Pseudo Doppler Direction Finding System for Localizing Non-Cooperative VHF Transmitters with a Hybrid UAS

William E. Gerhard III

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Masters of Science

in

Electrical Engineering

Pratap Tokekar, Chair

Joseph B. Baker

Alfred L. Wicks

June 27, 2019

Blacksburg, Virginia

Keywords: Radio Direction Finding, Pseudo Doppler Direction Finding, UAS, Wildlife Tracking

Copyright 2019, William E. Gerhard III

Pseudo Doppler Direction Finding System for Localizing Non-Cooperative VHF Transmitters with a Hybrid UAS

William E. Gerhard III

(ABSTRACT)

Current radio direction finding techniques are limited in flexibility and focus on specific applications. Commercial off the shelf systems exist for a wide range of applications from navigation to search and rescue and wildlife tracking. However these systems rely on commercially available VHF receivers and are limited in transmission modulation techniques and frequency ranges. The majority of these systems are expensive which places them outside the reach of most individuals while the current open source designs require specialized skills and knowledge to build. The goal of this work was to design a low cost system capable of determining the approximate location of a non-cooperative VHF transmitter that could easily be implemented on a variety of unmanned systems. One unmanned aerial system was designed, built, and evaluated. Existing open source hardware and software systems were utilized for the development of the pseudo Doppler direction finding system, and work was conducted utilizing recursive Bayesian techniques to estimate the VHF transmitter's location. Results and explanations of system behaviors are presented along with limitations and possible modifications to improve performance and reliability.

Pseudo Doppler Direction Finding System for Localizing Non-Cooperative VHF Transmitters with a Hybrid UAS

William E. Gerhard III

(GENERAL AUDIENCE ABSTRACT)

Radio direction finding uses specialized radio equipment to determine the direction that a radio signal is coming from. Commercial systems are often expense, and existing hobbyist designs require specialized skills, and both are not flexible in application or frequency. The same is true for commercially available drones, which tend to be expensive or face other limitations. In this work a low cost radio direction finding system that uses easily found open source hardware and software was built and evaluated, along with a low cost unmanned aerial system. Then using the data collected, a computer algorithm was tested that could estimate the transmitting radio's location. After testing it was determined that all systems did work, but still had room for improvement. Future steps and system modifications are presented that could improve the system's performance.

Dedication

To my Mom, Dad and all who helped me along the way, without whom I would not have gotten this far.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unmanned aerial systems (UAS) have many applications in field biology, and their potential as a tool for wildlife tracking in rugged, difficult to access terrain has been demonstrated in several studies. Linchant et al. [20] (2015) provide a comprehensive survey of how UAS are changing the monitoring and tracking of populations of wildlife. UAS ranging from military grade to hobbyist are being used around the world to supplement or replace manned aircraft. This has huge benefits ranging from reduced logistical burdens of supply and managing aircraft in remote areas, to lower operational costs and reduced risks for field biologists. [20] Reducing risk is a huge benefit, as the current leading cause of death for field biologists is aircraft accidents. [26] UAS also allow different payloads, ranging from IR cameras to radio tracking equipment, to be flown low and slow, producing high spatial and temporal resolution data that can be shared and reviewed with other researchers. [20]

Beyond simply replacing manned aircraft, UAS are a potent tool for the active tracking and localization of wildlife. The ability to monitor where wildlife go, how they move and react to changes in the environment, and other behaviors not normally observed are critical to conservation efforts and to the management of wildlife populations for future generations. The vast majority of wildlife tracking is done through simple VHF radio collars that transmit an auditory tone.

Several recent works have explored using the unique capabilities offered by UAS combined with radio direction finding techniques to track wildlife. Vonehr et al. [30] (2016) utilized

Figure 1.1: Example of a VHF Tracking Collar

a rotating multirotor UAS with a directional antenna to determine the angle of incidence of an incoming signal. In the same work the premise of a pseudo doppler direction finder was investigated but determined to be too complex to implement. Bayram et al. [11] (2017) expanded upon the idea of using directional antennas on-board multirotor UAS by creating a path planning algorithm that enabled three UAS to work together to triangulate the signal source and calculated its position. Their algorithm was able to estimate the signal's source with positional errors of 3.7 meters, 10.7 meters and 18.2 meters, depending on the starting parameters and the experiment setup. More recently Bayram et al. [12] (2018) explored the use of a single UAS with a directional antenna to estimate the position of a signal source. By using an online strategy to choose measurement locations, the UAS was able to localize the signal source to a user specified 10 meters of error.

Another method used for localization of tagged wildlife is measuring the characteristics of the received signal and extrapolation the signal's origin from the signal's gradient. Bayram et al. [10] (2016) developed a local search algorithm that drives the UAS towards the signal source through measuring characteristics of the received signal. By using this algorithm in an open field, the signal source was able to be located with a positional error of 50 meters.

Other work in this area done by Desrochers et al. [15] (2018) produced similar results with tests conducted in boreal forests of Canada, but with a larger error of 94 meters.

Radio direction finding as a genre is as old as the radio itself and presents a unique data set for localization applications. Radio direction finding is used in commercial aviation for navigational assistance as well as in emergency response, search and rescue, wildlife tracking and item recovery. [8] Amateur radio enthusiasts have been using radio direction finding for decades in events such as fox hunts (searching for a hidden transmitter) and geocaching. [2] The most common radio direction finding technique is rotating a highly directional antenna in a circular pattern and measuring the signal gain. The direction that has the highest gain is the direction oriented towards the signal source. However it is not the only method of radio direction finding.

Broadly speaking, there exist two categories of radio direction finding techniques, phase comparison and magnitude comparison. Within the phase comparison family of radio direction finding there exists two primary methods, Doppler (Or Pseudo Doppler) direction finding and interferometery. The other two methods are magnitude based, the Watson-Watt technique and the above mentioned directional antennas. All of theses techniques allow the angle of arrival of an incoming RF (radio frequency) wave to be measured, with various levels of hardware complexity, accuracy, and cost. [8]

Commercially available unmanned systems follow in the same vein, and all are compromises between complexity, operational ability, and cost. There exist three configurations of unmanned aerial systems: Fixed Wing, Rotor or Multirotor, and VTOL (vertical take-off and landing). Fixed wing platforms are traditional aircraft, featuring a fixed aerofoil that provides lift and some form of propulsion to provide forward velocity. Typically fixed wing aircraft are the most efficient, but suffer several operational drawbacks. Rotor based aerial systems have become popular in recent years, and use one, or more, rotating aerofoils to

provide lift. The company DJI is a worldwide provider of multirotor UAS, such as the S900 which is a high capacity UAS used extensively in aerial photography and academic research. [1] Hybrid systems are a mixture between a tradition fixed wing platform and that of a multirotor. Hybrid systems utilize a fixed aerofoil for lift during forward flight, but use at least one rotating aerofoil to take off and land vertically. This style of unmanned system is used by Wing, an experiment drone delivery company, since it offers both the operational benefits of the rotor system and the longer flight duration and range similar to that of a fixed wing system. [7]

Radio direction finding combined with UAS create a rich, if noisy, data set. This data set, combined with recursive bayesian estimation (RBE) techniques, can be used to localize the signal source. RBE techniques are a family of approaches that estimate an unknown probability density function using a mathematical model recursively over time using incoming measurements. Theses techniques are broadly used in everything from communications to computer vision. [16] Broadly, there are three main categories of RBE techniques that trade between their ability to handle non-linearity and computational efficiency. Kalman filters typically are the most computationally efficient, but do not handle non-linearity well. Particle filters or Sequential Monte-Carlo filters are the least computationally efficient; but they can be adapted to a wider range of problems. Grid based estimators fall somewhere in between. [16]

The focus of this work was threefold, to develop a low cost radio direction finding system, a low cost hybrid UAS, and implement an RBE filter to use the data from the direction finding system and the UAS to estimate the location of the transmitter. A motivation for this work was the unique needs of a system to track bats. Due to the high speeds and long distance bats travel, a fixed wing UAS was needed both to extend the system flight duration and range as well travel with the swarm. However due to heavy vegetation inhabited frequently

by bats, the ability to take off and land vertically was required, resulting in a Hybrid UAS. Chapter 2 covers the background information regarding radio direction finding techniques, UAS platforms, and RBE techniques. In addition a more in depth exploration of some related work is given. Chapter 3 provides in depth information regarding the construction of the Hybrid UAS, PDDF systems, GNURadio software, and the particle filter. Chapter 4 provides a per unit validation and performance of each component, simulations of the UAS's impact on the PDDF and the particle filter and gives preliminary results of field testing.

Figure 1.2: Finished Hybrid UAS

This work contributes to the field through

- Development of a low cost, easy to build hybrid UAS
- Development of a low cost software defined radio doppler direction finder using existing products
- Validation that a simplistic particle filter with angle of incidence measurements and GPS information can localize a non-cooperative transmitter.

In addition, through the development and use of the PDDF, this work enables orders of magnitude more angle of incidence data to be collected than any other method discussed,

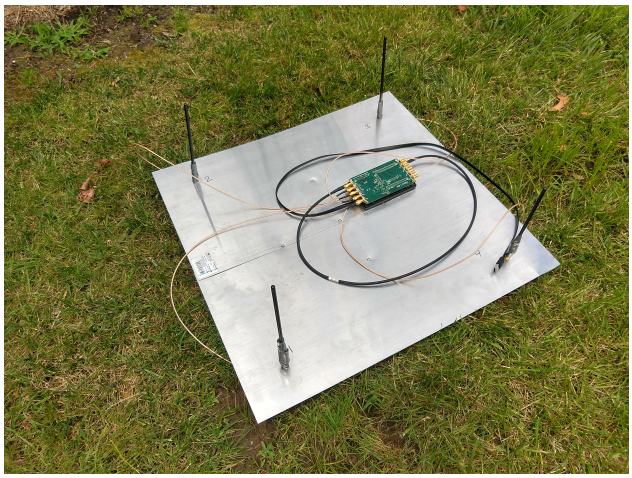


Figure 1.3: Finished PDDF

which speeds up sampling, is implementable using a fixed wing UAS, instead of a Multirotor UAS, all of which improves estimator performance and flight range and duration.

Chapter 2

Background

This work covers a broad range of topics and this section covers appropriate background information required to understand design choices made in Chapter 3. A brief survey of radio direction finding methods is covered along with related works that use those methods. Fixed wing, multirotor, and hybrid UAS weakness and strength are discussed and finally high level concepts around recursive bayesian estimation techniques are covered.

2.1 Radio Direction Finding

Due to the nature of how electromagnetic waves propagate, a variety of methods can be utilized in order to determine the direction of propagation of that wave. Some of these methods have been around since the discovery of electromagnetic waves, while others required the recent advancements in computers and digital processing in order to become practical.

2.1.1 Directional Antennas

By far the simplest method of determining the direction of propagation of an electromagnetic wave is through the use of directional antennas. Directional antennas, through the physical design of the antenna itself, tend to reject signals that do not conform to some physical parameter, such as direction or polarization. Polarization is property of transverse waves

that specifies the geometrical orientation of the oscillations. The earliest direction finding systems, polarization direction finders, used polarization to determine the angle of incidence. In these designs a rotatable electric or magnetic dipole, whose axis was in line with the direction of the electric or magnetic field, was rotate. [8] By moving the antenna, the polarization of the incoming signal could be determined and used to find the azimuth or direction of incidence. The best known example of this kind of directional antenna direction finding system is the rotating loop direction finder, which was in widespread use during the first half of the twentieth century. [8]

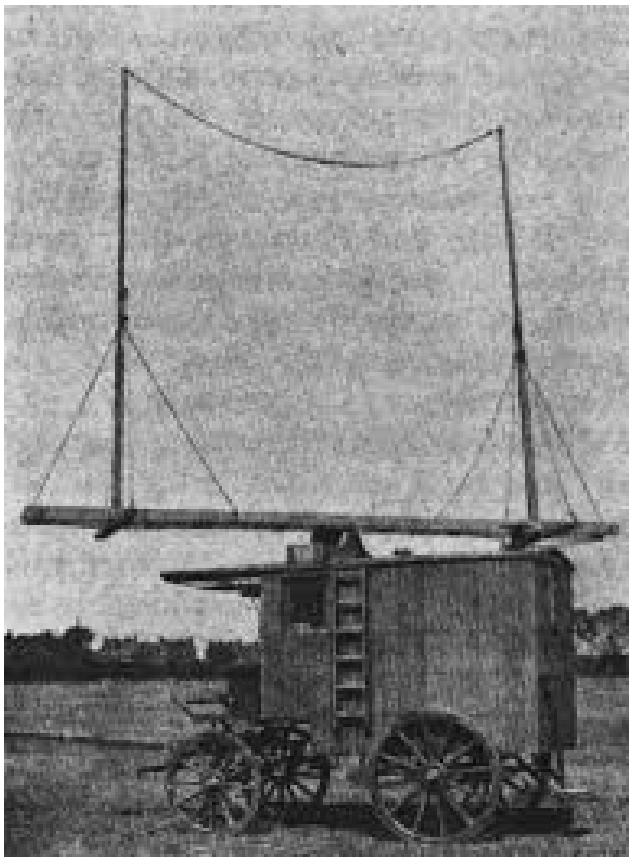


Figure 2.1: Mobile rotating loop direction finder for military use, approx 1918 [17]

A more modern approach is to use a directional antenna. These antennas increase the signal strength in one direction, while decreasing it in other directions. The most common direc-

tional antenna is the Yagi–Uda antenna. Developed by Shintaro Uda and Hidetsugu Yagi of Tohoku Imperial University, Japan in 1926, the Yagi–Uda antenna consists of multiple, parallel elements, typically quarter dipole antennas, spaced in a line. [31] Only one of these elements is connected to the receiver or transmitter, while the rest, so called "parasitic elements", are left unconnected. These "parasitic elements" are responsible for shaping the gain of the antenna system by constructively and destructively interfering with the electromagnetic waves to maximize the signal in one direction while minimizing it in all others. By rotating the Yagi-Uda antenna and measuring the received signal strength, the direction of incidence is determined. [8]

The advantage of the directional antenna approach is only one RF processing channel is required since only one antennas output is being measured. However both directional antennas methods suffer the same drawback. For any directional antenna approach, the physical antenna must be rotated mechanically in order to determine the direction of incidence. Since the directionality is based on the physical design of the antenna, the only way to measure the angle of incidence is to move the antenna. While easily accomplished manually, rotating the antenna precisely and accurately quickly enough requires expensive and heavy components, such as stepper motors or servos, or requires a platform that can rotate in place. The accuracy of a directional antenna is only as good as how accurately it can be rotated. Error in the rotation directly leads to errors in determining the angle of incidence. This limits the utility on unmanned systems or of the unmanned system itself. Yet directional antennas onboard UASs is a popular area of research. [28] [30] [11] [12] Work done by Vonehr et al. [30] (2016) showcases some of the limits created by the use of directional antennas on UASs.

Vonehr et al. [30] (2016) implemented a radio direction finding method utilizing a multirotor UAS and a Yagi-Uda antenna, and evaluated its performance for tracking wildlife. Their system was composed of a software defined radio (SDR) and embedded linux computer which

calculated signal to noise ratio (SNR) as a function of UAS yaw. A SDR is a radio system where a some portion of the traditional analog components have been replaced with software that fulfills the same purpose. As a result a SDR is more flexible that a traditional radio, as its core functionality can be changed by changing the software. This enables different modulation, demodulation or filtering techniques to be applied to the same hardware. Since the UAS itself was required to rotate in place to determine the direction of incidence, UAS platforms are limited to systems that have the ability to hover, which significantly reduces system range and run time. In addition, the system's accuracy is directly tied to the beamwidth of the Yagi-Uda antenna selected. The narrower the gain pattern, the more accurate the final direction of incidence will be, but the system must hover longer and rotate slower to ensure the signal is detected. The Vonehr et al. [30] (2016) experiment utilized a Yagi-Uda antenna with a 40 degree main beam lobe (angle over which the signal is amplified), which required a forty second hover and rotation evolution per sample in order to ensure detection of the signal. [30] Work done by Bayram et al. [12] (2018) and Bayram et al. [11] (2017) also used this directional antenna approach to localize targets. Bayram et al. [11] utilized three UAS equipped with Yagi-Uda antennas that worked together with an online algorithm to optimally place each UAS to minimize the error in the estimated position and achieved errors of 18.2 meters, 3.7 meters and 10.7 meters depending on the starting parameters of the algorithm. However each angle of incidence measurement took 2 minutes to perform. Bayram et al. [12] (2018) used a similar approach, but with a single UAS instead of three, and was able to locate a stationary target within 10 meters, with each angle of incidence measurement taking 2 minutes to perform. Since most multirotor UAS's flight time is under 30 minutes, all of theses approaches result in a trade-off between samples taken and area covered. Most importantly, fast moving wildlife, such as bats, can conver a large distance in 2 minutes.

2.1.2 Watson-Watt technique

The Watson-Watt radio direction finding technique was implemented by Watson-Watt in 1926 and consists of comparing the amplitudes received from two antennas with sine or cosine gain patterns. [8] By taking the real component of the arc tangent of the sine gain patterned antenna V_{sin} and the cosine gain patterned antenna V_{cos} an azimuth α can be calculated as shown in the equation below. If there exists a phase delay δ between the two signals due to interference or multipathing, the bearing can still be calculated. [8]

$$\alpha = Re(arctan\frac{V_{cos}}{V_{sin}}) = \frac{1}{2}arctan\frac{2|V_{cos}||V_{sin}|cos(\delta)}{|V_{sin}|^2 - |V_{cos}|^2}[8] \tag{2.1}$$

A limited number of antennas that have sin and cos gain patterns exist. The most common antenna is the crossed-loop antennas, but these antennas suffer from errors due to the small antenna size when operating in multi-path prone environments. Ferrite antennas and Adcock antenna arrays also have the required gain patterns, but are not typically built for higher frequency applications and not widely available commercially. In addition, a direction finder using the Wattson-Watt technique requires at least two RF processing channels, three processing channels if an unambiguous bearing is desired. [8]

2.1.3 Interferometery

The interferometer direction finder was first used for radio astronomy [8] but has applications in terrestrial radio direction finding. In its simplest form, interferometery compares the phase information from three or more antennas spaced no more than $\lambda/2$ apart, where λ is the wavelength of the signal of interest. The spacing is critical in ensuring the phase change between two adjacent antennas is not greater than 180 degrees. With a being the antenna

spacing and Φ_1,Φ_2,Φ_3 the phase of the received signal at each antenna, the azimuth (left and right) α can be calculated as

$$\alpha = arctan\frac{\Phi_2 - \Phi_1}{\Phi_3 - \Phi_1}[8] \tag{2.2}$$

The elevation angle, (up and down) ϵ can also be determined.

$$\epsilon = arccos\frac{\sqrt{(\Phi_2 - \Phi_1)^2 + (\Phi_3 - \Phi_1)^2}}{\frac{2\pi a}{\lambda}}[8] \tag{2.3}$$

In addition, correlative interferometery exists, which is very similar to traditional interferometery, but uses a calibrated array and compares the received signal phases to a reference data set and calculates the azimuth as the value where the correlation coefficient is at a maximum. This allows antennas to be spaced farther than $\lambda/2$ apart and utilize far more antenna elements. However this radio direction finding technique requires a reference dataset, and both interferometery techniques require at least three RF processing channels. [8]

2.1.4 Pseudo-Doppler and Doppler Direction Finding

Doppler direction finding utilizes the Doppler effect to determine the direction of incidence. If an antenna element is rotated in a circle of radius R, then the received signal ω_0 is frequency modulated with the rotational frequency ω_r due to the Doppler effect. As the antenna rotates towards the direction of incidence, the received frequency increases; while when the antenna rotates away from the direction of incidence the received frequency decreases. [2]

This means the instantaneous amplitude of the received signal $u(t)$ is related to both the antenna radius R, direction of incidence α and wavelength of the frequency of interest λ. θ

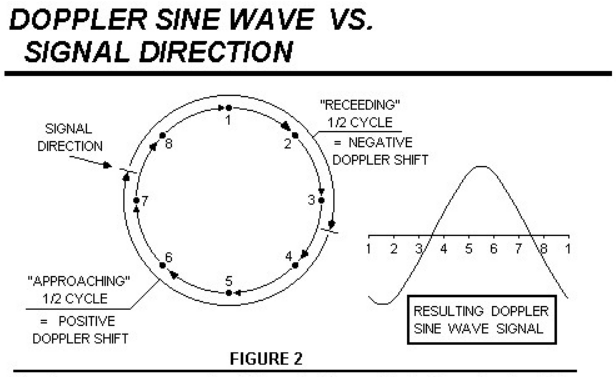


Figure 2.2: Diagram illustrating how the phase changes on a rotating antenna allow direction of the incoming signal to be determined [2]

is the phase content of the signal of interest.

$$u(t) = arccos(\varphi(t)) = arccos(\omega_0 t + \frac{2\pi R}{\lambda} cos(\omega_r t - \alpha) + \theta)[8] \tag{2.4}$$

By taking the derivative of the instantaneous $\varphi(t)$, the Doppler signal can be extracted.

$$\omega(t) = \frac{d\varphi(t)}{dt} = \omega_0 - \frac{2\pi R}{\lambda} sin(\omega_r t - \alpha)[8] \tag{2.5}$$

After removing the DC content ω_0, the demodulated signal S_D is obtained.

$$S_D = \frac{2\pi R}{\lambda} sin(\omega_r t - \alpha)[8] \tag{2.6}$$

Now if the phase of the modulated signal S_D is compared to a reference signal S_r of equal center frequency to the antenna rotation frequency ω_r then the angle of incidence α can be extracted.

$$S_r = -sin(\omega_r t)[8] \tag{2.7}$$

The Doppler direction finder still suffers from the same issues as the directional antennas, mechanical rotation. There is a method that avoids this issue, however.

Pseudo-Doppler direction finding is an alternative approach to Doppler direction finding that uses the same principles but replaces a single rotating antenna with an electronically sampled circular antenna array. Each antenna is sampled in sequence at a fixed frequency, equivalent to ω_r. By combining the samples into a single signal, the resulting superimposed phase shift is the same as a single, physically rotating antenna. This means the angle of incidence can still be determined by comparing the phase of the demodulated Doppler signal to a fixed frequency sinusoid. [2]

The advantages of Pseudo-Doppler direction finding is threefold. By keeping the antenna spacing under $\lambda/2$ (or equivalently keeping radius R under $\lambda/4$) the angle of incidence can be determined unambiguously, and an antenna array of any size is possible. This allows the Pseudo-Doppler direction finding system to have high immunity to multi-path reception and relatively high sensitivity. In addition, this method only requires a single RF processing channel and a method of antenna switching to operate. [8]

2.1.5 Signal Strength Mapping

While not technically a method of radio direction finding, measuring received signal intensity is an incredibly common method in the literature of tracking wildlife and determining the location of unknown transmitters.[10] [15] Desrochers et al. [15] provides a comprehensive look into this approach to wildlife tracking as well as its pitfalls. In an ideal world, measuring the received signal strength at two points would allow the transmitters position to be determined, since ideal electromagnetic waves propagate in a uniform sphere from the transmitter. Unfortunately, that is not the real world behavior. Radio signals are attenuated

non-uniformly as they propagate through and are reflected by structures and terrain between the transmitter and receiver. Walls, cliffs, and other features impact waves, creating local minimums and maximums. Even the antennas themselves have non uniform gain patterns, and their orientation compared to the transmitter impacts the received signal strength.

This is shown in the results of the Desrochers et al. [15] experiment, where a UAS with an omnidirectional VHF receiver flew seven times over five transmitters for a total of thirty-five data sets. Only in two sets did the strongest signal come from a location where the UAS was closest to the transmitter. In addition, only seven datasets had the signal strength increase the nearer the UAS was to the transmitter. [15] All of these tests were conducted outdoors in a boreal forest to simulate real wildlife tracking conditions. Of the 10 usable data sets, the average location error was 94 meters when a quadratic, least-squares regression algorithm was used to estimate transmitter location. Bayram et al. [10] also utilized this method to localize a transmitter, but through use of an online path planning algorithm was able to reduce the average location error to 50 meters.

2.2 Unmanned Systems

Only in the last decade have developments in lithium battery technology and low cost electronics enabled the widespread use of unmanned systems. In 2016, there were 788,570 UASs in the US. By the end of 2018 there were over 1.1 million, and this number is projected to continue increasing. [3] While the vast majority of these UASs are in consumer hands as hobbyist toys, commercial and industrial use continues to rise as new applications are found. [3]

2.2.1 Types

There exists three basic types of UAS platforms.

Multirotor

The most commercially widespread UAS in the consumer market, a multirotor UAS is composed of three or more propulsion units controlled electronically by a microcontroller or single board computer. [3] Such a configuration is inherently unstable, and as a result the platform control system must contain several sensors to enable active and continuous control. Multirotors are often sold as toys, but are also used for industrial, commercial and research use. [3] [1]

Figure 2.3: DJI S800 Multirotor [1]

They have the advantage of being able to change directions near instantaneously, and can hover over one location. The draw back of this design is all lift required to keep the system aloft comes from the propulsion units, and as a result suffers from shorter flight times and lower top speeds.

Fixed Wing

Fixed wing UASs resemble tradition aircraft, featuring one or more wings with one or more propulsion units. Lift is generated by air moving over the airfoils of the wings, while the thrust required to move the system is generated by the front or rear facing propulsion units. The majority of commercial off the shelf UAS of this type are intended for commercial and industrial use, and are pricier than the hobbyist multirotors. However many remote operated fixed wing platforms (remote control planes) exist and are converted to enable autonomous operations. [3]

Hybrid

The final UAS platform type is the hybrid platform, which combines one or more elements of both multirotors and fixed wing UAS platforms. While the designs can vary widely, they typically follow two design styles. The first utilizes the same propulsion units for both hover and forward flight, typically tilting the main propulsion unit or other methods of thrust vectoring. The second utilizes separate propulsion units for hovering and forward flight.

Figure 2.4: Experimental VTOL UAS developed by NASA [6]

Both designs offer the benefits of taking off vertically and the ability to hover, while offering some of the speed and range benefits that come from a fixed wing platform. [6]

2.3 Recursive Bayesian Estimation Techniques

Recursive Bayesian estimation (RBE) techniques are a powerful set of mathematical tools that allow software the flexibility to work not with absolutes results (the target is or is not there), but with probabilities (50% chance the target is there). [16]

2.3.1 Hidden Markov Models

Hidden Markov models are the fundamental principle behind recursive Bayesian estimation techniques, and are intellectually fairly simple to understand. A Markov model (or chain) is a stochastic model that describes a sequence of possible events where the probability of any event happening is solely dependant upon the current event. Such a system is memory-less, i.e. the process's future and past states are independent. [16]

A hidden Markov model is a special case of the Markov model where the model itself is assumed to have unobservable, i.e. hidden, states. In such a process, the state the process is in is not visible to the user, only the output of that state. Since each state has some probability distribution over the process outputs generated by the hidden Markov model, the sequence of outputs does give some information about the sequence of states in the process. The adjacent "hidden" refers to the state sequence the model progresses through, not the parameters, or probability distributions, of the states. By dynamically changing these parameters, a HMM can output the best set of state transition and output probabilities. This enables systems with noisy, imprecise, or less than optimal data inputs to generate useful data outputs in addition to a confidence measure in the accuracy of the data output. [9]

2.3.2 Sequential Monte-Carlo Methods

Monte-Carlo methods are a broad class of algorithms that use repeated random sampling to compute results, the name is in reference to the famous grand casino in the principality of Monaco. [22] Unlike traditional algorithms were the output is expected to always be correct, within Monte-Carlo methods the output can be incorrect within a certain probability. When combined with signal processing and Bayesian statistical inference, sequential Monte-Carlo methods can estimate the internal states of a dynamical systems from partial observations that contain random perturbations in the sensors as well as in the dynamical system. Sequential Monte-Carlo Method uses a set of particles to represent the posterior distribution of some stochastic process given noisy and/or partial observations of the state-space. The state-space model can be nonlinear; while the initial state and noise distributions can take any form required. Sequential Monte-Carlo methods are good at generating samples without requiring assumptions about the state-space model or the state distributions. However, these methods do not perform well with high-dimensional systems. [19]

Chapter 3

System Design

Several systems were designed for this work, ranging from radio direction finding system to a hybrid UAS, along with several pieces of software. This chapter discusses the research done, design choices made and construction techniques used for each component used in this work.

3.1 Hybrid UAS

Unmanned Aerial Systems are fast becoming a major component of commercial air traffic. They provide a cheap and convenient solution to problems that normally require the charter of expenses and potentially dangerous aircraft, and are used in a range of applications such as wildlife tracking, surveying, and even real estate. Since the focus of this work is related to wildlife tracking, a UAS was an obvious platform choice.

3.1.1 Derivation of Requirements

A critical portion of this work was to ensure the lowest deployment cost for the UAS platform. While numerous commercial platforms are available, they all suffer from the same issue, cost. The lowest cost commercial off the shelf (COTS) UAS capable of autonomous flight is DJI's Mavic. This multirotor UAS costs $1500 and only has listed operational flight time of up to 31 minutes as stated on the DJI website. [1]

For fixed wing or hybrid UAS platforms, the cost is even higher. Since fixed wing and hybrid UAS never received the mainstream consumer push that the multirotor platforms received, they cost thousands or tens of thousands of dollars, putting them out of reach of many users. Space is also a premium. Most commercial platforms are designed for one specific application, usually photography, and as a result are difficult to reconfigure for changing mission requirements. As a result, a custom platform was needed.

Based on feedback from the local wildlife science department, such a UAS platform must at least match the flight time of the standard multirotor drone, be capable of vertical takeoff, and have a large enough payload to carry radio equipment required for wildlife tracking. In addition, the platform must be reproducible without specialized tools or skills. As with any platform intended for real world use, simple is always better, as complex mechanical systems tend to break down. Based on these requirements, the ideal platform is a hybrid UAS with two propulsion systems, one for vertical flight and one for horizontal flight.

Therefore the requirements for the system are as follows:

- UAS must be a Hybrid UAS
- Maximize use of COTS components
- Reproducible without specialized tools or skills
- Capable of hosting radio direction finding systems

3.1.2 Component Selection

There are three major components to a UAS platform, the frame the UAS is built on, the propulsion systems, and the guidance and control system.

UAS Frame

The advantage of purchasing a commercial off the shelf frame is threefold. First, it's cheaper and stronger than attempting to build a frame from scratch. Replacement parts are readily available and it's easier to modify a frame than build it from scratch. Depending on the requirements, there are a vast number of platforms that exist at various cost points that are all valid choices to build a system off of.

For this project, two frame selections were evaluated. In order to potentially host the Doppler direction finding array in the future, a 2 meter wingspan was a minimum to fit the antenna spacing required. In addition, to support the hardware and other items required for VTOL, a payload capacity of 5lb or more is mandatory. The first frame explored was the Ranger Pusher Glider 2000mm, shown in figure 3.1 below.

Figure 3.1: Ranger Pusher Glider 2000mm, Image from Hobbyking.com

The Ranger Push Glider features the required 2 meter wingspan and has a payload capacity of 5 kg (11 lbs) with a plastic fuselage and an expanded EVA foam wing construction. Most importantly, the Ranger frame contains two aluminium square tubes inside each wing and two aluminium rods that fit through the fuselage into each square tube. Theses aluminium frame pieces provide great rigidity and strength to the foam wings and would support additional

weight without causing wing failure. It makes an ideal platform and easily modifiable to fit the system needs for a total cost of around $120.

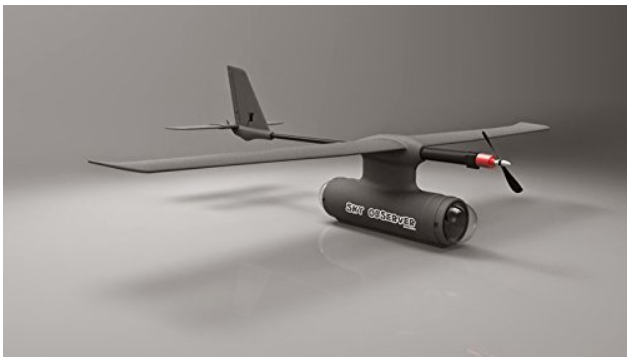


Figure 3.2: Zeta Sky Observer, Image from Amazon.com

The second platform that was evaluated was the Zeta Sky Observer. Featuring a 2 meter wingspan and an under-slung pod for mounting payloads, the Sky Observer also utilized a plastic fuselage and EVA foam wings. Very similar to the Ranger Pusher Glider, unfortunately the Sky Observer does not seem to be widely available and ranges in price from $120 to $320 dollars. While the under-slung pod would allow a greater payload volume than the Ranger Pusher Glider, the unavailability of replacement parts and greater cost eliminated it as a potential platform. As a result the Ranger Pusher Glider was selected as the frame for this work.

Propulsion and Power System

The Ranger Pusher Glider frame comes with a 2215 1400Kv brushless DC motor with a 8x4 plastic propeller for the horizontal thrust unit. Combined, this thrust unit provides a estimated 1038.9 thrust grams. [5] Since the platform's empty weight before modification is 1083 grams, this provides an almost 1:1 thrust to weight ratio. In order to reduce cost and complexity, using the propulsion unit that came with the frame was deemed essential. Therefore the main battery voltage was selected at 11.1V, or a 3S LiPo battery, matching

the expected voltage for the Ranger Pusher Glider's existing brushless motor and speed controller. In order to maintain the system's expected performance with the existing horizontal propulsion unit, any additions to the frame must not exceed 3 lbs, 1360.78 grams. Using this estimate as a high end system weight, the vertical propulsion units could then be selected. Using an online tool and past engineering judgement, 2830 1200KV brushless DC motors were selected for the vertical propulsion units. Utilizing a 11x4.5 prop, each motor at maximum thrust provides around 4000 newtons of thrust. [4] Between four motors, this results in 16000 newtons of thrust at maximum RPM, or allows the system to hover at an estimated 20% throttle. By being conservative, this allows flexibility in future payload requirements and acknowledges that sometimes large deviations exist between estimated and real world performance. The motors used were Turnkey PROPDRIVE v2 2830 1200K/V DC Brushless Outrunners, both for their availability and relatively low per unit cost of $18.34. Based on the current rating, 40 amp electronic speed control units were selected to give sufficient overhead to improve reliability and reduce thermal build up while still minimizing cost.

Guidance and Control System

In the open source UAS community, there exist two primary firmwares, Ardupilot and Pixhawk. Both are roughly equal in terms of capabilities for UAS, and both are compatible with the same hardware platforms. For our UAS, the Pixhawk firmware was selected primarily because it supports Hybrid UAS systems out of the box with minimal modification.

In addition the pixhawk 2.1 controller was selected due to its wide use in both the academic and hobbyist communities and several units were available for use in the lab. The pixhawk 2.1 is capable of completely autonomous operations, onboard data logging for performance validation and debugging, and can be linked to a laptop for realtime data visualization.

Figure 3.3: Pixhawk 2.1, Image from Pixhawk.org/

3.1.3 Construction methodology

The approach to constructing the UAS was to keep the process as simple as possible and not require specialized tools. The most important and complex portion of the design was attaching the booms that hold the vertical propulsion units to the wings in a robust fashion that still allows the UAS to be disassembled for transportation. To that end two carbon fibre plates were bonded underneath each wing, covering from the wing roots to the edge of the aerilions. These plates provide both a secure mounting point for the booms and stiffen the wings to prevent flexing due to the increased weight. Each plate has a hole drilled in it that allows access to the clips that secure the wings to the fuselage. The combination of the carbon fibre plates in addition to the existing aluminium superstructure created a very stiff wing system capable of supporting the VTOL propulsion system. Finally, the booms that hold the VTOL propulsion system were mounted under the wings near the wing roots and bonded to the carbon fibre plate.

The motors are attached to the booms via four screws and two brackets that fit around the boom and rely upon friction to stay in place. This allows the motors to easily be replaced if issues arise and allows the system to fail in a controlled fashion in the event of a hard

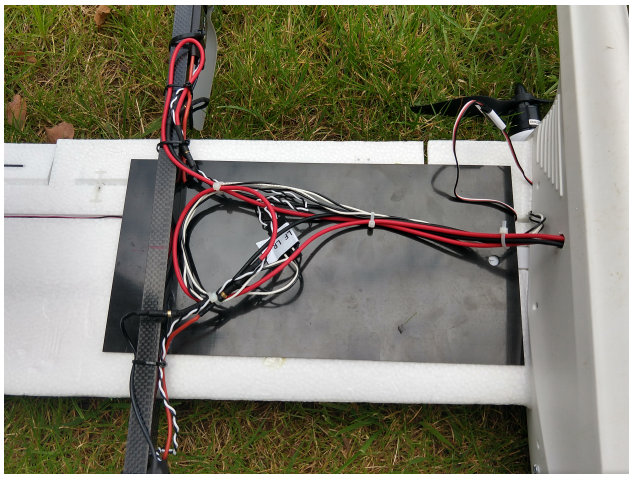


Figure 3.4: Carbon fibre plate bonded to wing root of the UAS

landing. Finally cabling with connectors were run from the motors and speed controllers mounted on the booms into the fuselage through two holes and connected to the pixhawk controller mounted in the fuselage.

Inside the fuselage the pixhawk was mounted to the wooden frame that was already inside the Ranger, while the communication and remote control radio transceivers were mounted with velcro to the interior top of the plastic fuselage. Finally the GPS receiver was mounted externally on top of the airframe and ran inside through a small hole. The airspeed sensor was ran out the nose of the airframe through the conveniently located hole designed for a small video camera. The total cost to build the Hybrid UAS excluding the batteries and Pixhawk 2.1 was $314.94 at the time of construction.

3.2 Pseudo-Doppler Direction Finder

The other major design component of this work was the pseudo-doppler direction finder (PDDF). While several professional and hobbyist grade COTS systems exist, all of them use an external VHF radio receiver, which limits the frequency flexibility and software ap-

Figure 3.5: Assembled UAS

Table 3.1: Bill of Materials for Hybrid UAS

Component	Quantity	Cost Per	Total
Ranger Pusher Glider 2000mm (78.7") (PnF)	1	$108.40	$108.40
PROPDRIVE v2 2836 1200KV Brushless Outrunner Motor	4	$17.23	$68.92
Hobbywing X-Rotor ESC - 40amp (OPTO)	4	$12.79	$51.16
NTM Prop Drive 28 Series Accessory Pack	4	$3.93	$15.72
Gemfan Propeller 11x4.5 Grey(2pcs)	3	$3.83	$11.49
Carbon Fiber Sheet 1.5mm*300mm*150mm	2	$16.95	$33.90
Carbon Fibre Square Tube 15 x 15 x 800mm	2	$9.80	$19.60
M4 Nuts and Bolts	1	$5.75	$5.75
Total			$314.94

plications. No existing commercial off the shelf system for pseudo-doppler direction finding existed when this work was started. However, the components to build such a system did exist.

3.2.1 Requirements

At minimum, to track tagged wildlife the PDDF had to operate in the 145 to 165 MHz band. This VHF band is commonly used by radio collars designed for larger wildlife, while the UHF

band is more typically used to track smaller wildlife like birds due to reduced antenna size. Therefore the PDDF should be flexible and re-configurable for tracking different targets in different radio bands. In addition, the system needs to be small, low power and light to maximize the Hybrid UAS's flight time and fit within its size requirements. Ideally the system should be low cost and not require any special tools or skills to build, however in reality this requirement was softened. Open source software was required in order to increase accessibility to the system and enable other interested parties to recreate the work.

3.2.2 Survey of COTS SDR systems

As previously discussed, there are very few COTS PDDF systems, and the majority are specialized for very specific applications. However what has seen a widespread growth in the past decade or so is the software defined radio. Unlike traditional radios where a vast majority of the functionality of the radio is implemented in hardware inside the radio itself, a software defined radio takes those components that enable that functionality and transfer them over to software. That means the same radio hardware that traditionally would be limited to a single role can now be dynamically changed by altering the software on a host system. As an example, one receiver can now support both wide band FM, narrow band FM, AM, and any other modulation method simply by changing some parameters in software. This flexibility complements the intended application of this work, allowing the sensor package to easily change frequency and other parameters based on mission needs, without having to remove and replace hardware.

The primary software defined radio used through academia is the USRP series developed by Ettus Research. Within the USRP series there exists a wide range of configurations intended for use in applications ranging from rack mounted to small portable embedded

systems. The downside of any of the USRP lines is cost. The USRP E310, which is the cheapest portable stand-alone SDR platform designed for field deployment, costs over $3,000 dollars. It features 4 RF channels, two inputs, two outputs, and covers a frequency range of 70MHz - 6GHz. The high cost does not make it a feasible candidate for this work. A suitable software defined radio would need to be found elsewhere.

Inside the hobbyist sphere, there exist quite a few different software defined radio (SDR) platforms that are widely available and relatively cheap. The cheapest and most widespread is the RTL-SDR platforms. More of a family of low cost SDRs than a singular SDR, the RTL-SDR uses one of six RF ICs to create a single RF input channel with a frequency range of around 54 Mhz to around 2200 MHz, depending on the exact IC. Originally a low cost digital TV tuner, work by the hobbyist community allowed open source software to retune and change the parameters of the RTL-SDR, creating a fully functioning SDR for around $20. The other major SDR with open source support is the HackRF One. Developed by Great Scott Gadgets, it has a 1 MHz to 6 GHz operating frequency with half-duplex transceiver and takes up to 20 million samples per second. In addition, the HackRF One has several daughter boards that plug into the HackRF One to provide additional capabilities. The HackRF One costs around $300.

3.2.3 Selection and construction

In order to fully explore what could be done with low cost SDRs, two systems were actually built. The first utilized a RTL-SDR and some custom circuitry, while the second uses a HackRF One and an Operacake board (a small addon board with GPIO controlled RF switches).

RTL SDR

Due to its wide availability and lost cost, the RTL SDR was selected to implement the first PDDF. This approach to building a PDDF was similar to the work done by the RasHAWK team as part of the AOC Tech Challenge. [24] For this work's implementation an Analog devices ADG904 4:1 RF Mux was used along with a RTL-SDR in conjunction with an ATMEGA328P micro-controller to change between four input feeds. The output of the RF switching IC was feed into the RTL-SDR which was connected to a Raspberry Pi 3. The Raspberry Pi 3 would be responsible for handling all the DSP calculations and was selected due to low cost, decent computation performance and widespread support. In addition, a LSM6DS33/LIS3MDL based IMU along with a Maestro Wireless A-2235H GPS receiver were included on the PCB to allow the system to directly measure position and orientation.

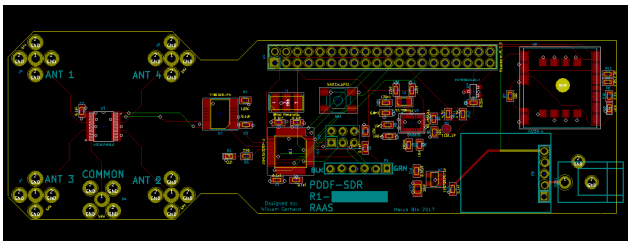


Figure 3.6: RTL-SDR switching PCB

This created an all in one system that was easily deployable and could log all information required.

Two major issues with this system that prevented it from being used further in this work, both DSP related. The first major issue was related to sampling and switching. Since the external microcontroller which handles changing between the RF channels is on a separate clock source from the RTL-SDR, the data is not synchronized. This results in changing numbers of samples from each antenna every cycle. The changing number of samples per antenna resulted in the introduction of phase errors that limited system accuracy. The

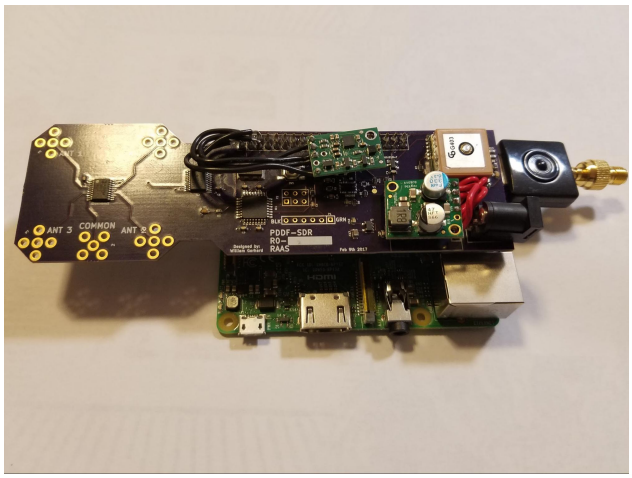


Figure 3.7: Assembled RTL-SDR System

second issue was related to the drift inherent in the microcontroller's clock. In order to simplify the design, the internal oscillator built into the ATMEGA328P was used instead of an external clock source, and this internal clock drifted by +-10% per the data sheet. This drift exaggerated the previous phase error.

HackRF One

The second PDDF system is based around the HackRF One and the Operacake board. At the beginning of this work, the Operacake was an experimental add on board that was not commercially available, but had fully documented open source hardware with schematics and board design files available online. However the Operacake is expected to be commercially available sometime in 2019 or early 2020. Therefore it was deemed acceptable to explore using the HackRF One and Operacake in the scope of this work.

The Operacake enables a HackRF One to switch between one of eight RF inputs. It does this by multiplexing two RF switches into a third RF switch that feeds the output. The MACOM MASWSS0129TR-3000 dual RF switching IC is the output RF switch, and is rated for 43 dB of isolation at 2.4GHz and has .7 dB insertion loss at 2.4 GHz. It operates from

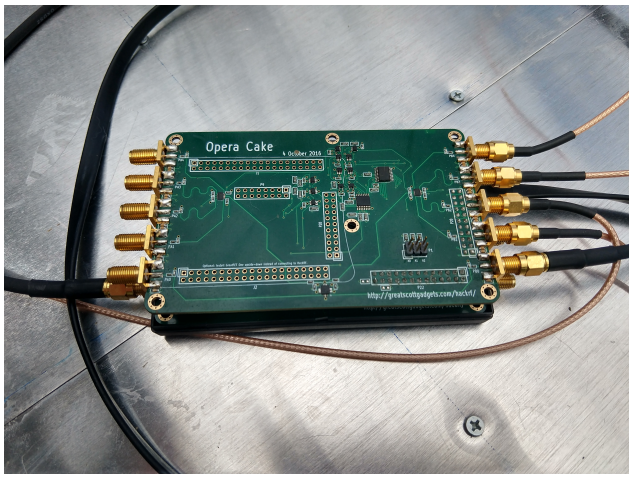


Figure 3.8: Assembled Operacake board

DC to 6 GHz, matching the performance specifications of the HackRF One. Two Skyworks Solutions, Inc. SKY13322-375LF SP4T RF switches are used to control each bank of four RF inputs. The SKY13322 has 28dB of isolation at 1 GHz and only .45 dB insertion loss.

The interesting feature of the hack RF One and Operacake, and what enables this system to bypass the issues that plagued the RTL-SDR system, is the use of the LPC43xx SGPIO peripheral built into the HackRF One. The LPC43xx SGPIO peripheral inside the HackRF One is simply a large bank of user defined and configurable 32-bit shift registers that are clocked at a higher rate than the MAX5864 ADC/DAC. By reconfiguring these registers, external GPIO can be driven by the registers' values, changing state in between samplings of the ADC. The Operacake, when plugged into the HackRF One, uses these GPIO to control the various RF switches. This means that the transitions between RF inputs occurs in between ADC samples and allows a uniform sampling per RF input.

The construction of the Operacake board was fairly straightforward, but required the use of a SMT stencil and reflow oven due to the small surface mount used in its design. The total cost of building one Operacake board was approximately $200.

In addition to building the Operacake board, firmware modifications to the HackRF One

were needed. While the code that handles using the SGPIOs to switch the Operacake was written by a HackRF One developer, it was based on firmware for the HackRF One that suffered from several bugs and other performance issues. To fix this, the SGPIO -enabled firmware and the current master firmware for the HackRF One were merged and debugged. In addition the SGPIO switching mode was enabled by default in the firmware and set to autostart at a user selected switching frequency. Due to the nature of how the SGPIO interacts with the HackRF One, the antenna switching rate is tied to the data sample rate S_{hackRF} and the number of samples per antenna $Samples_{antenna}$. This relationship is given in the equation below.

$$F_{AntennaSwitching} = \frac{2S_{hackRF}}{4Samples_{antenna}} \tag{3.1}$$

Antenna Array

The final component of the PDDF system was the antenna array. In order to fulfill the spatial Nyquist theorem, the antenna array cannot exceed $\lambda/2$ spacing between each antenna element, i.e. a square four element antenna array cannot exceed 0.35λ a side. However, experiments conducted over the last couple of decades by the amateur radio fox hunting community have shown a square antenna array with sides not more than 0.22λ provides optimal results for pseudo Doppler direction finding. This spacing allows a large enough phase change between the antenna to be measurable while not exceeding the spatial Nyquist limit. [23] Such an array provides acceptable performance at a range of frequencies centered at the frequency of interest $\frac{c}{\lambda}$ but ranging from $0.8\frac{c}{\lambda}$ to $1.2\frac{c}{\lambda}$.

The primary frequency of interest in this work is 150MHz, resulting in a λ of 78.72 inches (1.9986 meters) and therefore results in an antenna spacing of 17.3184 inches (0.44 meters). This spacing covers frequencies from 120 MHz to 170 MHz, which contains all the VHF frequencies used for wildlife tracking.

Each antenna element in the array was a commonly available monopole antenna. Since the HackRF One features an impedance matching circuit in the RF front end, selecting antennas with exactly 50 ohm impedance was not a requirement. Monopole antennas were selected rather than dipoles or some other commonly used antennas types primarily due to concerns surrounding availability. The majority of antennas found for commercial sale are 1/4 wavelength monopole antennas, while dipoles and other antennas are typically custom built. Therefore ensuring functionality with 1/4 wavelength monopoles was deemed a priority. To provide better performance and create a rigid surface for antenna mounting, aluminum sheeting and some wood was used to build the antenna array. The aluminium sheeting provides a ground plane for the monopole antennas and a rigid structure to ensure proper antenna spacing. Finally identical lengths of RG-316 coax cabling was used to connect each antenna to the Operacake board. RG-316 antunates approximately 11 dB per 100 ft, giving each 24 inch cable a .22 dB loss.

3.3 GNURadio Flow Graph

GNURadio is a widely used open source digital signal processing software designed for SDRs. Composed primarily of one large programmatic flow graph created by connecting blocks that perform certain processes, GNURadio enables rapid development and easy modification of software projects. Each block itself is a program written in C/C++ for performance with a python wrapper handling data inputs and outputs to the larger flow graph program.

For the pseudo Doppler direction finder the flow graph itself is fairly straightforward and is composed of four main sections: input, filtering, FFT calculation, and comparison.

The input section of the flowgraph consists of two blocks, an RTL-SDR read block and a frequency translating FIR filter block. The RTL-SDR read block is actually a generic

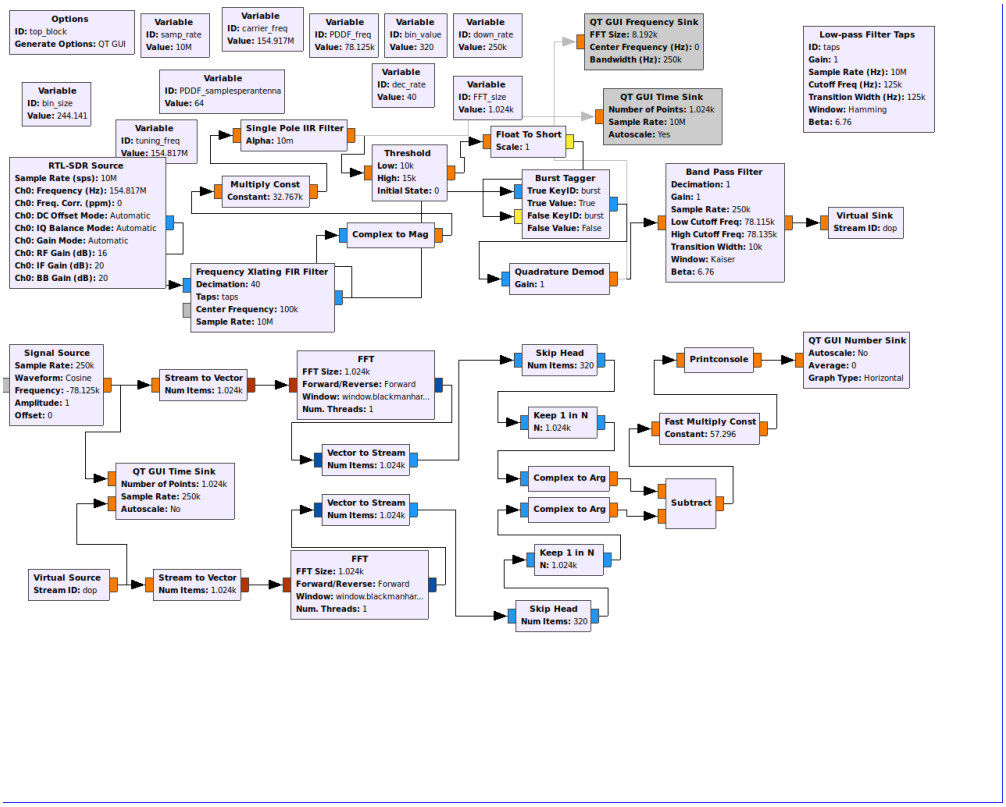


Figure 3.9: GNU Radio Flowgraph

block capable of reading data from any supported SDR, and sets both the SDR sample rate as well as the tuning frequency. The frequency translating FIR filter block performs three functions, a frequency translation, antialias filtering, and a decimation. One of the downsides of using a HackRF One is there is not a DC blocking filter built into the hardware, resulting in a large DC component at the tuned frequency of interest. By using a frequency translating FIR filter, the HackRF One can be tuned to a frequency close to the actual frequency desired, and the signal can then be filtered and shifted in software, eliminating the DC offset. The decimation allows the HackRF One to be set to a higher sample rate to achieve the Operacake switch frequency desired, and then have the sample rate reduced to something more manageable for lower performance systems.

The filtering section of the flowgraph is a simple notch filter. While the block is labeled as

a lowpass filter, the same block in GNU Radio depending on how its parameters are set, can be a high pass, notch, or low pass filter. For reasons unknown notch filter parameters still result in the block being named a low pass filter. The notch filter width is set to a tenth of the FFT bin size around the Operacake rotation frequency. This is intended to help suppress adjacent frequencies' phase random noise on the calculated phase of the Operacake rotation frequency in later portions of the flow graph. This filter does change the phase of the incoming signal, however it is a constant phase change and can be corrected through careful calibration with a known signal source.

To actually determine the angle of incidence of the incoming signal, the phase of the now filtered sinusoid at the Operacake rotation frequency must be compared to the phase of a non-offset cosine at the same frequency. First each stream of floats in broken into an array x_k of length $N = 1024$. Then a discrete Fourier transformation (DFT) is taken where N is the length of the array x_k, n is the current index in the array x_k and k is the sample rate in Hz.

$$X_k = \sum_{n=0}^{N-1} x_k e^{\frac{-i2\pi}{N} kn} \tag{3.2}$$

GNURadio implements the DFT as a fast Fourier transform, which is a computationally efficient version of the DFT, however the fundamental math is the same. The result of the calculation is a vector of complex numbers containing both amplitude and phase information calculated at each $k * n$ frequency. Given $k = 250KHz$ and $N = 1024$ then each calculation covers a frequency range of 244.141 Hz, and each step is called a bin. The rotation frequency of the array as given by 3.1 is 78.125 KHz, which falls exactly into the 320 bin (or $n = 320$).

The DFT is ran on both the filtered incoming signal and a non-phased shifted cosine at the frequency of rotation ($\omega = 78.125KHz$). The imaginary component θ of the 320 DFT bin

is extract and compared. The difference between the two values is the angle of incidence α in radians.

$$\alpha = \theta_1 - \theta_2 \tag{3.3}$$

Finally a custom GNURadio block logs the calculated angle of incidence as a number along with a microsecond UNIX timestamp, and saves it in a CSV file for later use.

3.4 Particle Filter

While the PDDF system provides angle of incidence information, that information by itself is unable to localize the transmitter. To determine the location of the transmitter, additional angle of incidence information is needed at the same time interval from different spatial locations as demonstrated by Bayram et al. [11] (2017), or multiple angle of incidence measurements at different spatial and temporal locations, with the strong assumption that the target did not move, as done in more recent work by Bayram et al. [12] (2018). A third approach using particle filters, that does not require the strong assumption that the target does not move, is used in this work. By combining multiple angle of incidence measurements at different spatial and temporal locations with models representing sensor and target behaviors, an estimated position and confidence in that estimate, can be calculated. This approach is similar to traditional triangulation where multiple bearings are taken at the same temporal moment at different spatial locations. A line is drawn from each spatial location along the bearing and where they cross is the location of the target. However this approach uses bearings taken at different spatial and temporal locations and models how the target might move to in effect do the same thing.

Inside a particle filter, each particle is a hypothesis of the origin of the incoming signal. As a result, each particle must contain 5 elements of information inside the state space X_n. This state vector provides information that describes the position, velocity, and angle to the origin of an object moving in the plane as shown by:

$$X_n = (s_n^x, u_n^x, s_n^y, u_n^y, \theta_n) \tag{3.4}$$

While the velocity and angle information is technically contained in the x and y location, calculating once and saving the information in each particle reduces the number of floating point operations per time step by half at a minor expense in memory. Imperfect observation of the angle, and thus the x and y position, but not velocity, is possible at each time step. This is due to the velocity of each particle being the exact opposite motion of the PDDF system in the plane combined with a random Gaussian variable.

3.4.1 Motion Model

Due to the various sizes, habitats and habits of wildlife, developing motion models for how wildlife moves in a difficult task. Most recent work has been focused on modelling how animals move during migrations [18] or using stochastic differential equations to model how groups of animals react to intruders or human activities [14] [13] or move over the period of a day. [27] Intuitively, a bird will move differently than a moose and both will behave differently during a migration than foraging for food. Without knowing the subject that is being tracked it is difficult to create a useful model. In these cases, the most common model used is a Brownian movement model with a cyclic movement dependency. [25] Since our particle filter is working on a short time scale, minutes instead of hours, the target is assumed to be moving in a random direction at a random speed at each time step, resulting

a Brownian-like motion model. Thus:

$$S_n = S_{n-1} + \int_0^{\Delta} U_n + Z_n U_{nx} = U_{n-1}^x + - \int_0^{\Delta} A_x + Q_n U_{ny} = U_{n-1}^y + - \int_0^{\Delta} A_y + Q_n \quad (3.5)$$

U is the velocity, S is the position, Z_n is a random Gaussian distribution with a mean of 0 and a variance of .5, and Q_n is a random Gaussian distribution with a mean of 0 and a variance of 25. A is the measured acceleration on the PDDF system in the X and Y directions. This model makes the PDDF system's current position the center of the particle filter's coordinate system, allowing a direct comparison between each particles bearing to (0,0) and the incoming corrected bearing measurement from the PDDF system.

3.4.2 Sensor Model

The sensor model of the particle filter is also quite simplistic. The only information being directly measured by the PDDF system is the bearing of the incoming signal of interest. Therefore the sensor model is one dimensional and simply applies a likelihood estimate based on how close each particle's saved bearing is to the measured bearing. With each measured bearing from the PDDF system, the absolute value of the measured bearing and the particle's bearing is calculated. The smaller the result sum, the more likely the particle is to being the correct hypothesis, resulting in:

$$- 0.5 log(2\pi\sigma^2) - \frac{\theta_n^2}{2\sigma^2} \quad (3.6)$$

Where σ^2 is the signal variance and θ_n is the difference between the particle's saved bearing and the measured bearing. This shape increases the likeliness of particles whose saved

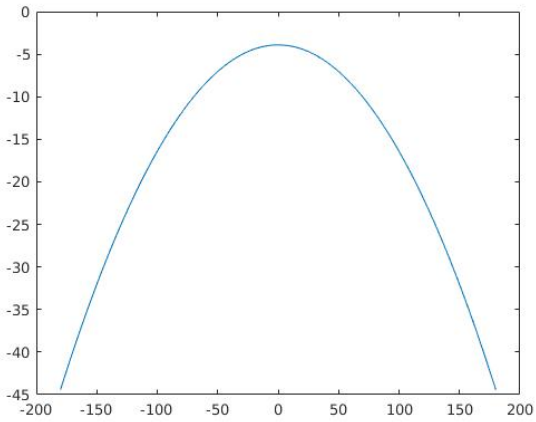


Figure 3.10: Plot of Sensor Model

bearings are close to the measured bearing, while decimating particles whose bearings are 180 degrees off. The variance can be adjusted depending on real world data collection results to account for errors and other noise.

3.4.3 Framework

The particle filter itself was implemented in C++ for maximum speed and portability, and utilizes a sequential Monte-Carlo framework designed for robust, cross platform development. [19] The particle filter uses each incoming bearing measure to add or subtract from the likelihood of each particle being the correct location. Once a certain number of particles have a likelihood close to zero, all particles are resampled. The resampling method selected was that proposed by Liu and Chen [21] after 50% of the particles have been eliminated as the hypothesis. To get the estimated position of the transmitting signal, a simple approximation of the particle cloud's integral was used to find the mean and variance of s_n^x and s_n^y. In this filter, each particle's position is multiplied by the particle's likelihood and summed. This sum is then divided by total likelihood over all particles and the resulting value is the most likely location of the target. To calculate the variance of the estimated position, the squared

difference between s_n^x and an estimate of its mean it used rather than calculating the mean of $(s_n^x)^2$ using the same method as above. This is faster and reduces computational complexity with the particle filter.

Chapter 4

Results

Each component designed and built for this work was tested individually to ensure proper operation. Outdoor testing of the PDDF system was conducted, and finally the results were fed into the particle filter.

4.1 Unit Validation

Each individual subsystem was tested to validate that it met the design parameters outlined in Chapter 3. By testing each subsystem individually, issues could be resolved and results could be measured with confidence that the system as a whole works as designed.

4.1.1 Hybrid UAS

To valid the performance of the Hybrid UAS and ensure it met the design criteria specified, several flight tests were conducted inside the Virginia Tech Drone Cage. Each test lasted approximately 2 minutes and consisted of three stages of flight. First, the UAS took of vertically in a hover, and gained altitude until approximately 10 meters above ground level was attained. The UAS then tilted forward and gained horizontal speed. After sufficient speed was gain the UAS transitioned into forward flight. After approximately 200ft, the UAS transitioned back from forward flight into a hover, slowly descended and landed. Unfortunately,

the first several tests were inconclusive beyond achieving a hover. This was due to the use of existing lab stocked LiPo battery packs. Unfortunately these packages had degraded due to use over their lifetime and were not able to hold a full charge. As a result the UAS was required to land immediately after takeoff to prevent an uncontrolled landing.

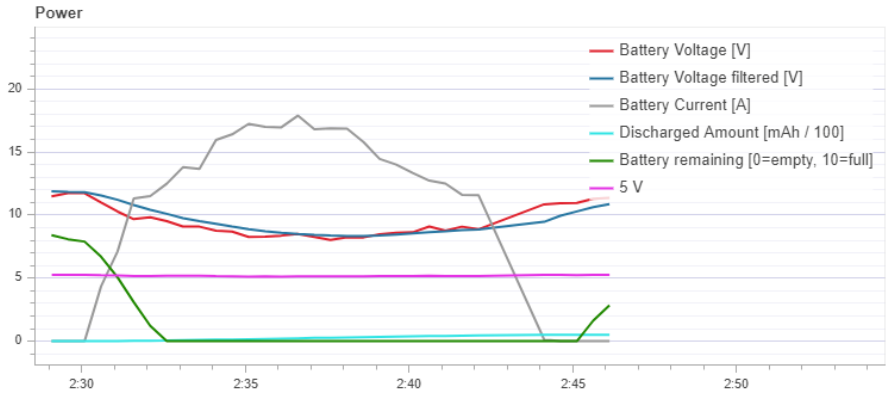


Figure 4.1: Hybrid UAS hover test current consumption

These tests did prove the UAS system air worthy and provided a baseline current consumption for hover. Later testing with new LiPo batteries provided additional data on forward flight.

Due to the limited size inside the drone cage and the nervousness of the operator, full transition to forward flight was not obtained. However as the transition to forward flight begins there is a 15% drop in current consumption as the UAS aerofoils begin to provide lift and the VTOL propulsion system decreases thrust production as shown in figure 4.2. A 15% drop in current consumption results in a 15% increase in total flight duration, as the capacity of the battery is the limiting factor for determining a UAS's flight time.

(a) Pixhawk flight mode control inputs (b) Current consumption

Figure 4.2: Further hybrid UAS testing results

4.1.2 RF Switching Circuit

Once the Operacake board was assembled, various tests were conducted to ensure proper operation and measure the real system performance. The first series of tests were conducted using a Rigol DSA1030A spectrum analyzer, and measured the RF input channel isolation characteristics. 50 Ohm terminators were placed on channels 1, 3, and 4 while the output of the spectrum analyzer was fed into channel 2. The output of the Operacake was then connected to the spectrum analyzer's input. Once the Operacake was powered, channel 3 was selected to be connected to the output.

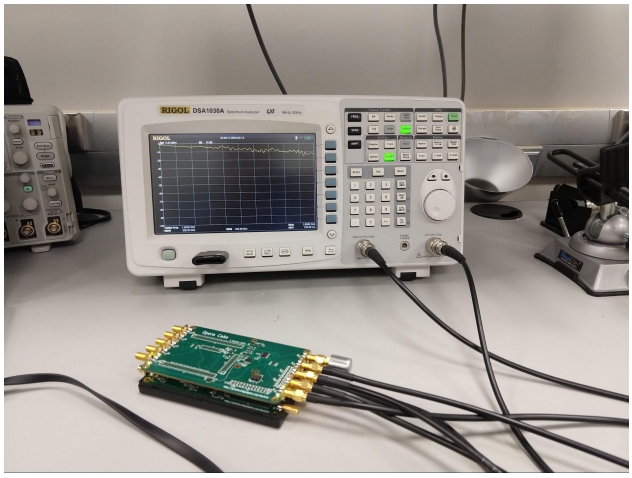


Figure 4.3: Operacake isolation and loss setup

Once connected, a frequency sweep from DC to 3 GHz in 30000 KHz steps was run, and the measured signal strength at the output of the Operacake was plotted.

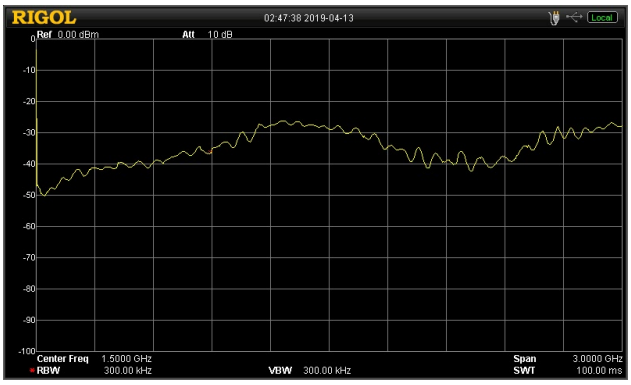
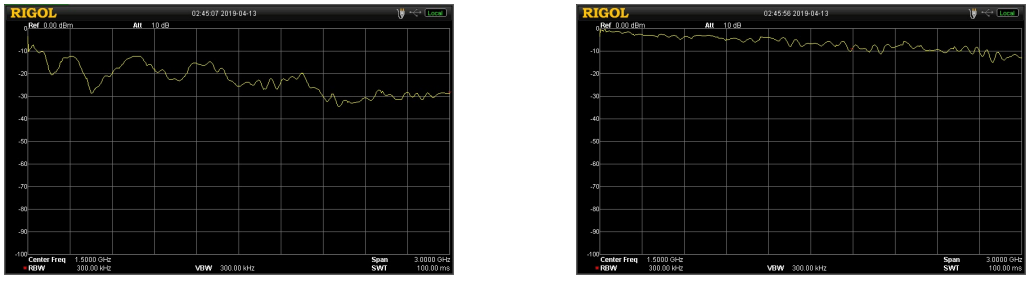


Figure 4.4: Operacake channel isolation measurements

This test measured the isolation between channels on the RF switches, with 28 dB of channel isolation at 120 MHz and 29 dB at 150 MHz as shown in 4.4. The next set of tests measure the insertion loss through the Operacake. Once again a DSA1030A spectrum analyzer was used and 50 Ohm terminators were placed on channels 1, 3, and 4 while the output of the spectrum analyzer was feed into channel 2 and the output of the Operacake feed into the spectrum analyser. When the Operacake was powered, channel 2 was selected as the input to connect to the output. Once again, a frequency sweep from DC to 3GHz in 30000 KHz steps was conducted. Two sets of tests were run, to measure the performance when the Operacake was powered and unpowered.



(a) Operacake unpowered (b) Operacake powered

Figure 4.5: Input loss test results

When unpowered, the insertion loss was -22 dB as shown in 4.5a, while when powered the insertion loss dropped to -6 dB, 4.5b. Once the performance of the RF switches was verified,

the next step was to verify the firmware modifications worked and the Operacake rotated between RF inputs. For this test a Agilent N5181B Signal Generator was used to create a -30 dBm 100 MHz signal that was fed into channel 1. Channels 2, 3, and 4 were connected to 50 Ohm terminators. The output of the Operacake was connected to Tektronix DPO 3034 Digital Oscilloscope.

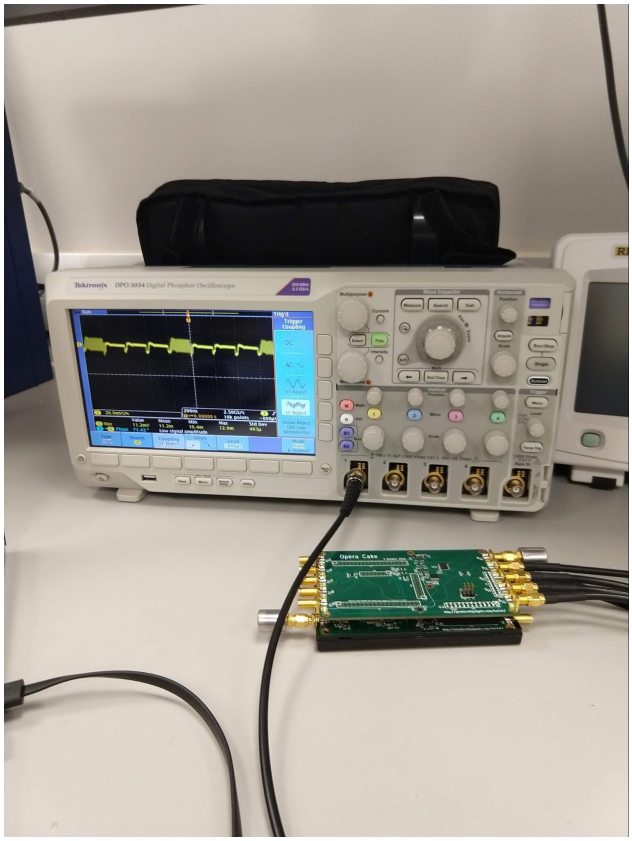


Figure 4.6: Operacake switching frequency test setup

The HackRF and Operacake was powered and set to sample at 10 megasamples per second with 4 samples per antenna. Using the equation in 3.1, the system should switch between antennas at a 5MHz rate. A 100 MHz signal was fed into channel 1.

The deltaT between two sections is 400 ns, resulting in a rotation rate of 5 MHz, which matches the calculated frequency. In addition, only in one of the 4 sections is a signal

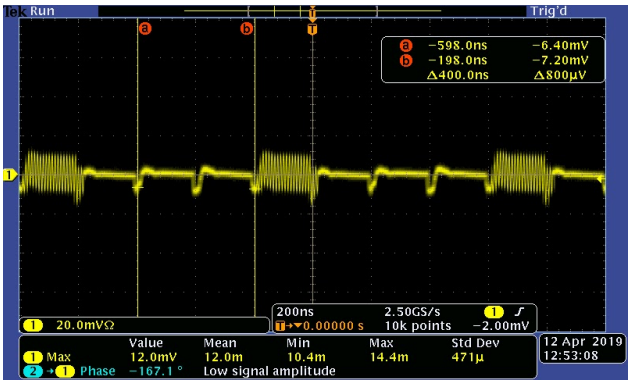


Figure 4.7: Operacake switching frequency test result

present. The results of this test are shown in 4.7.

4.1.3 Software Defined Radio

To test the SDR and validate the performance of the Operacake board combined with the HackRF a straightforward test was conducted. Two Tektronix AFG3000 two output signal generators were connected to channels 1, 2, 3 and 4. The function generators were set to create sine waves at 60, 61, 62, and 63 MHz. A simple GNURadio flow graph took the data from the HackRF One and plotted the FFT of the input signal. Each signal generator was turned on and fed the respective sinusoid into the Operacake and HackRF One. In the FFT there is a distinctive peak at 60, 61, 62, and 63 MHz. And as each generator was turned off, the respective signal dropped from the FFT plot.

4.1.4 Pseudo Doppler Direction Finder Software

The final test of the PDDF System and software was conducted using the anechoic chamber at Virginia Tech. A small antenna array was built using standard WiFi antennas spaced in a 1.2 inch (3 cm) radius circle. This array was connected to the Operacake and HackRF

One and placed in the chamber. A -10 dB 2.4 GHz far field signal was generated. The array was placed with antenna 2 approximately facing the far field source and data collected commenced.. After a few seconds data collection was halted and the array was rotated approximately 180 degrees. The far field signal was generated and data collect commenced again. The resulting bearing estimates were plotted in a histogram.

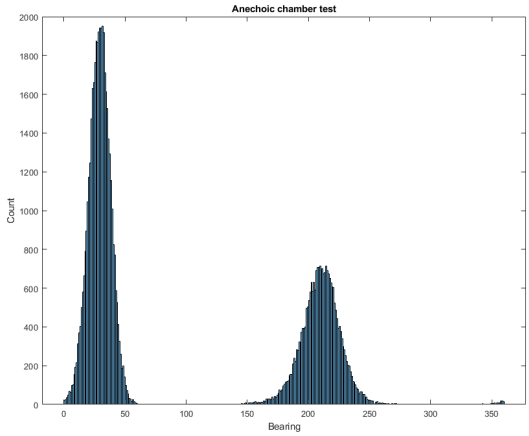


Figure 4.8: Histogram of angle of incidence calculations from anechoic chamber test

4.8 shows the 180 degree phase change, with two distinct signals close to 50 and 220 degrees. In addition the data showcased a Gaussian distribution of bearing estimates.

4.2 Simulations

Two sets of simulations were conducted as part of this work. The first was to understand more of the effects mounting the PDDF array on-board a UAS might have on the system's ability to measure phase change. This was in part due to how close the system will be to the signal source of testing, as Virginia Tech does not have any large, empty fields nearby. The second was to simulate and measure the accuracy of the Particle Filter.

4.2.1 UAS Mounted PDDF Simulations

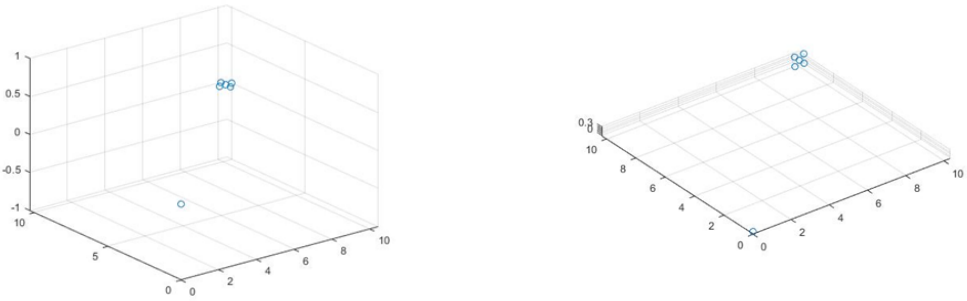
A set of simulations was created, to explore the impact of a PDDF system mounted on board a UAS and how the altitude and 3D orientation is space, as well as the drones motion, impacts the received signal. Since the major interest was near the signal source, the radio signal propagation was model using spherical wave fronts instead of assuming planer waves and assume the vehicle is moving in a straight line at a constant velocity. The equation for approximating the phase of the signal at any point is shown in 4.1 [29]

$$\phi_n(t) = -kr_n^{MS}(0) + \omega_{Dn}t + \frac{kr_n^{MS}(0)}{2}(\frac{vt}{r_n^{MS}(0)})^2 sin^2\theta_n \tag{4.1}$$

$r_n^{MS}(0)$ is the initial position of the UAS in a 2D plane, while ω_{Dn} is the Doppler frequency, equivalent to $kvcos(\theta_n)$. v is the UAS velocity, t is the current time step, and θ_n is the angle relative to the origin that the UAS is moving. k is the wave number.

The kinematics of the system were modeled using two reference frames, a global frame that contained both the signal source and the UAS, and a local frame centered on the UAS. The UAS-centric local frame is spherical, allowing the antennas' locations on the UAS to be calculated easily from the UAS roll, pitch, and yaw information. For each movement of the UAS, the antenna positions are recalculated in the local frame. these local frame positions are then converted to global frame coordinates and applied as offsets to the UAS's location in the global frame. 4.9 shows example orientations of the UAS and antenna array.

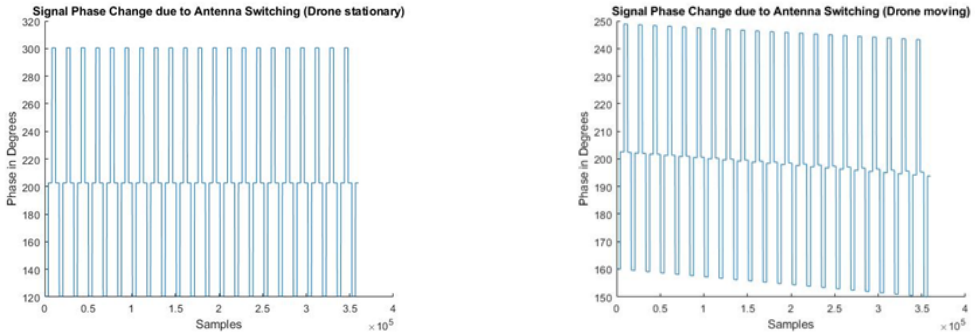
With the Doppler direction finding system, the only information needed is phase, as magnitude has no effect on the bearing calculation. As a result ideal omnidirectional antennas were used for both the transmitter and receivers, and only the phase information was calculated in the simulation to increase system performance. Phase was calculated directly as a



(a) UAS in level flight (b) UAS in steep bank

Figure 4.9: Examples of simulated UAS orientations

function of distance and wave number, with Doppler effects due to movement and changes in the UAS's orientation were applied.



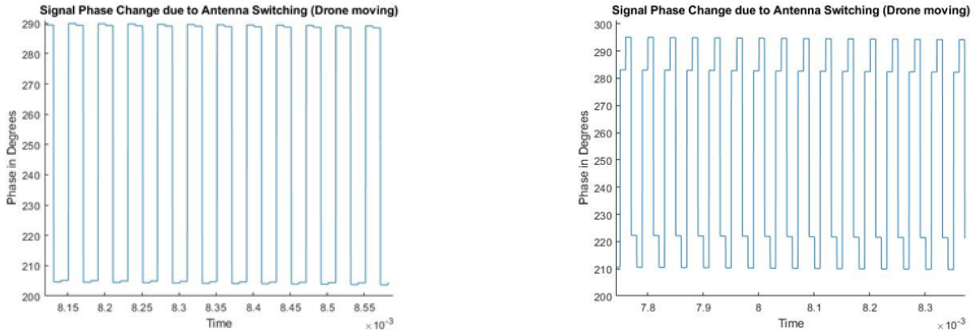
(a) Stationary UAS (b) UAS moving at 40 m/s

Figure 4.10: Simulation signals

In 4.10a, the UAS is stationary and is receiving good data. There is a constant phase at each antenna and an approximate 90 degree phase change when the antennas switch, going from 200 degrees at antenna 1 to 300 degrees at antenna 2 to 200 degrees at antenna 3 and 110 degrees at antenna 4 and so forth cyclically. The large phase change is easy to measure. When a constant velocity is applied to the UAS in 4.10b the resulting Doppler shift is very small, at 150MHz and 40 m/s forward velocity this shift is only .02 Hz.

However the most interesting result was the impact of the orientation of the array when close

to the transmitter. When the array was close to the transmitter, the near field spherical wave fronts were dominant over any kind of planer waves. At certain orientations, the phase of the two front antennas and the phase of the two rear antennas are almost the same, as seen in 4.11a. Instead, as in 4.10a, of a cyclic phase shift that imitates a sub-sampled sinusoidal function (0, +90, 0, -90) 4.11a switches between 205 degrees and 290 degrees. In effect, this creates a 2 element antenna array rather than a 4 element antenna array and 2.6 is no longer valid, meaning an unambiguous angle of incidence cannot be calculated. However if the orientation of the UAS is changed, a larger phase shift between antennas becomes present as seen in 4.11b and an unambiguous angle of incidence can be calculated.



(a) UAS directly above signal source (b) UAS tilted directly above signal source

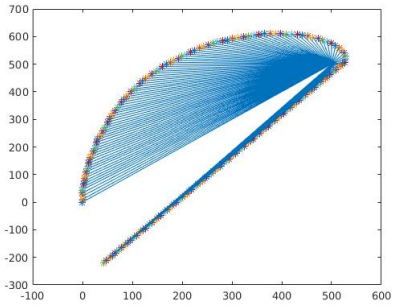
Figure 4.11: Simulation signals at various orientations

4.2.2 Particle Filter Simulations

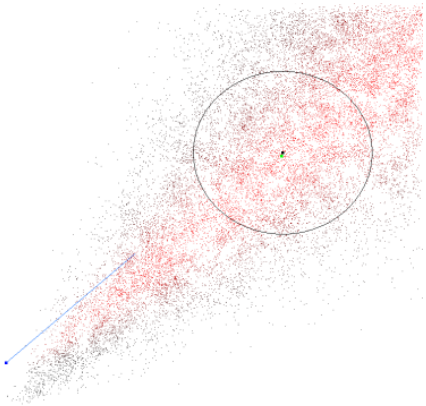
To evaluate the feasibility of a particle filter approach to tracking transmitting RF targets, several simulations were run using generated data. The motion of the target and the motion of the PDDF system were created and plotted in Matlab, and the bearing between the two calculated. Large noise was then added to the bearing measurement, on the order of +-15 degrees, and small noise to the position and acceleration parameters. The results were placed into a csv file that was fed into the particle filter. The visualizations below were created

using Matlab and OpenCV.

The first simulation shown in figure 4.12 was of a simple movement spiraling towards the target, the a constant movement away from the target once it was reached, as shown in figure 4.12a. The second simulation, shown in figure 4.13, was a simple straight line motion as shown in figure 4.13a.



(a) Ground true data

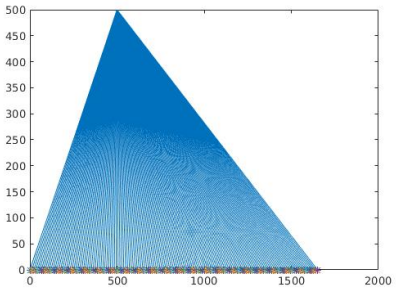


(b) Particle distribution after 1 update

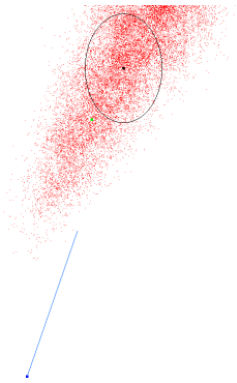
(c) End Result

Figure 4.12: Simulated particle filter with movement towards target. Green dot is the target, blue dot is the UAS, and black dot is the estimated target position.

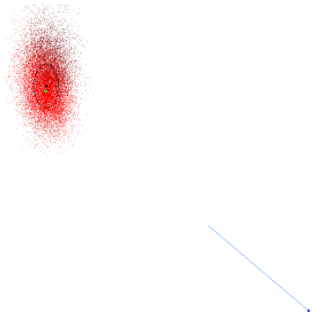
figure 4.13 shows that the particle filter approach has potential but suffers a degradation in accuracy unless the PDDF system's movement can produce a large bearing change. This is observable if the final particle distributions, figures 4.12c and 4.13c, are compared. figure 4.12c has a much smaller particle cloud and a estimation variance (black oval) than figure



(a) Ground true data



(b) Particle distribution after 1 update



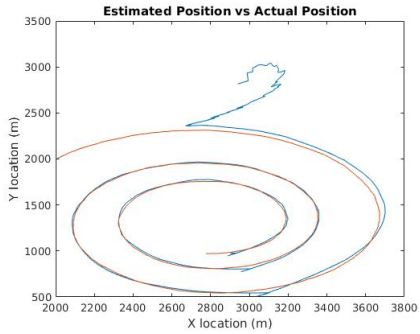
(c) End Result

Figure 4.13: Simulated particle filter with straight line movement. Green dot is the target, blue dot is the UAS, and black dot is the estimated target position.

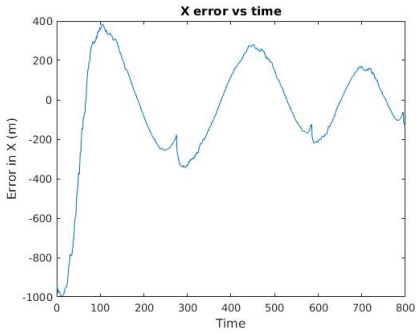
4.13c which means the target's position is being estimated with a higher confidence. The difference is due to how the simulated UAS moved, and how the larger changes between angle of incidence measures, the blue lines in figure 4.12a and 4.13a, allows a more accurate target position estimate.

Further simulations were conducted using a moving target, with one test being a target that constantly moves away from the PDDF system, figure 4.14, and the second being a random walk pattern with short-term linear behavior, figure 4.15.

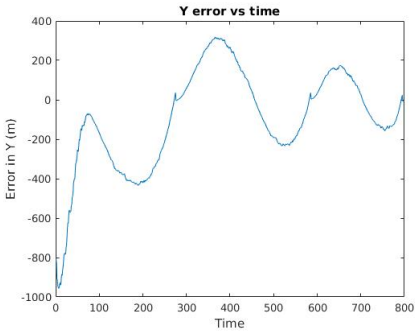
The avoiding target simulation, figure 4.14, resulted in some large errors early in the simulation as shown by the large distance errors in figure 4.14b and 4.14c, but as more data was



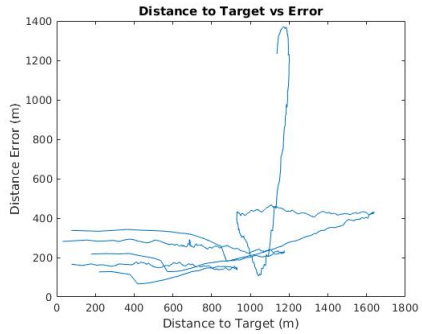
(a) Particle filter target estimated position (blue line) vs actual position (orange line)



(b) Error in X dimension between estimated position and actual position



(c) Error in Y dimension between estimated position and actual position

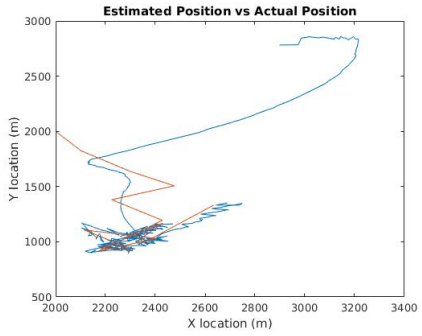


(d) Distance to the target vs total estimate error in meters

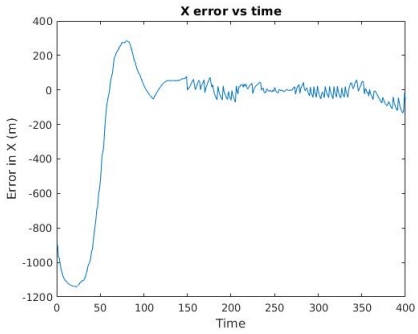
Figure 4.14: Active avoidance simulated particle filter

collected the error decreased, but remained cyclic as the nature of the particle filter created a lag when the target rapidly changed position.

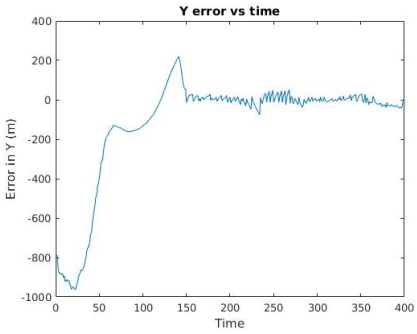
The random walk test, which used a target motion model similar to what Pozdnyakov et al. [25] propose, figure 4.15, performed much better, in a large part because the high speed of the PDDF system allowed it to catch up to the target and begin circling it, drastically reducing error, as seen in figures 4.15b and 4.15c.



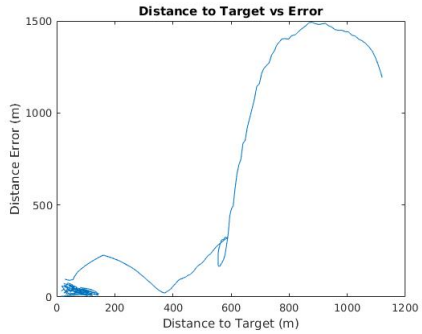
(a) Particle filter target estimated position (blue line) vs actual position (orange line)



(b) Error in X dimension in meters between estimated position and actual position



(c) Error in Y dimension in meters between estimated position and actual position



(d) Distance to the target vs total estimate error in meters

Figure 4.15: Random walk simulated particle filter

4.3 Stationary PDDF Tests

Once the PDDF system was confirmed to be functional, tests were conducted outdoors to determine the system's performance in a real world conditions. All tests were done in a large field located near the Virginia Tech campus due to the lack of a large open area without vertical surfaces on campus. The lack of vertical surfaces was important for these tests as data without large multipathing effects was desired. For the first set of tests, the PDDF was stationary and mounted on a small cart. In this fashion, a baseline for the system's performance could be measured before additional sources of error were added by moving the cart. For these tests a person was used to manually activate a small handheld radio to simulate a wildlife tracking collar.

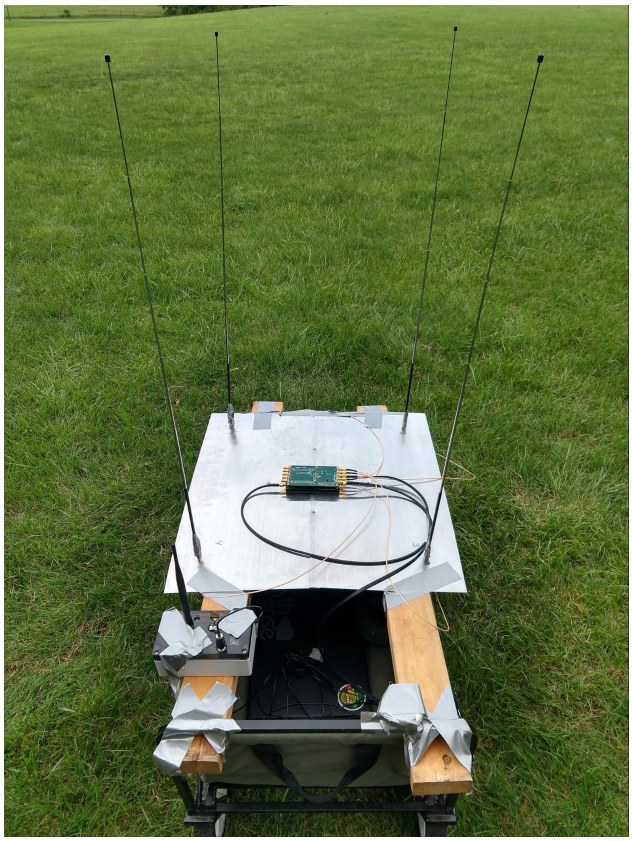
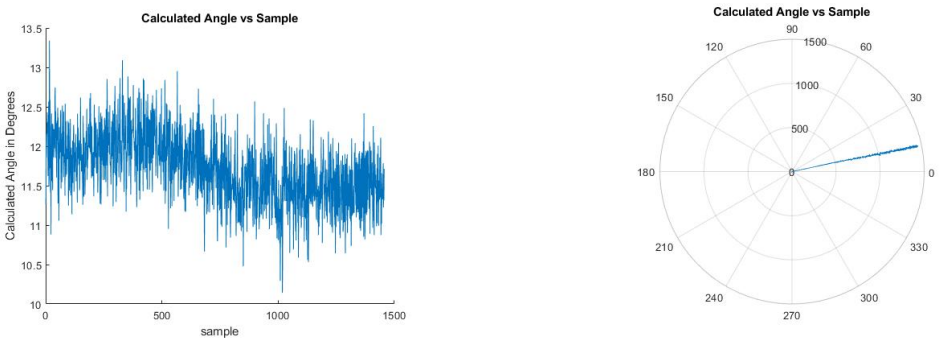


Figure 4.16: PDDF system mounted on cart

Three tests were conducted, the first being a simple measurement of noise in the calculated bearing. This test had the signal source stationary approximately 100 ft away from the PDDF array in line with antenna 1.

Once the data was captured, a histogram of the calculated angle of incidence measurements was created to visualize the distribution of measurements, shown in figure 4.18.

As shown in figure 4.17, the calculated angle of incidence for this test fluctuated around 11.9 degrees. As the signal source was aligned with antenna 1, the calculated angle of incidence should be 0 degrees. This difference is due to the phase change introduced by a combination of factors, including the antenna array spacing, coax cable length, and phase change due to software filtering. However this phase change is constant across all signals as long as



(a) Calculated angle of incidence plotted with Cartesian coordinates

(b) Calculated angle of incidence plotted with Polar coordinates

Figure 4.17: Calculated angle of incidence for a static signal and static PDDF

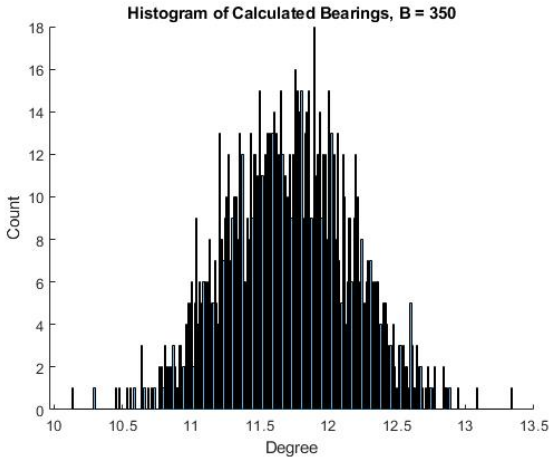
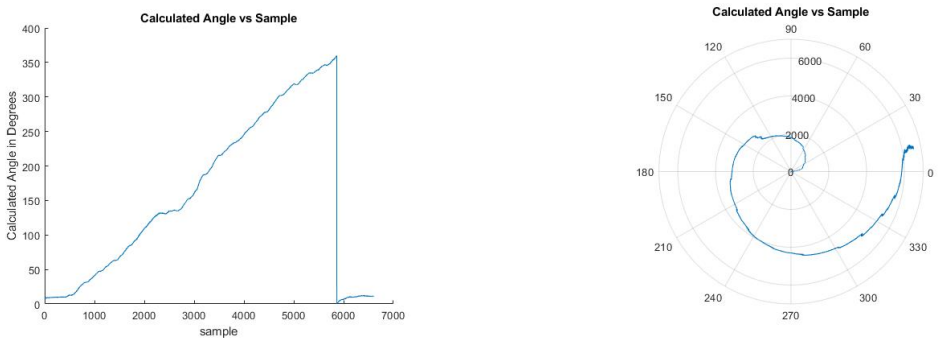


Figure 4.18: Histogram of calculated angle of incidence with 350 bins, given a bin width of 0.1 degrees

the above mentioned factors remain the same, meaning a simple offset can be applied to all future measurements conducted at the same frequency.

The next test was a full 360 degree walk around the array to ensure there were no issues with antenna gains, no nulls, and that the entire system worked as designed. As shown in figure 4.19, the system tracked the moving signal very well.

The final set of tests conducted was a stationary PDDF array and a moving signal. The signal source was located approximately 100 ft away from the PDDF array and walked from



(a) Calculated angle of incidence plotted with Cartesian coordinates

(b) Calculated angle of incidence plotted with Polar coordinates

Figure 4.19: Calculated angle of incidence for a signal completing 1 360 revolution around the array

antenna 4 towards antenna 1. A map with the signal's GPS track is shown in figure 4.20.

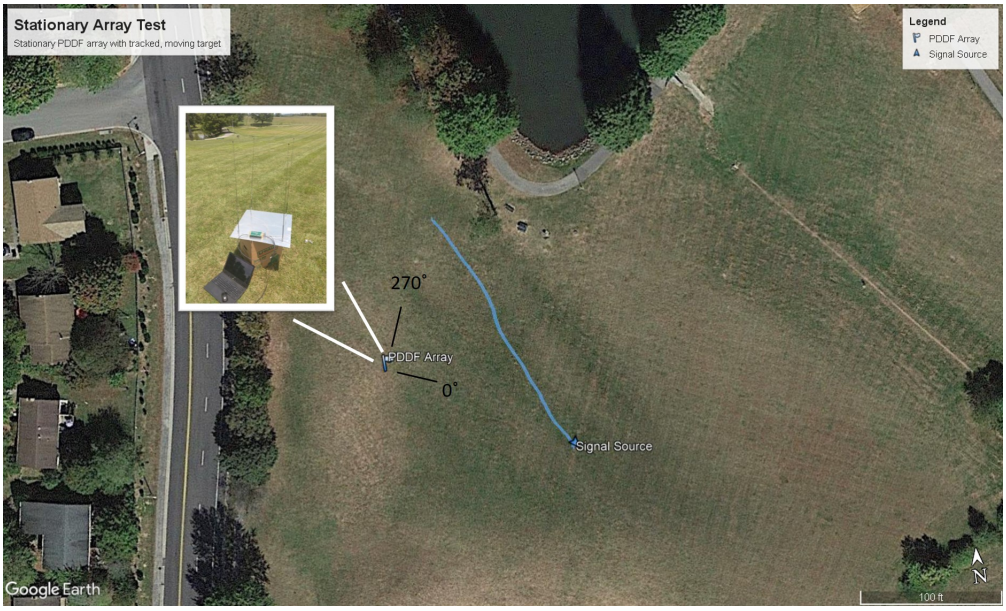
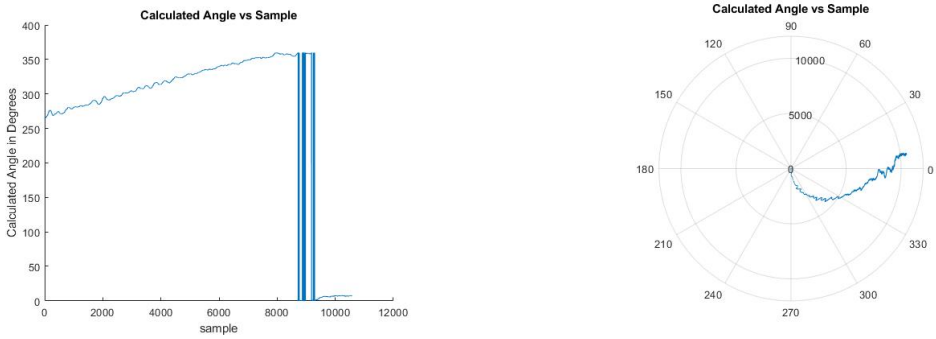


Figure 4.20: Map showing the PDDF position as well as the moving signal's path

The data was then plotted to see if the calculated angle of incidence was related to the motion of the signal source. The data aligned very well with the motion of the signal source, however there is a small ripple present in the signal, visible in figure 4.21a and 4.21b. This ripple results from the cyclic up and down motion of the signal source due to the walking

motion of the person carrying the transmitter.



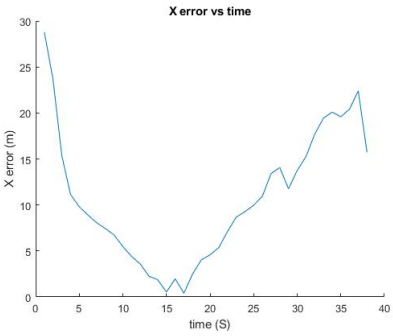
(a) Calculated angle of incidence plotted with Cartesian coordinates

(b) Calculated angle of incidence plotted with Polar coordinates

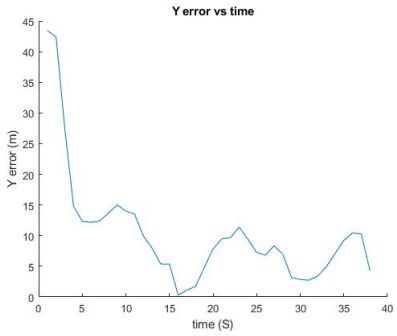
Figure 4.21: Calculated angle of incidence for a signal moving in a straight line

To confirm the particle filter was working and to evaluate its performance on a data set with minimal additional noise, the data shown in figure 4.21 was fed into the particle filter. Since the only information used in the particle filter is position, velocity, and angle of incidence; a stationary array and moving target can be treated the same as a moving array and stationary target. The particle filter was initialized with a 200m by 200m grid, shown in figure 4.22d.

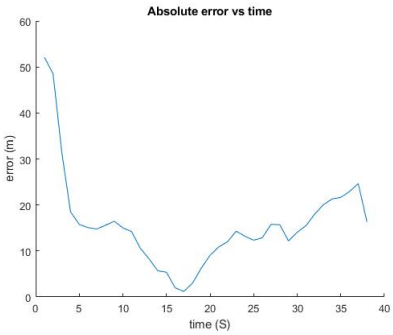
The results of the test, shown in figure 4.22 were very good. The straight line motion was able to localize the target to within 30 meters, shown in figure 4.22c after 5 angle of incidence measurements and was able to keep the error under 30 meters throughout the test. In addition, the final particle distribution, figure 4.22e, is very similar to the simulated particle filter test, figure 4.13c. The larger error in x dimension, figure 4.22a compared to the y dimension, figure 4.22b, shows how the one dimensional nature of the particle filters sensor model tends to overestimate the distance to the target. This overestimation can be limited by increases the change in angle between sequential angle of incidence measures, which is further explored in section 4.4.



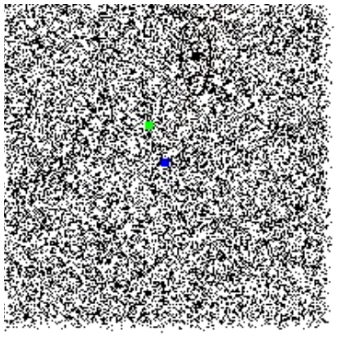
(a) Particle filter estimation error in the X for a stationary array



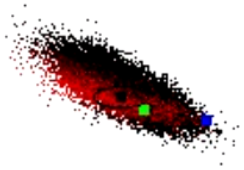
(b) Particle filter estimation error in the Y for a stationary array



(c) Particle filter estimation error distance for a stationary array



(d) Initial particle filter cloud distribution



(e) Final particle filter cloud distribution

Figure 4.22: Particle filter results of a stationary array and tracked target using a 200 meter by 200 meter grid. The blue dot is the PDDF array and the green dot is the signal source. The black dot is estimated location of the signal source

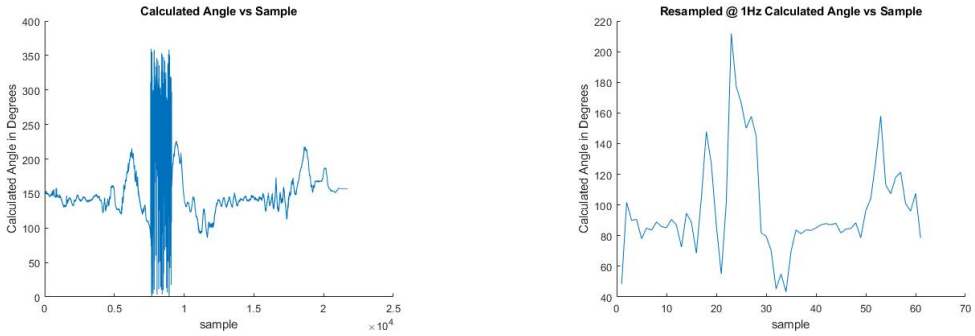
4.4 Moving PDDF Array Tests

Once the static cart and mobile signal tests were completed, another series of tests were conducted to generate additional data to be fed into the particle filter. GPS was used to record the speed of the cart along with its heading and position. For the first test the cart was pulled at a constant speed along a walking path with the signal source located 100 meters away as shown in figure 4.23.

A third of the way into this test the broadcasting radio signal was turned off to introduce noise into the particle filter to see if it would be able to recover. This noise is visible in figure 4.24a between 8,000 and 9,000 samples. The base data rate of the PDDF is approximately 330 Hz, which was down-sampled to 1Hz to match the GPS data. This down-sampled signal in shown in figure 4.24b.

Figure 4.23: Map showing the signal source and path of the PDDF system for the first particle filter data collection

The angle of incidence data along with the GPS location, speed and heading were combined using a matlab script to create the input csv file for the particle filter. The particle filter



(a) Calculated angle of incidence at full data rate

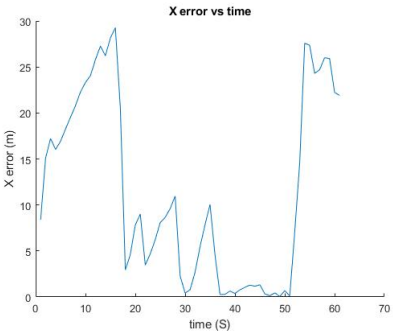
(b) Calculated angle of incidence reduced to 1Hz

Figure 4.24: Moving PDDF array data set with noise introduced

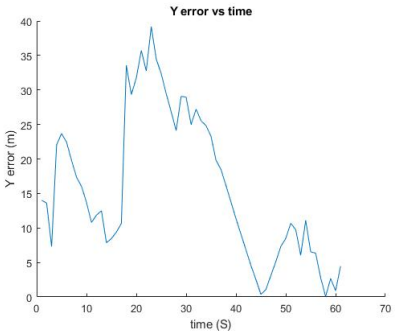
was run twice on each set of data, once with a starting grid 2km by 2km in size, and one with the starting grid set to 200m by 200m.

The results of the 200m by 200m particle filter using the first data set, shown in figure 4.25 was good. Figures 4.25a and 4.25b show the difference between the estimated signal position and the true signal position in meters as the test progressed. Even with the noise added to the data set, the particle filter was able to localize the transmitter within 40 meters for the entire test, as shown in figure 4.25c. The particle cloud itself started with a uniform distribution, shown in figure 4.25d and was able to converge, as seen in figure 4.25e. There is an offset between the true position (the green dot) and the estimated position (the black dot) in figure 4.25e, in part due to the noise added to the system intentionally and from the antenna geometry changing from both the motion of the cart and vibrations from rolling over obstacles. Geometric dilution of precision also played a part, exaggerating angle errors the closer the PDDF got to the target.

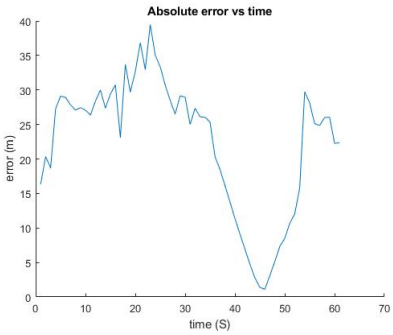
When the starting grid was increased to 2km by 2km, the results were very similar to the 200m by 200m grid results. The error was slightly larger, but still was under 50 meters as shown in figure 4.26c. Both particle filter results are very similar to the simulation results



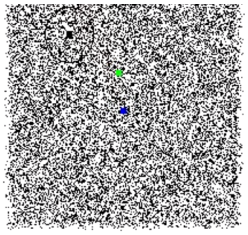
(a) Particle filter estimation error in the X for run 1



(b) Particle filter estimation error in the Y for run 1



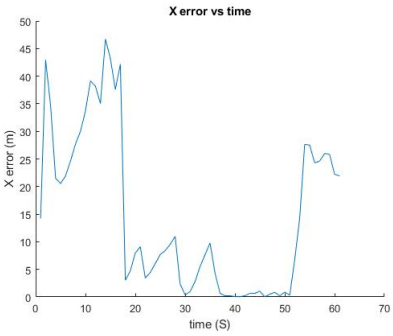
(c) Particle filter estimation error distance for run 1



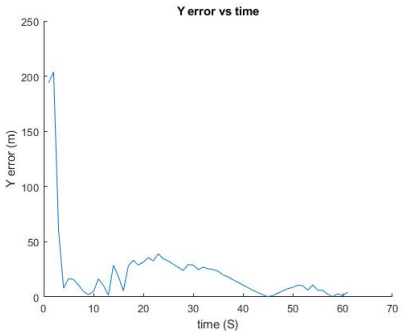
(d) Initial particle filter cloud distribution

(e) Final particle filter cloud distribution

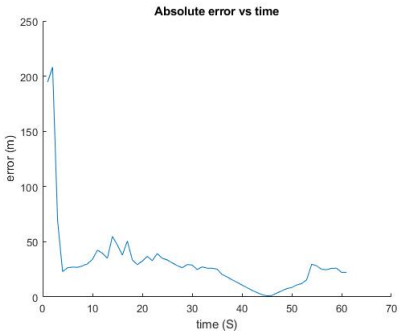
Figure 4.25: Particle filter results of data set 1 using a 200 meter by 200 meter grid. The blue dot is the PDDF array and the green dot is the signal source. The black dot is estimated location of the signal source



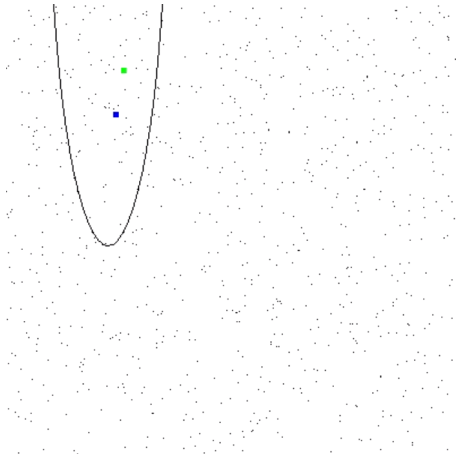
(a) Particle filter estimation error in the X for run 1



(b) Particle filter estimation error in the Y for run 1



(c) Particle filter estimation error distance for run 1



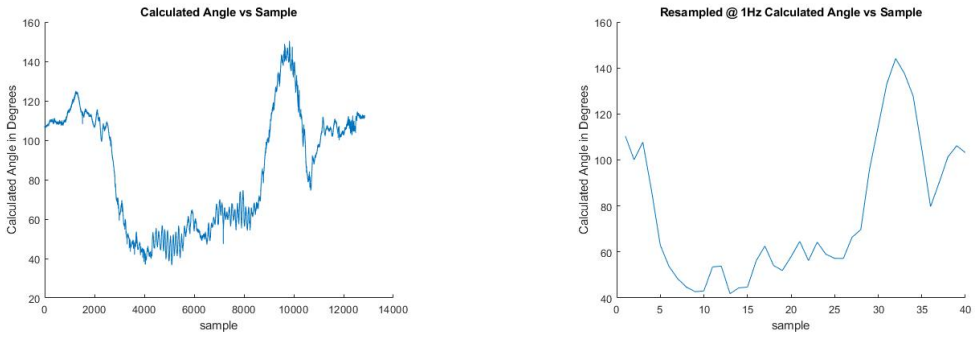
(d) Initial particle filter cloud distribution

(e) Final particle filter cloud distribution

Figure 4.26: Particle filter results of data set 1 on a 2000 meter by 2000 meter grid. The blue dot is the PDDF array and the green dot is the signal source. The black dot is estimated location of the signal source

shown in figure 4.13 and to that of the stationary array test, figure 4.22, and exhibit the same tendency to overestimate the distance to the target in the X dimension. Based on the simulations, the best way to minimize this overestimation is to have larger bearing changes in both the X and Y dimension. Such a path is shown in figure 4.27 was used to generate more data for the particle filter to see if the overestimation would be reduced.

Figure 4.27: Map showing the signal source and path of the PDDF system for test 2



(a) Calculated angle of incidence at full data rate

(b) Calculated angle of incidence reduced to 1Hz

Figure 4.28: Moving PDDF array data set circling around signal source

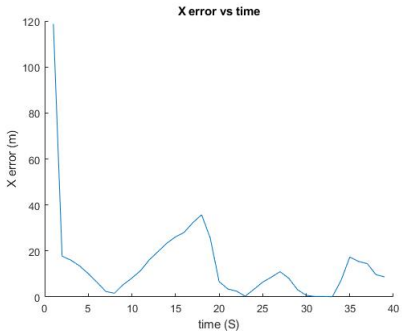
The second test, shown in figure 4.27, was conducted in a field and involved a stationary

signal with PDDF array moving completely around the source. Based on the simulations done in section 4.3, increasing the change between sequential angle of incidence measurements should reduce the error in the localization estimate. The same method of down-sampling to match GPS was done as in the first test and is shown in figure 4.24.

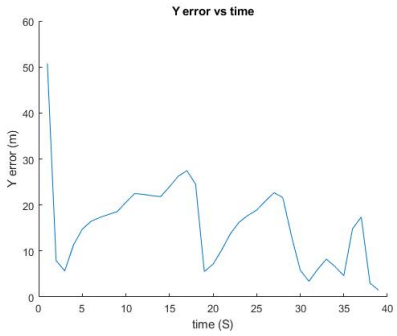
The results of the 200m by 200m particle filer is much better and confirms that a rapid change in angle of incidence is helpful for reducing estimated system errors, as shown in figures 4.29a and 4.29b. The total error distance between the actual location and the estimated location was 8.84 meters at the end of the test, and is shown in figure 4.29c and stayed under 40 meters for the duration of the entire test. In addition, the figures 4.29a and 4.29b show the same cyclic increase and decrease in error as the simulations presented in figures 4.14b and 4.14c. When the particle filter's starting grid was increased to 2km by 2km, the results, shown in figure 4.30 are nearly identical to the 200m by 200m results.

Two notes were made regarding the quality of data collected in these tests. As the cart moved, the antennas in the array sway due to vibrations resulting in the rippling in the calculated angle of incidence. These ripples resulted from the change in distance between the antennas creating a phase shift in the calculated signal and can be seen in figures 4.24a and 4.28a. In addition, since the ground was not level the plane of the antenna array moved vertical with respect to the stationary antenna, which also resulted in a change in phase and directly impacted the calculated angle of incidence. Several methods could reduce this impact and increase system accuracy, and will be discussed in Chapter 5.

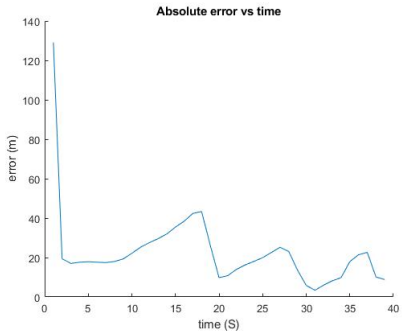
Geometric dilution of precision also played a factor in the estimation errors with the particle filters. While a 1 degree angle of incidence error at 300 meters has a fairly small impact relative to the arc-distance of that angle error, when closer to the transmitter that 1 degree of error has a much larger impact. Unfortunately due to the limitations of ground testing longer range tests were not possible which would have reduced this source of error. There



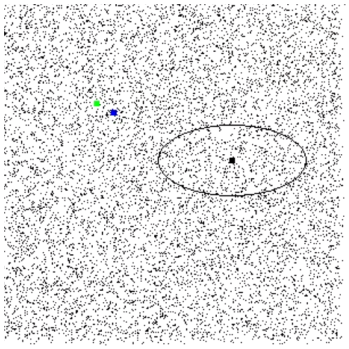
(a) Particle filter estimation error in the X for run 2



(b) Particle filter estimation error in the Y for run 2



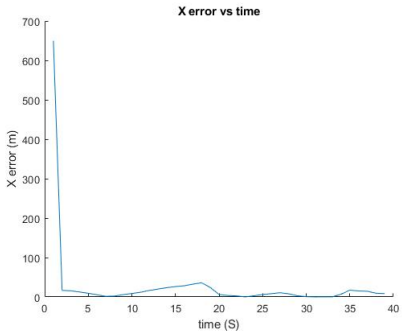
(c) Particle filter estimation error distance for run 2



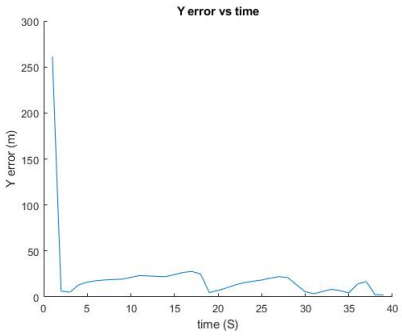
(d) Initial particle filter cloud distribution

(e) Final particle filter cloud distribution

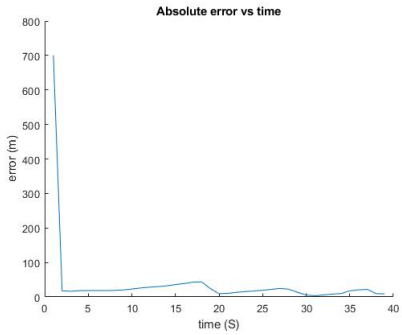
Figure 4.29: Second particle filter results in 200 meter by 200 meter grid



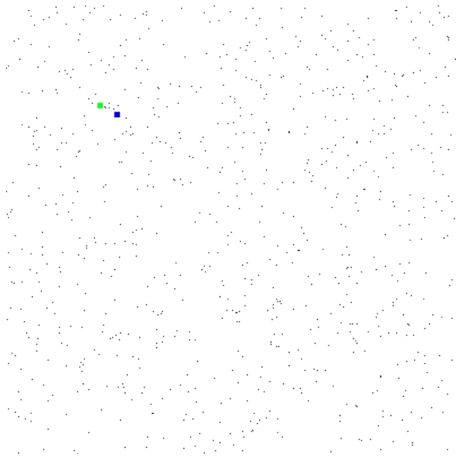
(a) Particle filter estimation error in the X for run 2



(b) Particle filter estimation error in the Y for run 2



(c) Particle filter estimation error distance for run 2



(d) Initial particle filter cloud distribution

(e) Final particle filter cloud distribution

Figure 4.30: Second particle filter results in 200 meter by 200 meter grid

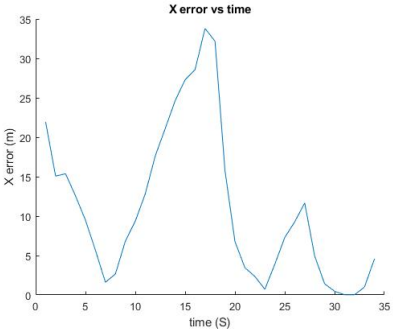
is not a large, level field without large vertical structures nearby the Virginia Tech campus available for use, in part due to the topology of Blacksburg. As a result, the PDDF array and transmitter were not be more than 100 meters apart.

Finally, with a little tuning the particle filter's performance can be improved, primarily by stopping the filter once a user specified variance in the estimated position is reached. Using this method, positional estimation error of under 5 meters was achievable with the data set in figure 4.28b as shown in figure 4.31.

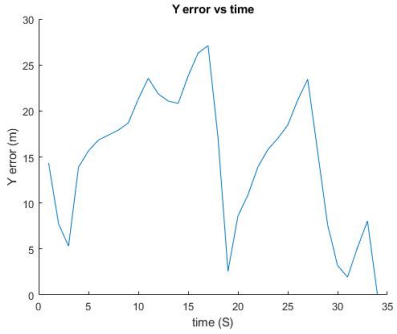
A summary of all the results is found in table 4.1.

Table 4.1: Particle filter estimation errors at last sample of filter

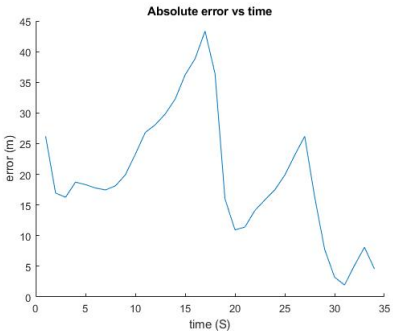
Data Set	Grid Size	X Error	Y Error	Total Distance Error
Data Set 1	200 m X 200 m	21.92 m	4.45 m	22.37
Data Set 1	2 km X 2 km	21.92 m	4.52 m	22.38
Data Set 2	200 m X 200 m	8.72 m	1.47 m	8.84
Data Set 2	2 km X 2 km m	8.58 m	2.28 m	8.87
Data Set 2 tuned	200 m X 200 m	4.57 m	0.024 m	4.57



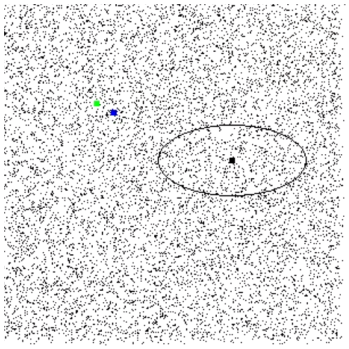
(a) Particle filter estimation error in the X for run 2



(b) Particle filter estimation error in the Y for run 2



(c) Particle filter estimation error distance for run 2



(d) Initial particle filter cloud distribution

(e) Final particle filter cloud distribution

Figure 4.31: Second particle filter tuned results in 200 meter by 200 meter grid

Chapter 5

Conclusions

The goal of designing a low cost Hybrid UAS has been accomplished. This design has all the operational benefits of a multi-rotor UAS, combined with the longer operational range and higher payload capacity of a fixed wing UAS. This design is suitable for tracking fast moving wildlife in rough terrain and heavy vegetation, such as bats. The VTOL capabilities allow the system to launch from areas with little horizontal space, such as woodland clearing or river banks. The forward flight characteristics allow the system to operate at high speeds and altitudes over longer distances while maintaining an ability to stop and hover, ideal for following birds and bats. In addition, this design is cheap to build and easy to maintain due to the low complexity and use of commercial off the shelf components.

In combination with the Hybrid UAS, a low cost Pseudo Doppler Direction Finder system is presented. This system, unlike other systems in literature, is able to acquire signals without mechanical rotation and has a 330Hz sample rate. This sample rate is four order of magnitudes faster than other wildlife tracking techniques. [28] [30] [11] [12] When used with the particle filter developed in this work, the PDDF system is capable of tracking a VHF signal's position under 40 meters or error, comparable to other works in this area. Desrochers et al. [15] [10] This high sample rate, when combined with the particle filter and an IMU system, would enable the UAS to track the location of fast moving wildlife in near real time. However two items still need to be completed to enable real time tracking.

All systems were functionally, but higher quality IMU data tightly coupled to the bearing

measurements would improve the accuracy of the particle filter's localization and enable the use of the PDDF systems full data rate. Bracing the antennas in a way to prevent changes to the geometry of the array is also necessary to improve angle of incidence accuracy. In addition, a method of rating the quality of the angle of incidence calculation is needed to order to prevent bad data from impacting the particle filters estimations. Two methods, one uses the HackRF One and one using a different platform, could result in more reliable data acquisition.

5.1 Future Work

Several avenues for future work exist that would improve the direction finder and the particle filter's performance. Foremost, a confidence rating of bearing quality is needed. Even with no incoming signal, or a poor incoming signal not suitable for high accuracy bearing measures, the PDDF system will still generate data. One method to provide that confidence rating using the HackRF One and Operacake combination is measuring the envelope of the output of the notch filter. During testing, it was noted that a good bearing measure always occurred while the output of the notch filter presented a constant envelope. If the output of the notch filter showed fluctuations in amplitude, the bearing result was poor. By using the envelope's rate of change as a measure of signal quality, and saving that information with the bearing, the particle filters performance could be improved by increasing the impact of the good bearing data and decreasing the impact of bad bearing data. This confidence rating would also allow multiple non-constant signals to be tracked reliably. Another method of providing a confidence rating and potentially improving system performance requires a different SDR platform. KerberosSDR is a 4-input coherent SDR that will be available late 2019 for $150.

Instead of using an RF switch to change between antennas, the KerberosSDR has four SDRs

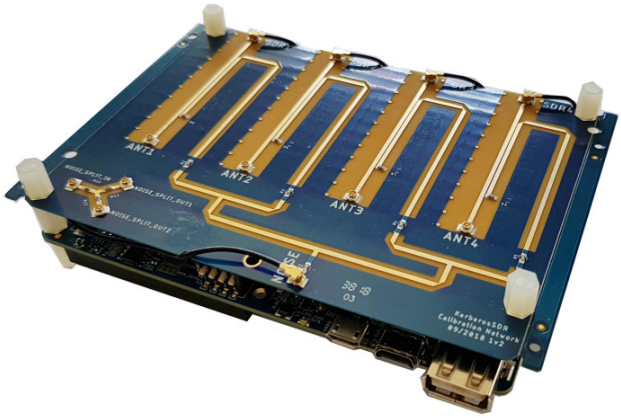


Figure 5.1: KerberosSDR platform, Image from indiegogo.com

sampled simultaneously. This allows three things. One, the KerberosSDR can use a linear array of antennas to do interferometery based direction finding by directly comparing the phases at each antenna. This method may be more immune to noise than the pseudo Doppler direction finding method, and potentially could be more compact. Second, if pseudo-doppler direction finding is desired, the rotation can be handled in software instead of hardware. Finally, the phase information at each antenna is being measured and as a result a simple confidence rating could be derived by subtracted the phases of the two antennas located directly across from one another. The difference in phase between two antennas directly across from one another should be constant, and in a perfectly spaced array 180 degrees apart, irregardless of the signal. If this phase change is not constant, then multipathing or other propagation effects are affecting the antenna array and the angle of incidence calculations will be less accurate. The final improvement would be a tightly coupled IMU/GPS/direction finding system. Having one system that could collect all the data at the same time would enable the particle filter to be run in real time onboard the system and present the estimated transmitter location in real time.

All software and supporting documents can be found at: https://github.com/raaslab/PDDFThesis

Bibliography

[1] Dji the world leader in camera drones/quadcopters for aerial photography. URL https://www.dji.com.

[2] About doppler dfs. URL http://silcom.com/~pelican2/PicoDopp/ABOUT_DOPP.html#DS.

[3] *FAA National Forecast FY 2019-2039*.

[4] Propeller thrust estimator. URL http://www.flybrushless.com/tools/thrustCalc.

[5] Scorpion sii-2215-1400kv motor propeller data. URL http://innov8tivedesigns.com/images/specs/SII-2215-1400-Specs.htm.

[6] Vtol uav with the cruise efficiency of a conventional fixed wing uav. URL https://technology.nasa.gov/patent/LAR-TOPS-241.

[7] X - wing. URL https://x.company/projects/wing/.

[8] Introduction into theory of direction finding. *RohdeI & Schwarz Radiomonitoring & Radiolocation | Catalog 2016*, page 62–85, 2016.

[9] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966. doi: 10.1214/aoms/1177699147.

[10] Haluk Bayram, Krishna Doddapaneni, Nikolaos Stefas, and Volkan Isler. Active localization of vhf collared animals with aerial robots. *2016 IEEE International Confer-*

ence on Automation Science and Engineering (CASE), page 934–939, Aug 2016. doi: 10.1109/coase.2016.7743503.

[11] Haluk Bayram, Nikolaos Stefas, Kazim Selim Engin, and Volkan Isler. Tracking wildlife with multiple uavs: System design, safety and field experiments. *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, page 97–103, Dec 2017. doi: 10.1109/mrs.2017.8250937.

[12] Haluk Bayram, Nikolaos Stefas, and Volkan Isler. Aerial radio-based telemetry for tracking wildlife. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 4723–4728, Oct 2018. doi: 10.1109/iros.2018.8594503.

[13] David R Brillinger, Haiganoush K Preisler, Alan A Ager, John G Kie, and Brent S Stewart. Employing stochastic differential equations to model wildlife motion. *Bulletin of the Brazilian Mathematical Society*, 33(3):385–408, 2002.

[14] DR Brillinger, HK Preisler, AA Ager, and MJ Wisdom. Stochastic differential equations in the analysis of wildlife motion. *2004 Proceedings of the American Statistical Association, Statistics and the Environment Section*, 2004.

[15] André Desrochers, Junior Tremblay, Yves Aubry, Dominique Chabot, Paul Pace, and David Bird. Estimating wildlife tag location errors from a vhf receiver mounted on a drone. *Drones*, 2(4):44–53, Dec 2018. doi: 10.3390/drones2040044.

[16] Pedro Domingos. *The master algorithm: how the quest for the ultimate learning machine will remake our world*. Basic Books, a member of the Perseus Books Group, 2018.

[17] Rudolf Grabau and Klaus Pfaff. Funkpeiltechnik franckhsche verlagshandlung, 1989.

[18] Mevin B Hooten, Henry R Scharf, Trevor J Hefley, Aaron T Pearse, and Mitch D

Weegman. Animal movement models for migratory individuals and groups. *Methods in Ecology and Evolution*, 9(7):1692–1705, 2018.

[19] Adam M. Johansen. Smctc: Sequential monte carlo in c++. *Journal of Statistical Software*, 30(6), 2009. doi: 10.18637/jss.v030.i06.

[20] Julie Linchant, Jonathan Lisein, Jean Semeki, Philippe Lejeune, and Cédric Vermeulen. Are unmanned aircraft systems (uass) the future of wildlife monitoring? a review of accomplishments and challenges. *Mammal Review*, 45(4):239–252, 2015. doi: 10.1111/mam.12046.

[21] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. doi: 10.2307/2669847.

[22] N. Metropolis. The beginning of the monte carlo method. *Los Alamos Science (1987 Special Issue dedicated to Stanislaw Ulam)*, page 125–130, 1987.

[23] Joe Moell. Wide-range antenna arrays for the roanoke doppler, Apr 2008. URL http://www.homingin.com/newdopant.html.

[24] Randeep Pannu, Antonio Samuel, Willie Thompson, Thomas Goodman, Patrick Wolfram, Brandon Luquette, Chris Hinkey, Joe Reinhart, and Jim McCloskey. Distributed em situational awareness based on raspberry pi and redhawk. *AOC Susquehanna Tech Challenge 2013*, Mar 2014.

[25] Vladimir Pozdnyakov, Thomas Meyer, Yu-Bo Wang, and Jun Yan. On modeling animal movements using brownian motion with measurement error. *Ecology*, 95(2):247–253, 2014.

[26] D. Blake Sasse. Job-related mortality of wildlife workers in the united states, 1937-2000. *Wildlife Society Bulletin (1973-2006)*, 31(4):1015–1020, 2003.

[27] Peter E. Smouse, Stefano Focardi, Paul R. Moorcroft, John G. Kie, James D. Forester, and Juan M. Morales. Stochastic modelling of animal movement. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1550):2201–2211, 7 2010. ISSN 0800-4622. doi: 10.1098/rstb.2010.0078.

[28] Amir Torabi, Michael W. Shafer, Gabriel S. Vega, and Kellan M. Rothfus. Uav-rt: An sdr based aerial platform for wildlife tracking. *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug 2018. doi: 10.1109/vtcfall.2018.8690711.

[29] Ekman Torbjörn. *Prediction of mobile radio channels: modeling and design*. PhD thesis, Uppsala University Signals and Systems, 2002.

[30] Kurt Vonehr, Seth Hilaski, Bruce E. Dunne, and Jeffrey Ward. Software defined radio for direction-finding in uav wildlife tracking. *2016 IEEE International Conference on Electro Information Technology (EIT)*, page 0464–0469, May 2016. doi: 10.1109/eit.2016.7535285.

[31] Hidetsugu Yagi and Shintaro Uda. Projector of the sharpest beam of electric waves. *Proceedings of the Imperial Academy*, 2(2):49–52, 1926. doi: 10.2183/pjab1912.2.49.