Worcester Polytechnic Institute

CERBERUS: Computer Enabled Robotic Base Enhancing Remote Unmanned Security

Marissa Bennett (RBE)

Kenneth Quartuccio (RBE)

Jeffrey Tolbert (RBE/ECE)

MQP-KZS-CRBS

Advised by:

Professor Kenneth Stafford (ME/RBE)

Professor Alexander Wyglinski (ECE)

April 26, 2018

Abstract

The Computer Enabled Robotic Base Enhancing Remote Unmanned Security (CERBERUS) is a semi-autonomous sentry robot for deployment to remote unmanned Air Force installations. The project's goal is to fufill the Air Force's need for quicker responses to remote installations while also removing the need to put humans in danger to investigate possible intrusions. The platform is based around the Action Track Chair provided by the Air Force with a control system designed by the Worcester Polytechnic Institute (WPI) student team.

Acknowledgments

The material is based on research sponsored by Air Force Research Laboratory under agreement number FA8650-09-2-7929. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on.

Disclaimer

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsement, either expressed or implied, of Air Force Research Laboratory, the U.S. Government, or OAI.

Table of Content

List of Figures

List of Tables

Executive Summary

War is an interesting concept. While all animals participate in the fight for survival, humans are the only species that takes it to such a massive scale that it could be considered warfare. As a exclusively human construct, a war is a long term battle waged between different factions due to a conflict of interest. Wars are fought for numerous reasons, but most of them are not usually important enough to justify the cost of life involved. Wars fought over religion or differences in belief are probably the worst offenders, and while wars over territory might make sense in a nationalist sense, better communication between people would most likely solve 90 percent of disputes without any need for violence. The underlying truth is a large portion of the global economy revolves around the militaries of the world and their assets. And as warfare drifts further from human combat and more towards drone and robotic engagements, power is less about the number of people and more about how effectively they are used.

Countries like the United States have military assets scattered across the planet, ranging from ammo depots to full nuclear missile installations. There are over 800 actively maintained bases, and thousands of other facilities that are active but unmanned. Protection of these assets is paramount to this countries military strength, but the sheer scale of that makes it almost impossible. There is a need for a base defense system that is versatile enough to be deployed anywhere in the world, advanced enough to respond to threats autonomously, and modular enough to adapt to the constantly changing conditions of war. Such a system would ideally have minimal human interaction, so that one person could monitor several installations remotely. In the event that an alert was triggered, the robotic sentry could autonomously get into an appropriate position and provide video feed and other information to the security crew. It also should be tele-operated, so that it can do mobile reconnaissance. When it is no longer needed, it should be able to return to shelter autonomously, to minimize human interaction.

The goal of this project was to design the platform described above; A security system for unmanned bases that is both global and intelligent. The focus was not to create a final design, but rather a concept to prove that the idea itself is feasible.

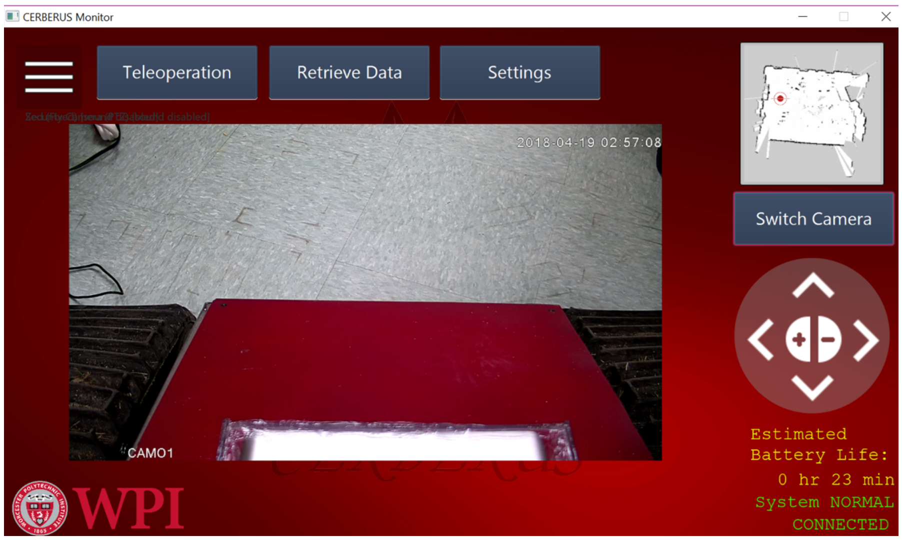


Figure 1: User interface with security camera feed displayed

1 Introduction

Recently robotics has been expanding into the security industry. Robots can remove the risk to military personnel and reduce costs which is why this move into security applications is driven by government funded research [15] [18]. The percentage of the countries GDP going to defense spending has been decreasing since the Korean war [19][20] which has been driving new research to focus on cost saving applications over traditional military applications. One area where research is starting to pick up is the use of security robots that can patrol areas and react to alarms.

Within the military security robot space there has been several indoor security robots [8]. The U.S. Army has also created outdoor security robots like the MDARS platform which is built on a modified Humvee [11]. This platform however has many limitations primarily due to its size. The Air Force Research Laboratory (AFRL) is interested in using a compact robot platform to respond to alarms at sites like underground missile installations. This would also cut down response times for alarms. In 2014 the Air Force failed to recapture one of these missile installations by seconds during a training exercise demonstrating the importance of quicker response time [13]. Currently the AFRL is just starting to do proof of concept research.

This project focuses on responding to the AFRL's need for a small all terrain robotics platform for these types of installations. Over the 2017-2018 school year this Major Qualifying Project (MQP) team will adapt an Action trackbot platform for this application. The team will also develop the control system that will be able to autonomously navigate around the installation and wait for remote operations. The project will ultimately create a deliverable platform for the AFRL and demonstrate its capabilities in May of 2018 at the AFRL facility in Ohio. This will serve as a proof of concept to for the AFRL so they can decide whether to fund future research in this area.

2 Overview and Mission Scenario

To formulate the project and platform requirements the team needed to better understand the state of the art and the actual challenge being presented by the Air Force Research Laboratory (AFRL). The team research the Action Track Chair being provided, similar operator interface, existing security robots, autonomous robot docking, various positioning systems, stereo vision, and the specific need of the Air Force for such a platform. By researching these topics the team was able to create a better set of requirements that will help guide the project to deliver a better product to the Air Force who is the client for this project.

2.1 Anticipated Mission Scenario

A typical job for the CERBERUS platform would be to respond to perimeter alerts at a ICBM missile silo like the Delta 9 missile silo (see Figures 2 and 3). While the specific type of site is not specified from the AFRL this missile site would qualify as underground storage at an unmanned location that has a perimeter less than 1200 ft (880 ft for the Delta 9 site). Using the Delta 9 site as an example is a good way to plan mission specifics.

In a typical mission the CERBERUS platform is woken up from a standby mode from a perimeter alarm on the site. After it wakes up it disconnects itself from the charger in its base station. Then the platform would travel to the area where the alarm was triggered within 2 minutes and wait for commands from an off site user. During this time an off site Air Force Security Force (AFSF) member would be getting notified of the alarm. They would then go check the base's cameras. After checking base cameras the robot would be in position for the AFSF member to take remote control of the CERBERUS platform. At this point the AFSF member can use the on board cameras of the CERBERUS platform to conduct an inspection of the area to assess the situation at the site. Based on this assessment the AFSF personnel will determine if military personnel should be sent to the site to either secure it from a trespasser or carry out the empty quiver scenario.[13] The empty quiver scenario is when a nuclear missile is lost or stolen.

In the past this mission would involve sending AFSF personnel to scout out the site in person. This not only wastes manpower due to common incidents like animals triggering alarms but also significantly delays response time in the event the site is actually under attack or control by adversaries. A failed recapture training exercise shows the difference this platform could make because the recapture failed by only a few seconds[13]. This is what the CERBERUS platform is designed to prevent.

Figure 2: Delta 9 missile silo site

2.2 Existing Robots

The US Army in collaboration with the Air Force research laboratory (AFRL) worked on a project to develop a Mobile Detection and Assessment System (MDARS) for use at army bases. This platform started in 1991 and has gone through many iterations and is still used today [11]. The platform is a small vehicle which drives on base roads and can use a variety of sensors to autonomously detect threats. There are many things to be learned from MDARS because of its extensive use in the field.

One of the first aspects of MDARS that was explored was how they used sensors to navigate. The most recent MDARS platform implements many sensors, including LiDAR, stereo vision, differential Global Positioning System (DGPS), wheel encoders, and laser range finders [11]. The DGPS is the only absolute position sensor on the robot for positioning. This helps confirm the original CERBERUS design because the field-tested MDARS also used stereo vision and DGPS for navigation. Another aspect of MDARS that was researched was the user interface provided to end users. The MDARS operator station originally used joysticks for control of the vehicle but eventually moved over to

Figure 3: Delta 9 missile silo site top down view (275 by 165 ft)

using Xbox controllers. They used a very simple two screen station in which one screen had various camera views and the other screen had relevant readouts about the vehicle's operation.

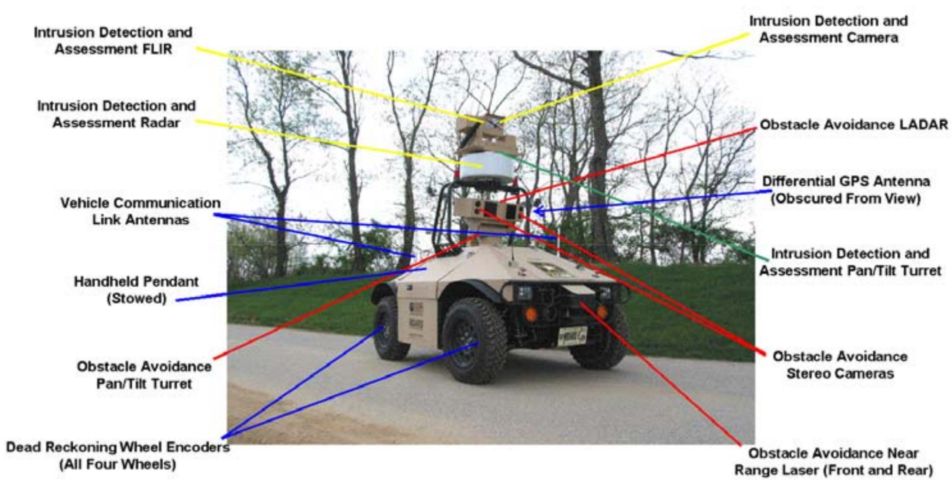


Figure 4: MDAR Vehicle with labels

Space Robotics Challenge (SRC) is one of four projects presented by NASA in their Centennial Challenges. The goal is to simulate and assist with an R5 robot in the procedure of a NASA mission[7]. The robots must be able to align a communications array, repair a broken solar array, and identify and repair a habitat leak. This is relevant because many aspects of these challenges apply to the Air Force Research Lab student challenge. The robots built for the Centennial Challenge can vary wildly, but are all designed to be durable and versatile.

The purpose of the Centennial Challenges are to attract the public to the process of advanced technology development, and are focused on drawing diverse people from non-traditional sources[12]. The program does not provide government funding, though offers incentive prizes to independent inventors, including student groups. This challenge is comparable to the AFRL Student Challenges, and though it is limited to only student groups, the purpose is very similar. The AFRL is in need of a growing skilled workforce, and so the Student Challenges provide students with opportunities for capstone projects with the AFRL, with solutions to their various projects, and with connections to a potential new wave of workforce. The comparable portions between the solutions to these challenges include autonomous mobile robots and path planning to a goal. A major difference is that this challenge does not limit the sensors we use.

2.3 Action Trackchair

The Action Trackchair being provided for this project includes two Duracell Ultra WKA12-100C/FR sealed lead acid (SLA) batteries. During communication with the Air Force Research Lab, it was mentioned that we were expected to use the provided batteries. These batteries are 6 cell (12v Nominal) and are wired in series making the nominal system voltage of 24 volts. Each battery is rated for 1200 Watt hours. A typical lead acid battery cell operates from 1.98 volts to 2.15 volts which makes the usable battery range of a six cell battery 11.89 volts to 12.9 volts (See Table 1 below) [10]. It is important to prevent overcharging in lead acid batteries as that can damage them. Both overcharging and fully draining these batteries will greatly reduce their lifespan. The WKA12 Duracell batteries are rated for 50-150 cycles at full discharge meaning that if deployed on a base regular maintenance will be required. The battery charger provided by Action Trackchair is a Pro Mariner Gen 3 Pro Sport 20 charger. This charger supports 20 Amp charging with two batteries [4]. This makes the typical charge time for the WKA12 batteries over five hours.

In order to maximize the lifespan of the batteries, it will be essential to control the battery charging and limit how much the batteries can discharge. The provided charger should allow the batteries to charge at the correct voltage, and the software will limit how drained the batteries by alerting the user when it drops below 20 percent of it's capacity.

Table 1: Contains important voltages for use when charging the WKA12 batteries. It is important not to over charge or over discharge the battery to maximize lifespan.

Operation Type	**Min Voltage**	**Max Voltage**	**Nominal Voltage**
Discharging	11.89	12.9	12
Charging	14.4	15	14.7

The drive train is one of the most important parts of any mobile robot because it is the base that the rest of the system is built on. The Action Track Chair was designed to accommodate a full grown human, providing freedom of motion in outdoor environments. As such, it is designed to navigate rugged terrain and harsh conditions. All of the weight is low slung, to prevent tipping, and the motors have an electromagnetic parking brake, to stop the platform from moving without input. All of these design considerations went into the modular platform that is provided for this challenge.

Building off the Trackchair platform, it would be ideal to continue using the methodology of Action Trackchair. It is important for the platform to maintain its low center of gravity because a low center of gravity will help keep the robot from tipping on inclines. The platform needs to be able to travel through snow and water, so keeping the electronics protected is also a high priority.

Table 2: This table shows the dimensions and weight of the Action Trackchair.

Weight	3̃15 lb
Height	22”
Width	32”
Length	39”
Clearance	8”

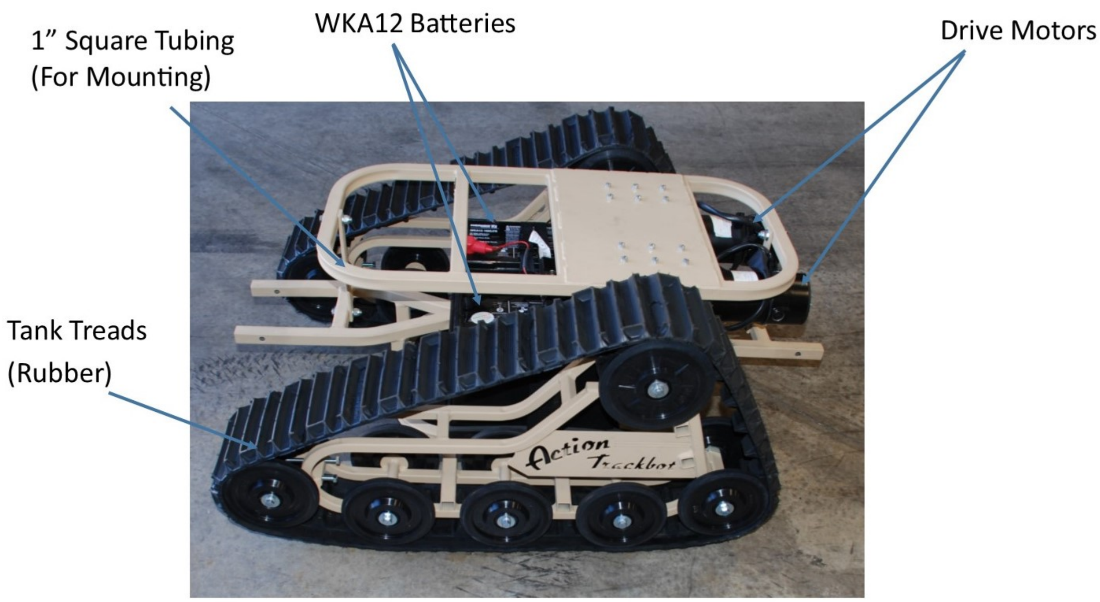


Figure 5: This is a labeled picture of the specific Action Trackchair platform provided to the team. It shows the location of the motors and the batteries.

2.4 Operator Interface

In moments where the robot may fail autonomously, or otherwise require human intervention, operators must be able to control and communicate with the system remotely. To do this, a user interface (UI) is necessary to convey information between the robot and the operator. This layer can include a video display, tele-operation, the health of the robot, status feedback, and anything else that may be relevant [9]. Figure 6 below shows a basic UI layout used for tele-operation of a robot.

Depending on the wants and needs of the operators, the interface may become more or less complicated as the one shown. MDARS (Mobile Detection Assessment Robotic Sentry), which will be touched upon more in the next section, uses four screens to show various camera angles. This is just another variation of forming a UI.

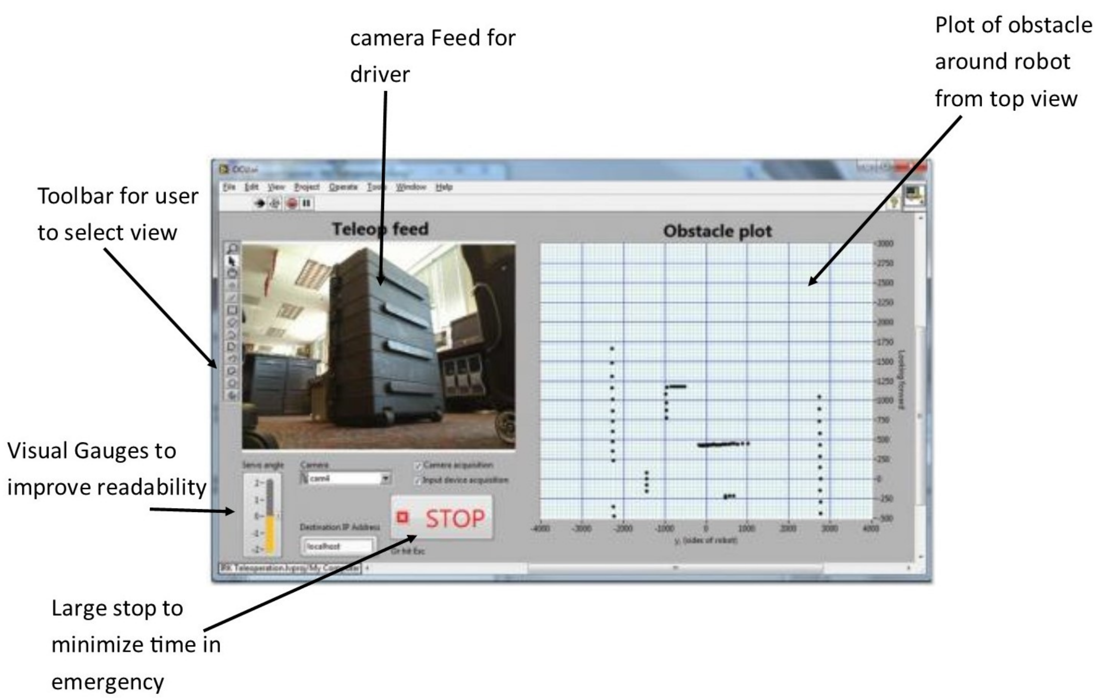


Figure 6: This is an example user interface for remote operation. It includes a top view showing nearby obstacles and a front camera view for the driver. Additionally there are tools for editing it and a large emergency stop button.

2.5 Robot Docking Stations

There are several different self-docking schemes used by robots today. Many of them use IR or radio beacons at the base station to align with the docking station. A Roomba, for example, has a charging base that emits a specific IR signal for each zone. The Roomba will look for these signals and based on what it receives, will adjust navigation in order to bring it closer to home. It knows it is properly lined up with the ground station when each sensor on the robot is receiving the right signal from the ground station.

Other docking schemes use passive docking stations with visual landmarks that the robot can see. The passive systems are cheaper, easier to deploy, and require no additional electronics on the docking station. A team from the University of Stellenbosch published a paper in 2015 discussing the implementation and testing of a QR code-based docking system [2]. In addition to the QR codes at the docking station their robot had a camera and two IR range finders. Their approach was to use the camera to approach the charger and use IR when they got within range. The algorithm for docking involved three distance zones: very close, close, and far. In the far zone they would just center on the QR code and drive toward it. When they were in the close zone they still used only the camera but they would not center on the QR code. They would turn to put themselves perpendicular to the QR code then move toward the docking station. Finally in the very close zone they used the IR range finders to check that their angle of attack was good and to perform the final docking. See Figure 7 below for diagram of this approach. The IR sensors were important in the very close region because the ability of the robot to read the QR codes becomes compromised.

This passive docking system has several advantages over an active system for the CERBERUS project. First is it keeps the docking station as simple as possible because only the battery charger and passive components are needed. Additionally the use of stereo cameras would allow the robot to see the QR code and get distance without the QR code thus removing the need for the IR range finders the University of Stellenbosch used.

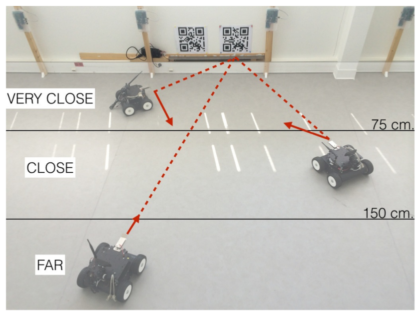


Figure 7: Different Docking Approach angles are all easy to compensate for using stereo camera's and IR sensors.

2.6 Positioning Systems

The Global Positioning System (GPS) is a geo-positioning system based on a network of at least 24 orbiting satellites (Figure 8)[14]. The satellites orbits the Earth at 12550 miles up, and completes an orbit twice per day.The satellites contain an atomic clock which is calibrated daily, because they run ahead of land based atomic clocks by 38 microseconds every day [6]. Each satellite transmits its Coarse/Acquisition (C/A) code as well as a Pseudo-Random sequence (PRN) [14]. Hand-held GPS devices are able to determine what satellites they are receiving C/A code from. The C/A code is used to quickly lock onto a GPS satellites signal and get basic navigational information from the satellite. After this Coarse Acquisition the receiver uses the PRN and timing to calculate the precise distance from the satellite. It takes four acquired satellites to have a full lock in GPS. This is because there are four variables to solve for: latitude, longitude, altitude, and time. Modern GPS Receivers get a typical accuracy of within 3-5 meters (9.8-16.4 ft) [5]. The main advantage of using GPS for positioning is that it doesn't require the user to transmit any data. Hand-held GPS receivers do not need to be connected to cellular data networks, and do not need to broadcast to the satellites themselves.

There are several different GPS receiver systems that use a nearby fixed point base to provide corrections to a roaming receiver. One of these methods is Real Time Kinematic (RTK) GPS, also known as Carrier-Phase GPS (CPGPS). This system relies on the GPS satellite's carrier signals primarily whereas traditional GPS uses the data being transmitted. The RTK system tries to determine the exact signal phase at both the base station and the roaming station. The distance between the base station and the roaming station(s) are then computed using the difference in phase. This corrects for most atmospheric effects if the base and roaming stations are near each other (Figure 9). Typical RTK systems vary in accuracy anywhere from +- 4 cm to millimeters in high end systems. The real limitation of RTK is that current systems are expensive. The lowest cost system available is the Emlid Reach RTK which cost $235 per module [3]. When you incorporate the cost of GPS antennas and the need for two stations as a minimum the base cost for this system is $570[3]. This is in contrast to a typical standalone GPS receiver which can cost as little as $40 [16].

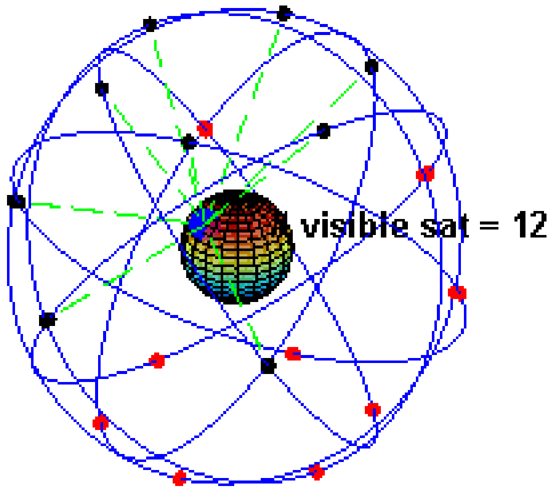


Figure 8: GPS Constellation showing how the satellites orbit to ensure at least 4 satellites are in view of the receiver at all times.

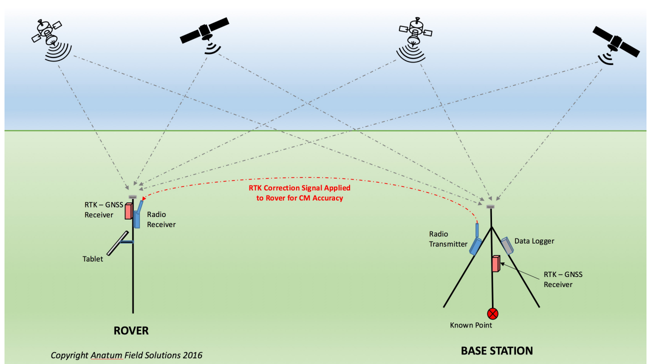


Figure 9: Typical RTK setup. This shows the base and receiver connecting to standard GPS satellites. It also shows in red the communication link between the base and the receiver.

2.7 Stereo Vision

With the exponential increase in graphics processing power in recent years, robots that operate with camera systems have become increasingly more commonplace. However, traditional cameras are very limited in how they process depth, which is where stereo cameras come in. With two cameras acting like human eyes, robots are able to build a depth of field. Using this, they are more easily able to isolate objects and determine distance. Graphics processing allows for accurate object recognition, such as knowing that a tree is a tree and being able to separate it from it's surrounding. Because the distance between cameras is fixed and known, a simple triangulation calculation will be able to determine the distance to the object.

Camera systems all contain different specifications, such as focal length, frames per second, and interface type that can all affect the quality of data produced by the unit. Cameras should ideally be wide field of view, have good nighttime capabilities, and work at a high enough fps for the robot to be able to respond in real time. Some camera systems, like the ZED camera, are designed for easy implementation and have libraries of code to help with accessibility.

While this method of computer vision is relatively straightforward, it is not without flaws. Human eyes are far more advanced than modern camera technology. Cameras don't work well in adverse lighting conditions, such as intense brightness or darkness. Cameras that are designed to work in one area might not work well in other conditions. Systems that utilize stereo vision often maintain strict control over the environment, or use other sensors to compliment the camera system. As graphics processing becomes more advanced and sensors are developed that are more accurate and versatile, more and more robots will start using stereo vision.

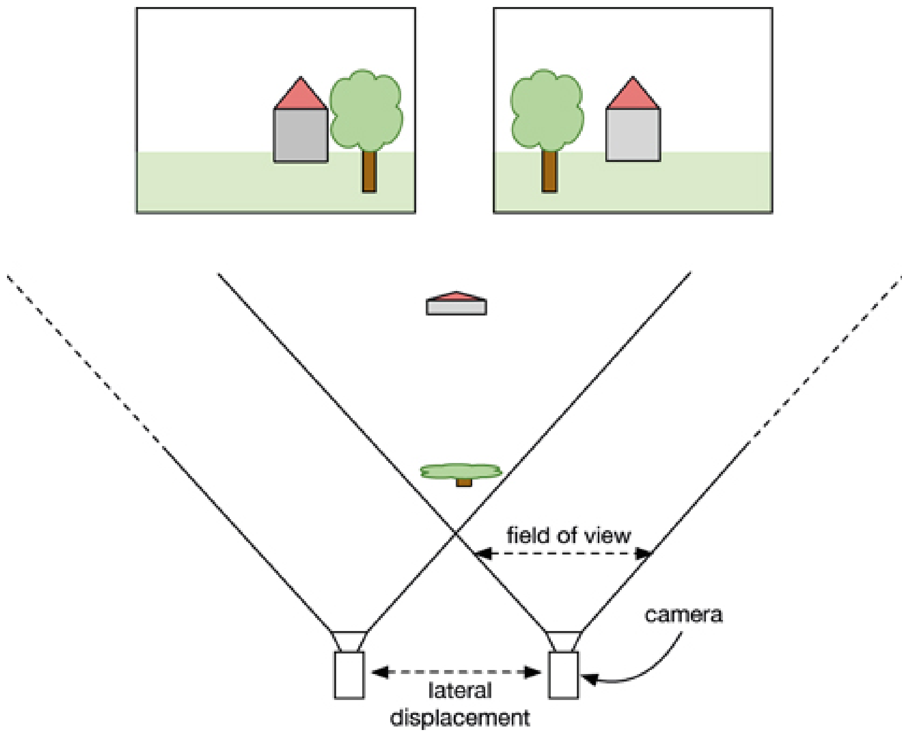


Figure 10: Simplified diagram of how Stereo cameras see. It calculates the overlapping region and given the lateral displacement, can compute distance.

3 Proposed effort

3.1 Project Requirements

Given the complexity of this project, it was necessary to layout the scope and specific requirements for assessing the projects success. Below are those requirements broken up into must haves, nice to haves, and reach goals.

Must Haves:

- Travel at least 600 ft in 10 minutes (1 ft/sec) in normal conditions
- Travel in 1 ft deep snow, through 1 ft deep puddle, and over gravel and grass
- Detect/avoid obstacles (Examples in detailed reqs. below)
- Stop in visual range of target
- Have all wireless communication encrypted with AES 256 or better
- Detect loss of reliable communication within 1 minute of loss
- Display remote video for use by operator investigating alert
- Have remote operation (Teleoperation)
- Have a way to remotely set target robot pose and start navigation
- Have at least 90°Field of View (FOV) in 1080p video for operator
- Keep all electronics within operating temperature limits
- Have a safe-charge charging station

Nice to Haves:

- Have a standalone communication system

- Use solid state electronics where possible to minimize maintenance
- Have 50 Mbps or greater bandwidth
- Have user interfaces that are cross platform for Windows, Linux, and Mac
- Be able to provide diagnostic data including battery level, signal strength, and critical sensor readings
- Have safe default behavior in event of communication failure

Reach Goals:

- Be able to autonomously return to its shelter
- Log all data received on remote operation system
- Log all sensor data on robot

3.1.1 Requirements Breakdown

For maneuvering requirements, the must haves are to travel at least 600 ft in 10 minutes (1 ft/sec, 0.68 mph) and to travel in 1 ft deep snow, through 1 ft deep puddle, and over gravel and grass. This speed requirement is set based on running the robot at 33 percent of its maximum speed, and this other requirement comes directly from the AFRL challenge document (Appendix A).

For Navigation/Localization Requirements, the must haves are to detect/avoid obstacles and to navigate to target location autonomously. While the AFRL challenge document (Appendix A) doesn't specifically require object avoidance, it is required to be an effective sentry and was part of the AFRL proposal (Appendix D) that was approved. Obstacles could include vehicles, barrels, boulders, and humans. The AFRL wants the robot to be waiting for future commands at the incident site within 2 minutes. This means the robot has to be able to navigate to arbitrary positions/poses within the compound autonomously. Other reach goals are autonomously returning to it's shelter. The AFRL challenge document (Appendix A) and AFRL proposal (Appendix D) make no mention of autonomously returning to base. The overall goal is to have a fully autonomous system so it would be nice if it could autonomously return to the base.

For communication requirements the must haves are all wireless communication encrypted with AES 256 or better, and detect loss of reliable communication within 1 minute of loss. Commercial standards for encrypted communication is currently AES 256. Military wireless communications tend to use higher levels of security but for a sentry robot in the field AES 256 is adequate. It is also important that the end user can detect that communication failure has happened so they can react. Additionally the robot is suppose to have a default response to communication failure so it needs to detect if the communications system is failing. Some nice to haves are 50 Mbps or greater bandwidth and to have a standalone communication system. This minimum bandwidth requirement was calculated using a 1080p video streams with H.264 encoding. It is also important that the communication system be stand alone because there is no guarantee of coverage for other more dependent networks like cellular.

For the operation station requirement the must have is to display remote video for use by operator investigating alert. The sentry robot is design to help an end user investigate incidents and decide whether to deploy personnel to a site or not. Displaying the video recorded by the robot allows the end user to do this task. The nice to haves are user interfaces that are cross platform for Windows, Linux, and Mac, and to provide diagnostic data including battery level, signal strength, and critical sensor readings. The AFRL uses many different platforms and having the operator software compatible with Linux, Windows, and Mac will allow the AFRL to run the robot from any machine they have available. As this robot will occasionally operate in harsh environments over long periods of time it is important that its status can be checked and problems diagnosed remotely. Reach goals are to log all data received on remote operation system. This is a security robot so all data sent back to the operator should be logged to review incidents later. Additionally the logs might need to be used in a prosecution in the event that an intruder is captured.

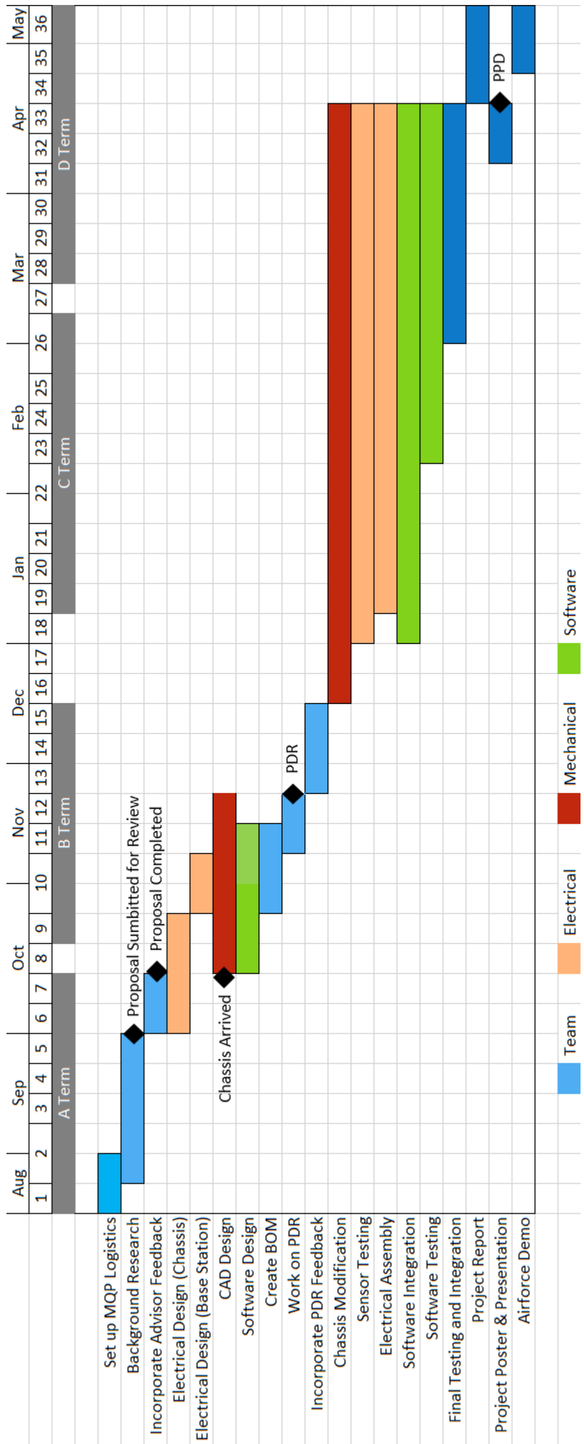
For software requirements the must haves are to have remote operation (Teleoperation), and to have a way to remotely set target robot pose and start navigation. This sentry robot is designed as a tool to help an off site operator decide if deployment of personnel is necessary. To do this the end user will need to be able to look around the area where the alarm was triggered. This mandates a remote operation mode. It is still unclear how this robot will be integrated into a bases alarm system. Making the navigation and target settable remotely it makes later integration into a complete security system easier. A nice to have is to Have safe default behavior in event of communication failure. Communication failure is always a possibility especially in a security application which is why it is important that the robot have a safe and reasonable response to such a situation. A reach goal is to log all sensor data on robot. Logging of all sensor data and software outputs will help with debugging possible problems that arise. This is especially important for problems that are hard to replicate.

For the robot electrical system requirements the must haves are to have at least 90°Field of View (FOV) 1080p video for operator, and to keep all electronics within operating temperature limits. This first requirement will make it easy for the end user to remotely operate the robot without crashing. Additionally this wide FOV helps with detecting threats. Proper thermal management is important in any electrical system but this robot will be operating in harsh environments with differing temperatures and it is important that it never under performs due to thermal limiting in components. A nice to have is to use solid state electronics were possible to minimize maintenance. The action track chair will have decent system vibrations so it is important that only solid state components are used to prevent unnecessary maintenance and to keep system reliability high.

For docking/charging station requirements the must have is to have a safe-charge charging station. The AFRL challenge document requires that the robot be stored on charge for a long time. Proper battery charging requires a safe-charge (a.k.a. trickle charge) application. Also having a charger that is easy to disconnect from comes from the AFRL challenge document (Appendix A).

3.2 Timeline

In order to stay on schedule, it is important to keep an active list of what needs to be done. Using a Gantt chart will allow the team to keep track of what needs to be done, as well as what is on schedule and what isn't. It also makes breaking up work much simpler. Below is our overall Gantt chart for the project. The numbers under each month represent weeks.



Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
A Term
B Term
C Term
D Term
Set up MQP Logistics
Background Research
Incorporate Advisor Feedback
Electrical Design (Chassis)
Electrical Design (Base Station)
CAD Design
Software Design
Create BOM
Work on PDR
Incorporate PDR Feedback
Chassis Modification
Sensor Testing
Electrical Assembly
Software Integration
Software Testing
Final Testing and Integration
Project Report
Project Poster & Presentation
Airforce Demo
Proposal Sumbitted for Review
Proposal Completed
Chassis Arrived
PDR
PPD
Team
Electrical
Mechanical
Software

3.3 Funding & Budget

The primary source of funding for this MQP is the $8,374 the AFRL is providing for this project. Additionally the AFRL is providing the $10,800 Action Track Chair. This external funding source significantly helps the project because there isn't a need to spend time fund-raising. This funding does however bring with it more team responsibilities such as periodic progress reports, two visits by the AFRL representative, and a final demo at Wright Patterson Air Force Base. In addition to the external AFRL funding there is $250 per team member provided by WPI and a matching $250 per team member that is expected from each team member in place of book costs.

Table 3 below shows the proposed budget broken up by categories. The Action Track Chair is the most expensive category at $10,800 but that is a fixed cost and paid by the AFRL. The computer and GPU allocation at $1,000. There is $1,700 allocated for sensors. A large portion of the project is adding controls to the robot so having accurate and reliable sensors is important. There is $1,400 allocated for the team to travel to Wright Patterson Air Force base to demo the robot at the end of the year. Finally there is a category for miscellaneous electrical costs ($500), a category for prototyping ($1,000), and a category for hardware($1,000). Shipping for the chair will be $300 which leaves $474 unallocated that can be used as needed later in the project. This makes the total budget for this project $20,674.

Table 3: Proposed Budget

Category	**Budget**
Computer w/GPU	$1,000
Communication	$1,000
Sensors	$2,450
Misc. Electronics	$500
Prototyping	$1,000
Hardware	$1,000
Action Track Chair	$10,800
Travel to AFRL	$1,400
Chair Shipping	$300
Unallocated	$474
Total	$19,174

3.4 Team Roles

The team composes of three WPI students: Jeffrey Tolbert, Marissa Bennett, and Ken Quartuccio. Table 4 below shows what major(s) each student is and what their primary focus on the project is. All students will work on mechanical, electrical, and software aspects of the project because of the small team size and because all students are Robotics Engineering Majors.

Table 4: Team Roles

Team Member	**Major(s)**	**Primary Focus**
Jeffrey Tolbert	RBE & ECE	Electrical Systems, Communication, Sensing, Controls
Ken Quartuccio	RBE	Mechanical Systems, Navigation, Controls
Marissa Bennett	RBE	Operator Station, Navigation, Controls

4 Mechanical Design/Analysis

One of the biggest challenges to this project was making a mechanical system that was capable of handling all the stresses of military life and protect a delicate computer system. The teams solution was to redesign the chassis to make it easier to add weatherproofing as well as fit the electrical and computer systems more efficiently.

4.1 PDR Design

The preliminary design for CERBERUS involved modifying the track chair chassis to better accommodate the batteries and electronics, creating sensor mounts that were waterproof and stable, and protecting all of the hardware as much as possible. In response to feedback from the PDR, the team decided to redesign the plating system. The fenders were removed, and replaced with simple side covers. All plating is made out of 16 gauge steel. While the basic design of the computer case has remained the same, the proportions were changed from the original design to better suit the components that were being put in the case. The concept behind the electrical case is that all of the drive line components would be contained in something waterproof that doesn't need to be opened. While originally this involved a box on the back of the robot, it was later moved to the front in order to better utilize the space available and keep the center of gravity low. The unusual shape it took is the result of this move and the space that is available at the front of the chassis.

There has not been a major revision of the design for the ZED stereo camera mount. The camera mount that was designed hasn't been fully tested but performs well. In the PDR design, we specified a very different security camera option than what we ended up purchasing. The design of the mount has changed similarly.

After intensive testing, we determined that in order to better protect the Walabot unit, it would be better to relocate it to beneath the front support beam. Testing will need to be done in order to determine if this location will change the performance but the radar will be more secure.

4.2 Chassis Design

The final design for the chassis is very similar to the Preliminary design. The only major change is the removal of the barrel jack connector from the top of the robot and the addition of the security camera post. The barrel jack connector was removed from the design in favor of a much simpler and more rugged magnetic charging system. The security camera post was added in order to implement a much higher resolution panning and tilting night vision security camera.

4.3 Liquid Cooling

Using liquid cooling over traditional passive cooling was decided very early into the project. The choice was made not because of the advantages of liquid cooling, but rather the drawbacks of passive cooling. The platform is going to be exposed to temperatures ranging from -30C to 100C, and because it is waterproof, the electronics are all sealed in an airtight box. In lower temperatures this might be fine, but in warmer weather the platform is going to be almost impossible to cool passively. Any heat sinks that could be used to dissipate heat will also absorb heat from the environment. Using a liquid cooled loop allows isolation of the electrical system from the platforms thermal characteristics. Coolant has a much larger mass heat capacity than aluminum (4.1813 vs 0.897) which means the fluctuation of temperature will be much smaller. These smaller temperature changes will help protect the components from breaking due to expansion and contraction.

4.4 Mechanical Summary

Overall, the mechanical design was completed to satisfaction. The end result is a platform that is rugged enough to survive falls, weather, and the rigors of military life, while being delicate enough to protect the sensitive electronics on board. Because it is a prototype and proof of concept, several things could be improved that would serve to make the design better in some regard. However, the mechanical design meets all of the requirements and does not hinder the other elements of the CERBERUS platform.

5 Electrical Design/Analysis

The electrical design of the CERBERUS platform can be divided into three primary categories. The first category is the power electronics. This category includes the breakers, the E-stop, the motor controllers, and the DC-DC converter. The second category is the data path. This section includes anything that carries data on the robot such as the CAN bus, the UART communications, the USB communications, and the wireless communication. Finally the third category is sensing. This category includes interfacing sensors to ROS and testing the sensors performance limitations.

5.1 Initial PDR Design

The power electronics for the CERBERUS platform have three primary roles. The first role is protection which is why the design proposed at the PDR (see Figure 11) has two auto reset 30 amp beakers, a 1 amp fuse, a primary 120 amp breaker, and an emergency stop. The two auto reset breakers are each set at 30 Amps because they each protect one of the motors from shorting. The 30 Amp cutoff point was decided based on the motor curves and the expected draw according to Action Trackchair. The 1 Amp fuse is for the POE injector which powers the Bullet Radio. There is also an emergency stop button wired to a contactor which allows anyone to stop the robot in an emergency. Finally there is a 120 Amp master breaker that doesn't auto reset to protect from shorts.

The power electronics design also covers voltage conversion. The primary system power from the battery is a nominal 24 volts but most electronics need highly stable voltage rails at lower voltages. The proposed solution is an HD Plex 400 DC-DC ATX PSU. This DC-DC converter creates a 12v rail, a 5v rail, a and a 3.3v rail. The reason a DC-DC converter is used instead of a linear regulator is the efficiency is much higher limiting heat generation in the computer enclosure. See figure 11 to see the voltages that every device needs on the platform.

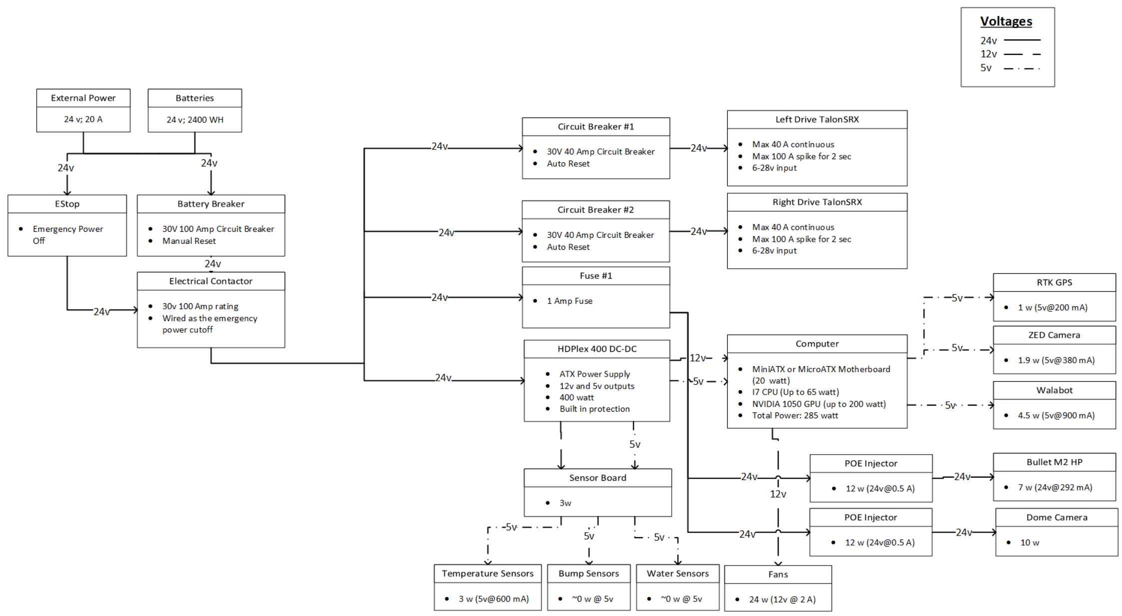


Figure 11: The PDR power Design Diagram

The data path design (figure 12) shows how all the devices communicate information to each other. The computer communicates to the primary sensors via USB and communicates to the lower sensors through the Hero development board. The Hero development board communicates via CAN Bus to the motor controllers. CAN is used because it allows for a clean two wire solution that is noise immune. The Hero development board also uses other less complicated communication protocols(I2C, DIO) to talk to the other low level sensors.

The other part of the data path design is the wireless communications that allow the robot to communicate to the remote operator. Given the bandwidth requirement (see section 3.1) there was a very limited number of options. It was quickly decided that the platform would use 802.11 B/G/N devices. While researching various communication systems employed by similar robot platforms, the Bullet Radio was decided on as the primary radio on the robot. This decision was heavily influenced by the WPI WALRUS MQP which also used the Bullet radio successfully on an outdoor all terrain robot. As part of the design process a Link Budget was calculated and an expected bandwidth was also calculated.

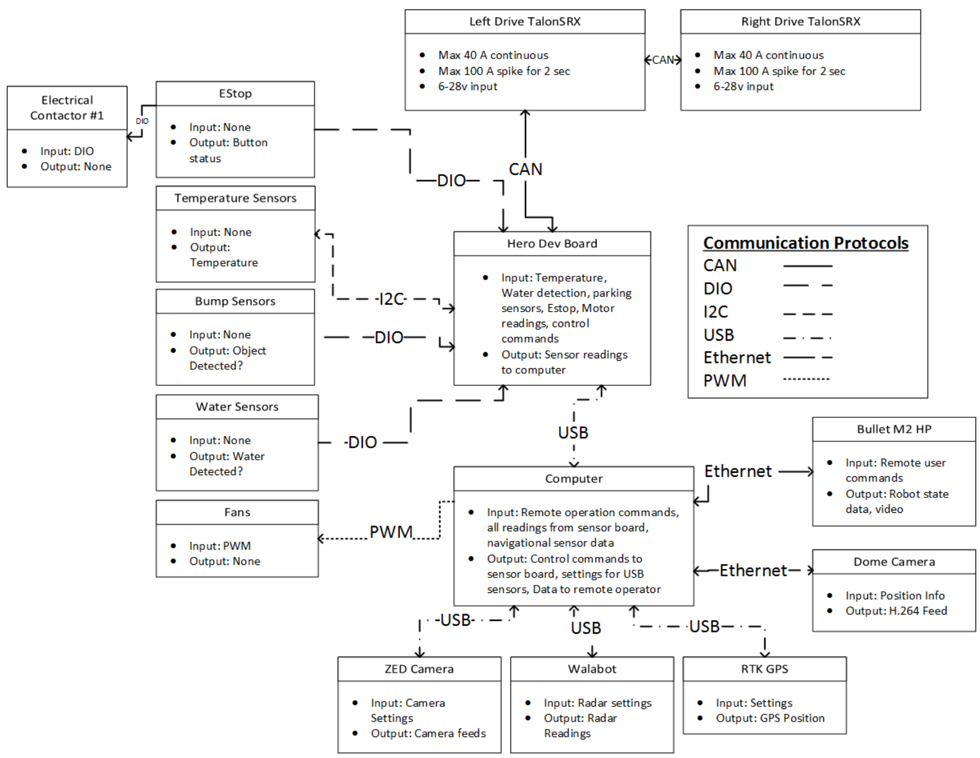


Figure 12: The PDR data path design diagram

The CERBERUS platform is primarily a navigational robot which makes localization and mapping two of its primary roles. As seen in figure 12 above there is an RTK GPS module being used for localization. The decision to use the RTK GPS was based on the positioning system research (section 2.6. The RTK GPS module chosen for the design is an Emilid REACH RTK GPS module. The mapping part of navigation is critical because it is what allows our robot to detect and avoid obstacles.

The proposed sensor scheme for object detection is designed to be incorporate multiple sensors for increased robustness. This was achieved by using a stereo camera (ZED Camera) and a phased array radar (Walabot). The goal is that in the event of visual obstructions (heavy snow, heavy rain, etc.) there is a backup system that can still detect objects enough to avoid them.

5.2 Final Power Design/Analysis

After creating the initial power electronics design for the PDR, there was only a handful of other decisions needed during implementation. The primary implementation level decision are based around which connectors to use and what wire gauges to use for various components on the robot. After looking up the current ratings. Table 5.2 below shows the current ratings for the three primary wire gauges on the robot. Everything between the battery and the power distribution bars uses 6 AWG wire. After the power distribution anything on a 30 Amp breaker is using 12 AWG wire and anything on a 1 amp fuse uses the 18 AWG wire.

After implementing and testing the power electronics on the robot during remote operations there is only one outstanding concern related to the power electronics. This is that the electrical contactor gets warm when it is in use. After reviewing the datasheet the reason for the warmth became clear. The coil used to hold the switch in place draws approx. 600 mA when powered at 24v. This means it draws 14.4 watts of power which it is dissipating as heat. Ultimately this was determined to be a non-issue as the contactor is in the primary electrical box, away from highly temperature sensitive electronics. Also the contactor isn't approaching it's maximum operating temperature.

The other significant redesign for the final product was the integration of the

Table 5: AWG Wire Current Ratings

AWG Gauge	**Amps**
6	101
12	41
18	16

charger. The metal contact plates used for charging created a need to protect shorting the batteries out. To protect the batteries a simple ideal diode approximation circuit was created using a P channel MOSFET, a zener diode, and a resistor. The circuit was tested in NI Multisim and after it was implemented was tested again using a DC power supply before it was installed on the robot.

5.3 Data Path Design Implementation

The primary issues related to realization of the data path had to do with the Hero Development board and interfacing it with the computer. The original design had the development board interfacing with the computer over USB but after starting implementation it became clear that writing the library for USB communication was going to be a lot more development time than originally planned. This lead to the decision to change the communications to UART (Serial communications). While sending and receiving raw bytes was easy because of the provided C# library, there was other challenges that this introduced. Two of the primary challenges include frame synchronization, and frame validation.

The first problem with using UART is frame synchronization. Basically, the UART receives a stream of individual bytes but because you don't know when the transmitter started the frame, there is no way to know what bytes relate to what information being sent. The frame synchronization technique that was used to overcome this challenge is called constant overhead byte stuffing(COBS) encoding. COBS encoding entails replacing payload data that is 0x00 with a count to the next zero or the end of frame. By replacing all the payload zeros with non-zero numbers, the frame can use 0x00 as an end of frame byte. This allows the receiver to select data between two 0x00 in the input buffer as aligned frame data. This allows the raw stream to map to individual frames but the problem of frame validation still remains.

The frame validation problem that UART introduces is a result of bit errors. These bit errors occur very rarely but on a military robot it is important to have reliable control messages going to the lower level components and the motors. An 8 bit Cyclic Redundancy Check (CRC-8) is used to check each frame to ensure all bits are correct.

Setting up and using the Bullet radios was easy but the overall bandwidth and range still had to be tested before it was installed in the robot. Table 5.3 below shows the measured values at 600ft and 300 ft line of sight(LOS) locations. The expected SNR is calculated based on the free space path loss model and the link budget. The $\tilde{3}$ db offset is probably a result of the coax cable added to the design after the initial link budget calculations. This proved that the Bullet M2 radios would work for this application.

Table 6: Bullet Radio test data

Range (ft)	**SNR (db)**	**Expected SNR(db)**	**Bandwidth (Mbps)**
600	36	39.71	24.2
300	42	45.73	43.7

5.4 Sensor Implementation/Testing

The sensor scheme includes only three primary sensors(ZED, Walabot, RTK GPS) but has many implementation challenges. The Walabot required creation of a new ROS node that could interface the radar with existing ROS messages. After the Walabot was connected to ROS there was also testing that was needed to determine its performance specs for the project. In contrast the ZED camera

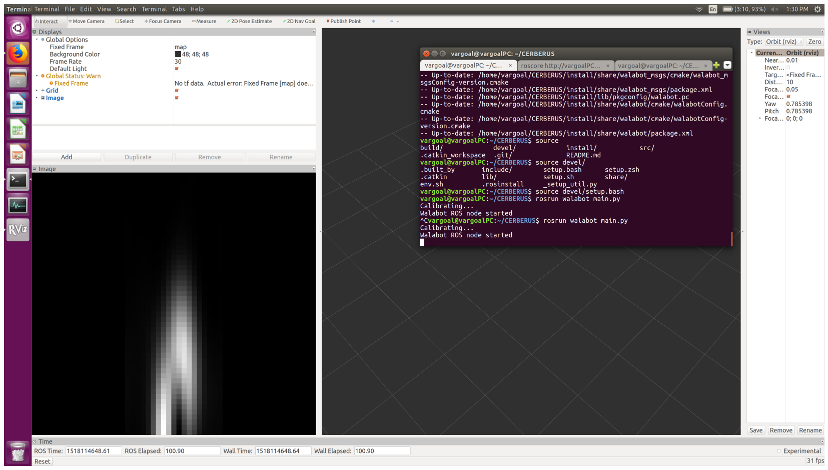


Figure 13: Running the RVIZ viewer and the Walabot ROS Node

came with a ROS node but testing was still required. The development of the RTK GPS involved both making a ROS node and testing.

The walabot ROS node is designed to read custom messages from the "/walabotSettings" ROS topic and publish image messages to another ROS topic. A test of the node can be seen in the figure 13. In the figure you can see the launched ROS node on the right and the image of the data on the left in RVIZ. RVIZ is a data visualization tool for viewing ROS message data.

The range of the walabot radar was a key performance metric that needed to be tested before the device was added to the robot. The range was tested against different targets (see table 5.4). These test results prove that the radar will work for detecting the primary obstacles set in the requirement section (Human, car, boulder, barrel). Additionally it defines the limit of the radar, especially against non metallic objects like the PVC piping. Another limitation is detecting targets when there is a lot of clutter which is why the building range was also limited.

Table 7: Walabot Range test results

Target	Max Range (m)
Human	5
Tree	3
Metal Fence	2
Bar Fence	4.5
Building	2
PVC Piping	2
Steel Barrel	5
Boulder	5

The Emilid RTK module connects to the rover PC using UART. The corrections from the base station are sent using TCP to the rover PC which sends them to the Serial comm port. After this was tested, the actual location data coming from the module had to be parsed. The messages are NMEA formatted so an existing python library was used. The values are then converted to a ROS message and published so the ROS navigation stack can use them for robot localization.

5.5 Battery Estimation

One important feature of the robot is battery life estimation. As part of the MQP, lead acid battery capacity models were research and simulated. Additionally sensing of voltage and current draw was added to the robot using the Talon SRX. Below is a summary of some of the research used in the battery model and a discussion about the state of implementation at the end of the project.

The first battery model that was considered was the one in Kenneth Stafford's paper titled *Electric Vehicle Simulation for Design Optimization* [17]. It requires battery mass, ambient temperature, discharge level, load current, nominal voltage, and loading history. The robot however has no way of measuring the battery mass. A less complex model was required so further research was conducted which lead to the IEEE explore article titled *An Improved Lead–Acid Battery Pack Model for Use in Power Simulations of Electric Vehicles* [1]. That technique discussed in the paper only requires five parameters: the relationship between state of charge and open circuit voltage, the battery resistance, the voltage drop coefficient, and an exponential term for battery recovery. This model is much simpler but had a 3% error when they tested it in the paper. Unfortunately due to time constraints in D term, the parameters were not able to be calculated from experimentation so the battery estimation was never implemented. The robot does have the battery voltage which the driver can look at but it is not as useful as a capacity estimate.

In summary the electrical design of the CERBERUS platform implements basic protection and power control using the primary electrical box while leaving communication and the DC-DC conversion in the computer case electronics. Additionally analysis and testing has shown that the wireless communication setup and sensing work.

6 Software Design/Analysis

6.1 PDR Design

During our PDR, a proposed an initial user interface and ROS architecture plan was displayed. These were critiqued by the advisors as well as the contact at the AFRL to create a professional final product.

The proposed user interface was created with Microsoft paint in order to provide a simple and quick visual representation of the different components. This helped to demonstrate a potential layout of the controls and features that could be implemented. Below in the initial image of our UI, we separated it into four main sections. The camera views, a ROS map, controls, and the status of the robot.

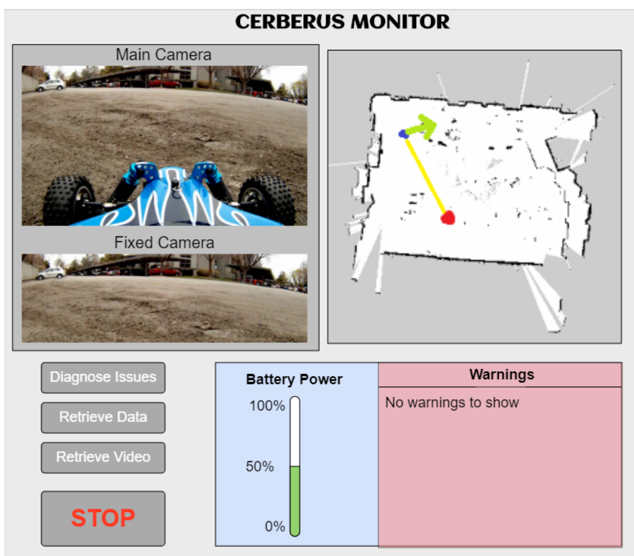


Figure 14: Original concept for our UI

The cameras would be used to see around the robot, and for teleoperating the robot. The ROS map would be able to visually show the user the path of the robot, including the direction it will point in, and any obstacles that it may encounter. The controls included an electronic emergency stop button, diagnose issues button, and buttons to retrieve data and video. Finally, the robot status area pictured the remaining battery life in percentages, and any warnings that may occur on the system.

For the proposed ROS architecture, we designed a flow chart to graph out where the nodes connect, what items they will send and receive, and what nodes we need to create ourselves. This flowchart can be seen below.

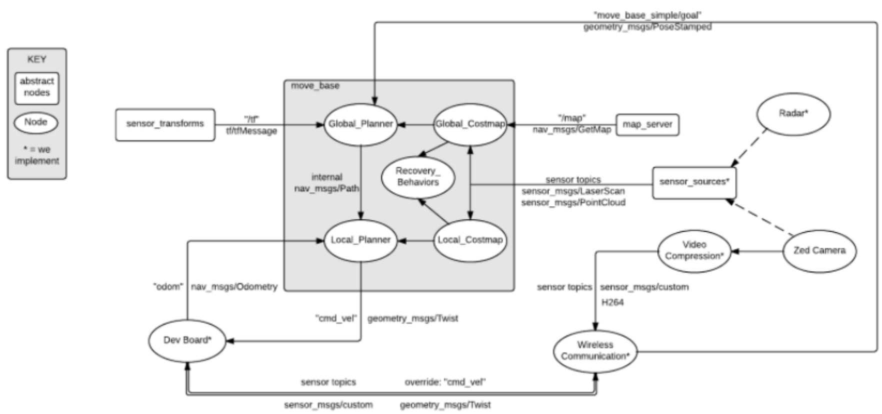


Figure 15: Flowchart of nodes in our ROS architecture for the robot and its shelter

The move base section of the chart is from the navigation stack in ROS. Most of the nodes exist in packages, but there are four nodes in which will need to be implemented. These nodes are the Dev Board, Wireless Communication, Video Compression, and Radar nodes. Starting at the wireless communications node, it connects the robot to the user at a remote location through the robot's shelter on the base. From there, our development board will send and receive commands and information to and from the wireless communications node to control all processes on the robot. The video compression and radar nodes interact with the Zed Camera and radar sensors on the robot, respectively. With this flowchart, it encompasses the design of our remote systems, the robot and its shelter, and the connections between them.

6.2 UI Development

As the first iteration was meant to get an idea of what was possible for a design, feedback and criticism was expected. This iteration received many needed improvements. These included having a way to teleoperate the robot and cameras, having a larger screen for the camera views, a smaller map, and accurately calculating the battery life to show on the screen. We also needed to create the application at this stage. For the second UI iteration, it incorporated most of the suggestions given. Additionally, the image was created into a functioning JavaFX application. This transition, along with having obtained feedback from the advisors and the AFRL contact, influenced a new design of the UI. Many of the old elements and features stayed in the UI design, however unnecessary elements, like two retrieve buttons, and a bar for the battery consumption were removed. The teleoperation button was added, which creates a new window to allow the user to start and stop the teleop thread for receiving signals from a connected USB gamepad. No other buttons on the gamepad functioned at this time.

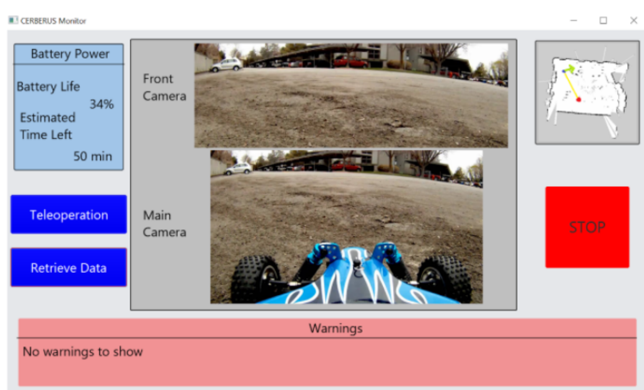


Figure 16: The second iteration of the UI

Transitioning from to the third UI iteration created a large jump in the layout and design compared to the others. This can be seen in the image of the third iteration below. The screen size was enlarged, and overall the UI was made more visually appealing. We also added the team's logo and WPI's logo to the screen as per advice from the advisors.

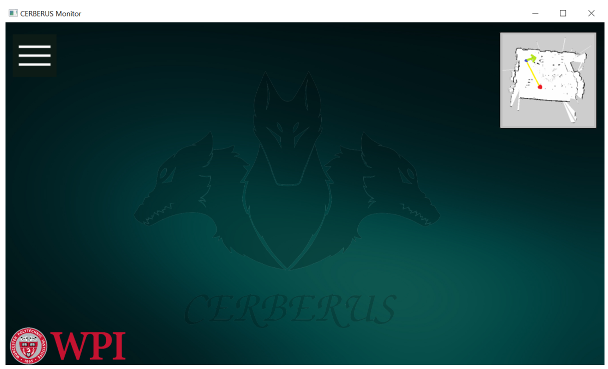


Figure 17: The third iteration of the UI

In the next image, the menu bar is fully expanded, showing the three buttons available to the user. The map in the corner is also a click-able entity for using autonomous functions. The Teleoperation and Retrieve Data buttons have the same functionality as from before. The settings button creates a pop-up window with options to change the theme and the IP address at this stage.

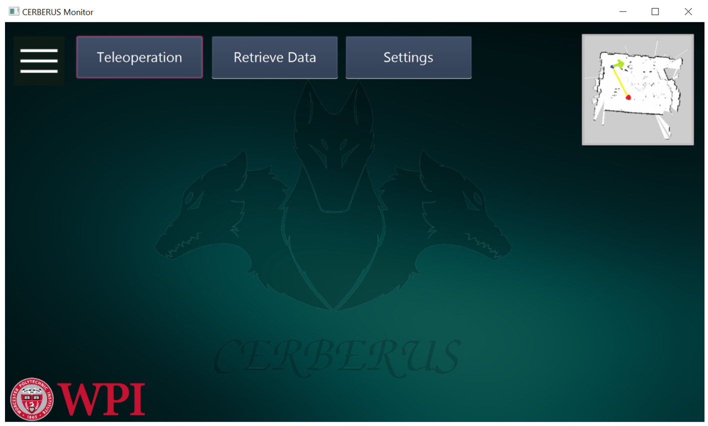


Figure 18: The third iteration of the UI

In the final UI, final additions to the functionality are added. The security camera feed is shown, as well as camera controls, a button to switch camera views, and system diagnostics in the bottom corner. In this view, the UI uses the red color theme.

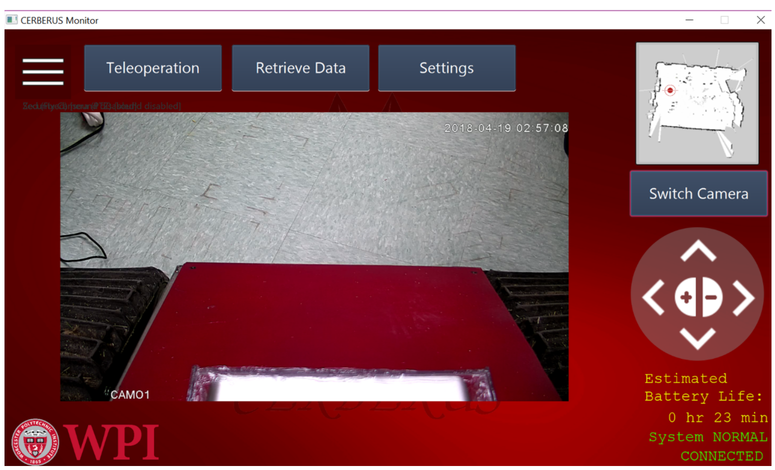


Figure 19: The forth iteration of the UI

By creating a user interface from scratch allows for a lot of custom design choices and features. As not every single task could be completed for the UI, some future development could include an advanced warning system, receiving more than just the log file from the robot, or a better way to estimate the battery life of the robot.

6.3 Server Development

The server code is run on the base station in the system. This program is written in Python 3, and incorporates the use of sockets and serial ports (UART). It will stay persistent even if the end user closes out of the UI, meaning it will continuously run the server unless it is manually shut off electrically. The UI (client) connects to the server on runtime. The server code communicates back and forth between itself and the client sending messages for the state of the HERO and itself, and receiving messages for teleoperation.

6.4 HERO Board Development

The HERO code communicates between the server and the Talon SRX motor controllers. It creates a feed over UART from the server in order to receive messages from the gamepad on the UI to then convert and send messages to the Talon SRX's for changing the speed of the motors. It also sends status updates to the server to be sent to the client to update parts of the UI. This program was written in C# and referencing the Cross the Road Electronics (CTRE) sample programs.

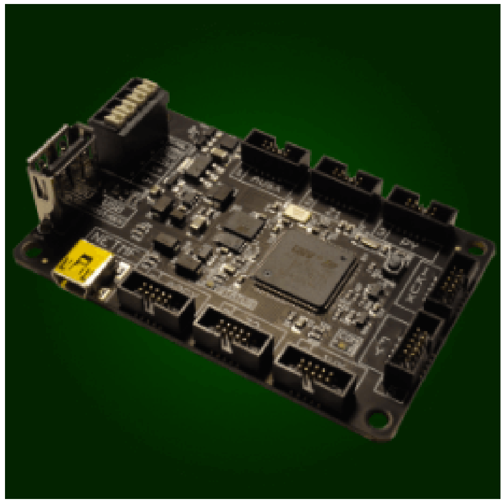


Figure 20: HERO Development Board by Cross the Road Electronics

6.5 Chapter Summary

The software development for CERBERUS has had great progress as everything was created from scratch. The user interface has all the features that had been asked to be implemented, and the design looks sleek and commanding. All programs including the code for the server and HERO board are standalone products.

7 Performance Evaluation

This section goes through all project requirements previously stated and reflects on their completion. The following requirements are essential to the operation of the CERBERUS platform. They detail the main functionality required to meet the mission goals.

7.1 Must Haves

Be able to travel at least 1 ft/sec

In order to test the movement speed of the robot, we marked a start and end point 50 ft apart. At full speed, the robot was able to travel this distance in 20 seconds. This would mean that our maximum movement speed is 2.5 ft/sec.

Travel in 1 foot of snow, through a 1 ft deep puddle, over grass, and gravel

Driving over grass and gravel were tested during our early tele-op testing. The platform's ability to drive over 1 foot of snow was demonstrated during a separate test. While moving the robot from the lab space to campus, the robot was driven through several puddles that were over 1 foot deep. It was also left out in torrential rain for over 6 hours, then booted up and driven back to the lab space after one of the demos.

Detect/avoid obstacles

The Walabot and ZED camera sensors were tested to ensure they could detect the required obstacles. Some integration into the ROS navigation stack was made.

Stop in visual range of target

The robot is able to stop in visual range of a target when driven by the user. The hardware and electrical components are in place and tested to allow this action in autonomous navigation.

Have all wireless communication encrypted with AES 256 or better

The Bullet M2 radios are currently configured to use WPA2 with AES 256 encryption.

Detect loss of reliable communication within 1 minute of loss

The detection of communication loss is virtually instantaneous. This is detected from the closing of a TCP socket that is opened between the driver station and the robot.

Display remote video for use by operator investigating alert

Video is streamed through bullet radios, and can be accessed using either the CERBERUS Command Software or VLC.

Have remote operation (Teleoperation)

Using the CERBERUS Command Software, it is possible to remotely drive the robot with an HID compliant gamepad, such as an XBOX controller.

Have a way to remotely set target robot pose and start navigation

The target pose is set in the driver station and sent to ROS. The connection to ROS needs to be established.

Have at least 90° Field of View (FOV) in 1080p video for operator

Both of the camera's on the platform support 1080P video streaming. The zed camera has a field of view of 105°, and the security camera has about 330°. The streaming of the cameras and the control of the pan-tilt were also tested and work.

Keep all electronics within operating temperature limits

All electronic systems were tested for 1 hour durations at maximum loads. No systems failed.

Have a safe-charge charging station

The base charger is designed to be safe and has inline fuses for protection. Additionally a protection circuit was designed, created, and tested that prevents the charger pads from being able to short the battery. The battery protection circuit acts as a diode but with a very low voltage drop to prevent unnecessary power loss when charging.

7.2 Nice to haves

Have a standalone communication system

The system is internally connected through wireless streams over the Bullet M2 radios. No outside networks are used while running CERBERUS.

Use solid state electronics where possible to minimize maintenance

All electronics used in CERBERUS are solid state or designed to work in high vibration environments.

Have 50 Mbps or greater bandwidth

Channel bandwidth was 150 Mbps. TCP Bandwidth to CERBERUS can be made across roughly 300 ft with 43.7 Mbps. Any distance closer than 300 ft increases the bandwidth. By 600 ft, bandwidth drops to about 24.2 Mbps.

Have user interfaces that are cross platform for Windows, Linux, and Mac

Any methods, classes, or files used in the creation of the UI are cross compatible on Windows, Linux and Mac.

Be able to provide diagnostic data including battery level, signal strength, and critical sensor readings

Data on all communication health, run time, and sensor voltage is recorded in log files or on the user interface.

Have safe default behavior in event of communication failure

In the case of communication failure, CERBERUS is trained to cease all movement.

7.3 Reach Goals

Be able to autonomously return to its shelter

Although CERBERUS can autonomously drive, there is currently no action for autonomous docking to charge.

Log all data received on remote operation system

All data received from CERBERUS is logged on a file on the remote user's computer.

Log all sensor data on robot

All data on CERBERUS is logged in a file on the remote system.

8 Future Work and Recommendations

While there is many aspects of this project that were complete, there is still room for improvement. Below is some of the suggested improvements for those seeking to continue this project in the future.

8.1 Mechanical Suggestions

The Action TrackChair platform that was provided by the Air Force Research Lab served as a useful starting point. However, as the project moves away from proof of concept and towards final product, there are several improvements that could be made.

1. Modify chassis to maximize usage of internal space on platform
2. Use Lithium Ion batteries in the place of sealed lead acid cells. Lithium batteries do not decay when deep cycled, have greater energy density, and can be stored with less discretion than lead acid cells. A custom designed battery pack could move the center of gravity much lower, and free up internal space for other electronics. If the batteries were designed with the chassis in mind, it would be possible to insulate them, and possibly incorporate them into the thermal loop.
3. Redesign computer case. Use a smaller form factor motherboard to reduce the size requirements of the case.
4. Redesign electrical box to allow for easy maintenance
5. Modify side plates to avoid friction with tread wheels
6. Add vibration damping
7. Finalize mounting of Walabot radar

8.2 Electrical Suggestions

While the overall state of the electronics allowed for basic operation of the platform, there are many ways this platforms electronics could work in the future.

1. Add additional sensors
2. Clean up wiring in computer case
3. Change the motor breakers to 40 Amps
4. Consider making wireless communication more robust to threats like jamming
5. Fully implement battery estimation(see electrical design section for algorithm)

8.3 Software Suggestions

Although most of the software has been established and thoroughly refined, there are still a few improvements that can be made to further enhance the overall system.

1. Finish ROS Navigation Stack Integration
2. Fix library issue with camera and teleoperation controls (If it still persists)
3. Add network security to connect base station to public facing internet
4. Finish methods to change between Walabot and ZED for object detection
5. Process more of the RTK NMEA messages. Currently only a few message types are processed.

References

[1] *An Improved Lead–Acid Battery Pack Model for Use in Power Simulations of Electric Vehicles*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6081918.

[2] *Docking autonomous robots in passive docks with Infrared sensors and QR codes*. https://hal.inria.fr/hal-01147332/document.

[3] *Emlid REACH RTK Kit*. https://emlid.com/shop/reach-rtk-kit/.

[4] *Gen 3 Pro Sport 20*. http://www.promariner.com/en/43020.

[5] *GPS Accuracy*. http://www.gps.gov/systems/gps/performance/accuracy/.

[6] *GPS and Relativity*. http://www.astronomy.ohio-state.edu/~pogge/Ast162/Unit5/gps.html.

[7] Jennifer Harbaugh. "NASA's Centennial Challenges: Space Robotics Challenge". In: *NASA* (Aug. 2017).

[8] *Indoor Surveillance Security Robot with a Self-Propelled Patrolling Vehicle*. https://www.hindawi.com/journals/jr/2011/197105/.

[9] Meghan Kerry. "Simplifying Robot Software Design Layer by Layer". In: *RTC Magazine* (Sept. 2011).

[10] *Lead Acid Battery Charging Basics and Chargers*. https://www.powerstream.com/SLA.htm.

[11] *Mobile Detection Assessment and Response Systems (MDARS) A Force Protection, Physical Security Operational Success*. http://www.dtic.mil/dtic/tr/fulltext/u2/a449408.pdf.

[12] Lee Mohon. "NASA's Centennial Challenges Overview". In: *NASA* (Aug. 2017).

[13] *Nuclear missile base security failed takeover drill.* https://www.cbsnews.com/news/nuclear-missile-base-security-failed-takeover-drill-ap-reports/.

[14] Ahmed El-Rabbany and Skillsoft Books EngineeringPro Collection. *Introduction to GPS: The Global Positioning System.* English. Norwood;Ipswich; Artech House, Incorporated, 2002;2006;2001; ISBN: 158053547X;9781580535472;1580531830;978158053183

[15] *Robotics Industry Insights.* https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Robotics-in-Security-and-Military-Applications/content_id/3112.

[16] *Sparkfun GPS Receiver.* https://www.sparkfun.com/products/13670.

[17] Kenneth Stafford. *Nuclear missile base security failed takeover drill.*

[18] *The next frontier: How robotics are changing the security industry.* https://tinyurl.com/y7nrtsaj.

[19] *United States Air Force FY2016 Proposal.* https://www.defense.gov/News/Special-Reports/FY16-Budget/.

[20] *United States Air Force FY2018 Overview.* https://tinyurl.com/y9avbwtl.

A AFRL Challenge Document

Robotic Sentry Using All-Terrain Wheelchair Platform

Background: The need for active security around facilities such as prisons, high value storage, missile silos, hazardous waste storage and political borders has created a demand for robots that can respond to breaches, alarms, or general inspection to reduce the workload on humans. A new generation of wheelchairs from Action Trackchair is designed for use in all terrains and offers a starting point to address a robust, robotic monitoring and response capability for these situations.

A New Design Concept: The Robotic Sentinel

Using the Action Trackchair (provided by the manufacturer) as a platform, develop a robotic sentinel that meets the security requirements of secure facilities, as indicated by the design scenario below. Design sensors, communications, control systems, programming, and systems management (power, maintenance, control) approaches that can be tested on the Trackchair as part of the project.

Scenario:
You've been asked to provide 24/7 security coverage of remote fenced in areas around an underground storage facility. The facility has two perimeter fences with a 10ft gap between them. Most of the surface between the fences is a 3ft sidewalk but there are areas that have gravel, grass and in some cases puddles up to 1ft deep. During winter months, snow drifts up to 1ft deep may be encountered. These facilities are square, 2 acre compounds which means you have a 1200ft perimeter to monitor.

There are dozens of these facilities so it is impossible to provide 24/7 human coverage of each site. You have stationary sensors and closed circuit TV so you know when there is a disturbance at each site. However, when there *is* an alarm you must quickly investigate the reason and if possible send a sentry to the breach within 2 minutes of an alarm to collect and provide more detailed information. The design goal is to design and build a robot that can do the job.

The robot shall be able to be stored in a standby condition in a small shelter that provides it with power, communication and shelter from the elements. Upon deployment, the sentry shall disconnect from the electrical connections, leave the shelter, proceed to a point designated before leaving the shelter via the shortest route, and position itself facing the area of interest to standby for further instruction. The approach to calculation and response to these requirements (i.e., real time onboard calculation vs. preprogrammed routes, obstacle avoidance, and location determination) will be part of the design process.

Skills Needed for This Work: (1) Basic understanding of robotics. (2) Control system theory for closed loop navigation. (3) Mechanical and Electrical design skills for mounting and constructing components. (4) Understanding of wireless audio/video data transmission and communications. (4) Test planning and testing skills to ensure capability and durability of a finished design.

Sponsor: James W. Poindexter, AFRL /RXMS, james.poindexter@us.af.mil, (937) 904- 4596

B End User - UI Questions

Answering these questions is completely voluntary. They reflect personal opinions and not the opinions of any governing body.

A robotic sentry is being used as 24/7 coverage of unmanned military installations. The security robot's job is to respond to potential threats by navigating itself to the location where the threat is occurring. At this point, a human operator (you) would take control of the situation remotely.

These questions pertain to the user interface portion you'll see of the robot on a standard commercial computer. The application will be used to communicate with the speaker, camera, and drive controls on the robot.

1. If an alert on the base occurs, how would you like to be notified? (Ex: computer shows a blinking notification and beeps/ *show that robot is responding to the alert)

2. Would a computer application that fills up the entire screen be prefered, or one that could be resized/is smaller and only covers a portion of the screen?

3. Would a pan, tilt, zoom camera be useful, or would you prefer a fixed camera? (Separate control)

4. What would your preference be for operating the robot? Joystick, controller, mouse,etc

5. Do you think you would prefer a more visual or textual layout to receive information?

6. Do you prefer to look at one video stream at a time, or would you rather have all of them open at the same time?

7. Is there anything else you might want in this application? If so, what?

C End User - AFRL Questions

1. Did you find out more about the chair?

Out in public with it, action track logo, BALL logo has to be visible
At least 1 picture with the group within 30 days and do a writeup, PR shit
We can get more if we wanted,
Weighs 2-300 pounds, 4x4 skid, shipping is usually $1000/6 chairs = $200-$300
Unsure how shipping is working

2. There is a place for progress reports on the AFRL student challenge website. How often do you want progress reports? Is the website your prefered way for us to submit them to you? Is there a format? Would you like gantt charts?

   Format of our choice.
   Like emailed progress reports.
   Monthly Call - Written every few months

3. What is the nature of the alert? (Type?)

Vague on purpose, we dont want to focus on that, come up with boundary condtions about optimal design. Inspect a part of the fence. Just a point on the map?

4. How close to the alert do we need to be able to get to it?
   Seems flexible

5. How long would a typical mission last?
   Minimize the time, want to try to get there within 2 minutes, depends on the nature of the alarm and how much data there is to collect, DONT THINK ABOUT IT

6. How often would the robot be responding to alerts?

7. How much maintenance would be acceptable? (once a year, every 6 months, every 3 months, etc.)

8. Would the robot need to make rounds on the base? If so, how often?

9. How will the base be maintained in the winter? (Would we need to account for snow blocking the entrance/exit to our shelter?)

10. When will the demo be at the AFRL? What does it consist of?

Obstacle course outside of Airforce base, not designed yet, maze, 90 degree turns and various obstacles, person people have teleoperated it in the past

Pictures coming soon

11. When will your two trips to WPI be?

    Whatever works for us, if at all, hahahaha. Near jan 1,

Discussion in next month or two about design

D AFRL Proposal

Computer-Enabled-Robotic-Base-Enhancing-Remote-Unmanned-Security (C.E.R.B.E.R.U.S)

Robotic Sentry Using All-Terrain Wheelchair Platform

Marissa Bennett , Kenneth Quartuccio, Jeffrey Tolbert

1. Proposed Design

In response to the Air Force's need for an autonomous sentry robot, we are proposing a stereo vision based sentry robot. Using stereo cameras doesn't require augmenting the robot's environment, providing an additional layer of security by not depending on external signals or beacons. Differential GPS will also be used to correct drift in the robot's position.

The track chair platform will be modified to better suit the challenge. An electronics housing will be added that will house the computer required to autonomously control the robot as well as additional electronic boards required to control the robot. This electronics box will also have shocks and dampeners to help stabilize the electronics within it to reduce board vibration and improve the robots reliability and reduce maintenance.

The software architecture chosen for the robot is the open source Robotics Operating System (ROS). This architecture provides much of the base functionality in thoroughly tested software packages. This will both reduce development times and create a more reliable product for the Air Force Research Lab (AFRL).

2. Timeline

Milestones

Project Begins - Aug 24, 2017
Hardware Modifications complete - Oct 12, 2017
Electrical Modifications complete - Oct 12, 2017
ROS setup with remote control working - Dec 15, 2017
ROS Navigation Stack with DGPS/Stereo Vision - March 2, 2018
Testing Complete - April 20, 2018
Documentation Complete - April 20, 2018
Project Ends(Project presentation Day) - April 20, 2018
AFRL Visit/Robot Demonstration - TBD

Deliverables

PDR Presentation
CDR Presentation
AFRL Final Report
MQP Paper
Modified Trackchair sentry robot
Remote Control and Monitoring UI

3. Team Background

Worcester Polytechnic Institute (WPI) started its Robotics Engineering program in 2007, it has maintained its position as a forefront in the robotics field. The university has participated in and won challenges ranging from NASA Centennial Challenges to DARPA's humanoid robotics competition. The team of students consists of three robotics engineering undergraduates, each with a specialization in either Mechanical Engineering, Electrical Engineering, or Computer Science. The primary adviser, Professor Stafford, is the director of the robotics resource center at WPI, a retired Air Force Colonel, and has successfully advised many senior design projects involving autonomous off-road vehicles for competitions such as the NASA RMC, RASC-AL Robo Ops, IGVC, Cornell Cup, and DARPA Grand Challenges. The co-adviser is Professor Wyglinsky who runs the wireless communication laboratory on campus and who has a lot of experience working with the military through his work with MITRE.

4. Budget Breakdown

The proposed total budget is $8,374 plus the provided Action Trackchair. This includes approximately $2,475 for the computer. The computer is the most expensive part of this project(excluding the trackchair) because it needs powerful graphics cards to process the stereo camera data and because it has to be watercooled to keep it weatherproof without overheating. The stereo cameras, web cameras, and differential GPS costs approximately $1,720. The electronics boards and motor controller cost approximately $1,745 total. The hardware required to adapt the trackchair is approximately $1000. Finally the cost to travel to Wright Patterson Airforce Base to visit the Air Force Research Lab (AFRL) is approximately $1,400.

item	cost	quantity	sub total
GPU(gtx 1080Ti with waterblock)	800	1	800
Processor	340	1	340
Motherboard (Z270 XPOWER)	300	1	300
RAM(32 gb)	237	1	237
1000 watt power supply	120	1	120
M.2 Drive (250 gb)	108	1	108
500 gb SSD	170	1	170
2000 watt inverter	160	1	160
Watercooling kit	400	1	400
Radio(BM5-Ti)	115	1	115
Base station radio	135	1	135
Robot antenna	100	1	100
Zed Cam	450	2	900
Webcams	60	2	120
Motor controllers (Talon SRX)	90	2	180
usb to CAN adapters	25	2	50
DGPS/RTK GPS supplies	700	1	700
PCB Fab(4x4in boards, set of 3, 2 layer)	80	3	240
Digikey Comps per board	50	3	150
Misc electronics	500	1	500
USB PCI card	30	3	90
USB Cables	2.5	10	25
45 lbs * 80mm Gas Strut	12	2	24
Vibration Dampener	15	2	30
Misc Hardware	100	1	100
16 Ga 4x8 Aluminum Sheet	100	2	200
0.22 Acrylic Sheet 2x4	60	1	60
1x1 1/8 aluminum square tube (6ft)	30	4	120
Misc Extra	500	1	500
Travel Expenses	1400	1	1400
		Total	8374
		Total w/t chair	23374