Russell Silva
5/22/2018
AMRUPT, Spring ‘18

Final Report

I. Executive Summary

Animal Movement Research Using Phase-based Trilateration (AMRUPT) is a technology being designed for the localization of small animals in the field of ecology. This technology is proposed to suit ornithological studies involving individual- and species-level migratory patterns and social interactions. The system utilizes Phase Interferometry for use in estimating the Angle of Arrival (AoA) of radio signals. These systems are substantially more accurate than other common methods, although performance scales strongly with the spatial scale of the receiver network. To accomplish this, a low weight radio tag is being developed to transmit signals to radio base stations. These tags will transmit sub 1-GHz UHF frequencies. To keep our system low-cost, we will be using inexpensive software defined radios (SDRs) for our basestations to obtain radio-wave measurements required for AoA calculations. Each basestation will consist of a three RTL-SDR network and each RTL-SDR in the network will be connected to an antenna, an RF switch for synchronization purposes, and a Raspberry Pi. The Raspberry Pi serves as an embedded device which will collect and analyze the characteristics of the received radio signal. Each Raspberry Pi will have an open-source radio coding platform called GNU radio installed, equipped with the software protocols necessary for obtaining accurate AoA measurements, which will then allow us to triangulate a signal location.

II. Introduction

The accurate and real-time localization of small-bodied animals in the field of ecology is imperative to determine individual- and species-level migratory patterns, social interactions, and other key behaviors. Many attempts[1,2,3] have been made to determine the positioning of animals temporally and spatially in the past, but have been either inaccurate (errors over five meters) or have required constant manual human intervention. Since direction finding requires wireless telecommunication, measurements have been thwarted by multipath interference from vegetation, electromagnetic interference, or other environmental conditions. Our objective is to develop a cost effective and automated system to track animal movements within the range of five meters while taking into account expected causes of error. Our proposed system consists of a receiver architecture that is built specifically for phase interferometry direction finding to facilitate accurate measurements from radio tags on tracked individuals.

III. Review of Literature

Many different techniques have been explored to achieve localization. Transmitted signals at antenna array elements can be quantized at receivers to provide phase difference information such as in phase interferometry [4,6]. Phase based measurements can be skewed by multipath effects in the environment by constructive and destructive interference for line of sight signals [6]. Received signal strength can be also be used for localization through several different algorithms; however, these metrics have been found to be unreliable [5] and thus do not have high accuracy. Time Difference of Arrival (TDoA) [1] is a method in which the position of a transmitter is determined from the time differences between the arrivals of short-lived transmissions at receiver stations in a network. TDOA systems are not as susceptible to multipath effects; however, obtaining precise positioning from close proximity transmitters in TDOA is difficult because nanosecond synchronization is required to compare lightspeed propagated signals. Difficulties in achieving nanoscale synchronization in time difference of arrival can be mitigated by subsample interpolation at signal correlation peaks [1]. Subsample interpolation improves the resolution of the arrival time estimate to an accuracy that is better than a specified sampling period by interpolating between the samples of a correlation peak [1 - p. 40]. Another bottleneck with TDOA is sampling rate as our timing difference measurements are only as accurate as our maximum sampling rate. Finally, TDOA systems suffer from the requirement that signals are extremely broadband, which could make RF transmissions more susceptible to interference from multiple narrow-band signals in the broadband transmission's range. Also, broadband signal transmissions would make a system incompatible with a multi-frequency ranging approach. However, a simple transmitter in TDOA could be constructed to transmit wideband signals instead of broadband [1].

Although lightspeed propagation substantially helps with obtaining real time results, it adversely affects the collection of synchronized data at antenna array elements in the radio frequency direction finding systems mentioned. More intensive hardware synchronization can be avoided by using a time-modulated array to switch between antennas in a direction finding system [7], shown in Figure 1. However, these systems lack the precision in location tracking which we desire for our system, because unexpected time gaps (+/- nanoseconds) could make such an approach unsynchronized among phase difference results.

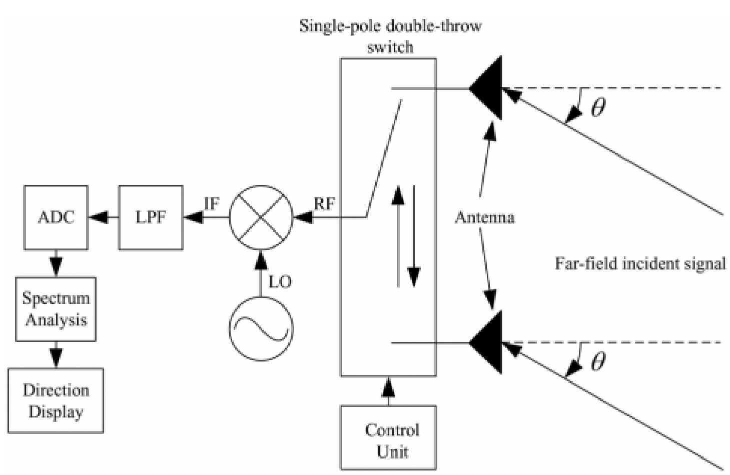

Fig. 1. Block diagram for direction finding of time-modulated array with harmonic characteristic analysis [7]

Even in systems with two ADC’s sharing a common clock signal, sampling does not start at the same time, resulting in bulk delays [9]. These synchronization errors can be corrected by incorporating a signal generator input or another type of external noise source to each channel [8].

A phase interferometry system with real time operation on multiple receivers driven by a common local oscillator was able to identify an angle of arrival within degrees for all emitter distances from 1 km to 100 km [1]. Because this system was driven by a common local oscillator, the resulting intermediate frequency at each receiver could be compared without phase-offset synchronization. The error .5± 2 threshold from this system is less than the degree angle of arrival threshold error determined ± 5 in the time-modulated approach [21]. Although, this error rate is promising, this receiver system was designed to receive line of sight airborne signals. This does not account for the adverse effects of a ground environment, which will add additional factors to AoA error such as multipath interference.

The phase interferometry system in [4] was primarily developed to handle UHF frequencies for airborne sources. Because the frequency of this system was relatively high for phase interferometry, antennas were spaced at distances larger than half the wavelength of transmitted signals. Thus, the system was optimized to handle phase ambiguity, which is explored more in the technical section of this proposal. Testing protocols and optimization were handled in Matlab and C, modeling the effects of antenna spacings on AOA accuracy under worst-case conditions. In the hardware setup, three antennas were used to resolve phase ambiguities and determine the azimuthal AOA in a synchronized three channel system with RF mixers driven by a common local oscillator. The design of this system provides a basis for a three antenna architecture as discussed in the technical section of this proposal. The block diagram of this system is shown in Figure 2.

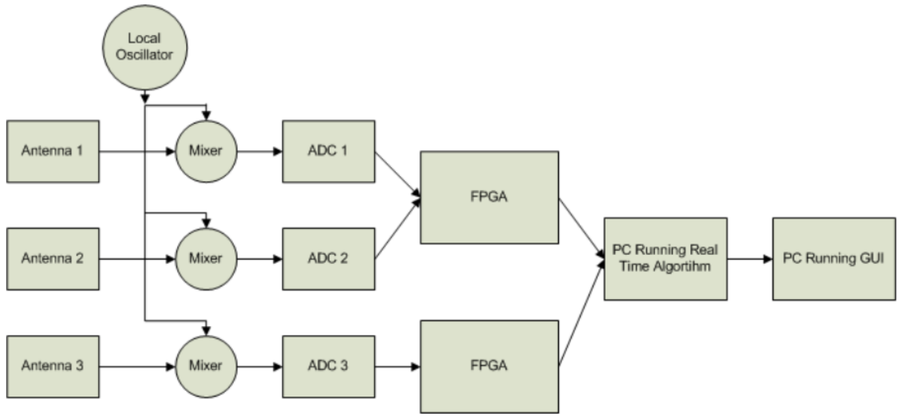

Figure 2: Block diagram for Phase Interferometry in Guerin, Jackson, and Kelly [4]

Ma, Hui and Kan [6] proposes a 3D indoor passive tag localization method with an accuracy of a few centimeters in a multi-frequency identification system. The paper leverages nonlinear elements in passive devices to generate second or higher-order harmonics which are used for location detection. This paper introduces a novel approach in mitigating multipath interference, defined as the occurrence when radio waves reach a receiver via two or more paths. This causes a constructive and destructive interference of the signal, as well as phase induced error in Figure 3.

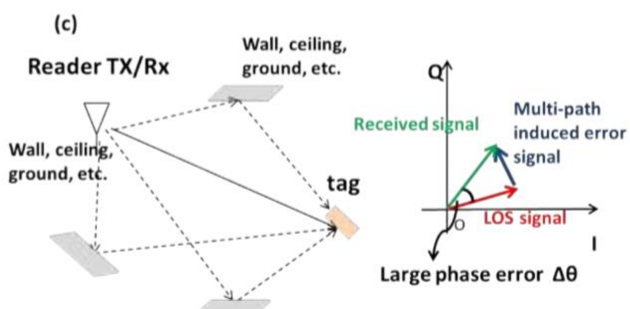

Figure 3: Dense indoor multi-path induced phase error [6]

In order to combat multipath interference, a phase error threshold defined by the bandwidth of operation and the number of frequencies used for ranging is used. An algorithm is implemented which uses heuristic multi-frequency continuous wave (HMFCW) ranging algorithm, where the frequency is dynamically adjusted to identify an optimal combination of frequencies based on the phase error threshold. An optimal frequency is more likely to yield a correct phase cycle integer (when the measured phase is θ, the actual phase can be $\theta + 2n\pi$) to determine a correct line of sight path. Ultimately, 3D positions are localized using differential distances modeled with optimized hyperboloid functions. The larger the bandwidth of the multi-frequency transmission, the more robust the system will be to multi-path induced phase error. The forward compatibility of our system to this one is discussed in the design objectives section of this proposal.

In order to improve the cost effectiveness of direction finding, [1] has used low-cost RTL SDRs to extract in-phase and quadrature samples from incoming radio signals for TDOA and AoA calculations respectively. Direction Finding Implementations using RTL SDRs are promising alternatives to more expensive options by achieving up to 3.5m accuracy in TDOA [1] and by having an extensive hobbyist base with multiple Github repositories such as this one [10], demoed here. The advantages of having this repository available to us is that it will provide us with a point of reference when implementing our code and hardware. This specific repository was a precursor to the RTL SDR system developed by Sam Whiting in [11].

The TDOA approaches discussed above have been promising, as ultra-wide bandwidth transmissions would be less susceptible to signal refractions from a cluttered environment. The TDOA approach implemented with RTL SDRs in [1] does have error thresholds within 5 meters; however, this TDOA system is limited by transmitter complexity.

In phase interferometry, triangulation error can be scaled lower with an increased number of receivers per unit area with angle of arrival calculations. This option will be effective with low cost and on site programmable receiver units such as the RTL SDR. Also, phase interferometry's potential for high accuracy with low frequency transmissions in cluttered environments can be further improved upon by the multi-frequency techniques in [6].

We choose a phase interferometry approach initially before coming across [1], and this project may switch into a TDOA approach in the future if we determine a TDOA system like [1] is computationally less expensive, more accurate, and can be altered to have more narrowband transmitters.

IV. Design Requirements

We have proposed the following objectives in the design:

1. The receiver system is low-power and can track up to 50 lightweight and low-power radio tags
2. System architecture is resilient in cluttered environment (unsusceptible to multipath interference, electromagnetic interference, and other environmental conditions)
3. System is able to achieve two dimensional high spatial accuracy (error for triangulation results is limited within 5 meters) with a 100-300m distance between receivers
4. Forward compatibility: Must be compatible with and adaptable to a multi-frequency-phase-integer-disambiguation approach for future versions.
5. System is cost-efficient (almost all components are commercially off-the-shelf)

The first objective is to successfully track the locations of 50 individuals in the testing environment. We need to design the tags as lightweight as possible since the individuals are small in size and heavy tags may affect the individuals' biological activities. To allow for the least possible human intervention during the tracking process, both the receivers and tags need to operate with minimal power consumption to increase automatic tracking period. In addition, both the transceivers (ground nodes) and tags (mobile nodes) follow a communication protocol in which the mobile nodes will go to sleep when they are not communicating with the ground nodes to reduce power consumption.

The communication protocol is an intended route for development, but has not yet been designed. It specifies that mobile nodes wake up every 5 minutes to prepare for data transmission to the ground nodes. The mobile node will receive a 5-second countdown signal once it wakes up. As soon as the mobile node is verified to be within the receiver's range and has good link, it will be synchronized to global time before it is given a scheduled transmission time by the receiver or sent back to sleep again. If the mobile node is not within range of any receiver, it will go to sleep and wake up every 5 minutes to check whether it's within range again. The complexity of the ground-node to mobile-node communication protocol will be governed by how accurate our receivers are when taking angles of arrival measurements. If angles of arrival from at least three base stations intersect to a triangulation area of no more than 5 meter error (discussed further) over the specified tracking area, then tags will not have to be linked to different receivers depending on location and the same set of receivers can be used for all tags. The communication protocol will also be used for a multi-frequency system, which is a possibility in the future of this project.

Furthermore, the system must be able to obtain accurate results in a cluttered environment. We agreed that a real environment would have substantial multiple interference as there will be trees and rocks that can reflect a wireless signal. Multipath interference could result a false transmit signal which would give us incorrect information about the location of the tags. The effects of multipath are more amply discussed in the literature review section.

We agreed to set the tracking accuracy of our system to 5 meters because this is a minimum requirement to monitor the social interactions and movements of small mammal species (trackable hosts can be expanded upon). We propose a triangulation algorithm for Phase 1 of the project in order to acquire this accuracy: Phase 1 will be the development of the necessary base station algorithms and hardware setup to achieve an accurate angle of arrival measurement. In a future semester, we plan to modify our system to better overcome the effects of multipath interference by frequency hopping to obtain minimum variation results (Phase 2). If necessary, we plan to implement a multi-frequency phase integer disambiguation system based on [6] that trilaterates positions of mobile nodes if a 5-meter accuracy level has not been achieved by previous efforts (Phase 3). Figure 4 illustrates error minimization with relation to AOA calculations:

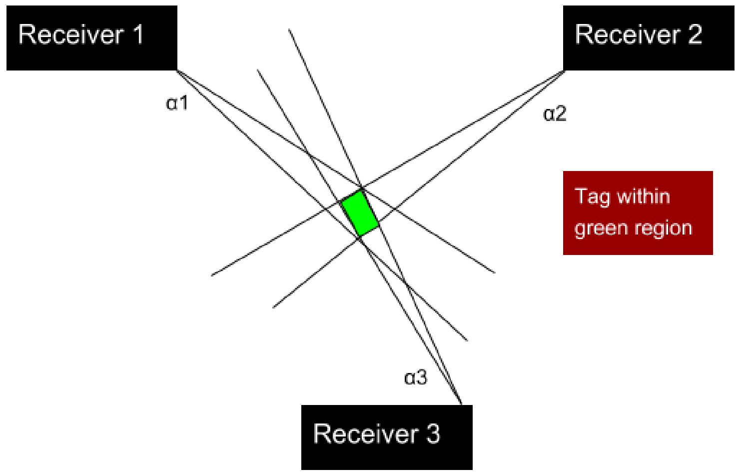

Figure 4: Error in triangulated area

In order to accomplish at least a 5 meter accuracy, a line of more than five meters cannot be drawn within the triangulated area of error. This area of error will be determined by α1, α2, and α3 (Figure 4) which resemble the angle of arrival error from receiver 1, 2, and 3 respectively. α1, α2, and α3 will be determined by phase difference errors from a transmitting RF signal to multiple antennas. Sources of AOA error are further discussed in the technical section of this proposal, and simulations have been planned to find algorithms that can make additional steps in minimizing this error.

To address forward compatibility, we must have a system in mind to distinguish tags. Code-division multiple access (CDMA) is a very popular method for doing this due to the coding infrastructure. We plan to use a less complex TDMA approach instead. In order to refine AoA measurements against multipath effects, HMFCW algorithms can be used to gather more angle of arrivals for less distorted line of sight signals.

After synchronizing samples and computing raw AoAs in our system, error thresholds as described in [6] can be computed for multiple sets of frequencies. Then, phase and consequently AoA measurements can be computed heuristically by HMFCW which is able to compute a correct cycle integer with 100% reliability when multi-path induced phase error is within $\pm 90^\circ \times$ BW%, and thus can resolve our multi-path interference issues.

V. Design Implementation

To accomplish our design objectives, we first needed to implement a rudimentary AoA estimator that could be improved upon in future iterations of this project. Since our localization system would certainly fail to be effective with a low two-dimensional spatial accuracy, we decided that obtaining AoAs with less than +/- 2.8° error with receiver distances at ~100 meters was paramount. Unfortunately, we did not achieve this objective this semester, but we did make ample progress towards this intermediate goal on the software end of the project, which includes having an Ubuntu OS fully prepared for the digital signal processing (DSP) of received radio signals and getting quadrature sampling working at coherently receiving RTL SDRs so that we have the radio signal characteristics we need for further DSP analysis and AoA calculations.

For increased readability and understanding, the design implementation section has been streamlined into six parts: System Architecture and Hardware Implementation, Software Overview, Sampling from the RTL SDR, Ettus OutOfTreeModules – Installation and Overview, GNU Radio Complete Flowchart, and Getting the GUI to Display Stable AoA. The System Architecture and Hardware Implementation section encapsulates the practical knowledge necessary to assemble the receiver basestations. The Software Overview section considers theoretical considerations (e.g. MUSIC algorithm costs and benefits) before stating the high-level software design. This section is included so that major software choices stated later in the Design Implementation have already been backed with explicit reasoning, and so that the reader can have a background knowledge of some of the major software components used (GNU Radio, Ettus library, etc.). The remaining parts refer to specific and detailed steps towards completing a software implementation that is close to generating accurate AoAs.

The following Empirical Testing section details the current state of the system, with suggested alterations of the previous design implementation section that would correct hardware/software shortcomings.

V.i. System Architecture and Hardware Implementation

Each basestation consists of interconnected RTL-SDRs to coherently receive RF signals from radio tag sources. Each RTL-SDR is a low-cost software defined radio which uses quadrature demodulation and receives frequencies from 500 kHz to 1.75 GHz [12]. A company at coherent-receiver.com has constructed this multichannel coherent basestation with four RTL SDRs, antenna switches, and integrated clock card. Figure 5 displays this device. The clock card is used to send a common clock signal to all four RTL SDRs. The common clock signal is a 28.8 GHz signal used to synchronize samples received at each RTL SDR. Even in systems with ADC's sharing a common clock signal, sampling does not typically start at the same time (bulk delays) [9]. Therefore, a noise generator (which can be replaced by an external function generator) is utilized during noise switching. In the noise switching process, antenna switches synchronously transition from a common noise signal to individual antenna signals. This antenna switching solves bulk delays by cross correlating samples from all four single channel receivers. Unfortunately, this has not been integrated yet into the current system. The effects of this missing, but fundamental function in our proposed system architecture is further discussed in the Empirical Testing section.

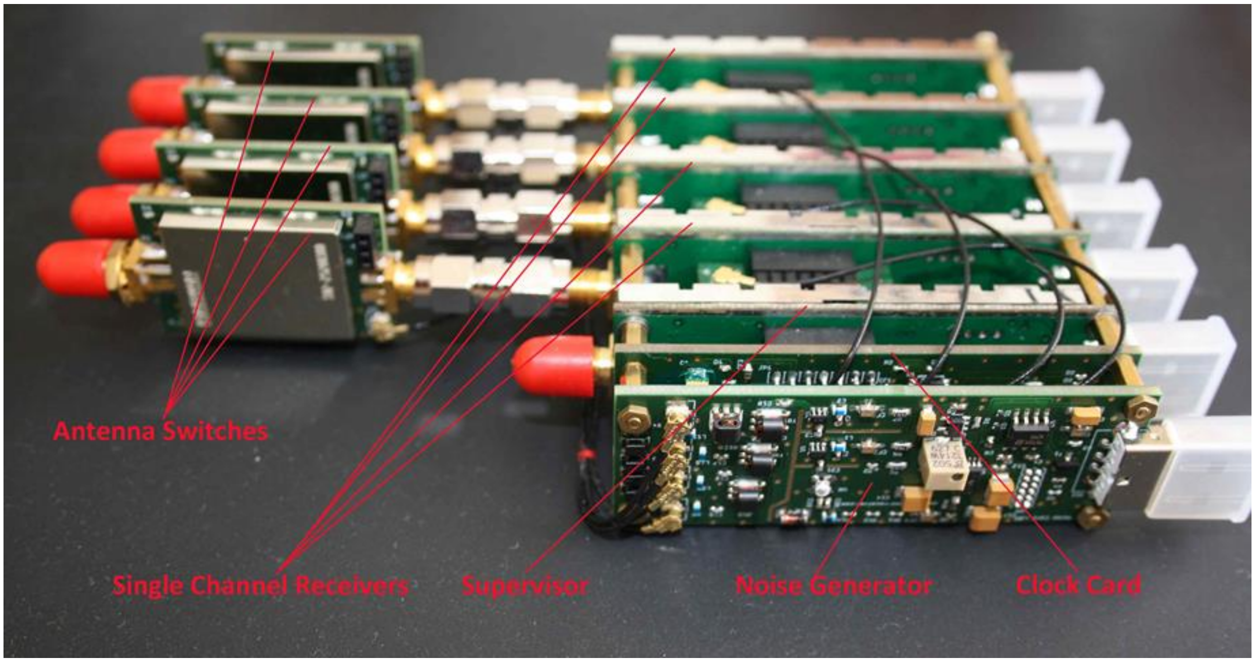

Figure 5. Coherent Receiver Components

In addition to the coherent receiver, low loss male to female SMA Connector wires were used to separate antennas connected to the RTL SDRs on the coherent receivers. The lengths of the connectors' insulated wires accommodated distances around half the wavelength of the incoming VHF RF signal (anywhere from 0.5 to 2.5 meters for VHF signals). In other words, the SMA connector wires have lengths that are long enough so that antennas can be separated at half the wavelength of the incoming RF signals.

Each RTL SDR has a USB connection intended for an individual port on a Raspberry Pi. The Raspberry Pi serves as an embedded device which collects data samples from each RTL SDR. To more efficiently prototype, we decided to use laptops as computing devices (which had usb connections to each RTL SDR). We intend to port our whole software from VirtualBox to the Raspberry Pi by creating a VirtualBox image file of the OS compatible with the Raspberry Pi, and storing this image file into an SD Card intended for the Raspberry Pi.

V.ii. Software Overview and Direction Finding Algorithms

Our software implementation consists of sampling received RF signals at the RTL SDRs in the coherent receiver and performing DSP methods on these samples to generate AoA measurements of these signals. GNU Radio is a free, open source software that provides signal processing blocks, and can display a streamlined interaction between these blocks in a GUI called GNU Radio Companion. We have used GNU Radio to determine Angle of Arrivals using MUSIC. MUSIC is a performs an eigenvalue decomposition on a covariance matrix of inputted data samples obtained from the received signal. From the orthogonality of signal and noise subspaces taken from the decomposition, peaks in an estimator function are found which determines a pseudospectrum vector.

Even though more computationally expensive, MUSIC was chosen over traditional digital beamforming for its higher accuracy. Moreover, the computation of MUSIC can be minimized through autocorrelation through rate-limiting (described in part V. v.).

In our implementation, the MUSIC algorithm is completely encapsulated into a GNU Radio Custom Block developed by Ettus Research. GNU Radio blocks are classified DSP functions that are used in a flowchart to accomplish an overall task from received RF samples. GNU Radio custom blocks are usually developed by an individual or a small group of researchers and are not included in the main GNU Radio directory, where widely used DSP functions can be found. Ettus custom blocks are successfully integrated after an in-depth installation process (V. iv.). This installation process installs an OutOfTreeModule, which is a component that extends GNU Radio with additional functionality provided by the module's custom blocks. [15]

Ettus Research is a National Instruments company that is the world's leading supplier of software defined radio and publishes software that can be used with its products online. We choose to use custom blocks developed by Ettus Research because they are very well documented and have been empirically tested (see [16]).

V.iii. Sampling from the RTL SDR

Since RTL-SDRs use a quadrature demodulator, radio signals are sampled into in-phase and quadrature values corresponding to the real and imaginary components of an incoming RF signal. Parallel I/Q extraction from multiple RTL SDR receivers can be retrieved in three ways: writing to a file from Ubuntu terminal, using a file sink (type of block) in GNU Radio, or samples can be monitored within GNU Radio using a QT GUI Sink (type of block). You can write demodulated I/Q samples from the coherent receiver to a file in Ubuntu Terminal using tutorial 1 in Appendix A. The Output of a QT GUI Sink within GNU Radio is displayed in Figure 6.

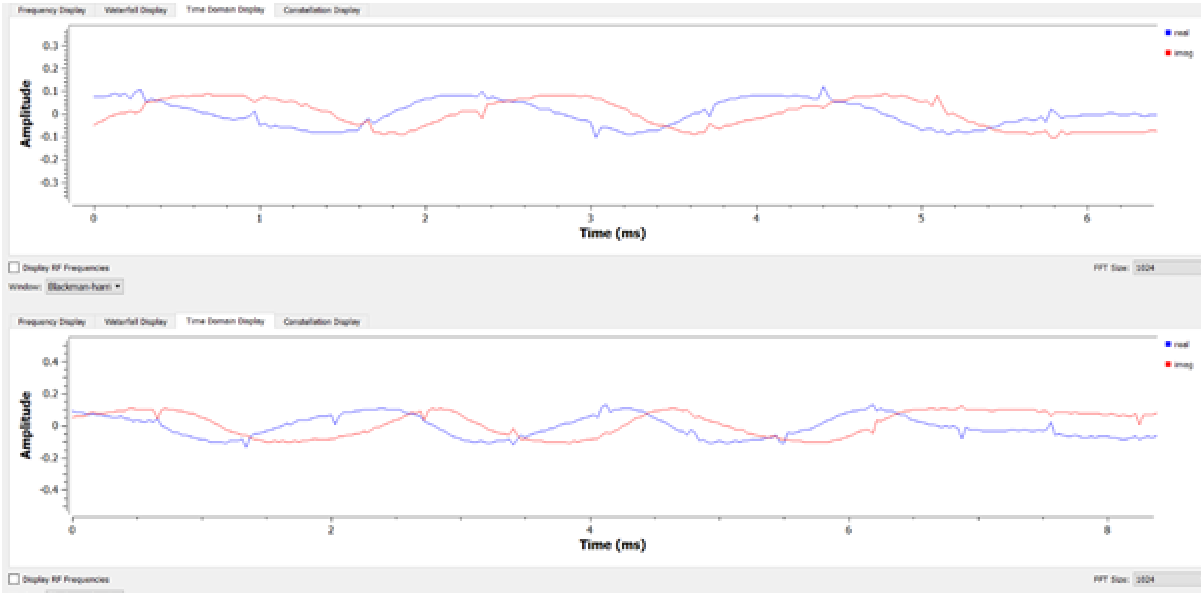

Figure 6: The parallel output of I/Q values from two RTL-SDRs in the time domain.

It is important to note that I/Q sampling process is done automatically by RTL-SDR Source blocks in GNU Radio. The RTL-SDR Source blocks allow a user to input a desired sampling rate, received frequency, and received bandwidth for each RTL-SDR in the coherent receiver. I/Q values will be transmitted in a stream of complex values from the RTL-SDR source block to the next block in sequence at the user specified sampling rate.

V. iv. Ettus OutOfTreeModules – Installation and Overview

During the four week implementation period, a large portion of the time was dedicated to installing and understanding the workflow of GNU Radio's OutOfTreeModules. We discovered that properly creating and installing out of tree modules in GNU Radio required an understanding of gr_modtool and cmake, which are much more efficient in Ubuntu file systems. Gr_modtool creates all the block files necessary from user-written Python or C++ code, and places the files in the right directories within an Ubuntu system. Cmake is an open source software used for managing the build process of software using a compiler-independent method [13]. For the installation purposes, cmake can be primarily used to compile and install an OutOfTreeModule using cmake … commands.

The Ettus Github library consists of all the cmake files required to build and compile Ettus custom blocks used for our software implementation for direction finding. The entire installation guide for getting RTL SDR Drivers, GNU Radio, and Ettus custom blocks working on Ubuntu is listed in Appendix A tutorial 2.

V. v. GNU Radio Complete Flowchart

We used a GNU Radio flowchart design using custom blocks from Ettus Research (https://github.com/EttusResearch/gr-doa) as shown in Figure 8.

Two RTL-SDR Sources are used with a specified sampling rate of 1 Msps, 433.9 MHz, and 98 kHz bandwidth. These complex values are transmitted to an autocorrelation block which uses a forward-backward averaging method to compute a sample correlation matrix (which is used in the MUSIC algorithm) from 2048k length vector of inputted samples. Another purpose of the autocorrelation block is that it is rate-limiting – since MUSIC is computationally expensive, would want to feed inputs into this algorithm at a lower sampling rate. The rate-limiting level is directly correlated to snapshot length (block parameter) [16]. The autocorrelation block outputs to a MUSIC Linear Array block which uses the MUSIC algorithm to determine the AoAs of the received signal parameters norm_spacing (distance between antenna elements divided by the wavelength of the carrier signal) and inputs (number of elements in the antenna array). The result of this flowchart is an AoA which can be more effectively tested/viewed from a GUI Compass.

In order to accommodate the above protocol for two RTL SDR receivers instead of four, the autocorrelation overlap size has been adjusted to 1024k instead of 512k (2048k total vector size/2 rtl-sdr receivers = 1024k overlap size) and the MUSIC Linear Array's Num_Ant_Elements variable to 2 instead of 4.

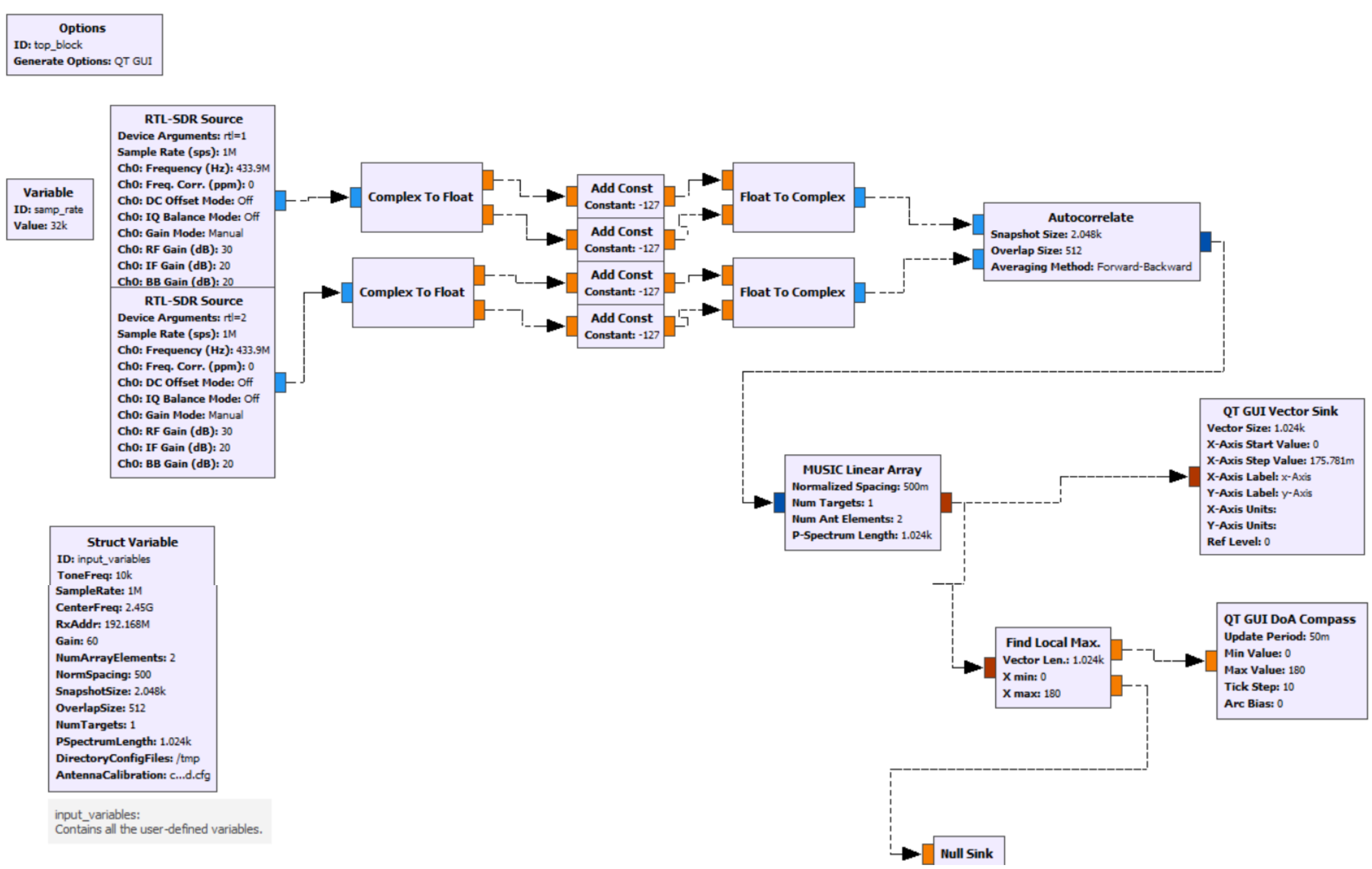

Figure 8. GNU Radio Digital Signal Processing Flowchart - Based on Ettus Research.

V. vi. Getting the GUI to Display Stable AoA

The GUI compass is used to more effectively view AoA values during prototyping and testing. This is because reading float values from the console would be tedious compared to viewing the direction of the incoming signal when testing AoA values at different locations in the proximity of the receiver basestation. The GUI compass is updated with a 50-millisecond period, so that AoA values can be viewed in real-time. A moving average block can be used to make the compass' inputted values more stable; therefore, reducing fluctuations in displayed AoA values. However, if too many values are averaged, the compass can become slower and less responsive to a moving transmitter. The GUI compass is displayed in Figure 10 (Empirical Testing section).
VI. Empirical Testing

An unmodulated RF signal was continuously transmitted at 434 mHz at a bandwidth of 98 kHz from CC1310s during testing. This 434 mHz frequency was used as the lowest default option for transmitter frequencies in SmartRF Studio.

Because the testing phase was short, not enough numerical results were obtained for testing AoA values at specific transmitter locations. However, our preliminary results indicate an error threshold relationship that is correlated to the distance between the CC1310 and the antenna array. This relationship is shown in Figure 9.

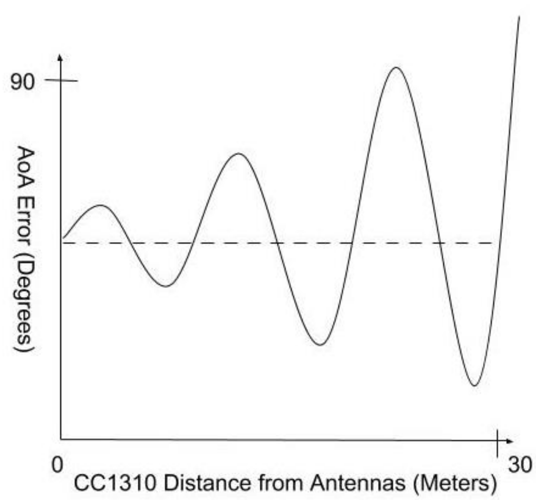

Figure 9: As the CC1310 approaches the antennas, variance from the true error (dotted line) increases. In other words, AoA measurements become more sporadic as the distance between the CC1310 and the antennas increases. As the distance between the CC1310 and the antennas decreases, AoA measurements become more concentrated, but still varied around a specific AoA

error value. The true error/specific AoA error (dotted line) changes based on the position of the CC1310.

Figure 10: Test results were displayed on a GUI compass

VII. Conclusion

In order to accomplish the localization of small animals, we plan to develop a cost effective and automated system to track animal movements within the range of five meters while taking into account expected causes of error. Our proposed system consists of a receiver architecture that is built specifically for phase interferometry direction finding to facilitate accurate measurements from radio tags on tracked individuals. In order to accomplish this, a low weight radio tag is being developed to transmit signals to radio base stations. These tags will transmit sub 1-GHz UHF frequencies.

Appendix A. Tutorials

Tutorial 1:
The following code is used for parallel I/Q data extraction within the Ubuntu Terminal:

```
rtl_sdr -d0 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture0-2.dat &
rtl_sdr -d1 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture1-2.dat &
rtl_sdr -d2 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture2-2.dat &
rtl_sdr -d2 -f 1125000000 -g 35 -s 2500000 -n 50000000 -N FMcapture2-2.dat &
```

Tutorial 2:
The entire installation guide for getting RTL SDR Drivers, GNU Radio, and Ettus custom blocks working on Ubuntu.

Commands with the command prefix “sudo apt-get …” installs a widely used library on the Ubuntu filesystem. These commands can be done on an Ubuntu OS installed to any machine (PC or Raspberry Pi). The machine used will determine the file system space (a memory space greater than 16 Megabytes is adequate for a correct installation of all the necessary libraries/modules in this tutorial, and a 16/32 Megabyte SD card is often used among Raspberry Pi users).

The purpose of this installation is to get the operating system to recognize RTL SDRs, acquire GNU Radio for DSP, and get Ettus custom blocks working in GNU Radio. The end goal of these steps will give the user the complete software framework for getting a functional AoA estimator running on a Raspberry Pi or PC (to the level we have achieved).

Important Note: Steps 2 and onwards use the terminology "generic cmake instructions" to refer to this set of instructions to install github libraries into Ubuntu:

$ git clone "*github library here*"
$ cd "*installed library name*"
$ mkdir build
$ cd build
$ cmake ..
$ make
$ make test
$ sudo make install
$ sudo ldconfig

It is critical to know that **Ubuntu versions 14.04 and 16.04** are the only compatible distributions with the Ettus Doa library at this time.

0. Make Sure to update the operating system before the following installations:
   pi@raspberrypi ~ $ sudo apt-get update
   pi@raspberrypi ~ $ sudo apt-get upgrade

1. Install GNU Radio by using the command `sudo apt-get install gnuradio-dev`

   Next, install the RTL SDR drivers, the following commands have been included in this report in case this page (https://gist.github.com/floehopper/99a0c8931f9d779b0998) gets lost in the future:

```
pi@raspberrypi ~ $ cat <<EOF >no-rtl.conf
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
EOF
pi@raspberrypi ~ $ sudo mv no-rtl.conf /etc/modprobe.d/
```

```
pi@raspberrypi ~ $ sudo apt-get install git-core
pi@raspberrypi ~ $ sudo apt-get install git
pi@raspberrypi ~ $ sudo apt-get install cmake
pi@raspberrypi ~ $ sudo apt-get install libusb-1.0-0-dev
pi@raspberrypi ~ $ sudo apt-get install build-essential

pi@raspberrypi ~ $ git clone git://git.osmocom.org/rtl-sdr.git
pi@raspberrypi ~ $ cd rtl-sdr/
pi@raspberrypi ~/rtl-sdr $ mkdir build
pi@raspberrypi ~/rtl-sdr $ cd build
pi@raspberrypi ~/rtl-sdr/build $ cmake ../ -DINSTALL_UDEV_RULES=ON
pi@raspberrypi ~/rtl-sdr/build $ make
pi@raspberrypi ~/rtl-sdr/build $ sudo make install
pi@raspberrypi ~/rtl-sdr/build $ sudo ldconfig
pi@raspberrypi ~/rtl-sdr/build $ cd ~
pi@raspberrypi ~ $ sudo cp ./rtl-sdr/rtl-sdr.rules /etc/udev/rules.d/
pi@raspberrypi ~ $ sudo reboot
```

After sending the above commands, you should be able to perform a simple test which generates a list of RTL SDRs recognized by the operating system. The test also checks the possible tuning range of each RTL SDR and the maximum sampling rate possible on the device (computer or Raspberry Pi) being used.

```
pi@raspberrypi ~ $ rtl_test
Found 1 device(s):
  0:  Generic, RTL2832U, SN: 77771111153705700
```

2. Install the bias_tee software from https://github.com/rtlsdrblog/rtl_biast (this will be used to turn the bias_tee on/off when performing RF Switching (not yet completed)) using the generic cmake instructions.

3. Install armadillo-code from https://github.com/conradsnicta/armadillo-code/ using the generic cmake instructions.

4. Install EttusResearch gr-doa custom blocks from https://github.com/EttusResearch/gr-doa using the generic cmake instructions. On some Ubuntu distributions, a list of missing dependencies is generated during the execution of the cmake instruction. If a 16.04 Ubuntu version was installed, and steps 0-2 were completed correctly, no additional dependencies need to be installed besides armadillo-code (step 3).

5. Relevant GRC files can be found from https://github.com/jakapoor/AMRUPT. The only file that needs to be installed from the Github at this time is the master flowchart for obtaining rudimentary AoA measurements called musicRTL.grc. Download this file and

open it with GNU Radio Companion. GNU Radio Companion should show a complete flowchart resembling Figure 8 from the Design Implementation section. If steps 0-4 were followed correctly, no messages exclaiming “missing block: ‘block name’ not found” should be displayed.

Appendix B. Works Cited

[1] Krüger, S W. “An Inexpensive Hyperbolic Positioning System for Tracking Wildlife Using off-the-Shelf Hardware.” May 2017.

[2] He, Chong, et al. “Direction Finding by Time-Modulated Array With Harmonic Characteristic Analysis.” IEEE Antennas and Wireless Propagation Letters, vol. 14, 2015.

[3] Řeřucha, Š., Bartonička, T., Jedlička, P., Čížek, M., Hlouša, O., Lučan, R. and Horáček, I. (2015). The BAARA (Biological Automated Radiotracking) System: A New Approach in Ecological Field Studies. PLOS ONE, 10(2), p.e0116785.

[4] D. Guerin, S. Jackson, and J. Kelly, “Passive Direction Finding: A Phase Interferometry Direction Finding System for an Airborne Platform,” Oct. 10, 2012. https://web.wpi.edu/Pubs/E-project/Available/E-project-101012-211424/unrestricted/DirectionFi ndingPaper.pdf.

[5] D. Zhang, J. Ma, Q. Chen and L. M. Ni, "An RF-Based System for Tracking Transceiver-Free Objects," Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07), White Plains, NY, 2007, pp. 135-144.

[6] Y. Ma, X. Hui, and E. Kan, “3D Real-time Indoor Localization via Broadband Nonlinear Backscatter in Passive Devices with Centimeter Precision,” Oct. 3, 2016. https://dl.acm.org/citation.cfm?id=2973754.

[7] He, Chong, et al. “Direction Finding by Time-Modulated Array With Harmonic Characteristic Analysis.” IEEE Antennas and Wireless Propagation Letters, vol. 14, 2015.

[8] Bartolucci, Marco, et al. “Synchronisation of Low-Cost Open Source SDRs for Navigation Applications.” 2016 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2016.

[9] Whiting, Sam, et al. “Time and Frequency Corrections in a Distributed Network Using GNURadio.” 2017.

[10]“Tejeez/rtl_coherent.” GitHub, 6 July 2016, github.com/tejeez/rtl_coherent

[11]Whiting, Sam, et al. “Time and Frequency Corrections in a Distributed Network Using GNURadio.” 2017

[12] “About RTL-SDR” https://www.rtl-sdr.com/about-rtl-sdr/

[13] https://stackoverflow.com/questions/17525153/how-exactly-does-cmake-work

[14] Optimization of MUSIC algorithm for angle of arrival estimation in wireless communications, NRIAG Journal of Astronomy and Geophysics,Volume 2, Issue 1, 2013, Pages 116-124,

[15] https://wiki.gnuradio.org/index.php/OutOfTreeModules

[16] “Phase Synchronization Capability of TwinRX Daughterboards and DoA Estimation” https://github.com/EttusResearch/gr-doa/blob/master/docs/whitepaper/doa_whitepaper.pdf