the command to paint a cable marking, the spray arm successfully painted a marking, as seen in Figure 39. While this paint marking was not able to be implemented into the cable following algorithm, the robot was able to mark the ground on command, demonstrating its ability to complete the task independent of its other tasks. One issue noticed was due to the height of the end effector, where on a windy day, the marking became hazy because overspray was spread farther than it should have been. Also, the amount of pressure left in the spray can determined how much paint would come out of the nozzle. This affected how much the release valve should be opened or closed. Overall, the shape of the marking was able to come together and be implemented with the opening and closing of the paint's release valve.

V. Conclusions and Future Work

Overall, this project provided an initial proof of concept and basic functionalities of an electric cable marking robot. Much was learned about integrating an electromagnetic sensitive sensor with a robot. The robot was able to create identifiable spray markings to denote where buried cables are located. The robot also has simulation-tested obstacle avoidance software ready to be implemented in a real-world setting. While there has been a lot of work, more work must be done before the result is a fully functioning system.

Some next steps for further development include improving short obstacle detection (such as detecting a curb), incorporating autonomous linear actuator control, improving the human-robot interface, and implementing the obstacle avoidance algorithm into the real-world environment. These are all steps that should be wrapped up before starting more laborious tasks. In addition, combining the work of this year's team with the work of last year's team would enable the system to access GIS information and serve as a foundation for future work.

Down the road, future teams may consider replacing or designing a custom locator that is not affected by the interferences caused by the robot itself. This may be possible by creating a custom filter which neutralizes the noise caused by the motors. This could also enable implementation of sensitivity control with the robot. In addition, implementing dynamic obstacle avoidance into a real-world environment would be a large step towards a finalized system. Interfacing the robot with GIS mapping software would also allow the robot to better detect, follow, and mark cables, as well as update existing records with more accurate information. Finally, the development of a mobile charging station for the robot between job sites would be a beneficial quality of life element.

VI. Acknowledgments

We thank Professor Greg Lewin and Professor Jing Xiao for their guidance, feedback, and dedication towards this project. We thank Karter Krueger for developing the cable following algorithms and working on the software side of this project. The many long days and nights did not go unnoticed. We also thank our sponsor Eversource and in particular, Umair Zia for providing feedback and guidance. Eversource provided funding to the project through ROSE-HUB, which allowed the purchase of components necessary to develop this project further. We also thank James Loiselle and the Washburn shops staff for helping in the machine shop and running the shop both late and early. Finally, we thank Zach Rivernider for SMD soldering vital components when they broke. Without everyone's help, this project would not be where it is today.

VII. References

[1] "What is DIG Safe®?" Dig Safe System, Inc. - MA ME NH RI VT - Call 811, 03-Jun-2021. [Online]. Available: http://www.digsafe.com/. [Accessed: 14-Oct-2021].

[2] B. Moak, "811 averts underground utility disasters," Clarion Ledger, 30-Jun-2017. [Online]. Available: https://www.clarionledger.com/story/business/2017/06/30/811-averts-underground-utility-disasters/429572001/. [Accessed: 14-Oct-2021].

[3] K. Martindale, "Keep workers safe, costs low: Why Ameren Missouri is leveraging the power of robotics," [Online] Available: https://go.gale.com/ps/i.do?id=GALE%7CA609381960&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=0032082X&p=AONE&sw=w&userGroupName=mlin_oweb&isGeoAuthType=true [Accessed: 24-Jan-2022].

[4] J. Allen, "GE Hitachi Ultrasonic Robot Completes Inspection of Buried Pipe at Texas Nuclear Power Plant, " GE, 9-Dec-2014. [Online]. Available: https://www.ge.com/news/press-releases/ge-hitachi-ultrasonic-robot-completes-inspection-buried-pipe-texas-nuclear-power [Accessed: 24-Jan-2022].

[5] M. Saccone, "Eversource worker hit by a car in carver while working to restore power," Boston 25 News, 14-Apr-2020. [Online]. Available: https://www.boston25news.com/news/local/eversource-worker-hit-by-car-carver-while-working-restore-power/KRBSO2RCCFFG5LLN4YXIRAK6ME/. [Accessed: 15-Oct-2021].

[6] W. I. T. N. W. Team, "Man marking underground utilities dies after hit by car," https://www.witn.com. [Online]. Available: https://www.witn.com/2020/08/09/man-marking-underground-utilities-dies-after-hit-by-car/. [Accessed: 15-Oct-2021].

[7] P. S. Division, "DIG safe statistics," *Mass.gov*. [Online]. Available: https://www.mass.gov/info-details/dig-safe-statistics. [Accessed: 15-Oct-2021].

[8] Vivax Metrotech, *vLoc3 RTK-Pro User Handbook*. Vivax Metrotech.

[9] Santos, Adam R. et al. *Demining Autonomous System*. Worcester Polytechnic Institute, 2020. Print.

[10] "Sleeve Bearing Carriages and Guide Rails," *McMaster Carr*. [Online]. Available: https://www.mcmaster.com/linear-bearings/sleeve-bearing-carriages-and-guide-rails-7/. [Accessed: 17-Dec-2021].

[11] "Belt Drive or lead screw?," *Belt drive or lead screw? The answer is in the application.* [Online]. Available: https://www.pbclinear.com/Blog/2020/February/Lead-Screw-or-Belt-Drives. [Accessed: 17-Dec-2021].

[12] "Scara Robot: How to build your own arduino based robot," *HowToMechatronics*, 01-Dec-2021. [Online]. Available: https://howtomechatronics.com/projects/scara-robot-how-to-build-your-own-arduino-based-robot/. [Accessed: 17-Dec-2021].

[13] "State Of Massachusetts Dig Safe Rules." Available: http://www.digsafe.com/documents/marules.pdf [Accessed: 6-Mar-2022].

[14] "Roadway surface - Boston." [Online]. Available: https://www.boston.gov/sites/default/files/imce-uploads/2019-01/s.3_-_sidewalk_-_hma.pdf. [Accessed: 07-Mar-2022].

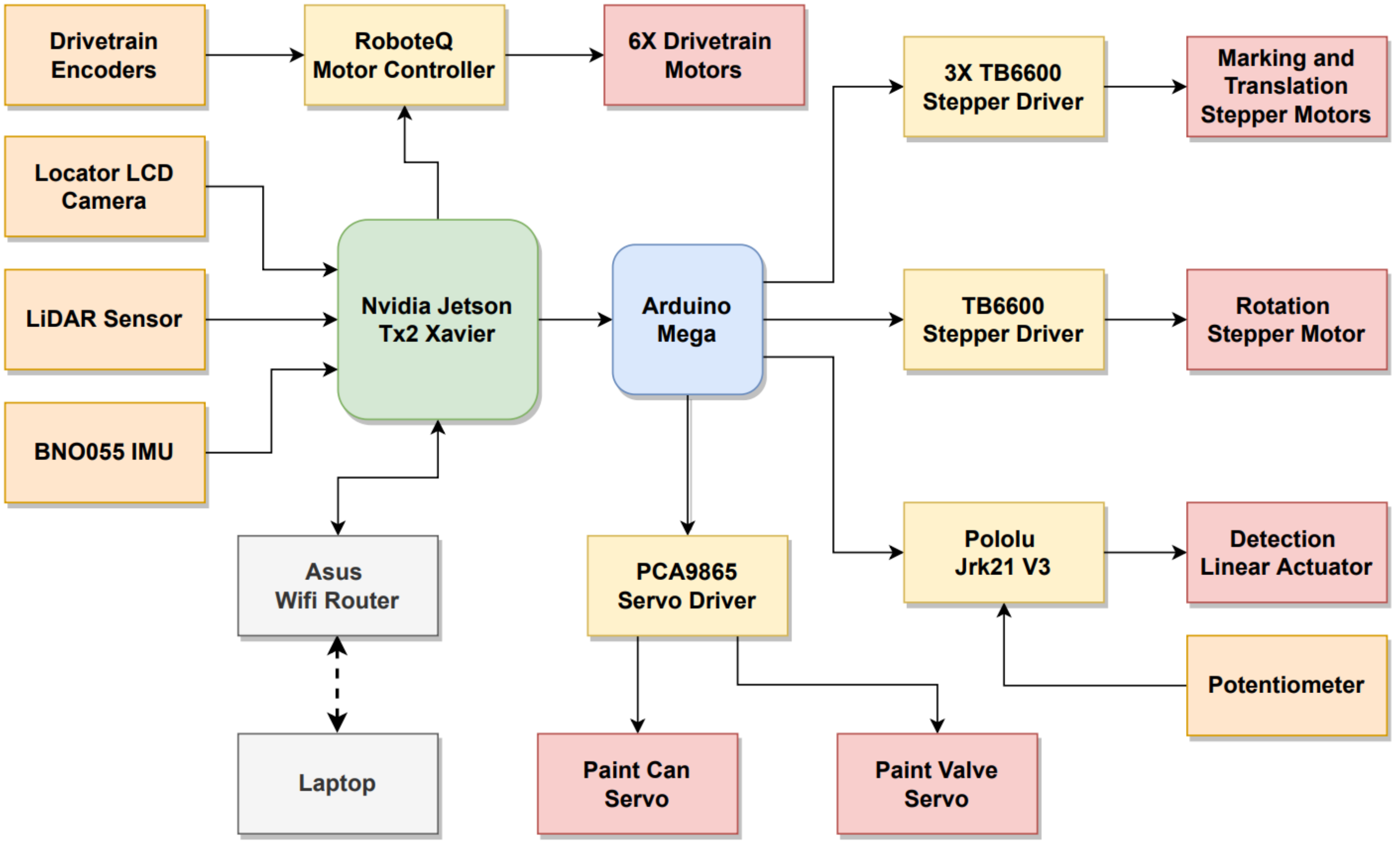
[15] "AccelStepper library for Arduino" Available: http://www.airspayce.com/mikem/arduino/AccelStepper/ [Accessed: 24-Apr-2022].

VIII. Appendices

A. *Appendix A: Chassis Decision Matrix*

Platform	Pros	Cons
Custom Design	- Made to our specifications - Can customize parts to improve battery life	- Very time consuming - Likely to require several iterations - Potentially more expensive than a prebuilt robot
Clearpath Husky	- Fully supported in ROS - All terrain - Known and trusted company - Customizable - Easy to attach sensor to	- More expensive than other options
Stanley Innovation RMP440	- All terrain - ROS capable - Long battery life - Somewhat customizable	- Larger than the other options and likely more expensive
SuperDroid Robots HK1500	- Fully supported in ROS - All terrain - Customizable - High ground clearance - High payload capacity - Fully assembled	- Not well-known company

B. Appendix B: System Architecture Flow Chart



The system is centered around an Nvidia Jetson Tx2 Xavier. The Nvidia Jetson handles commanding the drivetrain motors, image processing from the camera observing the screen of the cable locator, data acquisition from the IMU and LiDAR, and maintains the wireless network through the router. The Nvidia Jetson also communicates with an Arduino Mega, which handles the servo motor commands, the stepper motor commands, and the linear actuator commands.

C. Appendix C: RMM Dynamic Torque Analysis

Assumptions:

- The RMM motion profile will follow a quintic trajectory
- An approximate desired output velocity is 10RPM
- An approximate desired acceleration is 2 rad/sec^2

$I = Locator\ Moment\ of\ Inertia = 175.45\ lbin^2 = \ .051kgm\text{^}2$
$\omega_{out} = Max\ Velocity = 10RPM = 1.047\ rad/sec$
$\alpha = Max. Angular\ Acceleration = 2\ \ rad/\text{sec}^2$

$\tau_{out} = Required\ output\ torque = I\alpha = .051 * 2 = .102\ Nm$
$P_{in} = Required\ output\ power = \omega_{out}\tau_{out} = 1.047 * .102 = .107\ W$

$n = Gear\ Ratio = 12{:}\,80 = .15$
$\tau_{in} = Required\ input\ Toqrque = n\tau_{out} = \ .15 * .102 = .0153\ Nm$
$\omega_{in} = Input\ Velocity = \frac{\omega_{out}}{n} = \frac{10}{.15} = 66.67\ RPM = 6.98\ \text{rad/sec}$
$P_{in} = Input\ Power(double\ check) = \tau_{in}\omega_{in} = .0153 * 6.98 = .107\ W$

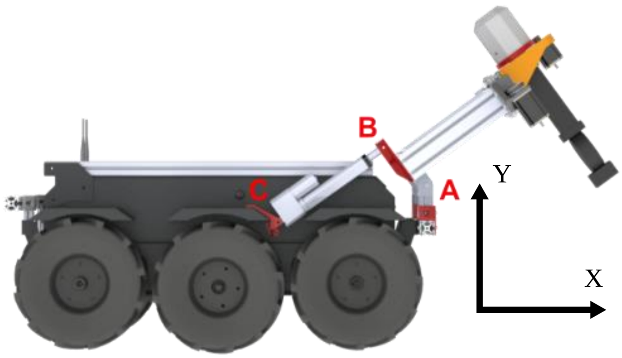
$Factor\ of\ Safety = 2$
$Final\ Required\ Input\ Torque = 2 * .0153 = 3.06\ Ncm$

D. Appendix D: Linear Actuator Static Analysis

Assumptions:

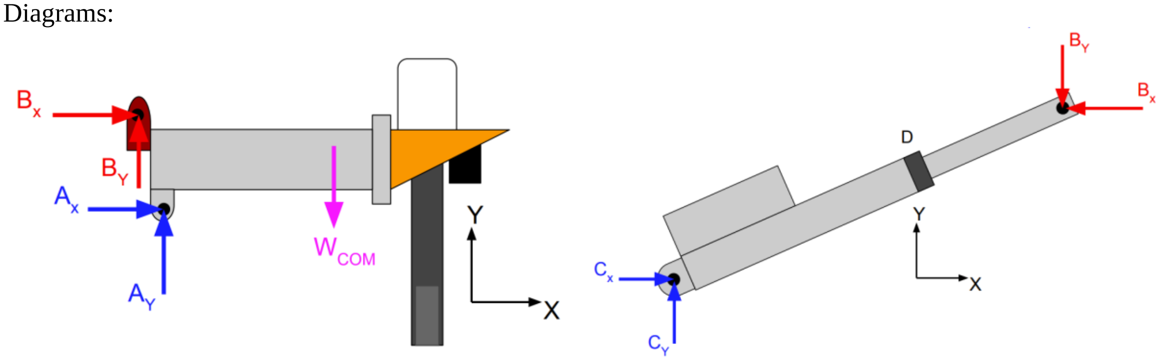
- The robot is stationary during operation
- The linear actuator moves at a constant velocity
- No manufacturing imperfections
- The robots home configuration is with the detection subsystem horizontal
- The input to the system is the linear actuator length, which ranges from 0 to .133 meters.
- The weight of the linear actuator is negligible
- The weight of the detection subsystem can be approximated as a center of mass.

The following coordinate system was used for calculations. Measurements of the initial joint positions were found using SolidWorks.



Nomenclature Table		
Variable (Bold is a vector)	**Description**	**Units**
$\boldsymbol{F_a}$	Force vector at joint A	N
$\boldsymbol{F_b}$	Force vector at joint B	N
$\boldsymbol{F_c}$	Force vector at joint C	N
$\boldsymbol{W_{COM}}$	Weight of the detection subsystem.	N
$\boldsymbol{r_{b\backslash a}}$	Vector from joint A to B	m
$\boldsymbol{r_{COM\backslash a}}$	Vector from joint A to COM	m
$\boldsymbol{r_{B\backslash C}}$	Vector from joint C to B	m
g	Acceleration due to gravity	m/s^2

Free Body Diagrams:



Static Equations of Equilibrium:

$\sum \boldsymbol{F} = 0 = \boldsymbol{F_a} + \boldsymbol{F_b} + \boldsymbol{F_{COM}}$

$\sum \boldsymbol{M_a} = 0 = \left(\boldsymbol{r_{B\backslash A}}\ x\ \boldsymbol{F_b}\right) + (\boldsymbol{r_{COM\backslash A}}\ x\ \boldsymbol{W_{COM}})$

$\sum \boldsymbol{F} = 0 = \boldsymbol{F_c} - \boldsymbol{F_b}$

$\sum \boldsymbol{M} = 0 = (\boldsymbol{r_{B\backslash C}}\ x - \boldsymbol{F_b})$

$\boldsymbol{F_a} = \boldsymbol{A_x} i + \boldsymbol{A_y} j$

$\boldsymbol{F_b} = \boldsymbol{B_x} i + \boldsymbol{B_y} j$

$\boldsymbol{F_c} = \boldsymbol{C_x} i + \boldsymbol{C_y} j$

$\boldsymbol{W_{COM}} = -\boldsymbol{W_{COM}} j$

MATLAB was used to solve for the various joint forces.
The plot of the joint forces with respect to position input can be seen below.