



Worcester Polytechnic Institute
Major Qualifying Project

UAVs for Search and Rescue Missions:
Optimal Path Planning and Triangulation of Signal Data

SUBMITTED BY:

Isaac Beeman
Sarah Elice
Griffen Spincken
Michael MacCormac

ADVISED BY:

Donald R. Brown III, Professor & Department Head
Matthew Amissah, Professor

Electrical and Computer Engineering

*A Major Qualifying Project submitted to the Faculty of WORCESTER POLYTECHNIC
INSTITUTE in partial fulfillment of the requirements for the degree of Bachelor of Science.*

March 25, 2020

Abstract

The search for a missing person in a disaster is time critical, and the efficiency of the search can be the difference between life or death. This project considers the design and testing of a search and rescue system utilizing an Unmanned Aerial Vehicle (UAV) and two Software Defined Radios (SDR) to systematically search for a missing person carrying a cell phone. Currently, UAV search and rescue missions rely mainly on the use of the UAV's camera and its aerial perspective. This project takes a different approach. By leveraging the maneuverability of UAV technology and the modularity of an SDR, together they can be used to measure and estimate the direction of signals emitted by cell phones and locate a missing person. To validate the proposed system, one SDR on the ground will be used to emulate the signals emitted by a cell phone while another SDR mounted on the UAV will collect directional signal strength data as well as UAV orientation and position data to estimate the direction of the emitted signals. The system operates by moving the position and orientation of the UAV over a repeatable pre-programmed flight path to search for the location of the transmitter emitting the signal. After the UAV returns from its flight path, the data collected by the airborne SDR is analyzed to compute the location of the emitted signal.



Table of Contents

Abstract	1
1.0 Introduction	3
2.0 Background	4
2.1 Humanitarian Based UAVs	4
2.2 UAV Regulations	4
2.2.1 FAA Regulations	4
2.2.2 Flight Test Location	5
2.3 Mobile Phone Emergency Beacon	5
2.4 Unregulated Frequency Bands	6
2.5 Triangulation of Signal Source	7
3.0 Methodology	8
3.1 Transmitter (Tx) SDR embedded hardware/software	8
3.2 Receiver (Rx) SDR embedded hardware/software	9
3.3 UAV and Rx SDR integration	11
3.4 iPad Software to control the flight path of the UAV	13
3.5 PC software to pre-process the data on the microSD card from the Rx SDR	15
3.6 PC software to perform the localization	16
4.0 Preliminary Validation Testing	19
4.1 Verification of the Tx SDR	19
4.2 Verification of the Rx SDR	19
4.2.1 GPS functionality within the Rx SDR	19
4.2.2 Testing of the Tx and Rx connection	20
4.2.3 IMU functionality within the Rx SDR	22
4.3 Verification of the UAV and Rx SDR integration	27
4.4 Verification of the iPad software	27
4.5 Verification of the PC software to pre-process the data	27
4.6 Verification of the PC software to perform the localization	27
5.0 Results	38
5.1 Football Field Test Results	42
5.2 Flight Test Results	42
6.0 Discussion	47
7.0 Future Work	50
References	51
Appendix A: GPS Antenna Product Specifications	54
Appendix B. Flight Path	55

1.0 Introduction

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, offer a distinctive solution to the search for missing persons. As of September 23, 2019, 1,300 people were still listed as missing after Hurricane Dorian hit the northern Bahamas two weeks prior [1]. After this natural disaster occurred, the unique aerial perspective of UAVs was used to assess and inspect infrastructure damage, prioritize efforts on the ground and plan safe routes for entry and exit of damaged areas [2]. These efforts rely on onboard cameras, CO₂ analyzers, and/or thermal imaging sensors. But with the advent of vast mobile networks and the popularity of mobile phones, UAV technology can also be used to search for missing people without solely relying on the UAV's aerial perspective. A cell phone, even when not in use, periodically transmits its signal strength through a Network Measurement Report (NMR) to a nearby base station [18]. Previous projects have used a mobile phone as a personal emergency beacon, and used a fixed wing UAV to successfully locate the mobile phone [28]. Our approach builds on this premise by using a drone with higher mobility than a fixed wing aircraft and with a phased array antenna allowing for a more efficient estimation of the location of the missing person.

In this project, we designed a system utilizing a UAV and an SDR to efficiently search for a transmitter that is emulating the signals emitted by a mobile phone. Through the use of a programmable SDR with two phased array antennas, we can monitor frequencies associated with the transmission and also vary the position and orientation of the UAV to experimentally search for the location of the transmitter emitting the signal. The SDR gathered the inphase and quadrature components of the incoming signal from both antennas as a function of the UAVs orientation and position. The UAV then proceeded to a new position to take additional measurements at different orientations. On a predetermined path, the UAV SDR combination collects and calculates information regarding the received signal, to be analyzed when the system returns to the ground. This project can be considered a first-step toward a fully-autonomous search and rescue system where the drone performs all processing on board and dynamically optimizes its flight path to reduce position uncertainty [14].

2.0 Background

This section of the report presents background on the ideas behind a humanitarian-based search device that can locate a signal transmitted by a mobile phone. A device of this nature would require regulatory compliance and location triangulation capabilities, discussed in this section. Previous experimental cases have alluded to the functionality of this device and facilitate an overview of the required theoretical background.

2.1 Humanitarian Based UAVs

The system created in this project can be used for humanitarian search and rescue missions. The idea of a humanitarian-based UAV is a relatively new concept, where the UAV is used to meet any kind of assistant needs. In our case, a UAV would be used to search for the signals emitted by a cell phone. This technology raises concerns regarding the inevitable increased costs for humanitarian organizations, the ethics behind a humanitarian “eye in the sky” and the mix of political agendas with humanitarian work. However, the UAV industry is gaining legitimacy with the help of the humanitarian sector, by showing the great achievements this new technology is capable of achieving [3]. Drones have been used to deliver needed supplies to remote areas, to survey regions in support operations, and to search for people after disasters. Currently, the latter is done in a highly inefficient way, through the use of a CO₂ analyzer, a camera, or a thermal imaging sensor. After a natural disaster, emergency response is time-critical and the current slow process of searching for a person can be greatly improved [4]. Through this project, we are hoping to innovate the use of a UAV to more efficiently search for a missing person using signals emitted from a mobile phone. However, there are many regulations in the United States for UAV use which need to be understood in order to complete the project.

2.2 UAV Regulations

Due to the seemingly unlimited potential of unmanned aerial vehicles, there are many restrictions on their flight in the United States. The Federal Aviation Administration (FAA) has enacted numerous UAV regulations. These regulations impacted our decision for the location of our flight test.

2.2.1 FAA Regulations

Unmanned Aerial Vehicles require licensing with the FAA to be operated. The FAA also has a series of guidelines that must be withheld to safely operate a drone [16]. These regulations dictate where, when, and how these vehicles may be piloted in the intended flight area. Areas around airports, for example, are restricted so that there can be no interference with inbound or outbound plane traffic. For recreation flyers, in an unrestricted area, a drone can be flown up to 400ft in altitude. This was more than adequate for the experiments in this project. Other regulations are the drone was required to stay within line of sight, never flown over groups of people or flown under the influence of drugs or alcohol [16].

2.2.2 Flight Test Location

The WPI campus was close enough to the Worcester Regional Airport to restrict the ability to fly UAVs on the campus. This, combined with the lack of open space in the areas around campus, led to the need to find a remote, unrestricted flight zone. The location we selected was the Central Massachusetts Radio Control Modelers (CMRCM) airfield in Northborough, MA, shown in Figure 1. This field is used for hobbyist remote control aircraft flights and allows us to conduct our experiments in a safe and minimally restrictive environment. The details of our experiments at this field were discussed in further detail in the following sections.



Figure 1. Aerial view of the CMRCM airfield in Northborough, MA (Google Maps).

2.3 Mobile Phone Emergency Beacon

For the project to be fully implemented, a mobile phone will need to be able to transmit some kind of signal for the receiver on the UAV to triangulate its location. Given the time constraint of our project, this idea was implemented using a radio to transmit a constant tone signal instead of an actual cell phone. This is similar to when a cell phone emits measurement reports containing the power level that is received from nearby cells used to help in handover decisions [28]. Another method used to determine the location

of a cell phone is through cell-ID positioning, where a person is able to determine the cell that is providing the coverage to the phone. Through this technique, a person can then calculate the time it takes for the signal to reach the base station in that cell and estimate the distance in between.

This project has gained inspiration from similar projects involving inventions created to find a missing person in the case of a disaster. A device carried by a person and used as an emergency locator beacon was implemented in a device called a Personal Locator Beacon (PLB). A PLB sends emergency distress signals to satellites. The satellite then is able to detect the alert and distribute it to a person on the ground that can help [28]. Similar to our project, an invention called a Cospas-Sarsat PLB is able to transmit a distress signal at a certain frequency with a complementary homing signal. The location of the transmitted signal can then be triangulated by search and rescue teams [28].

One report, *Your Phone as a Personal Emergency Beacon* [28], details a very similar solution as our project to the problem of locating missing people using their cell phone signals. This report creates a portable base station, using an SDR, that is able to be placed in any kind of vehicle to locate a person that has an active global system for mobile communications (GSM) phone. Essentially, the SDR in this report waits for the phone to connect to its base station. Then, the incoming signal's angle can be used to determine a measurement for the direction of the phone. The main difference between the implementation of this report is that it uses a fixed wing aircraft. When compared to a drone, a fixed wing aircraft has much less control over its flight path. Additionally, our method proposes a way to determine the approximate location of the phone, as opposed to this report which simply finds the direction of the phone relative to the aircraft.

2.4 Unregulated Frequency Bands

A major component of the project is the use of an SDR to locate the transmitting signal that is modelling a cell phone. In the United States, regulatory responsibility for the radio spectrum is divided between the Federal Communications Commission (FCC) and the National Telecommunications and Information Administration (NTIA) [10]. Radio frequency bands are allocated and regulated by these organizations to ensure the safety and successful operation of radio communication. Using this information, we selected frequency bands to transmit on without disrupting the signals of others while remaining within the confines of the law. The Industrial Scientific Medical (ISM) band was originally allocated to create a buffer for devices that created large amounts of radio noise at these frequencies. Objects such as microwave ovens and diathermy machines operate within these "unregulated" bands. More recently, these frequencies have been used for low power communication systems such as Bluetooth or Wi-Fi communications standards existing within the 900MHz, 2.4GHz or 5GHz ISM band [11] [12].

The specific frequency that a cell phone transmits on differs between cell phone carriers, but typically a cell phone transmits around the 2.4GHz frequency band [5]. However, our drone's controller also operates at the 2.4GHz band. Two or more transmitters operating at the same frequency creates interference, prohibiting the receiving device from functioning properly [6]. Therefore, we continuously transmit a tone signal at the carrier frequency of 900Mhz with a single antenna SDR, to model the emissions of a cell phone. A lower frequency signal of 900MHz versus the 5GHz band, can effectively go through objects like for example, leaves in the forest surrounding the field where we were flying. This is

because a low frequency signal has a longer wavelength than a high frequency signal and therefore is able to propagate more easily [6]. With the use of the 900MHz band to transmit the modeled mobile phone’s signal, the received frequency profile was analyzed around 900MHz in order to triangulate the source location.

2.5 Triangulation of Signal Source

The two major methods in signal localization are trilateration and triangulation. Trilateration uses the distance from the target to two or more known receive points. By measuring the time of arrival, it is possible to find the possible location of the incoming signal. The type of sampling we will perform is triangulation. Triangulation uses the known angle from multiple locations to find the transmitter. This uses the angle side angle identity to find the third triangle point [30].

By projecting lines at the measured angle at each point we will be able to find an intersection of these lines. This will be the approximate location of our signal source. A mockup of a location being estimated using triangulation can be seen in Figure 3 below.

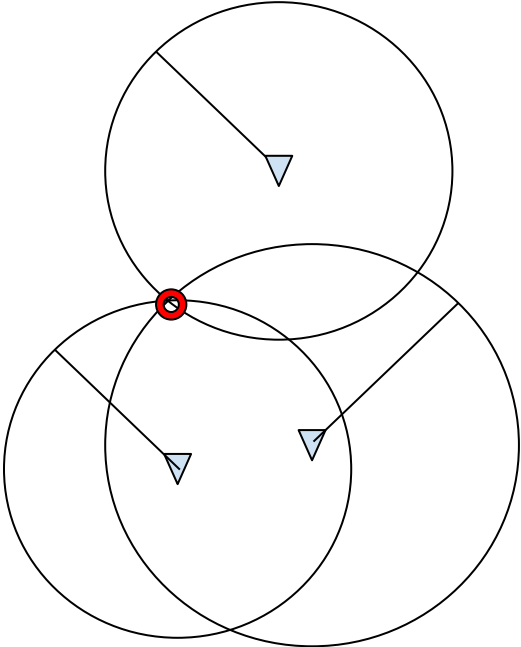


Figure 2. Sketch of trilateration localization.

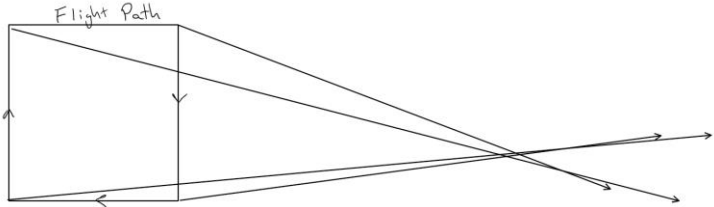


Figure 3. Sketch of triangulation localization.

The following section will discuss the project’s methodology in detail, which will put into action the ideas mentioned above.

3.0 Methodology

To determine where the SDR transmitter is located and find the “missing person”, we can monitor the received signal on the UAVs SDR at various points and triangulate the location of the transmitter. The main idea is as follows: First, a pre-programmed flight path is designed for the UAV. The flight path includes multiple waypoints at which the SDR collects data. At each waypoint, the UAV holds position and rotates its orientation at least 360 degrees. The airborne SDR collects timestamped radio frequency data from both antennas as well as timestamped orientation and position data and logs this data to files on the SDR’s removable SD card. After completing the flight path, the drone returns to home. The SDR’s SD card is then removed and a series of processing steps are performed on a laptop to produce an estimate of the emitter’s location.

The methodology of this project can be categorized into six separate programs and their mechanical counterparts:

1. Tx SDR embedded hardware/software
2. Rx SDR embedded hardware/software
3. UAV and Rx SDR integration
4. iPad software to control the flight of the UAV
5. PC software to pre-process the data on the microSD card from the Rx SDR
6. PC software to perform the localization

To develop a proof of concept for the design, a SDR was used on the ground to emulate a cell phone. The SDR was positioned on top of a tripod in the CMRCM field. The tripod was borrowed from one of the group members. Another SDR was attached to the Matrice 600 Pro drone, programmed to receive the emitted signals and log data to an SD card for later analysis. The team custom built a holder for the SDR and phased array antennas that connected to the drone. Both SDRs were USRP E312s bought off the shelf from the company, Ettus Research. The drone was also bought off the shelf from the company, DJI. The iPad used to control the drone was bought off the shelf from the company, Apple. A Dell laptop booted to Linux was used to write the programs for both SDRs. Separate laptops with the downloaded programs Eclipse and Matlab were used to perform the processing of the data. The following sections will detail each step of the project.

3.1 Transmitter (Tx) SDR embedded hardware/software

While the central concept of this project was to locate individuals by estimating the direction of arrival of signals emitted from their cell phone, we chose to use a USRP E312 SDR to emulate a cellphone emitter. This choice allowed us to control the frequency, format, and power of the emitted signal to simplify the overall design while providing a proof of concept of the core idea. We programmed the ground-based E312 SDR to transmit a constant tone unmodulated signal at a carrier frequency of 900MHz at high output power. All of the programming for the transmitter radio was completed using a Linux terminal in the C++ programming language. To interface with the USRP transmit channels we used various API functions and structs from the USRP Hardware Driver (UHD) libraries.

The UHD library contains specific API functions to interface with the USRP transmit streamer. Transmitter stream initialization includes setting the transmit channels, rate, center frequency, and analog gain. After initialization, the single-channel 900MHz transmitter was ready to stream. Next, the transmit API function, called `uhd::tx_streamer::send()`, was used to transmit data buffers containing the next set of prepared samples. A buffer used by the transmitter was the portion of the system's memory that was available to be used to send packets [19]. To produce a 900MHz tone signal, the buffer was filled with constant non-zero samples.

The transmitter radio used a microSD card that contained the program that starts the transmitter. The E312 radio also had a built-in rechargeable battery that enables us to place the emitter in a known location of the CMRCM field without the need for an external power supply. For the flight test at the CMRCM field, the transmitter was placed on a tripod to ensure that it was not resting in dirt or mud. We used a smartphone tripod adapter to attach the SDR to the tripod. A mockup of the SDR on the tripod can be seen in Figure 4.



Figure 4. Transmit SDR and tripod components.

3.2 Receiver (Rx) SDR embedded hardware/software

Once the transmitter was functioning, the next step was to construct a receiver on the UAV that can record information from its point in the flight path and save the data to a file. Another USRP E312 SDR was used as a radio receiver. Since the transmitted signal was around 900MHz, the receiver was searching for the signal around 900MHz as well. The receiver radio requires a team member to manually begin the data collection process through an USB cable connecting the serial console port on the radio with a laptop. The radio can be set to delay its start for a period of seconds before beginning collection, to allow us time to start the drone. The receiver radio is capable of compiling code in a C++ program development environment run on the SDR's embedded Linux operating system, which is contained on a microSD card. Various C/C++ libraries like UHD, libgps, and RTIMULib are used to interface with different hardware peripherals on the SDR.

Similar to the transmitter, the UHD library contains specific API functions to interface with the USRP receive streamer (RX). Receiver stream initialization includes setting the receive channels, rate, center frequency, analog gain, and frontend bandwidth through the API. After initialization, the dual-channel 900MHz receiver is ready to stream. Then the receive API function, called *uhd::rx_streamer::recv()*, was used to obtain data from the streamer by filling preallocated buffers with samples from each channel.

The next step was to access the onboard IMU and calculate the direction that the drone is facing. Samuel Prager from the University of Southern California (USC) helped the team obtain readings from the IMU through a test program that he wrote and corresponded to us about. The RTIMULib library was used to interface with IMUs on embedded Linux systems and was conveniently compatible with the IMU on the E312, the MPU9150. This library was not downloaded on the radio. Within the E312 SDK, a set of tools that can be used to develop software applications on a specific platform, the IMU library was able to be cross-compiled and installed by copying the compiled library files from the SDK to the E312 [26]. After downloading RTIMULib to the radio, it needed to be manually calibrated. The RTIMULib uses a program called Octave, which is very similar to Matlab and was easily converted into a script that could be carried out by Matlab. Using the RTellipsoidFit tool, we positioned the radio in every direction and orientation and were able to determine a set of 12 float values from a collected data set using the code. These 12 values were then manually added to the calibration document on the radio.

The RTIMULib library contains API functions to interface with the internal MPU-9150 inertial measurement unit (IMU). After initialization, the main API functions, called *RTIMU::IMURead()* and *RTIMU::getIMUData()*, were used to get the orientation of the SDR. The orientation is calculated automatically as a fusion of the sensors of the 9-axis IMU, which includes the accelerometer, gyroscope, and magnetometer.

The library libgps was used to communicate with the GPS daemon to interface with the GPS antenna. This library was already downloaded on the radio so it only needed to be imported into a C++ program to be able to interface with it and call its functions. Data that was collected as the drone is moving was ignored because the vehicular motion can create frequency dispersion due to the Doppler Effect [9]. To know when the drone was moving, we used the GPS location. The libgps library contains API functions to interface with the internal GPS. After initialization, the main API function, called *gps_read()*, was used to get GPS latitude, longitude, and altitude positional data, calculated speed and heading data, and an accurate GPS clock time.

While the data being collected was stored in the SDR's memory, the final data was saved to storage on the microSD card. A 128GB microSD card was used to enable the storage of at least an hour's worth of testing and collecting data. The SDR saves the received data into a separate binary file for each sensor. One file has the GPS data appended with the SDR's system clock time. Another file has the orientation data of the internal IMU appended with the system clock time. The last file contained the inphase and quadrature components of the received signal and the internal clock time. The data being written to these files are organized into custom structures and written in a binary format. The raw binary format was used for writing to files because the speed of collected samples is too fast to write in plain text. The structs are defined as follows:

```

struct rx_t {
    double ts_m;
    u_int64_t ts;
    u_int16_t samps;
    array<array<std::complex<float>, RX_BUFF_SIZE>, RX_N_CHANNELS> buffs;
};

struct imu_t {
    u_int64_t ts;
    u_int64_t ts_m;
    float x, y, z;
};

struct gps_t {
    double ts_m;
    u_int64_t ts;
    double latitude, longitude, altitude, speed, track;
};

```

In addition to writing data in the raw binary format, data collection and writing data to files was completed in parallel with multithreading to further improve system performance. Multithreading eliminates bottlenecks that arise from having slow functions and blocking calls in the program.

3.3 UAV and Rx SDR integration

The specific drone that was used in this project is the DJI Matrice 600 Pro, shown in Figure 5. This drone is known for its extended flight time, 5km long-range transmission, high-precision navigation and positioning system, and compatibility with the DJI GO app [13].



Figure 5. The DJI Matrice 600 Pro.

For the drone-receiver component, one SDR was placed on the drone along with two antennas used to receive the signal and an additional GPS antenna to accurately measure and record the drone's GPS coordinates. Both receive antennas have a band of 824-960MHz, suitable to receive our transmitted

900MHz [23]. While the USRP E312 radio comes with an internal GPS, in order to get a lock on a satellite, an external GPS antenna was required [30]. The product specifications for the GPS antenna used in the project is found in Appendix A.

Ensuring that the radio and its antennas are securely attached to the drone was very important, both for safety concerns and to ensure minimal vibration that can sway the results [20]. A 3D printed antenna holder was created to hold the two antennas, shown below in Figure 6. Since a 900MHz signal has a wavelength of 33.3 cm, the receiver's antennas were spaced half a wavelength apart (16.65 cm) so the signals cancel when the antennas are oriented so that a line drawn through the antennas goes through the emitter. An SDR can then be attached to the holder through the four screw holes depicted in the figure. This holder attaches to the bottom of the drone through the same four screw holes, as shown in Figure 7.



Figure 6. 3D Model of phased array antenna holder.



Figure 7. 3D Model of the SDR mounted on the antenna holder on the drone.

Special attention was taken to ensure that the radio did not interfere with the drone's gimbal camera or the drone's triple modular redundancy antenna system. The bottom of the drone where the antenna mount

was drilled is made up of carbon fiber. Carbon fiber is a very strong material that is also very lightweight. Carbon fiber is made from organic polymers, which consist of long strings of molecules held together by carbon atoms which are then fused together with a resin. This allows it to be five-times stronger than steel and twice as stiff. It is also highly chemically resistant and has high-temperature tolerance with low thermal expansion. All of these benefits do come with some challenges. Carbon fiber is relatively brittle causing a higher chance of cracking or shattering than a more malleable material like steel. This comes into play when drilling or mounting carbon fiber [21].

A common high-speed steel (HSS) drill bit allows for the drilling of 5-10 holes in a piece of carbon fiber before it is completely dull and starts to heat up and burn the resin. A more durable drill bit has tips that are coated with carbide or diamond allowing for many more holes to be drilled than HSS. A large consideration when drilling is drill speed both in terms of rpm and feed speed. When drilling carbon fiber, a high rpm is needed approximately 3000-5000rpm along with a slow feed speed. This allows the drill bit to take very small bites out of the carbon fiber and reduce the chances of cracking or shattering. Finally, the carbon fiber needs to be properly supported during the drilling process. This involves the use of a backing material. Backing material helps reduce deformation and splintering during the drilling process. The most common backing materials are wood and rigid foam insulation. These materials are clamped to a carbon fiber sheet to provide the backing support [22].

3.4 iPad Software to control the flight path of the UAV

Typically, the Matrice 600 Pro is paired with the DJI Go App for flight and camera control. However, we wanted the flight paths to be as repeatable as possible to ensure the same experiment can be completed at another time or a different elevation. The DJI Go App allows a person to fly a certain path and create waypoints as the path is flown. Then, the path can be repeated as many times as desired. This method relies on the human controlling the drone, possibly allowing the drone to fly at varying speeds throughout the test. This can lead to unpredictable collected data which can negatively affect the results. Additionally, to simulate the flight path ahead of time we would need each coordinate of the flight path. By creating a flight path in a third-party app, the exact same test can be completed another time with comparable results.

We looked into three different apps that are adaptable for the Matrice 600 Pro and allow the user to define a flight path. Autopilot for DJI Drones from Auto Flight Logic enables autonomous and semi-autonomous flight. However, it is priced fairly high, at \$29.99, and has poor reviews based on issues with crashing. On the other hand, the DroneDeploy - Flight App DJI is free but does not provide an option to correlate waypoints with GPS coordinates. The final and chosen app is the DJI GS Pro, also known as Ground Station Pro. This is an official DJI app that is free until eight predetermined paths. After eight, it requires a monthly subscription of \$8. This app uses Apple maps and allows the user to tap on the map to create waypoints, also providing the GPS coordinates of the point. Through this app you can control the orientation and spin of the drone. This app was tested at low altitude in Institute Park with a DJI Phantom drone and proved to be capable of flying in a predetermined path and turning 360 degrees at each point.

Using the DJI GS Pro app, we created multiple flight paths that test different scenarios. The different scenarios include an arc encompassing the transmitter, a square flight path, a triangle flight path, an arc

towards the transmitter, a square around the transmitter and an arc over the transmitter. The arc flight path is shown in Figure 8. Screenshots of the other flight paths created in the DJI GS Pro app can be found in Appendix B. The starting point is shown as the green circle, the flight path is the green line and the small red circle shows where the transmitter was located. Notice that each path starts on the area of mowed grass and moves towards the left, away from a nearby road and parking lot. At the CMRCM field a drone is not permitted to fly over the parking lot where it can be a danger to people standing by.

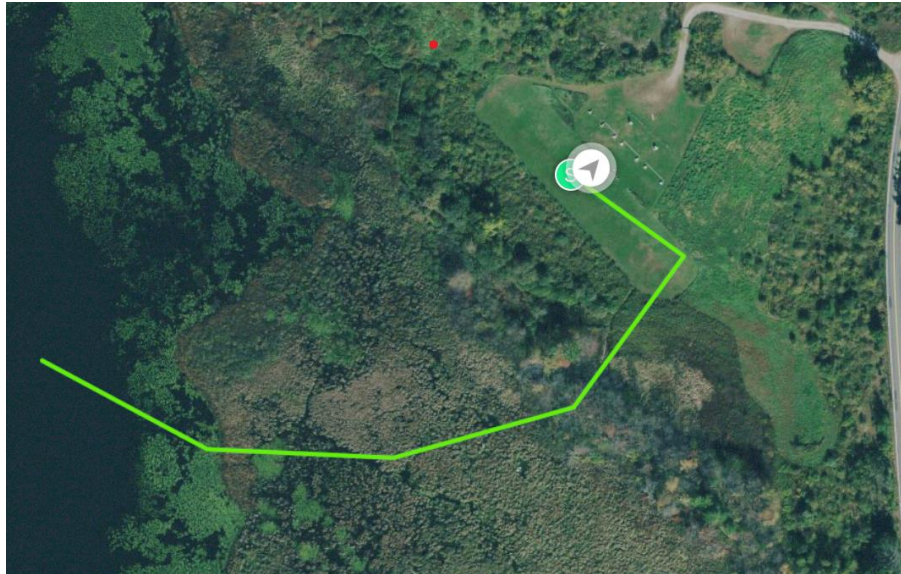


Figure 8. Arc flight path created in the DJI GS Pro app.

After creating each flight path, we sent the paths to Randy Holtgreffe, the president of the CMRCM Club for his approval. Through an email dialogue, we were able to formulate a more descriptive idea of the flight path, ensuring no flight path went over a nearby road. The data collected from the different flight tests will help us understand the relationship between a path and the localization of the transceiver. In addition, it allows us the opportunity to observe multipath propagation from woodland and water areas. For all of the tests, the transceiver was positioned next to the field on a tripod, as discussed in Section 3.1.

The exact determination of a flight path is dependent on a few factors including minimal path length, and the processing of acquired information capability of the software [15]. A minimal path length is desirable due to the duration limitation of the batteries on the UAV. Additionally, when used in a real-life situation of finding a missing person, minimal search time is critical resulting in a short but effective path. The Matrice 600 has an extended flight time capability due to its six onboard DJI smart batteries. If a signal battery fails, the remaining five ensures the UAV remains in flight. Additionally, the batteries enable the user to monitor their status during the flight [13]. With the payload of the attached camera and radios, the UAV is expected to have a flight time of 25 min. This was taken into account when determining the exact flight path. Ideally, a path planning algorithm should be coded in the software on the UAV so that the UAV can re-plan its path if needed based on the data it acquires. This requires a processor on the UAV that is efficient and capable of performing in real-time [14]. While this is necessary for the UAV to autonomously search for a missing person, it is beyond the scope of our project.

3.5 PC software to pre-process the data on the microSD card from the Rx SDR

At each pre-programmed waypoint in the flight path, the drone completes one or more 360-degree rotations. The incoming signal samples were mathematically viewed as inphase and quadrature components, Eq.1 for the first antenna, and Eq.2 for the second antenna, where k is the number of samples.

$$u_{1i}[k], u_{1q}[k] \quad \text{Eq. 1}$$

$$u_{2i}[k], u_{2q}[k] \quad \text{Eq. 2}$$

Using a program written in C, the inphase and quadrature components were pre-processed to prepare a set of data used in the next step. The samples from the two antennas were combined by their inphase and quadrature components to create the following function for the in phase, Eq.3, and Eq.4 for the quadrature.

$$x_i[k] = u_{1i} + u_{2i} \quad \text{Eq. 3}$$

$$x_q[k] = u_{1q} + u_{2q} \quad \text{Eq. 4}$$

Next, the received signal strength was computed by first finding the overall magnitude of the sample.

$$y[k] = |x[k]|^2 = x_i^2[k] + x_q^2[k] \quad \text{Eq. 5}$$

The magnitude was then averaged across all of the samples, determined by the following equation, Eq. 6 [29].

$$z[k] = \frac{1}{k} \sum_{n=0}^{k-1} y[n] \quad \text{Eq. 6}$$

To save time when writing the collected information from the Rx SDR, the information is all written in binary. The binary files from the flight tests are read by the C program to recover the data and converted into a csv file of ASCII characters. In the C program, each structure is defined exactly how it was used by the receiver. The C program also lines up the data obtained from the receiver, IMU and GPS by the time each sample was taken so that the information can be related to a graph using Matlab.

The following chart, Figure 9, shows the basic structure of the files produced in the project. The radio produces three binary files containing the received signal data, the IMU measurements, and the GPS data. These files are converted using the C programming language to a csv file. The csv file organizes the data by the time each sample was taken. This file is then entered into Matlab to determine the location of the transmitter, “fox”. The specifics for the localization step are described in the succeeding section.

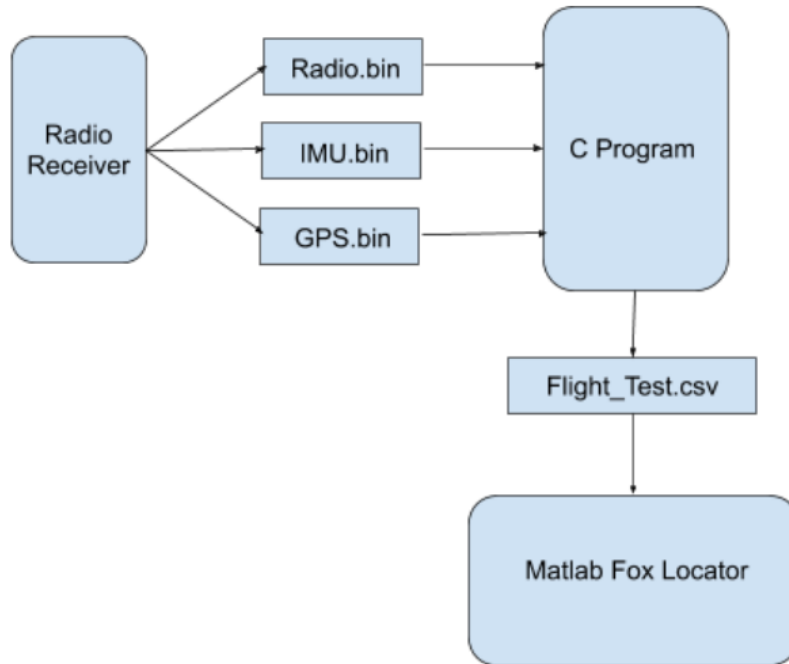


Figure 9. Outline of the project's architecture.

3.6 PC software to perform the localization

A key aspect of the approach considered in this project is the use of phased array signal processing techniques which are made possible by the fact that the E312 USRP SDRs used in this project have two receive antennas. This method uses the relative phases of each signal to each antenna to determine the direction of arrival of the emitted signal [8]. On the airborne SDR, the in-phase and quadrature signals from both antennas were recorded as samples along with the time of the sample, the GPS coordinates of the drone, and the orientation of the drone.

Using these vectors, a bilaterally symmetric polar plot can be created with magnitude weakest in the directions directly towards and away from the target signal, as shown in Figure 10. This is because at this point, the magnitude vectors from the two antennas canceled each other out, completed by computing the average of the inphase and quadrature of the samples from both antennas, described in Section 3.5. By using the onboard SDR magnetometer we were able to relate the resulting plot to the magnetic heading of the drone. From this plot, a vector can be calculated that is perpendicular to the two points of greatest magnitude, showing the direction of the transmitter [17].

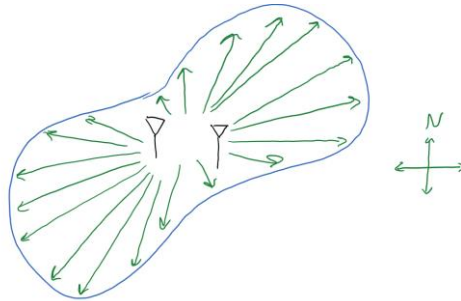


Figure 10. Sketch of predicted symmetric polar plot.

The average magnitudes at each point helped determine a direction to move, as shown in Figure 11. By calculating multiple vectors at different locations, the target can be located by mapping the intersection of these direction vectors. Outliers can be removed using statistical analysis and RSS. The average point was deemed the location of the target with an uncertainty based on the standard deviation of this list.

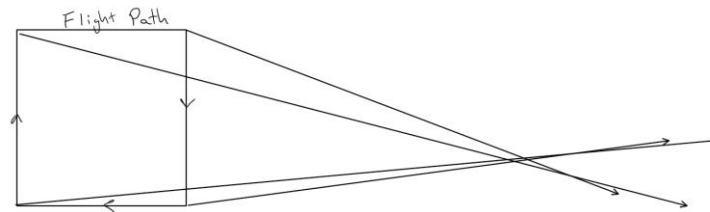


Figure 11. Sketch of intersection points.

We expect there to be some location uncertainty. This uncertainty may be due to multipath propagation where reflections of the transmitted signal from nearby objects are received. We will be calculating our location by taking an average in the x and y direction to find the center of all the line intersections. The PCA function in Matlab will give us the principal axis of this error and we will be able to plot an ellipse that aligns with this axis. Each standard deviation of the ellipse will account for a greater likelihood of capturing the data, shown in Table 3.6.1.

Table 3.6.1 Certainty Based on Standard Deviation

1 Standard Deviation	68% Certainty
2 Standard Deviation	95% Certainty
3 Standard Deviation	99% Certainty

The csv file created in the previous step was used as an input to a Matlab program designed to determine the location of the transmitter. Using this program for each flight path, the start and end time of each test point was established by recording the time when the longitude value changed. This helped the team better decipher the csv file to understand when the drone started spinning at each test point. Once the drone began to spin, the collected information was useful for the localization of the transmitter. After determining the time period of each spin, a polar plot at each test point was created. This was completed

by simply using the plot command in Matlab to graph the calculated magnitude times the cosine of the collected magnetometer reading against the magnitude times the sine of the collected magnetometer reading. This was done for each test point of each flight path. The Matlab program then determined the axis of each polar plot's null and drew a line on that axis. The intersections of each null line of the polar plots for each individual flight path was used to predict the location of the transmitter.

Each flight path was simulated first using the same Matlab program but with simulated data to test the functionality of the program. The intersections of the lines were used to create a confidence ellipse for the predicted location of the transmitter. To better visualize the collected data from each flight path test, we plotted the data on a graph using Matlab that has an overlap of a Google Map image of the field [24]. The same program was then used on the collected data and the results from both were compared.

4.0 Preliminary Validation Testing

Before performing a flight test, each part of the project's methodology was individually tested to ensure the proper functionality of the entire system. The results of each individual preliminary validation test and integration test are outlined in the following section.

4.1 Verification of the Tx SDR

Using a 900MHz antenna connected to the GSP-830 Spectrum Analyzer, we observed the signal and validated the operation of the emitter. The following figure, Figure 12, shows the recorded tone signal with a center frequency of around 900.006MHz.

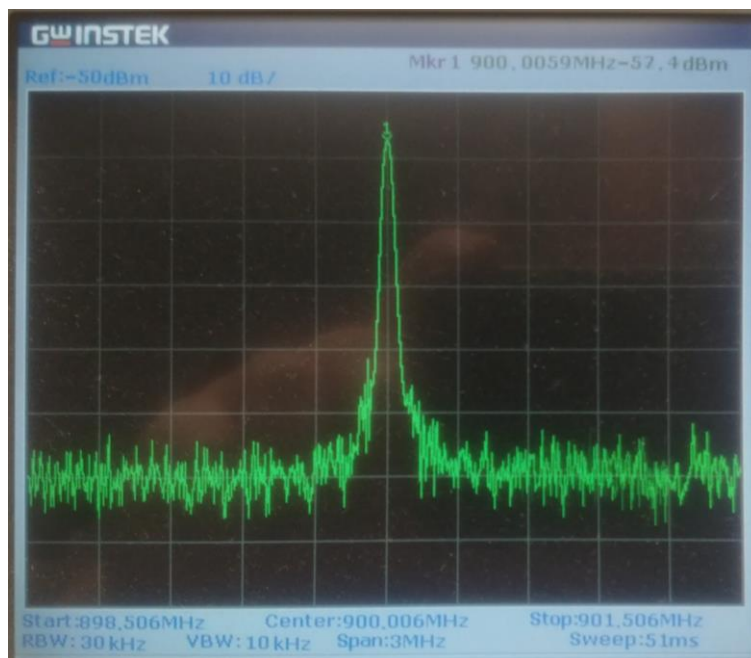


Figure 12. Tone signal viewed on the spectrum analyzer.

4.2 Verification of the Rx SDR

The receive functionality of the radio, along with the GPS and IMU measurements were the first to be tested for the receiver radio. These tests were conducted in a lab and proved the initial performance of the code created for the receiver.

4.2.1 GPS functionality within the Rx SDR

The completed test that verified the GPS measurements was to simply hold the GPS antenna against a window in Atwater Kent and observe the collected measurements from the SDR. Figure 13 shows the raw data that can be collected from the external GPS antenna in conjunction with the SDR. The main pieces of data that we extracted from this are the latitude and longitude values.

```

Time: 2019-11-07T16:57:04.000Z
Latitude: 42.27520394 N
Longitude: 71.80750095 W
Altitude: 586.811 ft
Speed: 0.01 mph
Heading: 287.9 deg (true)
Climb: 2.76 ft/min
Status: 3D FIX (9 secs)
Longitude Err: +/- 32 ft
Latitude Err: +/- 58 ft
Altitude Err: +/- 119 ft
Course Err: n/a
Speed Err: +/- 1 mph
Time offset: -1573142387.240
Grid Square: FN42cg
PRN: Elev: Azim: SNR: Used:
 5 17 095 28 Y
10 16 288 45 Y
15 73 047 30 Y
20 43 298 40 Y
21 64 279 45 Y
29 10 206 26 Y
133 34 216 43 Y
24 36 158 23 N
27 11 319 25 N
131 -91 000 41 N
138 30 226 44 N
operatioTPVor /etc/resolv.conf. mode":1, "time": "2019-11-07T16:56:24.000Z", "ept": 0.005}
" class": "TPV", "device": "/dev/ttyPS1", "mode": 1, "time": "2019-11-07T16:56:25.000Z", "ept": 0.005}
" class": "TPV", "device": "/dev/ttyPS1", "mode": 1, "time": "2019-11-07T16:56:25.987Z", "ept": 0.005}
" class": "TPV", "device": "/dev/ttyPS1", "mode": 1, "time": "2019-11-07T16:56:27.000Z", "ept": 0.005}
" class": "TPV", "device": "/dev/ttyPS1", "mode": 1, "time": "2019-11-07T16:56:28.000Z", "ept": 0.005}
" class": "TPV", "device": "/dev/ttyPS1", "mode": 1, "time": "2019-11-07T16:56:29.000Z", "ept": 0.005}

```

Figure 13. Received data through the GPS antenna.

4.2.2 Testing of the Tx and Rx connection

To verify the functionality of the receiver and transmitter system, first an ASCII graph was produced by the receiver showing the received data, after the transmitter had begun transmitting. The following image, Figure 14, shows the graph for a frequency range around 900MHz. As shown in the image, there was a spike at around 900MHz, the frequency of the transmitter radio.

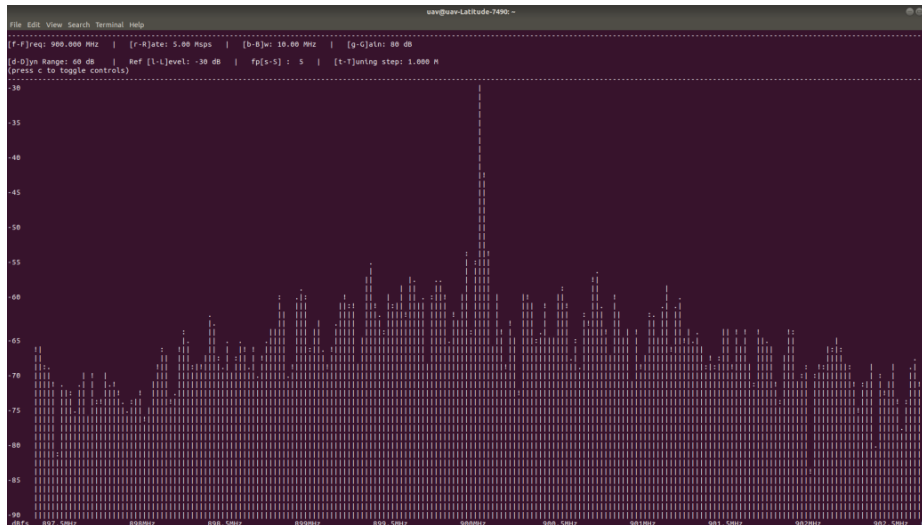


Figure 14. ASCII graph of received tone signal.

The next test was a wired test between the receiver and the transmitter. A picture of the test set up can be seen in Figure 15, with the transmitter on the left and the receiver on the right. Even though the E312 radio can function without an external power supply, both were connected to a wall outlet during the test. The transmitter and receiver were also both connected to the laptop to begin and end their respective programs. The transmitter output was connected to a splitter, separating the transmitted signal into two in order to test the two receive channels on the receiver.

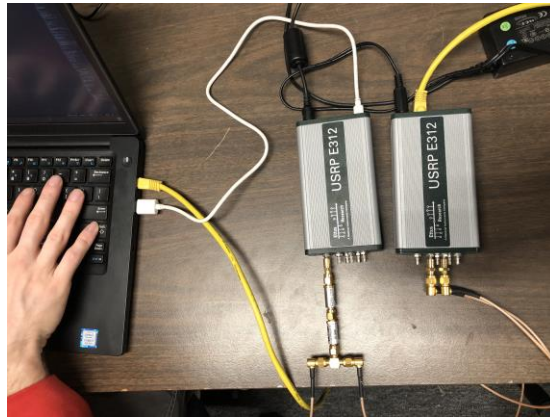


Figure 15. Setup of the wired test.

The results from the wired test can be found in the graphs on Figure 16. A graph showing the yaw of the IMU collected data vs time is on the top while the signal magnitude vs time is on the bottom. Since the radio was stationary during the test, the top graph is a relatively straight line, showing no movement. In the signal magnitude graph, the two waveforms shown are the inphase of both receive channels added together and the quadrature of both receive channels added together.

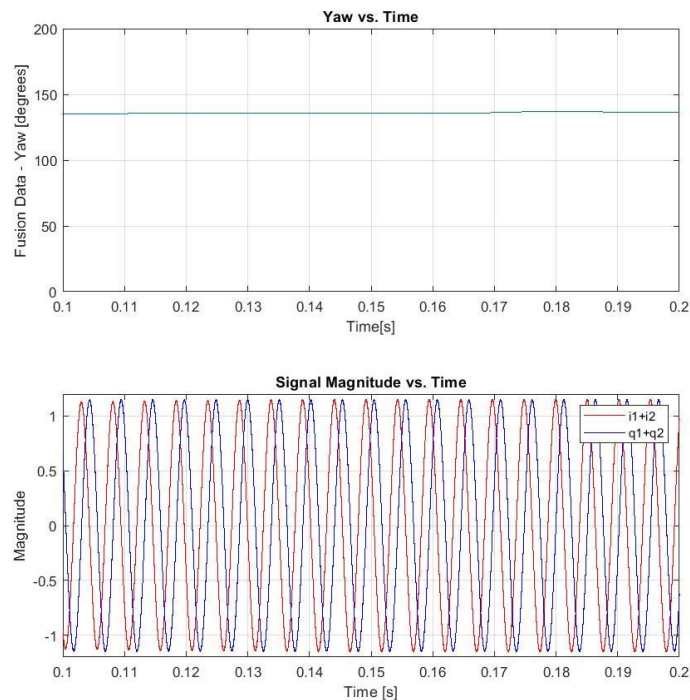


Figure 16. Results of the wired test.

The results of this test showed that the two receive channels of the receiver were capable of collecting the transmitted signal. To further test the IMU data, we performed another test with a variable yaw using a turntable.

4.2.3 IMU functionality within the Rx SDR

To test the functionality of the onboard IMU, we first plotted the calibration data from the calibration step of implementing the IMU code as ellipsoids [25]. This plot can be seen below in Figure 17a.

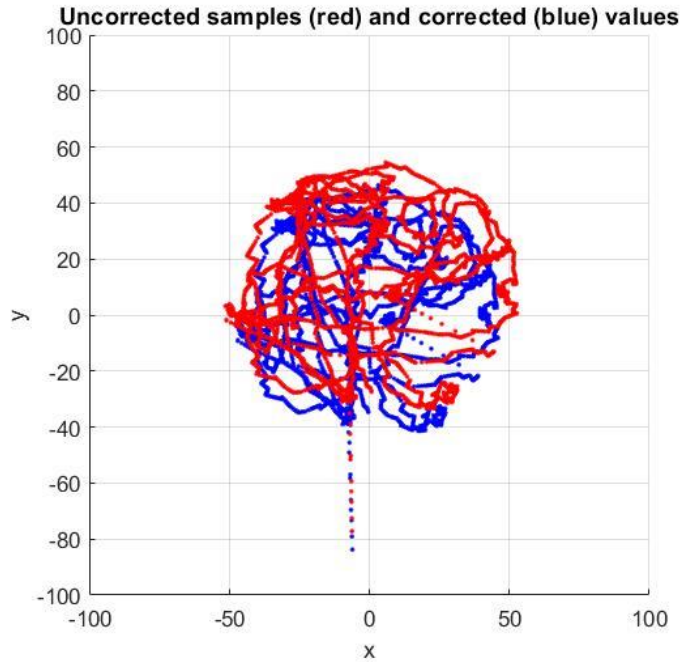


Figure 17a. Calibration of the IMU on the E312.

The same calibration procedure was followed with the radio in its position on the drone. The resulting graph shows the plotted ellipsoids of the calibration data, Figure 17b. As shown, the calibration was not as thorough as the previous calibration.

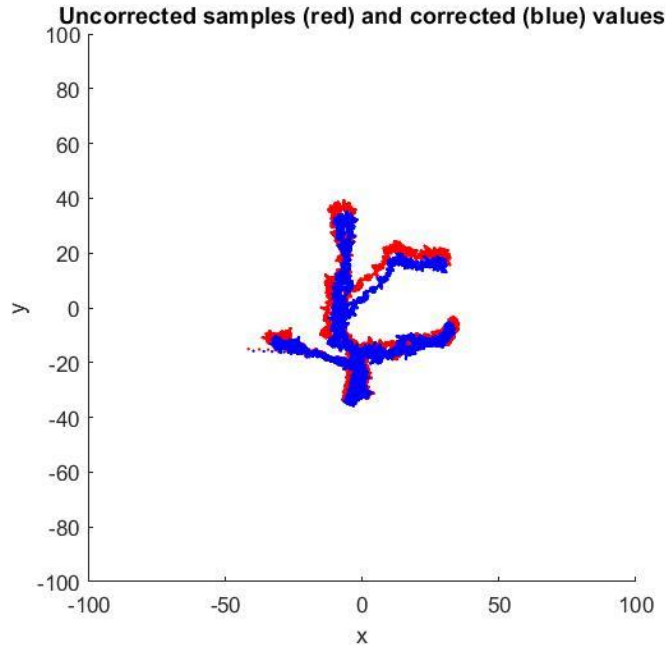


Figure 17b. Calibration of the IMU on the E312 positioned on the drone.

The IMU is capable of reading the time, log time, fusion values in terms of roll, pitch, and yaw, acceleration measurements, gyroscope measurements, and compass values. The following lines of code are an example of the data that the IMU returns after being properly calibrated.

Timestamp : Log Time : Fusion(roll,pitch,yaw) : Accel(x,y,z) : Gyro(x,y,z) : Compass(x,y,z)

19700102T224009.902102 : 168009902086 : (-000.5322,-01.0241,+010.5400) : (+00.1880,-00.1074,+09.8462) : (-000.0236,-000.0006,-000.0240) : (+43.0928,-07.8441,+32.0572)

We were most interested in the fusion numbers, yaw in particular. This was because the magnetometer, gyroscope, and accelerometer output raw data. Each individual sensor or partial combination of sensors is insufficient to provide both precise and long-term orientation data. Some potential sources of interference are objects with magnets or motors, including modern cell phones. This can affect the magnetometer's reading which would in turn affect the yaw measurement.

To further verify the functionality of the received IMU data, the information received from the SDR was used in Matlab to better visualize it. Using a Numark TT200 Turntable, we performed a few tests to visualize the raw IMU yaw data. In this test, the receiver is placed on a 12-inch tripod on top of a Numark TT200 turntable. The tripod was used to limit the electronics within the turntable from interfering with the receiver. Also used in this test were 900 MHz antennas, two on the receiver and one on the transmitter. The pictures below show the test set up, Figure 18 and Figure 19.

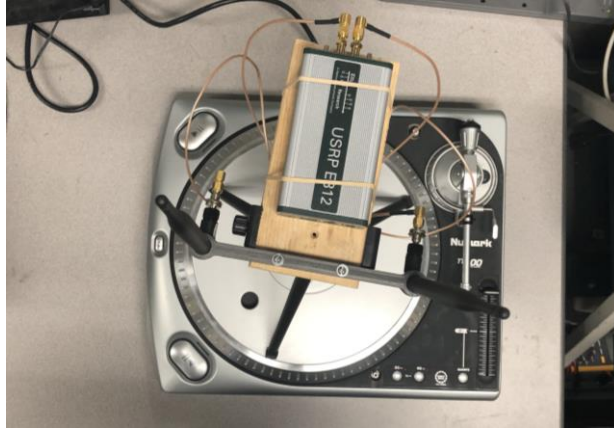


Figure 18. Turntable test setup (1).

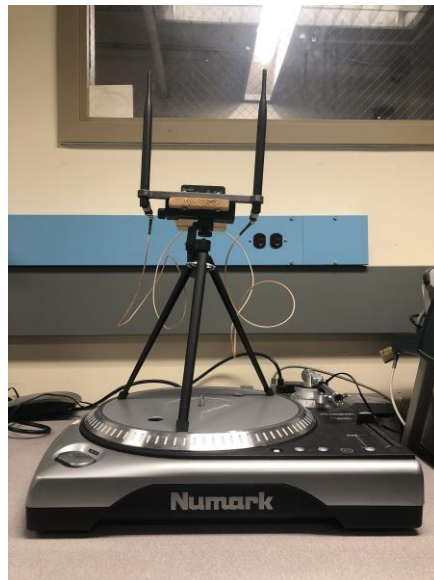


Figure 19. Turntable test setup (2).

The graph in Figure 20 shows the fusion yaw data versus time when the SDR was stationary. The yaw value shown in the graph remains fairly consistent at a value of 140 degrees.

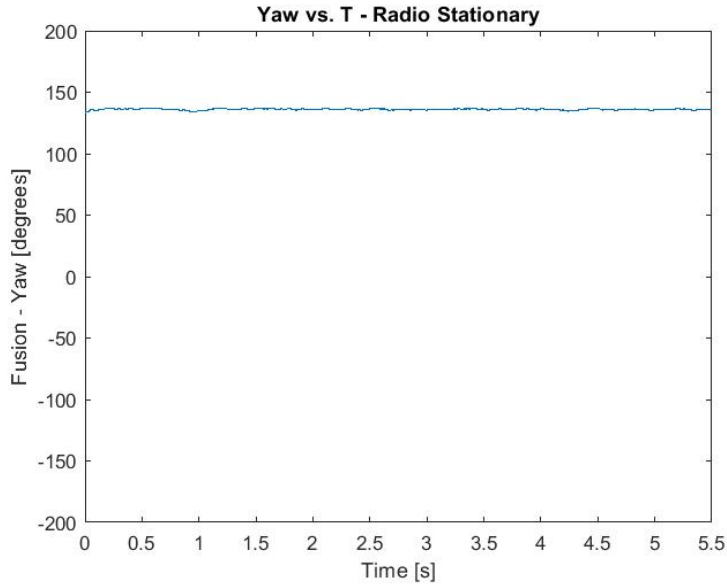


Figure 20. Fusion yaw data when the SDR is stationary.

The graph in Figure 21 shows the fusion yaw data that was collected from the SDR when the SDR is spinning on the turntable at 33 RPM, elevated on top of a 12-inch tripod spinning on the turntable. As shown, as the radio turns in a complete circle, the value of the yaw varies between 180 degrees and -180 degrees, verifying the completion of an entire circle on the turntable. The yaw data appears to not be linear, potentially due to various outside factors such as the magnets within the turntable or the other electronic devices in the room. Both of which may be interfering with the magnetometer reading.

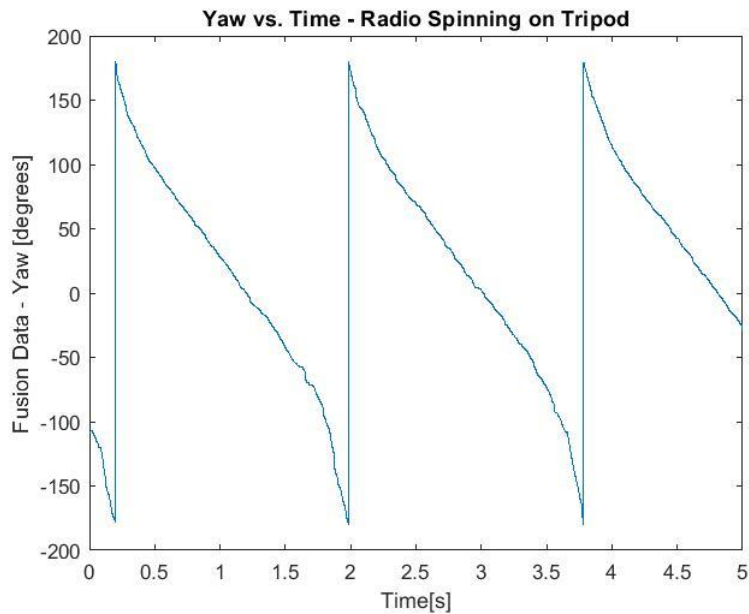


Figure 21. Fusion yaw data when the SDR is spinning at 33 RPM on the tripod.

The turntable was then spun at a constant 33 RPM and data was collected on the receiver while spinning. The resulting graph that displays the yaw data and the collected received signals is shown in Figure 22. This graph shows that the data repeats itself; it is periodic.

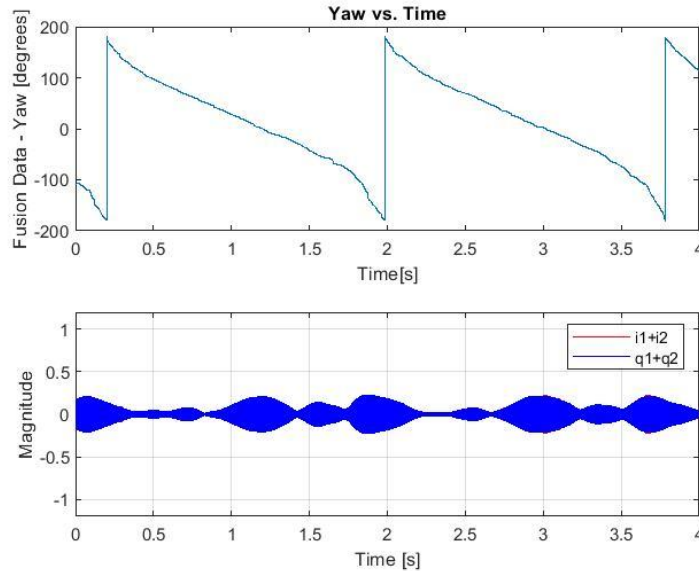


Figure 22. Results from constant spinning turntable test.

When the signal magnitude is graphed in a polar plot to determine the lines of the null, the resulting graph appears, Figure 23. This graph plots each period over each other. As shown, the polar plot is not a perfect peanut. This is due to the many effects of multipath occurring in the lab. Also, the turntable is electric and can be a source of interference as well. This test also showed that further work was required for syncing the time of the received signal with the measured IMU data. Each period did not line up perfectly, as shown by the multiple peanut plots superimposed on top of one another.

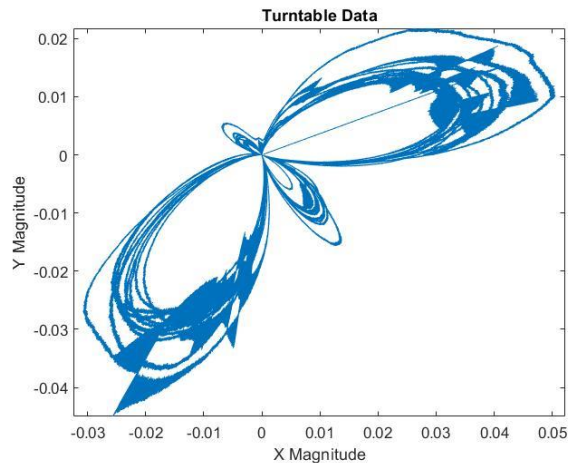


Figure 23. Polar plot of constant spinning turntable test.

4.3 Verification of the UAV and Rx SDR integration

The main concerns of the functionality of the SDR and antenna holder was that it would fall off of the drone mid-flight or that it would interfere with the drone's internal equipment. When the SDR/antenna holder was screwed into the drone along with the radio and antennas, the drone was manually picked up and moved around in different directions. When this was done, nothing fell off or came loose, ensuring that the holder was firmly attached.

4.4 Verification of the iPad software

The software chosen to fly the drone in a predetermined flight path was the app, DJI GS Pro. This app was tested at low altitude in Institute Park with a DJI Phantom drone and proved to be capable of flying in a predetermined path and turning 360 degrees at each point.

4.5 Verification of the PC software to pre-process the data

While in the lab, data was collected from the Rx radio at different orientations when the transmitter was turned on. This data was then used to write and verify the C program written in Eclipse. The sample data was written in the same format and structs as it would be in the actual field tests. Therefore, the data helped us better understand how to parse through the collected data, perform necessary calculations on the inphase and quadrature data, sync the different forms of data to one-time source and output a csv file with all of the collected data in a readable format.

4.6 Verification of the PC software to perform the localization

To prepare for each flight path, we simulated the data that we would receive. The data outputted from the C program that converted the received binary information was in the form of a csv file. Therefore, we created sample test data in excel which was exported as a csv. The sample data contained noisy magnetic heading and magnitude measurements. This data was then plotted in a polar plot, shown in Figure 24a. The lines shown on the polar plot are the lines of highest magnitude and the lines of the estimated null. The line of estimated null is the axis where the two received signals for each antenna have cancelled each other out, therefore showing the axis direction where the receiver predicts the transmitter to be located. This concept is also shown in Figure 24b, where the null of the polar plot on the right occurs when the two antennas (shown in blue) are directly perpendicular to the transmitter.

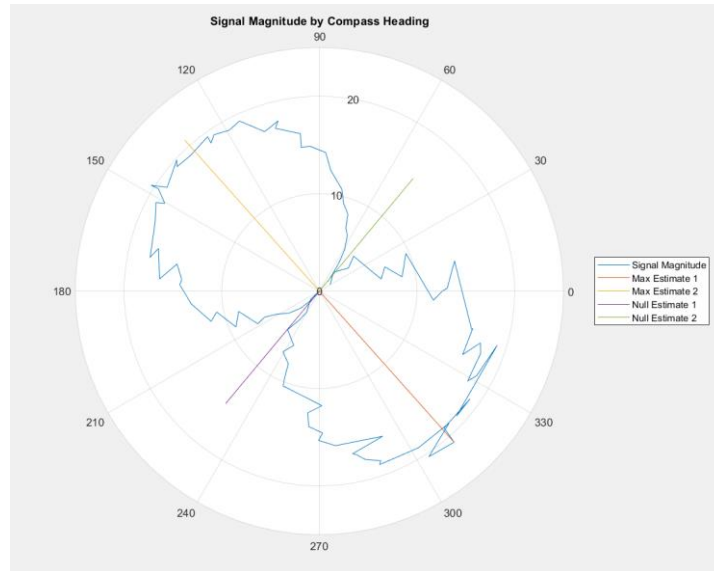


Figure 24a. Polar plot of sample data.



Figure 24b. Drone position in relation to polar plot and transmitter.

Using this sample data and an arbitrary flight path, we estimated the direction of the transmitter location through a simulation. This was completed by calculating the magnitude based on the relative heading to the fox itself. To increase realism this method was later changed to a system where magnitude was calculated based on the distance between the fox and each antenna in 3D space. In this scenario, we knew where the transmitter was located and working backwards, created polar plots leading to the transmitter location with a specific signal to noise ratio (SNR) of 30dB. Noise was also added to the direction vector to simulate noise from the onboard IMU compass. The polar plots drew their direction vector based on the strongest received magnitude from the antenna sum. The specific polar plot created at a point and the collective polar plots of an entire flight path can be seen in Figure 25.

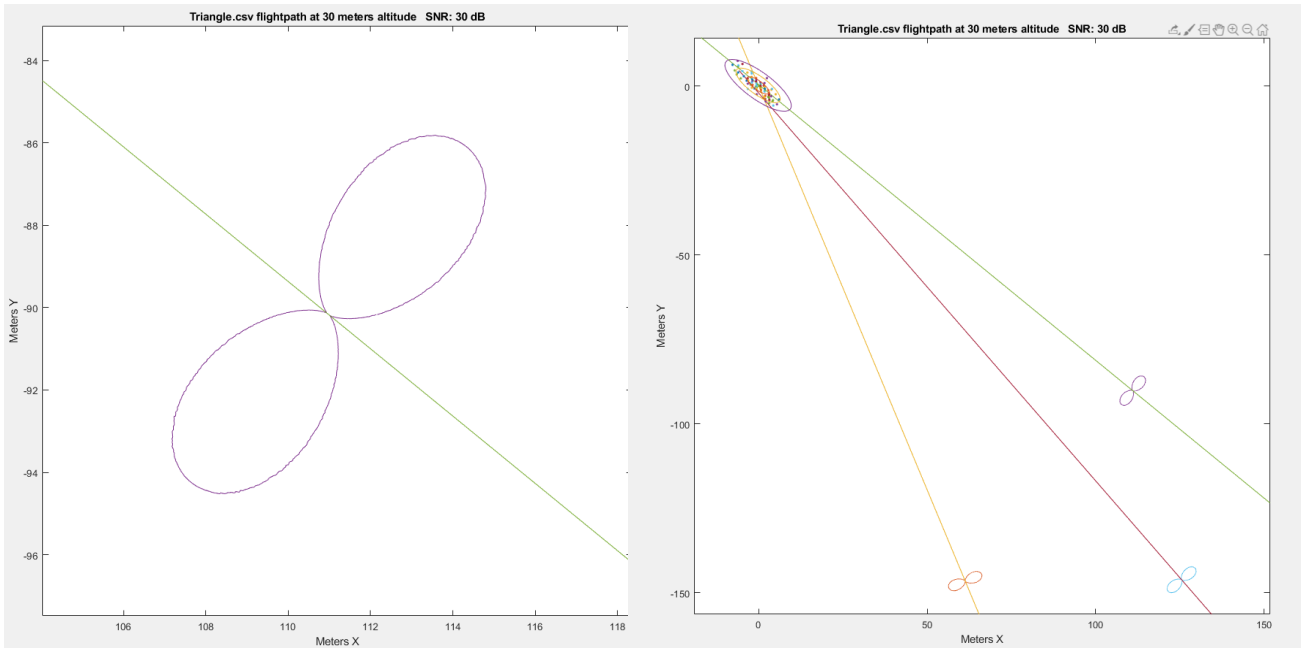


Figure 25. Transmitter location estimation using maximum magnitude.

The direction vectors used to actually locate the drone are found by drawing a line through the null in the signal magnitude. By averaging intersections in these vector lines, it is possible to estimate the location of the signal source.

The null was found by checking magnitudes that were 90 degrees from each other and attempting to maximize the ratio between them. This uses both the maximums and nulls to find the most accurate direction vector. Issues were found at lower SNR in terms of the nulls' accuracy but this was improved by using the included `smoothdata()` Matlab function. This uses a sliding window filter to remove higher frequency noise from the signal very effectively. This filtering was able to remove much of the noise and greatly improve the filter's accuracy.

We then performed a Monte Carlo Simulation for 500 iterations of the null searching method simulation. This displays points where the maximum magnitude for each polar plot intersects each other, and therefore can be a possible location of where the transceiver is located. This simulation created a point cloud of fox location estimations, as shown below in Figure 26. Then, using principal component analysis (PCA), we found the principal axis of the point cloud. This allowed us to plot an ellipse rotated to follow the principal axis slope and with dimensions that should capture one, two, and three standard deviations of the location estimation. This means that each ellipse should capture the result with a certain confidence with 1 standard deviation encompassing ~68% of the data while two and three standard deviations encompass 95% and 99% of the data. These confidence windows assume that the error is normal with the x and y axis. The error computed in this simulation can then be compared to the error we receive in the experimental results.

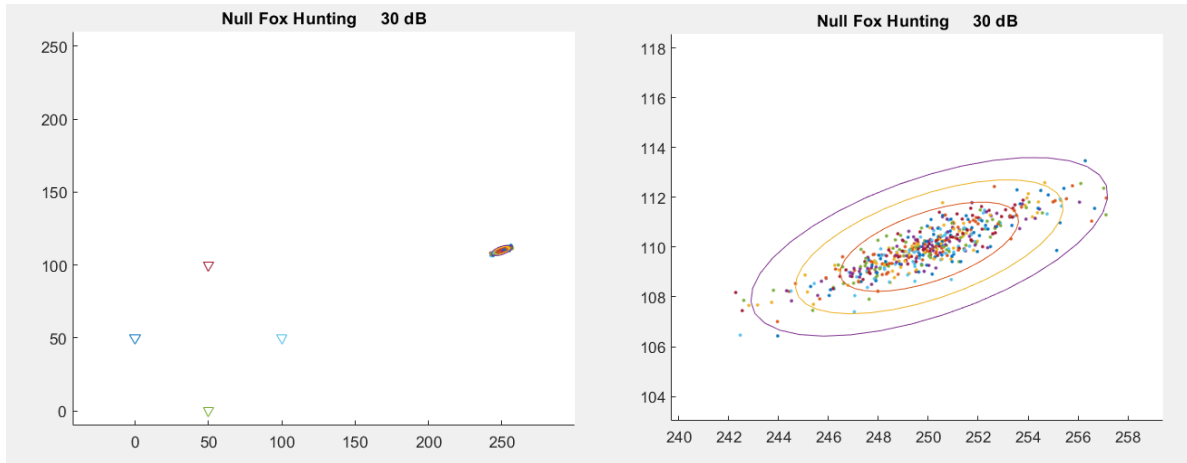


Figure 26. Transmitter location error approximation.

This location error approximation for another flight path can be seen below in Figure 27. Using the same method of PCA mentioned previously, we created an error ellipse for the estimated transmitter location. Shown is this error ellipse in both latitude and longitude and in meters. The point of the actual transmitter (“Fox”) is shown as the x in both graphs.

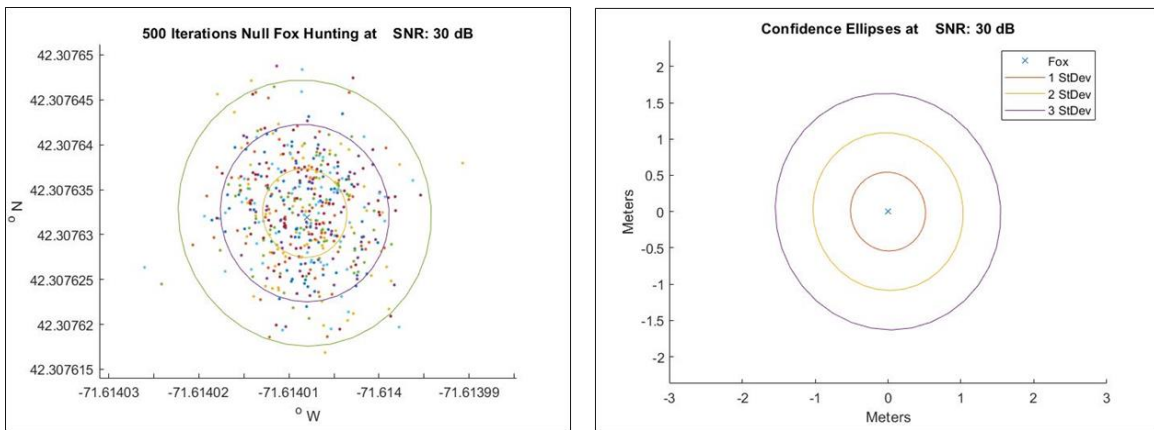


Figure 27. Transmitter location error approximation confidence ellipse.

Next, using the flight paths created in the previous section “Flight Path Planning”, we created confidence ellipses at the estimated location of the transmitter based on polar plots created at each point of the path. The confidence ellipse created for the six various flight paths are shown below in Figures 28a-f.

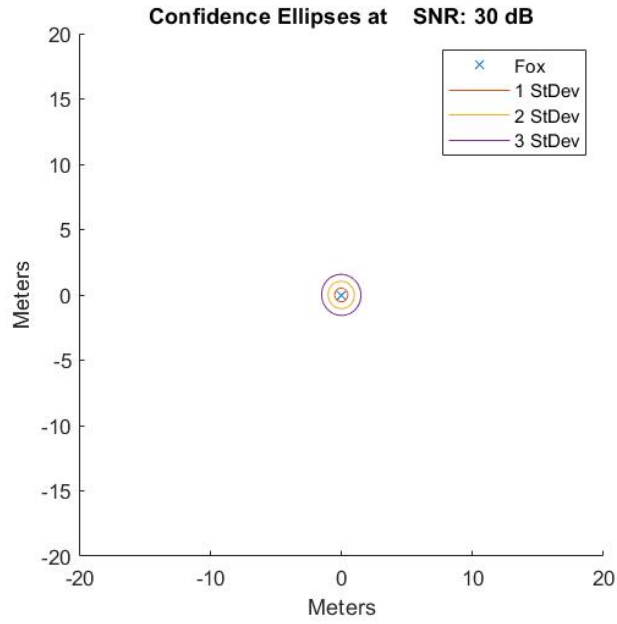


Figure 28a. Arc flight path corresponding confidence ellipse.

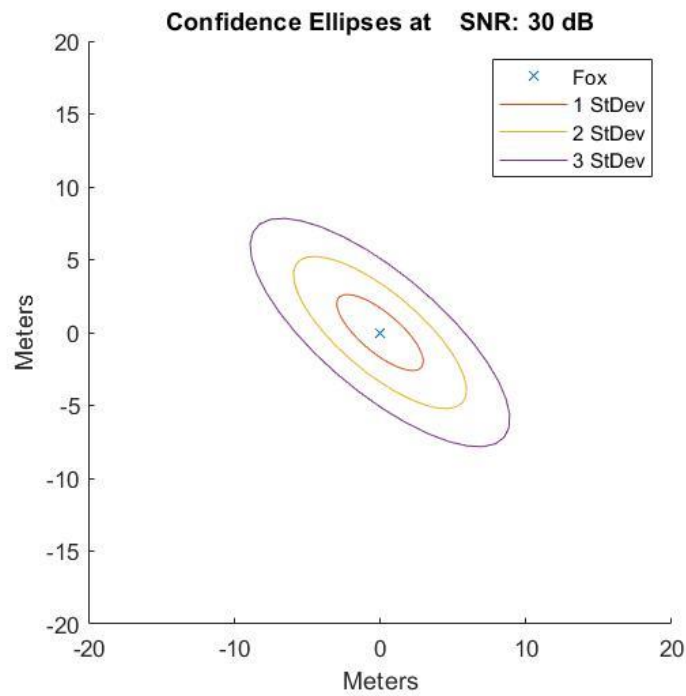


Figure 28b. Triangle flight path corresponding confidence ellipse.

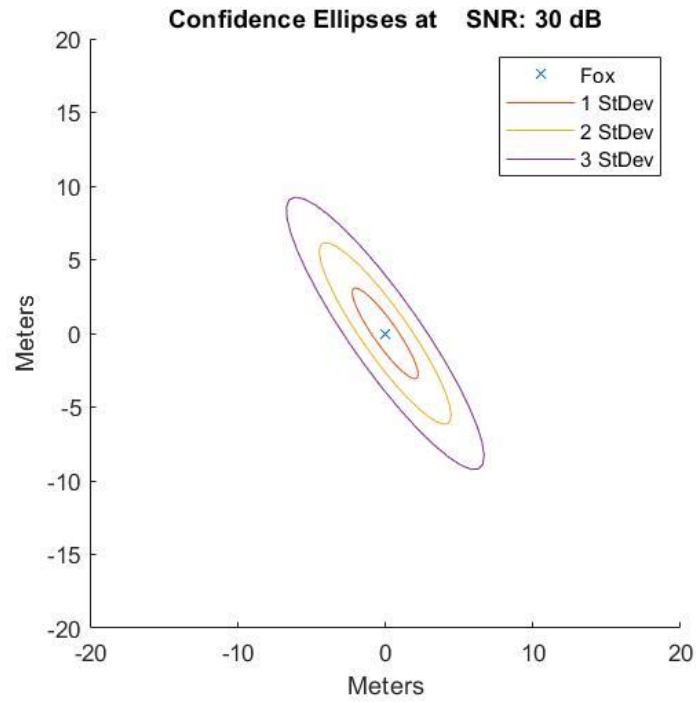


Figure 28c. Square flight path corresponding confidence ellipse.

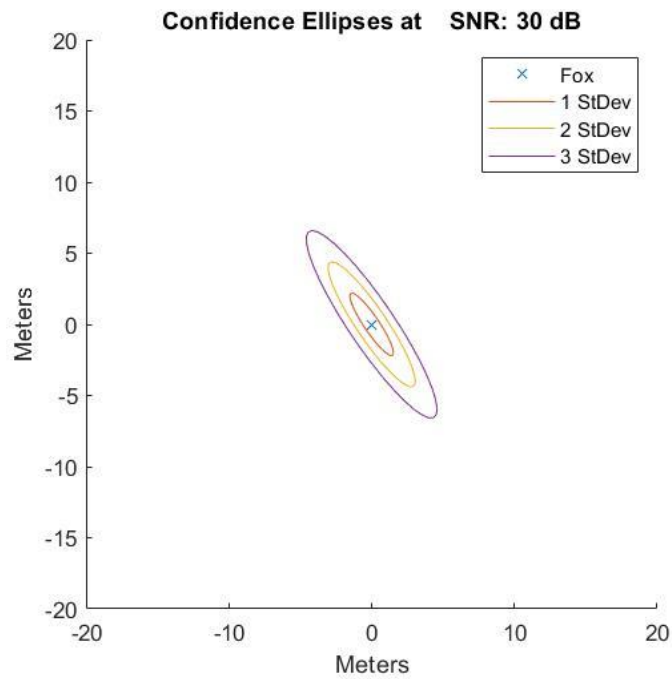


Figure 28d. Arc towards transmitter flight path corresponding confidence ellipse.

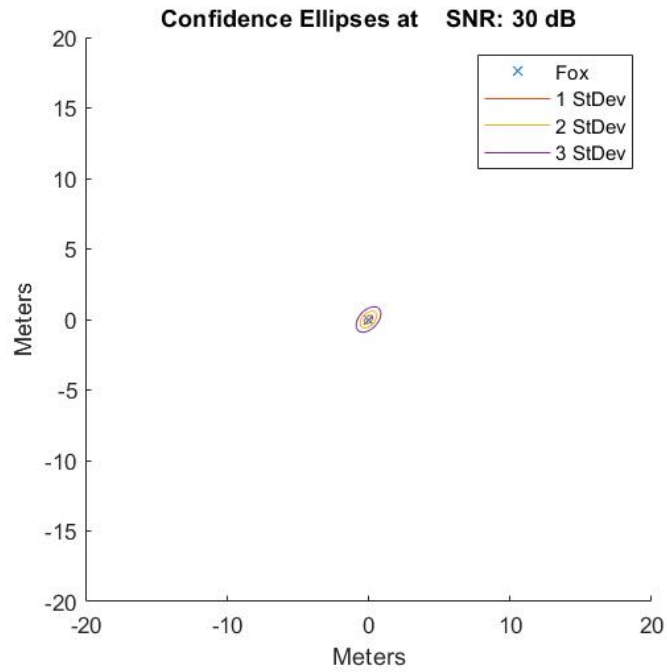


Figure 28e. Square around transmitter flight path corresponding confidence ellipse.

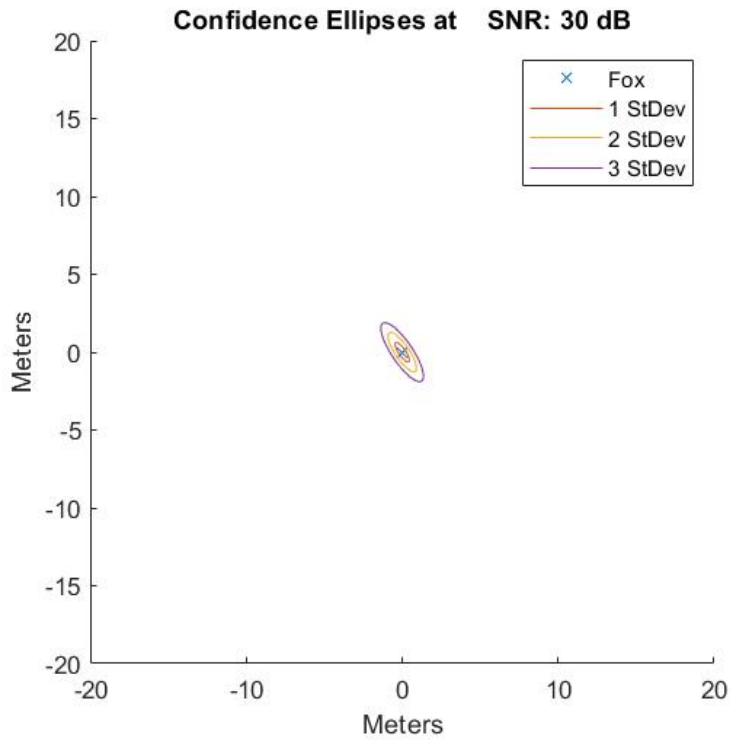


Figure 28f. Arc over transmitter flight path corresponding confidence ellipse.

While keeping the same axes for all of the graphs, the size of the error ellipse demonstrates the importance of picking a good flight path. The small confidence ellipses of the arc flight path, the square around the transmitter, and the arc over the transmitter are highly desirable and shows that these flight paths were simulated to be optimal for determining a very precise transmitter location. These flight paths have a simulated confidence ellipse of less than 5 meters in diameter, depicting an accurate predicted transmitter location. On the other hand, the very large confidence ellipses of the triangle, square and arc towards the transmitter flight paths show nonoptimal flight paths. These simulated confidence ellipses have almost a 20-meter diameter which is 4 times larger than the other three flight paths.

Using the simulated data, we created images shown in Figure 29a-f. For each flight path image, we plotted a polar plot at each point of the flight path that contained the simulated received signal data. A line of the null was drawn for each polar plot. The same Monte Carlo simulation described previously was utilized to determine potential locations of the transmitter. The inset plot of each image shows a zoomed in picture of the intersections and the error ellipses that result. A top view of a helicopter is shown to provide a scale. We expect to create a similar image from our test results.

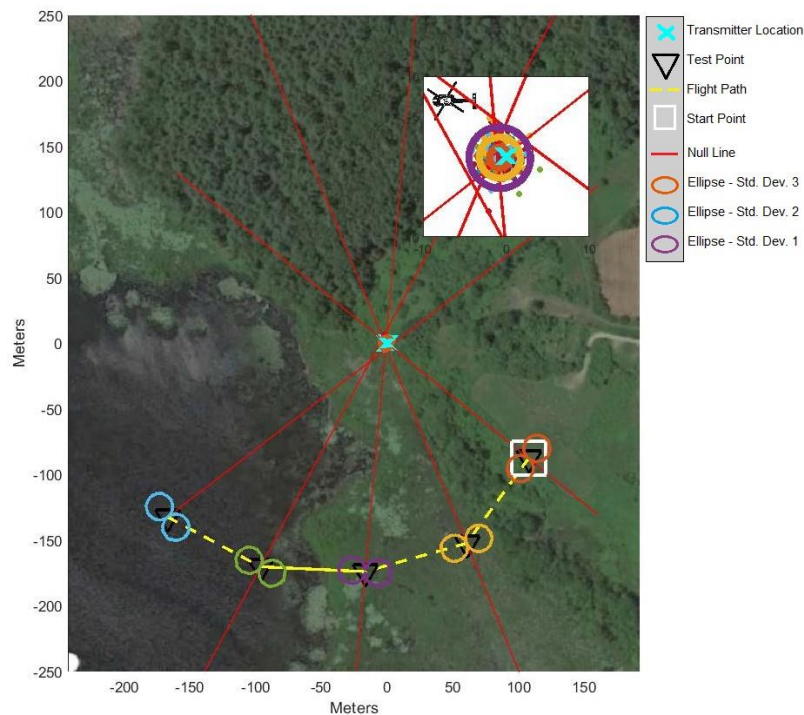


Figure 29a. Visual of entire simulated flight test for arc flight path.

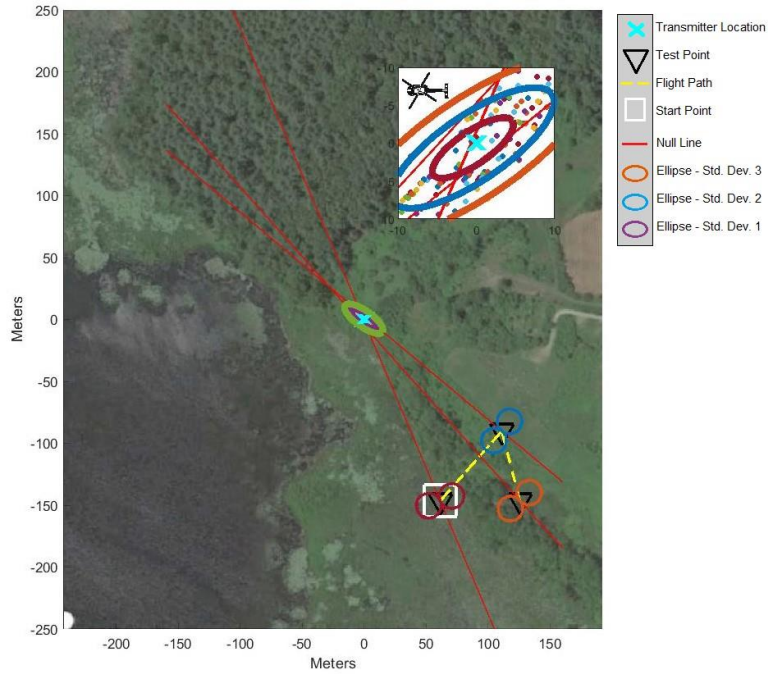


Figure 29b. Visual of entire simulated flight test for triangle flight path.

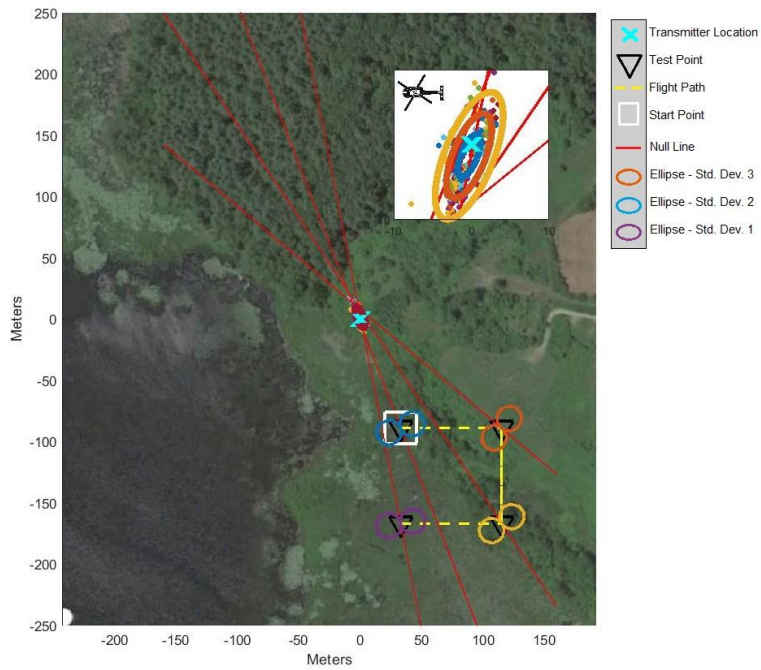


Figure 29c. Visual of entire simulated flight test for square flight path.

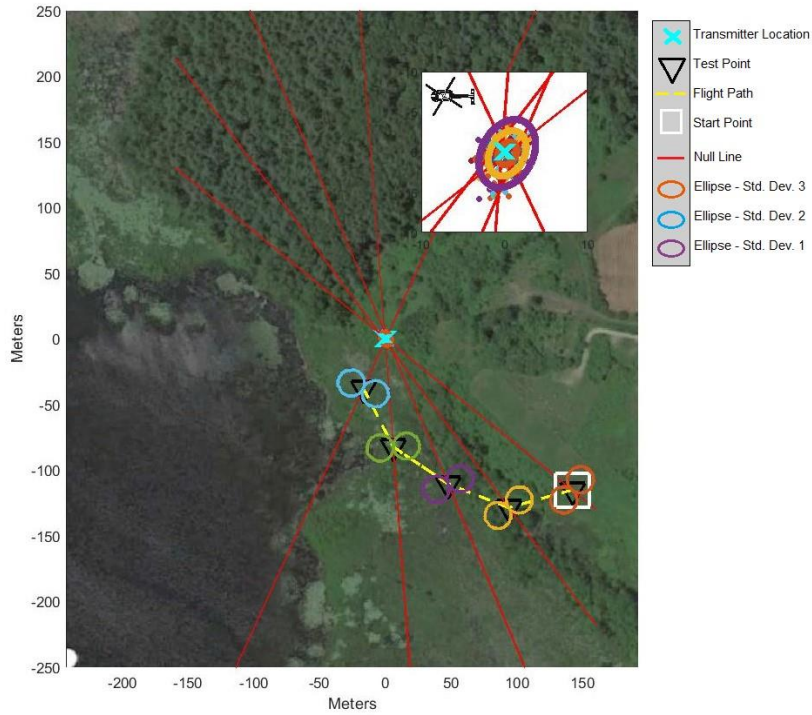


Figure 29d. Visual of entire simulated flight test for arc towards transmitter flight path

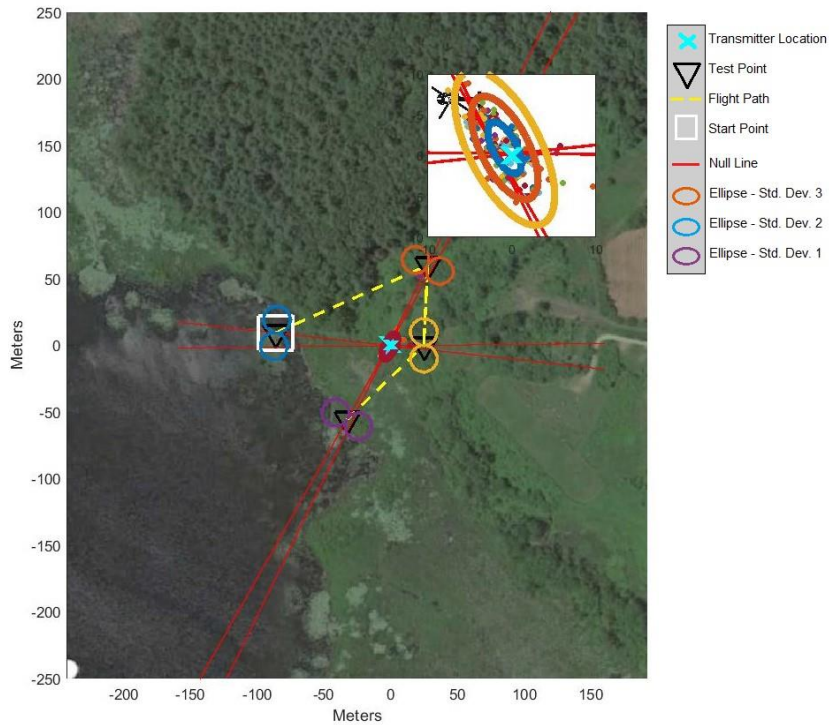


Figure 29e. Visual of entire simulated flight test for square around transmitter flight path.

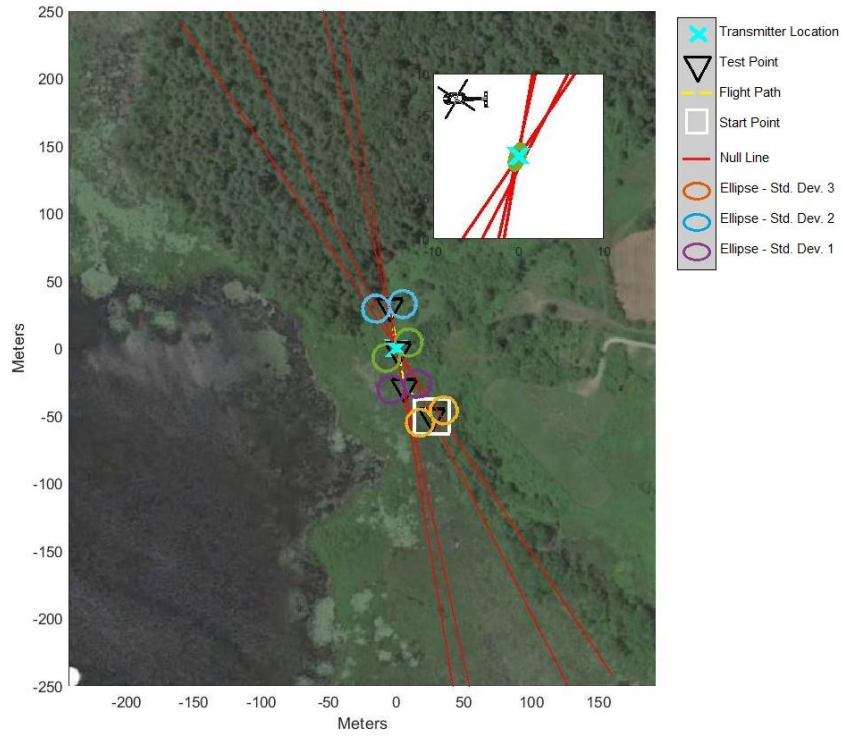


Figure 29f. Visual of entire simulated flight test for arc over transmitter flight path.

5.0 Results

After verifying the system's functionality indoors, our created system's functionality was examined through various outdoor tests. The first test completed was on the WPI football field. This validated the functionality of the transmitter and receiver from a far distance away with hopes to limit the multipath propagation effects of the lab. Additionally, it validated the collection of signal samples from the two receiver input channels and the saving of the time of each sample. This was followed by the flight tests, conducted using the drone to validate the entire system and complete the project's goal.

5.1 Football Field Test Results

After confirming the functionality of the receiver and transmitter in the lab, we completed a test on the WPI football field to verify the functionality of the receiver with respect to receiving signals from the transmitter, outputting IMU data, and outputting GPS data from a far distance away from each other and for a long period of time. The test was designed to imitate an actual flight test in a more controlled environment without the use of a drone and at the same height as the transmitter. The transmitter was placed on the tripod in the center of the football field on WPI's campus, shown in Figure 30, and shown as the blue circle in Figure 31.



Figure 30. Transmitter setup for the football field test.

Using a USB, a laptop with the transmit code was connected to the transmit radio to start the transmission. Then, the same USB and laptop was connected to the receiver radio to begin the receiving code. With a 10 second delay, the team was able to get into position before the receiver actually started receiving data. During these 10 seconds, one team member went to the test path's starting point holding the receiver-antenna device while the other team members stood away from the transmitter and away from the path to attempt to limit multipath propagation. The following image, Figure 31, shows the path that the person holding the receiver took during the test, starting at the bottom right point. Each point in the path is shown as a black triangle and the test path is shown as the dashed red line.

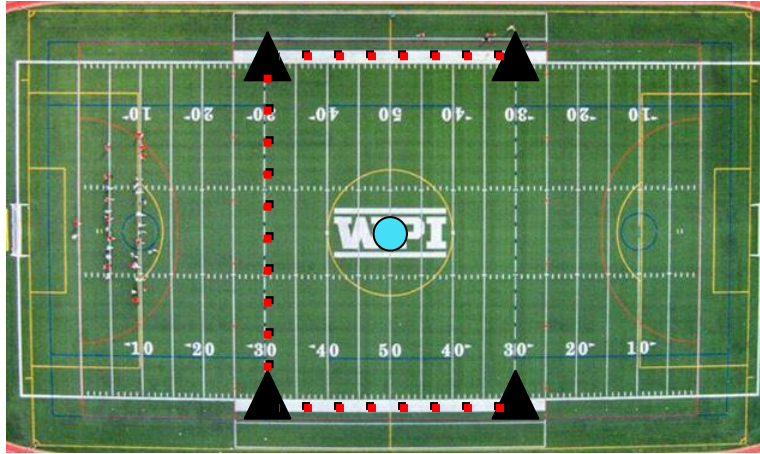


Figure 31. Football field test path.

The following graphs show the simulated results for the football field test. The first image, Figure 32, shows the confidence ellipses created for the test. Interestingly, even though the test points were close to the transmitter, the confidence ellipses are relatively large, especially when compared to the ellipses of the simulated arc test. Figure 33 shows the entire simulated path.

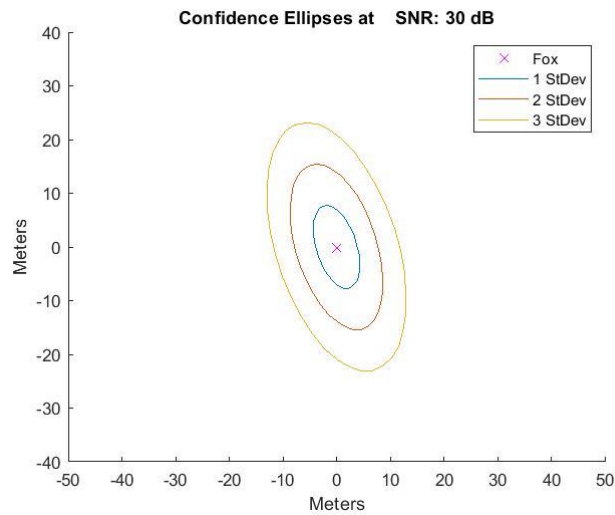


Figure 32. Football field test corresponding confidence ellipses.

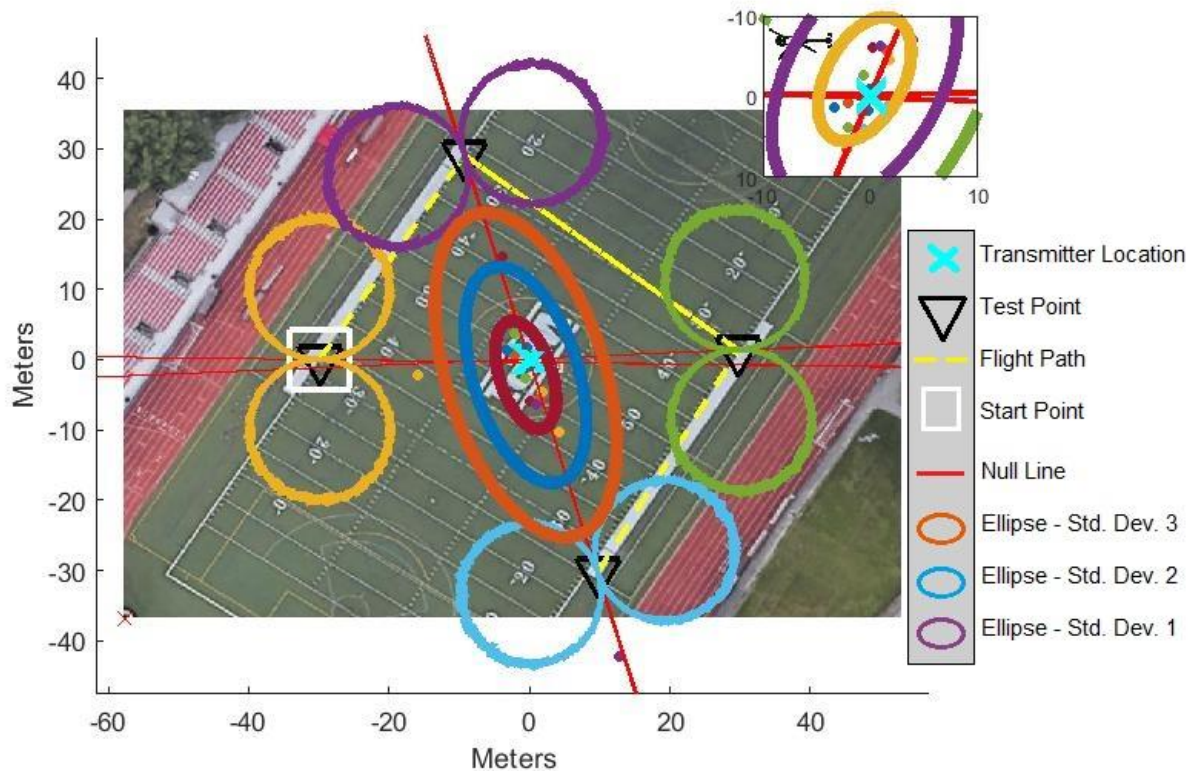


Figure 33. Visual of simulated football field test.

The collected data from the actual test was passed through the C program to convert the binary information to ASCII characters and organize the data by timestamp. The following graph shows the result of the collected data being graphed in Matlab, Figure 34. The first plot shows the received signal magnitude. The received signal remains around a constant magnitude in between test points. At each point, the receiver was spun in a 360-degree circle. When the signal magnitude nears zero, the radio is positioned at a null in relation to the radio. The second graph shows the radio's orientation throughout the test, using the IMU's yaw data. The yaw data ranged from π to negative π , depicting a full circle. At each point of the test path, the radio was spun in at least two circles. The last graph shows the speed of the radio, collected through the GPS antenna. Through this test we learned that the GPS antenna takes at least 30 seconds to sync with GPS satellites and therefore we were not able to collect the radio's speed for the first test point. However, after the first point, the speed slows to nearly zero at each point. This was useful because it shows when the radio is at a point of the path where the collected data is relevant to our test so that a polar plot can be created at the point.

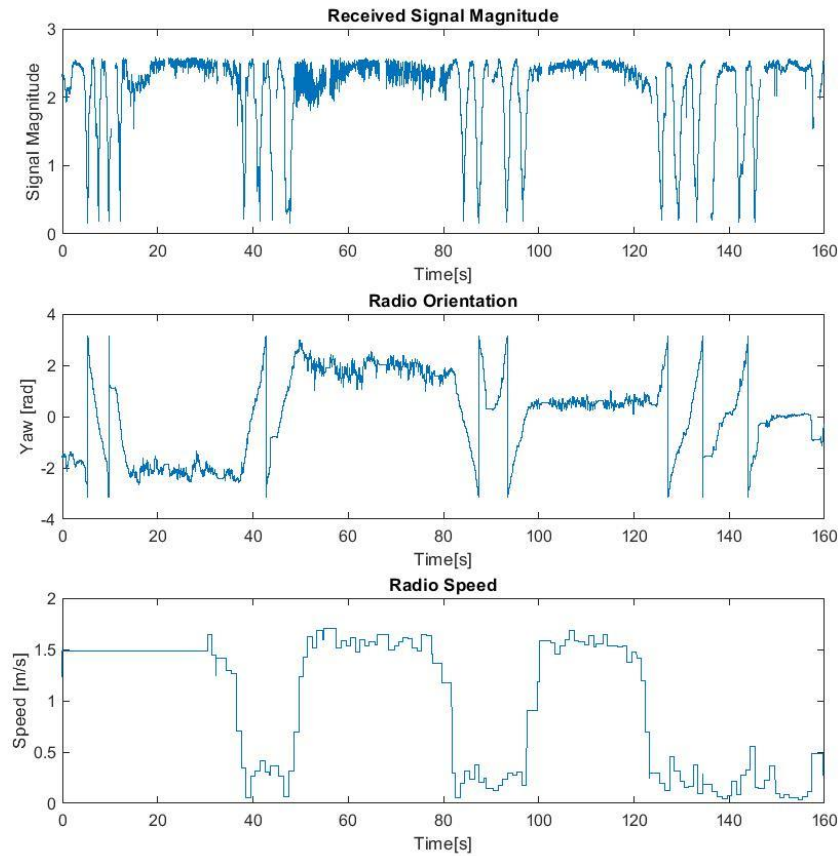


Figure 34. Football field test collected data graphed.

Using the collected data, we were able to create a graph shown in Figure 35 that appears similarly to the simulated graph of Figure 33. Shown in this graph are the polar plots created at each point of the path and the null line for each. The lines intersect in the center of the plot where the transmitter was located. The top left polar plot appears to be the farthest from the actual location of the transmitter. This may be due to a group of soccer players who were using that part of the field and contributed to multipath propagation. While the polar plots are not as smooth as the simulated data, the null lines can still be drawn with a high level of certainty for the location of the transmitter. Using a Matlab script, the convergence of the null lines was used to determine an estimated location for the transmitter. For this test, the estimated transmitter location was $(-71.81171, 42.27398)$, only 8.887m from the actual transmitter location.

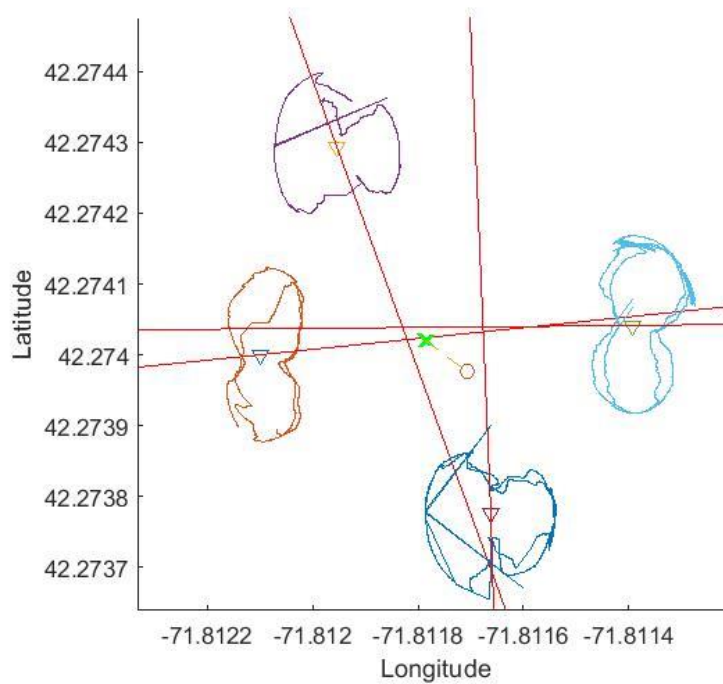


Figure 35. Football field test resulting polar plot graph.

5.2 Flight Test Results

In order to fully verify our created system, we conducted multiple flight tests at the CMRCM field as outlined throughout the methodology. Due to a time constraint and unpredictable weather, we were only able to test the arc, triangle and arc over the transmitter flight paths. The test was completed around noon, with a temperature of 49 degrees Fahrenheit, a wind speed of 18mph SSW and wind gusts up to 28mph. The following image captures the transmitter’s position during the test, located at latitude 42.3070, longitude -71.61340. Shown in Figure 36, the transmitter was placed on a tripod with an antenna facing straight upwards.



Figure 36. Transmitter setup for field tests.

The following graphs, Figures 37-39, show the received signal with respect to the drone's orientation for each of the three flight paths tested. These graphs show the proper functionality of the receiver code and the repeatability of the collected data.

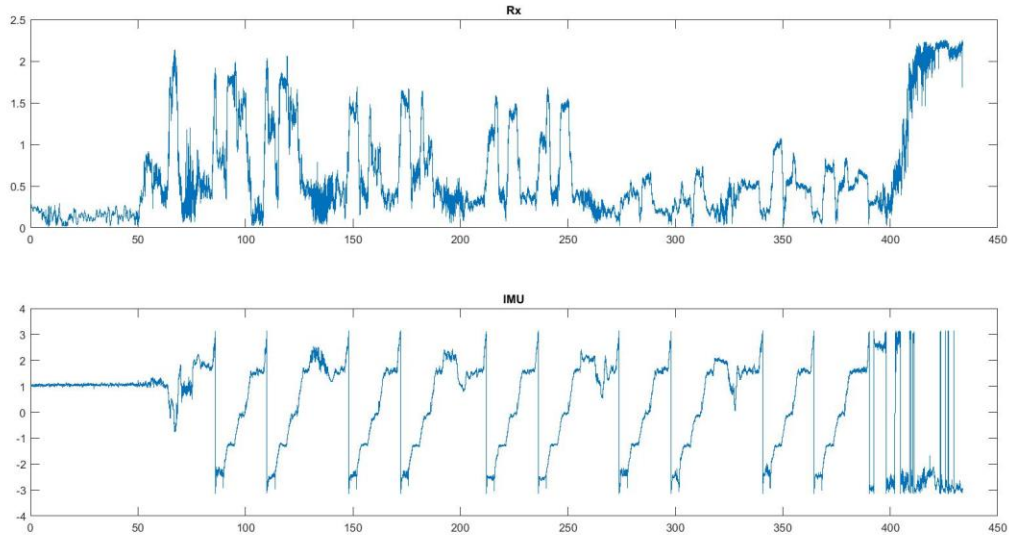


Figure 37. Arc test graphed collected data.

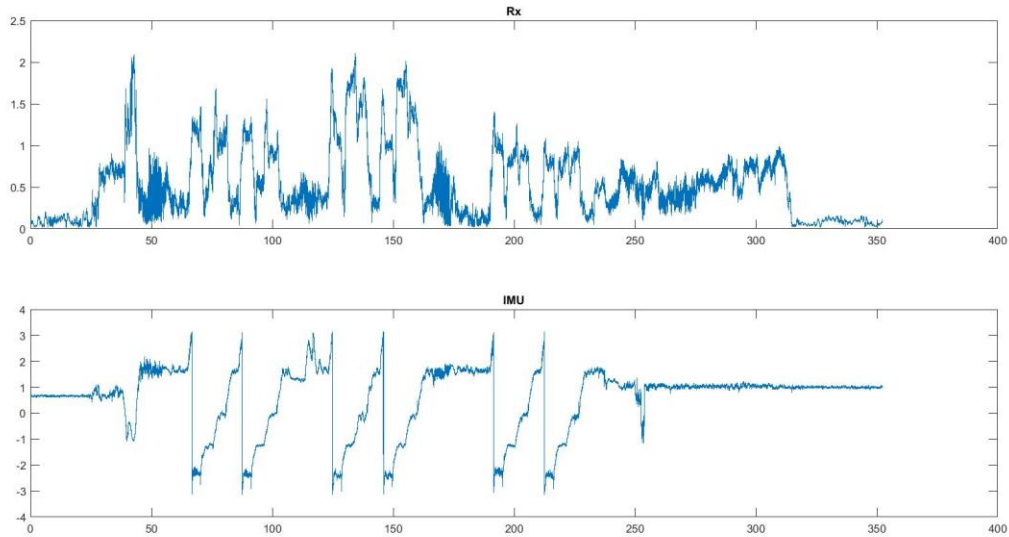


Figure 38. Triangle test graphed collected data.

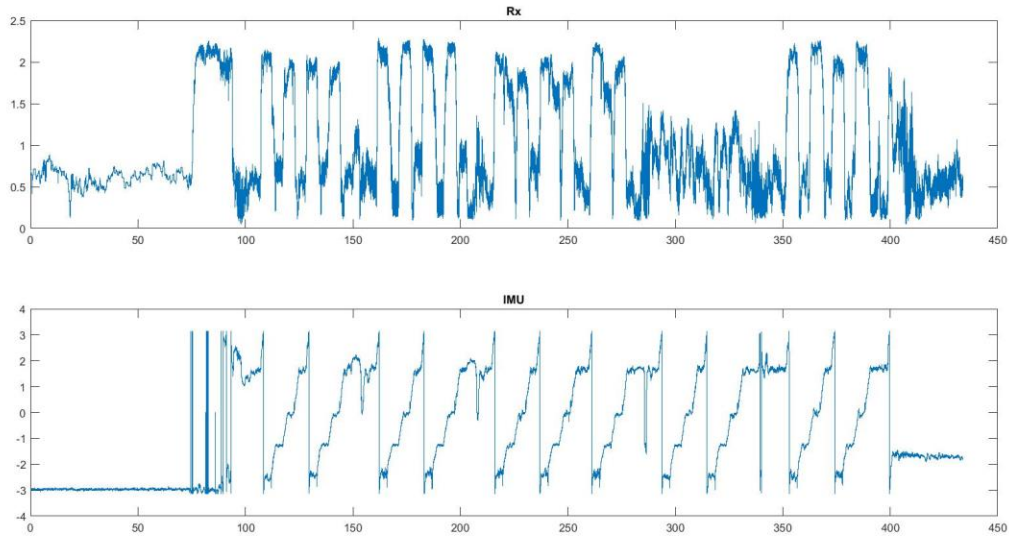


Figure 39. Arc over transmitter test graphed collected data.

Using the collected data, we were able to create polar plots at each point of the flight path for all three tests, shown in Figures 40-42. The polar plots use the orientation of the drone as the angle, and the signal magnitude as the magnitude. Points that are closer to the actual transmitter appear to have a larger polar plot because they have an overall higher magnitude. The red line drawn shows the line of null. The circle represents the predicted location of the transmitter while the “x” shows the actual location of the transmitter.

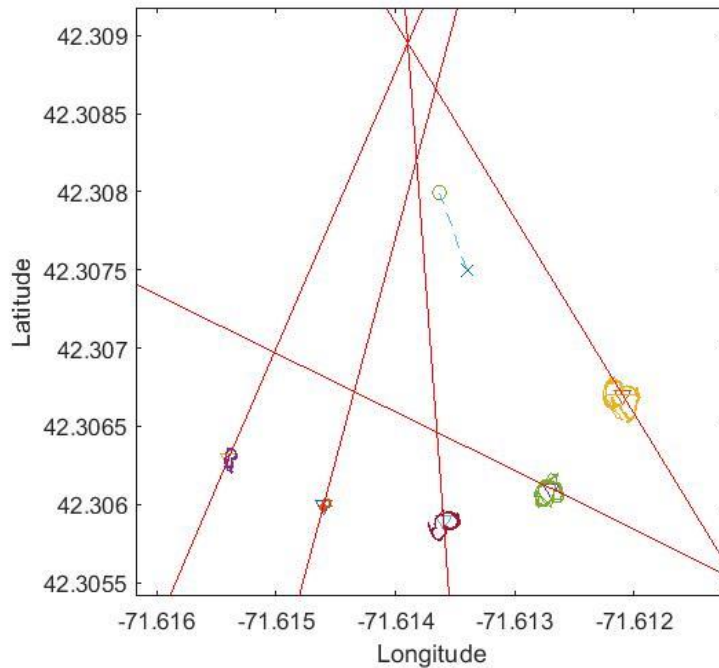


Figure 40. Arc flight resulting polar plot graph.

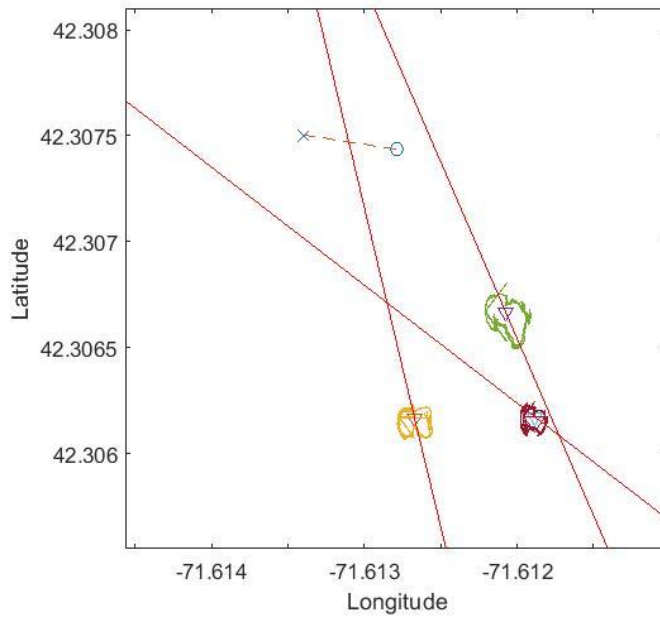


Figure 41. Triangle test resulting polar plot graph.

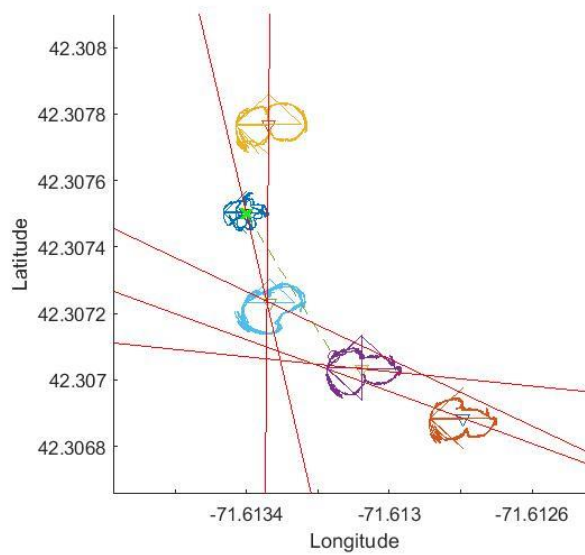


Figure 42. Arc over transmitter resulting polar plot graph.

After drawing the null line for each polar plot, the estimated location of the transmitter was determined using the same Matlab script described previously for the football field test. Then, the distance between the estimated transmitter location and the actual transmitter was calculated. The following table shows the results for each flight path's estimated transmitter location, along with the calculated distance.

Table 5.2.1 Results of the Flight Tests

Name of Flight Path	Estimated Tx Location (°W, °N)	Distance from Actual Tx [m]
Arc	(-71.61363, 42.30799)	31.3909
Triangle	(-71.61279, 42.30744)	68.238
Arc Over Transmitter	(-71.61316, 42.30707)	30.9524

The arc over the transmitter test appears to have the closest predicted distance from the actual transmitter, followed by the arc test and then the triangle test. These results can then be compared to the simulated results to determine how well our system performed.

6.0 Discussion

The goal of this project was to create a system consisting of a UAV, a transmitter and a receiver that models the search for a missing person in a situation where there is no cellular service either due to lack of infrastructure or natural disaster. A transmitter radio was used to model the cell phone. Our system was tested at various stages of the project to verify functionality, including a wired test, a turntable test, a football field test, and multiple flight tests. The results of each test acted as a building block for the following test.

To understand the accuracy of both the flight test and the simulation, the results from the flight test were compared to the simulated results. The simulation of each flight path shows that the arc over the transmitter would result in the most accurate estimated transmitter location. From our actual results, it appears that the arc test and the arc over the transmitter test have very similar results. These tests are able to estimate the location better than the triangle test, by nearly 40 meters. All three of the tests have a result that falls outside of their respective simulated confidence ellipses. This shows that the simulation did not account for enough noise that a real-world example might be exposed to. This can also show the need for additional flight tests to be completed on days with less wind.

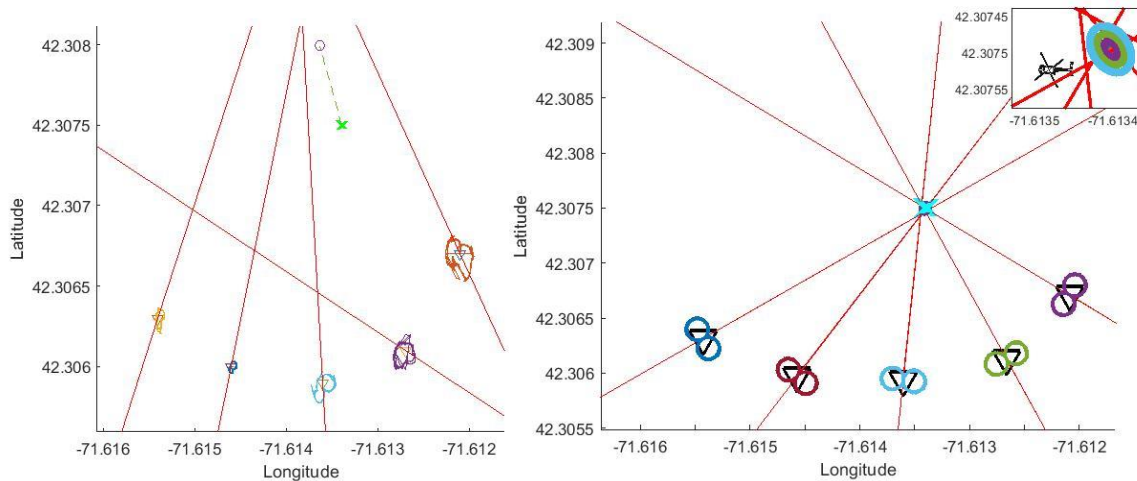


Figure 43. Comparison of actual (left) and simulated (right) arc test results.

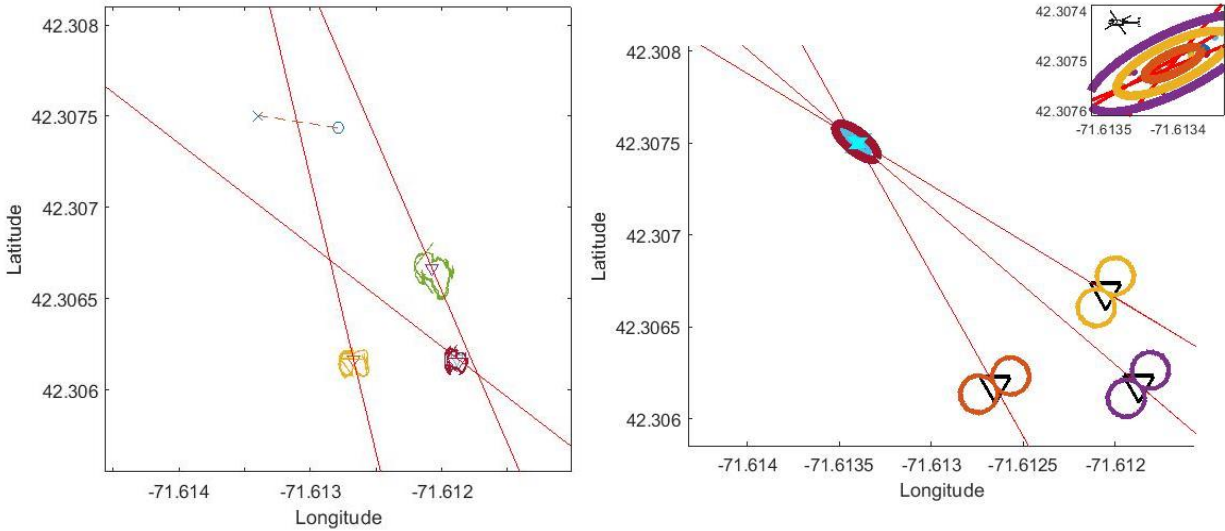


Figure 44. Comparison of actual (left) and simulated (right) triangle test results.

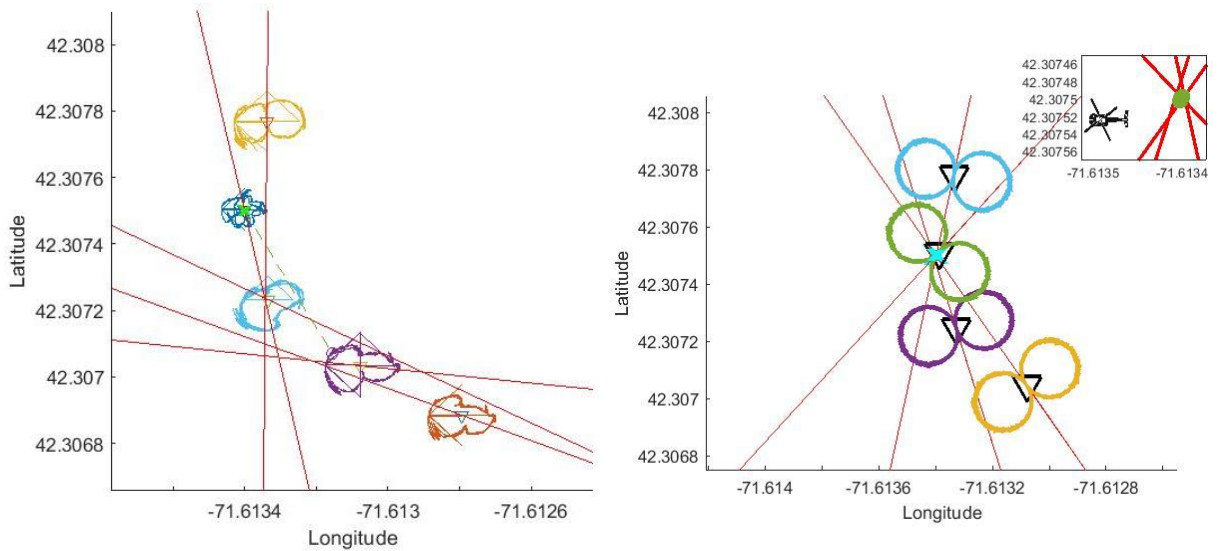


Figure 45. Comparison of actual (left) and simulated (right) arc over Tx test results.

As shown in the previous figures, there were discrepancies between the simulated and actual results but not in the way we were expecting. Our simulation was based around gaussian noise but the real-world tests have very random noise and a significant number of artifacts that were well out of proportion. When analyzing the data, we noticed that the IMU yaw value jumped to zero at multiple points. This could have contributed to some of the artifacts. One probable cause, as shown previously in Figures 17a and 17b, is the less consistent IMU calibration when the radio was positioned on the drone. Another possible source of error could be the landing gear on the drone. The landing gear stayed in the down position throughout the duration of the test. During the rotation of the drone the landing gear would pass in between the receiving antennas and the transmitter and could cause unknown variations in received signal strength and multipath propagation. Also, while the Matrice 600 Pro is capable of hovering in place, the high wind

speed and wind gusts could have forced the drone to move/tilt slightly at each test point. This causes the antennas to tilt/move as well causing possible reductions in received signal strength due to the propagation ellipse of the antennas.

7.0 Future Work

In a real-life scenario where a drone is using a radio to search for a missing person, it would be beneficial to be able to use collected data in real time to modify the flight path instead of using pre-programmed flight paths. This would allow the search to be more efficient and quicker. Future work can be completed on relaying information back to the ground using another antenna on a different frequency. A person on the ground can then use another radio to pick up the sent information. The information can then be viewed on Matlab while the test is still going on. By adjusting the flight path in real-time, the rescuer is able to reduce the size of the uncertainty ellipses by choosing the best next location to travel to. To fully automate the search for the missing person, additional work can be done to enable the drone to use feedback from the radio to design and update its own flight path. This would decrease the need for a person to be monitoring the received data from the receiver, creating a more efficient means to finding a missing person.

Additionally, it would be interesting to conduct a frequency sweep of the CMRCM field where the test was conducted to understand any nearby devices that may be transmitting at similar frequencies as our radio. A frequency sweep could also be conducted on the WPI football field to ensure a more thorough understanding of the frequencies being used there as well. Similar frequencies as the ones used in our system could have contributed to disturbance in the received signal, which in turn altered our polar plots and affected the calculations for the estimated transmitter location.

Other than possible interference from radio signals, the results were heavily reliant on an accurate measure of orientation. Further testing of the IMU on the receiver radio in relation to the magnetic interference of the drone's motors would be beneficial in improving our initial results. Better yet, the accurate positional and orientation data used by the drone could be merged with the data already collected by the receiver radio to get even more reliable results.

Furthermore, a mobile phone application can be created that can send an SOS signal which would be received by the radio on the drone in the case of an emergency. This would eliminate the need for our transmitter radio and would more closely resemble a real-life situation.

While further work would be required to more efficiently search for a missing person, we are confident that if implemented our system could be used as a building block to help in an emergency situation.

References

- [1] D. Coto, "Bahamians Look for Survivors With 1,300 Missing After Dorian," *Time*, 13-Sep-2019. [Online]. Available: <https://time.com/5676851/bahamas-hurricane-dorian-missing-people/>. [Accessed: 23-Sep-2019].
- [2] "DJI Urges Authorized Drone Users To Prepare For Hurricane Dorian," *DJI Official*. [Online]. Available: <https://www.dji.com/newsroom/news/dji-urges-authorized-drone-users-to-prepare-for-hurricane-dorian>. [Accessed: 23-Sep-2019].
- [3] K. B. Sandvik and K. Lohne, "The Rise of the Humanitarian Drone: Giving Content to an Emerging Concept," *Millennium: Journal of International Studies*, vol. 43, no. 1, pp. 145–164, 2014. doi: 10.1177/0305829814529470
- [4] P. J. Mosterman, D. E. Sanabria, E. Bilgin, K. Zhang, and J. Zander, "Automating humanitarian missions with a heterogeneous fleet of vehicles," *Annual Reviews in Control*, vol. 38, no. 2, pp. 259–270, Oct. 2014. doi: 10.1016/j.arcontrol.2014.09.008
- [5] "United States Frequency Allocation Chart," *United States Frequency Allocation Chart*, Jan-2016. [Online]. Available: <https://www.ntia.doc.gov/page/2011/united-states-frequency-allocation-chart>. [Accessed: 23-Sep-2019].
- [6] D. Yates, A. S. Holmes, and A. J. Burdett, "Optimal Transmission Frequency for Ultralow-Power Short-Range Radio Links," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 7, pp. 1405–1413, Jul. 2004. doi: 10.1109/TCSI.2004.830696
- [7] X. Li, "RSS-Based Location Estimation with Unknown Pathloss Model," in *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3626-3633, December 2006. doi: 10.1109/TWC.2006.256985
- [8] V. Giurgiutiu, "In Situ Phased Arrays with Piezoelectric Wafer Active Sensors," in *Structural Health Monitoring with Piezoelectric Wafer Active Sensors*, pp. 707-805, June 2014. doi: 10.1016/B978-0-12-418691-0.00013-7
- [9] D. Ogbe, D. J. Love, M. Rebholz, and P. Bidigare, "Efficient Channel Estimation for Aerial Wireless Communications," in *IEEE Transactions on Aerospace and Electronic Systems*, 2018. doi: 10.1109/TAES.2019.2894892
- [10] "Radio Spectrum Allocation," *Federal Communications Commission*, 07-May-2019. [Online]. Available: <https://www.fcc.gov/engineering-technology/policy-and-rules-division/general/radio-spectrum-allocation>. [Accessed: 29-Sep-2019].
- [11] "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area network--Specific requirements Part 11: Wireless LAN

Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Preassociation Discovery,” in *IEEE Std 802.11aq-2018 (Amendment to IEEE Std 802.11-2016 as amended by IEEE Std 802.11ai-2016, IEEE Std 802.11ah-2016, IEEE Std 802.11aj-2018, and IEEE Std 802.11ak-2018)*, pp.1-69, 31 Aug. 2018

doi: 10.1109/IEEESTD.2018.8457463

[12] IEEE Standard for Low-Rate Wireless Networks Corrigendum 1,” in *IEEE Std 802.15.4-2015/Cor 1-2018 (Amendment to IEEE Std 802.15.4-2015 as amended by IEEE Std 802.15.4n-2016, IEEE Std 802.15.4q-2016, IEEE Std 802.15.4u-2016, IEEE Std 802.15.4t-2017 and IEEE Std 802.15.4v-2017)*, pp.1-14, 9 July 2018

doi: 10.1109/IEEESTD.2018.8410916

[13] “Matrice 600 - Simply Professional Performance,” *DJI Official*. [Online]. Available: <https://www.dji.com/matrice600>. [Accessed: 29-Sep-2019].

[14] F. Kamrani and R. Ayani, “UAV Path Planning in Search Operations,” *Aerial Vehicles*, Jan. 2009. doi: 10.5772/6479

[15] S. A. Bortoff, “Path planning for UAVs,” *Proceedings of the 2000 American Control Conference*, Jun. 2000. doi: 10.1109/acc.2000.878915

[16] “Recreational Flyers & Modeler Community-Based Organizations,” *Federal Aviation Administration*, 13-Aug-2019. [Online]. Available: https://www.faa.gov/uas/recreational_fliers/. [Accessed: 30-Sep-2019].

[17] A. Montanha, A. M. Polidorio, F. J. Dominguez-Mayo, and M. J. Escalona, “2D Triangulation of Signals Source by Pole-Polar Geometric Models,” *Sensors (Basel)*, vol. 19, no. 5, p. 1020, Feb. 2019. doi: 10.3390/s19051020

[18] G. Chandrasekaran *et al.*, “Tracking vehicular speed variations by warping mobile phone signal strengths,” *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Seattle, WA, 2011, pp. 213-221. doi: 10.1109/PERCOM.2011.5767589

[19] H.-C. Yang and S. Sasankan, “Analysis of Channel-Adaptive Packet Transmission Over Fading Channels With Transmit Buffer Management,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 404–413, Jan. 2008.

[20] S. Kaul *et al.*, “Effect of Antenna Placement and Diversity on Vehicular Network Communications,” *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Diego, CA, 2007, pp. 112-121. doi: 10.1109/SAHCN.2007.4292823

- [21] Johnson, Todd. "How Is Carbon Fiber Made?" ThoughtCo. <https://www.thoughtco.com/how-is-carbon-fiber-made-820391> [Accessed October 1, 2019].
- [22] Latteier, Peter. "The Good, Better and Best Ways to Drill Carbon Fiber" Elevated Materials <https://www.elevatedmaterials.com/drilling-carbon-fiber/> [Accessed October 1, 2019].
- [23] "VERT900 Antenna" Ettus Research. <https://www.ettus.com/all-products/vert900/> [Accessed: December 10, 2019]
- [24] "Degrees Minutes Seconds to/from Decimal Degrees," *Federal Communications Commission*, 25-Aug-2016. [Online]. Available: <https://www.fcc.gov/media/radio/dms-decimal>. [Accessed: 11-Dec-2019].
- [25] "RTIMULib," *GitHub*. [Online]. Available: <https://github.com/RTIMULib/RTIMULib/tree/master/REllipsoidFit>. [Accessed: 11-Dec-2019].
- [26] S. Prager, "E312IMUTest," *GitHub*, 18-Nov-2019. [Online]. Available: <https://github.com/samprager/E312IMUTest>. [Accessed: 11-Dec-2019].
- [27] G. Hindman and A. C. Newell, "Reflection Suppression to Improve Anechoic Chamber Performance," *AMTA Europe*. 2006.
- [28] J. Munoz-Castaner *et al.*, "Your Phone as a Personal Emergency Beacon: A Portable GSM Base Station to Locate Lost Persons," in *IEEE Industrial Electronics Magazine*, vol. 9, no. 4, pp. 49-57, Dec. 2015. doi: 10.1109/MIE.2015.2484922
- [29] S. W. Smith, *The Scientist and Engineers Guide to Digital Signal Processing*. San Diego, CA: California Technical Pub., 1997.
- [30] The Editors of Encyclopaedia Britannica (Ed.). (2016, November 6). *Trilateration*. Retrieved March 24, 2020, from <https://www.britannica.com/science/trilateration>

Appendix A: GPS Antenna Product Specifications

Bingfu Vehicle Waterproof Active GPS Navigation Antenna with SMA Male Connector 3-5V DC for Vehicle Truck RV Motorhome Marine Boat Navigation Tracker Locator Car Stereo Unit

Specifications:

1. Dielectric Antenna

Center Frequency: 1575.42MHz \pm 3MHz

V.S.W.R: 2:01

Bandwidth: \pm 5 MHz

Impedance: 50

Peak Gain: > 3dBic based on 7x7cm ground plane

Gain Coverage: > -4dBic at -90 < 0 < 90 (over 75 volume)

Polarization: RHCP

2. LNA/Filter

LNA Gain Without cable: 28dB Typical

Noise Figure: 2.0dB

Filter Out Band Attenuation: (f0=1575.42MHz)

7dB Min f0 \pm 20MHz;

20dB Min f0 \pm 50MHz;

30dB Min f0 \pm 100MHz

V.S.W.R: < 2.0

DC Voltage: 3V to 5V

DC current: 7-10MA

3. Overall

Frequency Range: 1575/1609MHz

V.S.W.R: 2.0

Gain: 30 \pm 3dBic

Impedance: 50 Ohm

3. Mechanical

Weight: 60 gram

Size: 45x38x15mm

Cable: RG174

Length: 3m

Connector: SMA

Mounting: Magnet base

Housing: Black

4. Environmental

Working Temp: -40 to 85°C

Waterproof test: 4 hours under the tap, let the water rush over the housing;

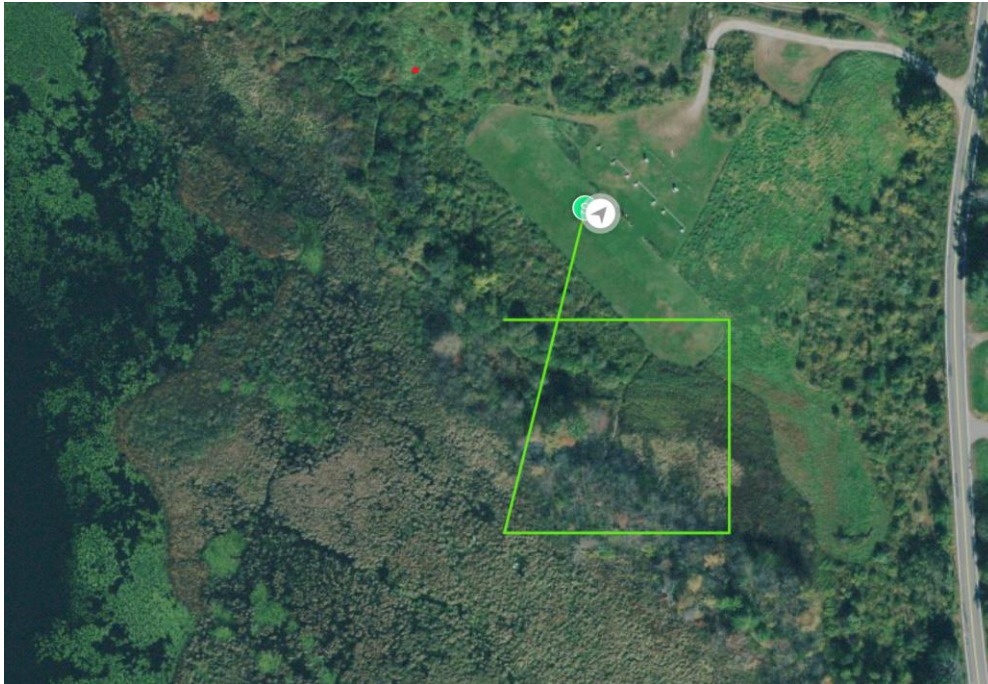
Vibration: Sine sweep 1g (0-p) 10 ~ 50 ~ 10Hz each axis

Humidity: 95% ~ 100% R

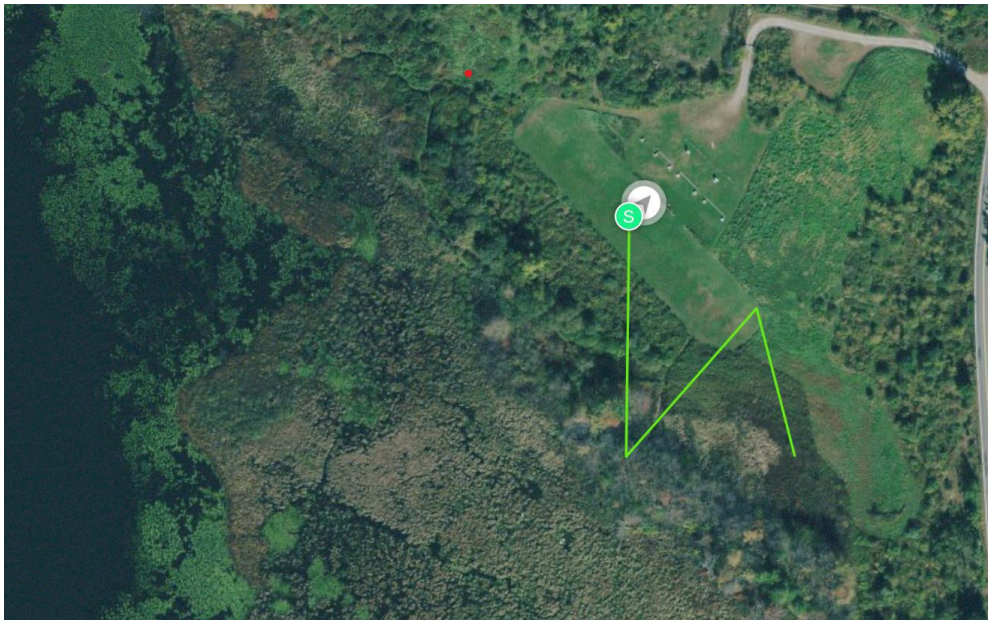
Package List: 1 x Antenna (As the Picture Shown)

Appendix B. Flight Path

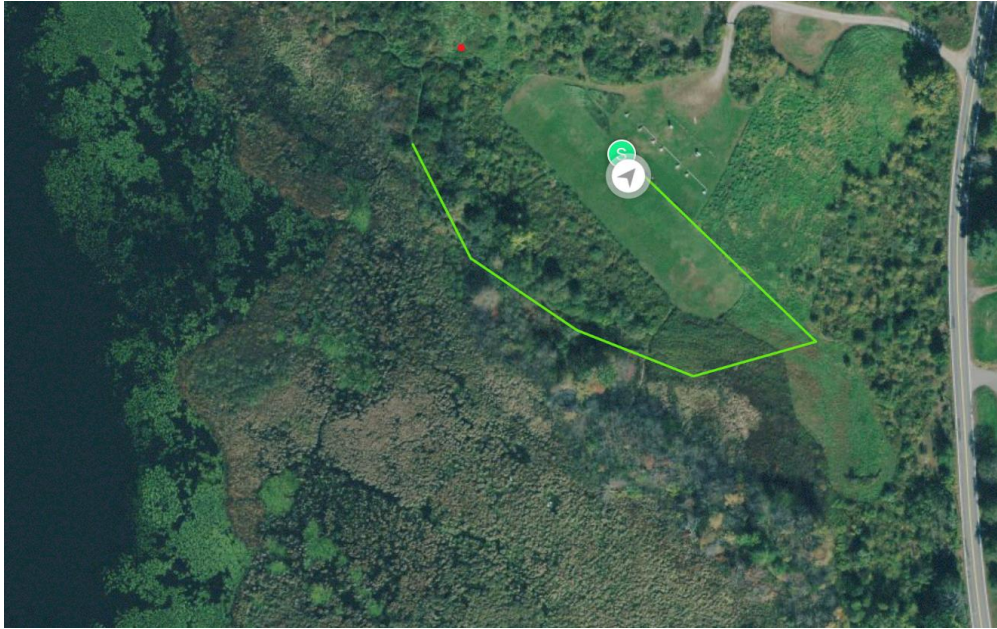
Square Flight Path:



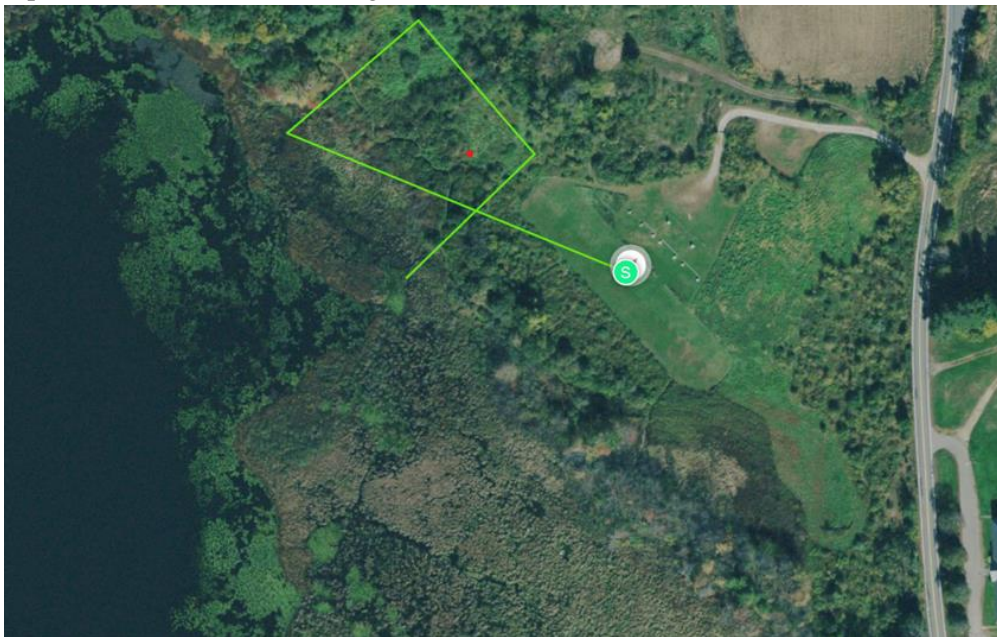
Triangle Flight Path:



Arc Towards Transmitter Flight Path:



Square Around Transmitter Flight Path:



Arc Over Transmitter Flight Path:

