AMRUPT PROJECT PROPOSAL:

Russell Silva - rms438
Mei Yang - mny8
Peidong Qi -pq32
Justin Cray - jgc232
Electrical and Computer Engineering, Cornell University
2/21/2018

Submitted to—
Dr. Julian Kapoor, Prof. Joe Skovira
Behavior/Electrical and Computer Engineering, Cornell University

I. Executive Summary

AMRUPT is a technology being designed for the localization of small animals in the field of ecology. This is used for the study of flight patterns, social interactions, or other biological attributes to most species. 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. Because many researchers are interested in small-scale movements of animals within populations, such a system may be extremely useful. 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.

RF signals phase information is calculated on the RTL SDR. This requires multiple receive antennas connected to an RF switch which is attached to the RTL SDR which communicates the I & Q values of the received to a Raspberry Pi for angle of arrival calculation. A three antenna system is used for angle of arrival. An angle of arrival will be computed between Antenna 1 and Antenna 2 that is ambiguous between Quadrants 1&2 and Quadrants 3&4. An angle of arrival will be computed between Antenna 1 and Antenna 3 that is ambiguous between Quadrants 2&3 and Quadrants 1&4. The angle of arrival will be determined to be in the quadrant that contains both AOAs.

II. Statement of Problem

II. i. The Localization of Small Animals in the Field of Ecology

The localization of small animals in the field of ecology is imperative to determining the flight patterns, social interactions, or other biological attributes to most species. Many attempts 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.

II ii. Literature Review

Transmitted signals at antenna array elements can be quantized at receivers to provide signal strength, phase difference, or time arrival information to be used in a Watson-Watt, Phase Interferometry, or Time Difference of Arrival (TDOA) systems respectively ([1] , [2], [16], and [22]). Phase based measurements can be skewed by multipath effects in the environment by constructive and destructive interference for line of sight signals [2]. 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. This flaw in time difference of arrival can be mitigated by subsample interpolation at signal correlation peaks [22].

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 [21], shown in Figure 1 [21]. In coherent receivers running on a single clock signal, synchronization errors from clock drift and other delays can be corrected by incorporating a signal generator input to each channel [20]. In addition, clock drifts and bulk delays can be mitigated by implementing cross correlation with virtual sources on a software level [23].

teristic Analysis

aojin Li, Junping Geng, *Member, IEEE*, and
or Member, IEEE

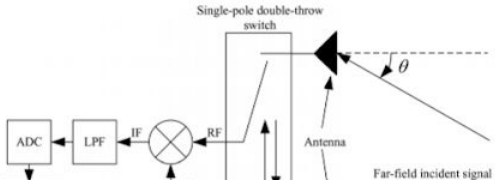

Figure 1: Time Modulated Array Setup

A phase interferometry system with real time operation on multiple receivers without an apparent phase offset correction/controlled noise source system was able to identify an angle of arrival within ± 2.5 degrees for all emitter distances from 1 km to 100 km [1]. The error threshold from this system is less than the ± 5 degree angle of arrival threshold error determined in the time-modulated approach [21].

The phase interferometry system in [1] 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.

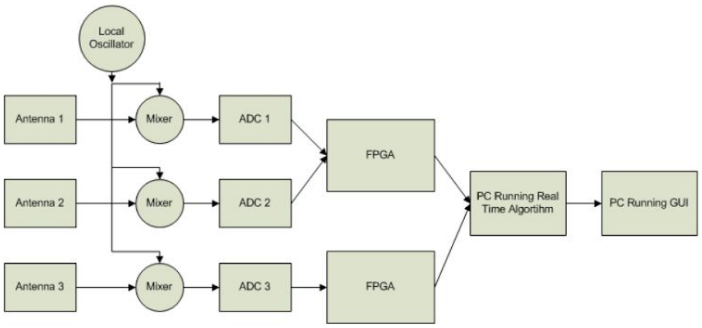

Figure 2: Block diagram for Phase Interferometry in Guerin, Jackson, and Kelly [1]

Ma, Hui and Kan [2] 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 backscatter which exploits nonlinear elements in passive devices to generate second or higher-order harmonics for an uplink response. 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 (Figure 3). In order to combat multipath interference, a phase error threshold is used within a hueristic multi-frequency continuous wave (HMFCW) ranging algorithm to find an optimal frequency combination that generates an undistorted line of sight path. HMFCW ranging correctly pinns down a phase cycle integer by transmitting a sequence of frequencies determined by a genetic algorithm to maximize an error tolerance equal to the percent bandwidth of the signal. 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 this system to ours is discussed in the design objectives section of this proposal.

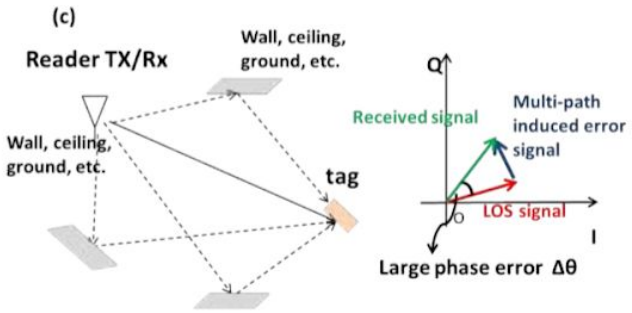

Figure 3: Dense indoor multi-path induced phase error [2].

In order to improve the cost effectiveness of direction finding, [22] and [23] have 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

[22] and by having an extensive hobbyist base with multiple Github repositories such as this one [24], 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 [23].

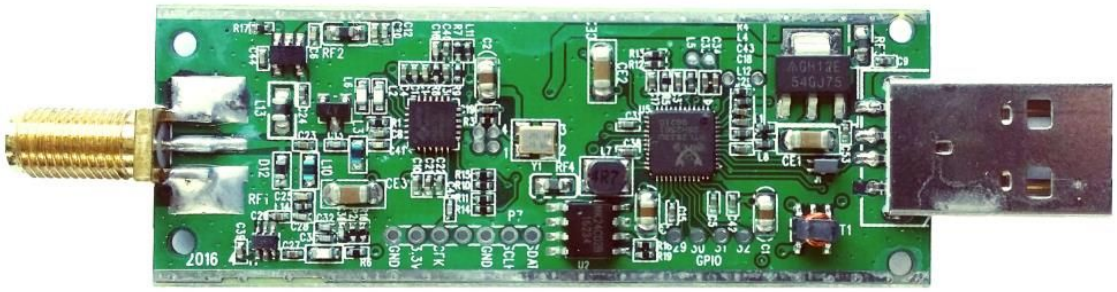

Figure 4: RTL-SDR (RTL2832U). Image Credit: rtl-sdr.com

III. Design Objectives

In order to accomplish the goals listed in the problem statement, 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 to node communication protocol will be governed by how accurate our receivers are when taking angle of arrival measurements. If angle of arrivals from a couple of 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. 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 wrong 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 and is already much more accurate than existing systems mentioned in the literature section. 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 that trilaterates positions of mobile nodes if a 5-meter accuracy level has not been achieved by previous efforts (Phase 3).

The diagram below is used to better understand error minimization with relation to AOA calculations.

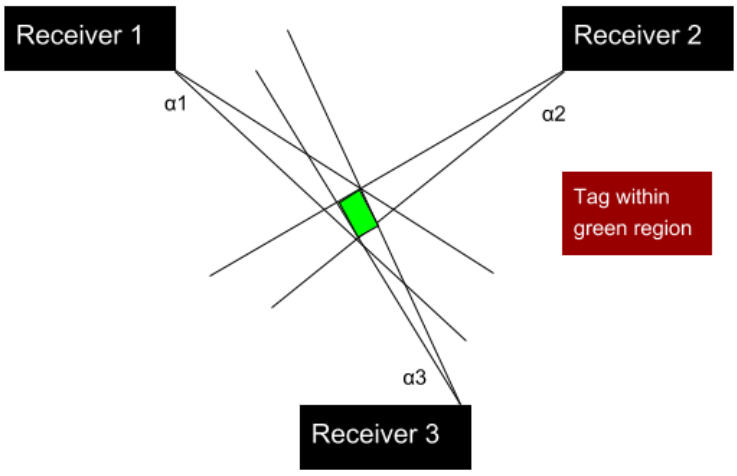

Figure 5: Error in triangulated area

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 5) 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, several design decisions are considered. One such design parameter is distinguishing tags. The CDMA scheme is a very popular method for doing this due to the coding infrastructure. However it is unclear if CDMA is compatible with making phase-based AOA measurements. Gold codes are mutually orthogonal to each other, so it is possible to distinguish multiple bit streams from multiple tags at the same time. However, gold codes may not be able to differentiate phases.

Last, but not least we have devised a system that is composed of cost-effective, off-the-shelf components. This is done to make this setup more reproducible in future works and more accessible to ecological hobbyists/researchers.

IV. Technical Approach

The entirety of the proposed direction finding system consists of radio transmitters and receivers. This section will focus primarily on receiver design as the lightweight radio tags are being developed by another party. In order to achieve our design objectives, the receiver architecture will require the most development.

IV. i. Receiver Architecture

We first propose a receiver architecture that consists of an RTL SDR to simplify wireless communication and improve the cost effectiveness of this project. The receiver architecture is outlined in Figure 6. Note that this now includes 4 antennas to complete Watson-Watt Angle of Arrival (AoA) measurements. Also note the decided clock board with highly stable (0.1 ppm) TCXO. This board must also have the power to drive multiple SDRs.

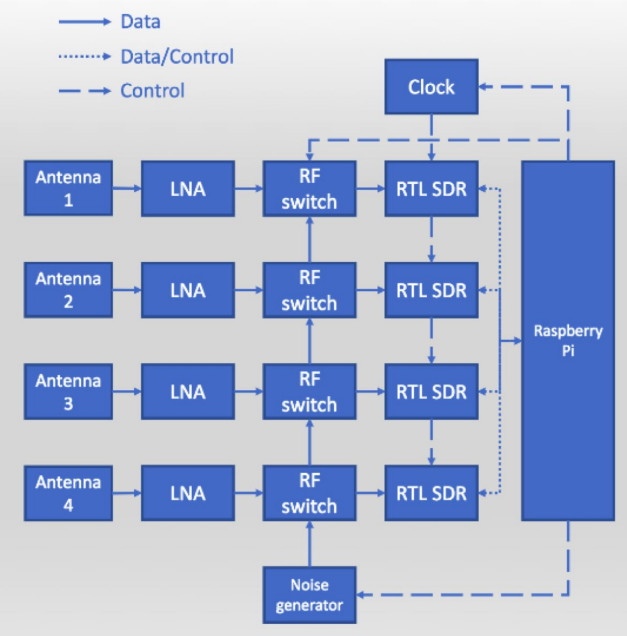

Figure 6: Receiver Architecture [27]

The ideal embedded device would include the following:

1. A sub 1-GHz device for VHF or UHF frequencies transmitted from radio tags. We choose a lower frequency band (relative to most RF applications) to mitigate multipath interference and better determine the phase difference of signals. Previous systems have used ~150 MHz as the operating frequency of transmitters because of the impact of large trees on multipath interference.
2. A very high sample frequency during the analog to digital conversion of RF signals. This is essential for mitigating adverse effect from noise when determining accurate phase differences from radio waves moving at the speed of light. However, a sampling frequency above twice the radio frequency (constant in a non-frequency modulated signals) is not needed. Sampling rates are further discussed in section IV. Vii. "RF Wave Reconstruction and Matlab Simulation".
3. Ample UART/I2C/SPI/GPIO connections for data logging and transfer
4. Contains every component necessary for receiving an RF signal from an external antenna – ADC, local oscillator, etc.
5. Extremely high RF sensitivity and blocking performance
6. Programmable and highly used by the public – helpful for finding more tutorials and readily available information on the device
7. Low power and low cost

From this list of specifications, the RTL SDR was chosen. The RTL SDR specifications are outlined in [28]. To address the above, the following specifications are highlighted. The maximum sample rate that does not drop samples is 2.4 MS/s. The RTL SDR uses USB to

interface directly with the Raspberry Pi. The frequency range depends on the dongles used. While we have not decided which dongle we will use yet, some options are outlined in Table 1 [29].

Tuner	**Frequency range**
Elonics E4000	52 – 2200 MHz with a gap from 1100 MHz to 1250 MHz (varies)
Rafael Micro R820T	24 – 1766 MHz
Rafael Micro R828D	24 – 1766 MHz
Fitipower FC0013	22 – 1100 MHz (FC0013B/C, FC0013G has a separate L-band input, which is unconnected on most sticks)
Fitipower FC0012	22 – 948.6 MHz
FCI FC2580	146 – 308 MHz and 438 – 924 MHz (gap in between)

Table 1: RTL SDR Frequency Ranges

IV. ii. RF Switch and Amplifier

The RF switch and amplifier are included in the "Antenna Switch Card" sold by coherent-receiver which includes a switch and an LNA. Controlled by I2C, this card has an LNA with a gain from 15-23 dB and a noise figure from .7-.8 dB. The MOSFET switch has unspecified insertion loss but since it is included in the hardware setup it should be compatible with the rest of the design.

IV. iii. Antenna

We seek an antenna that is in our target frequency range (the UHF band) and also has an SMA connection for ease of interfacing with other components in our system. To accomplish this, an antenna such as the ANT500 is compact high gain antenna and operates in the expected frequency range.
Note that the ANT500 is a monopole antenna which is used largely due to its simplicity in the architecture. While other antenna architectures such as a loop antenna and a dipole antenna are considered, a monopole antenna is chosen due to the fact that it's an isotropic radiator. An omnidirectional monopole antenna is also used in Kruger and Weiding.

IV. iv. RTL SDR to Raspberry Pi Connection and Datalogging

Figure 7: RTL SDR connect to Raspberry Pi [25]

The Raspberry pi has USB support to connected to RTL2832U. The Raspberry Pi also has software support for the RTL2832U. Once we have connected all four RTL2832Us from the coherent receiver platform to Raspberry Pi, we will need to download and install the RTL2832U driver. After we install the driver, we will be able to use the RTL SDR on the Raspberry Pi. We will follow the instructions in this video tutorial. [26]

Once we get AOA data from the RTL SDRs, we will run a program on the Raspberry-pi to log the data. The program will build a file for the data we have received. Then the program creates a tag label and a time label each measurement on the file. The tag label is going to indicate which tag sent the signal. There will be a real time clock onboard to monitor the time and help to create the time label for each measurement. We are planning to implement a short term memory cache for AOA measurements. Once at least 10 consecutive locations have been triangulated by the AOAs from one tag, the program will clear AOA data from that tag id. The position data will be stored in a primary text file for later data analysis.

IV. vi. Phase Disambiguation and Angle of Arrival Calculation

The phase difference between two antennas models the extra distance that a radio wave has to travel to reach one antenna over the other. The geometry that demonstrates this phenomenon is displayed in Figure 8, where an angle of arrival is dependent on a signal's wavelength, the separation between two antennas, and the phase difference. Using these fundamental notions of phase interferometry, the angle of arrival will be calculated at this stage using the phase differences obtained from the CC1310.

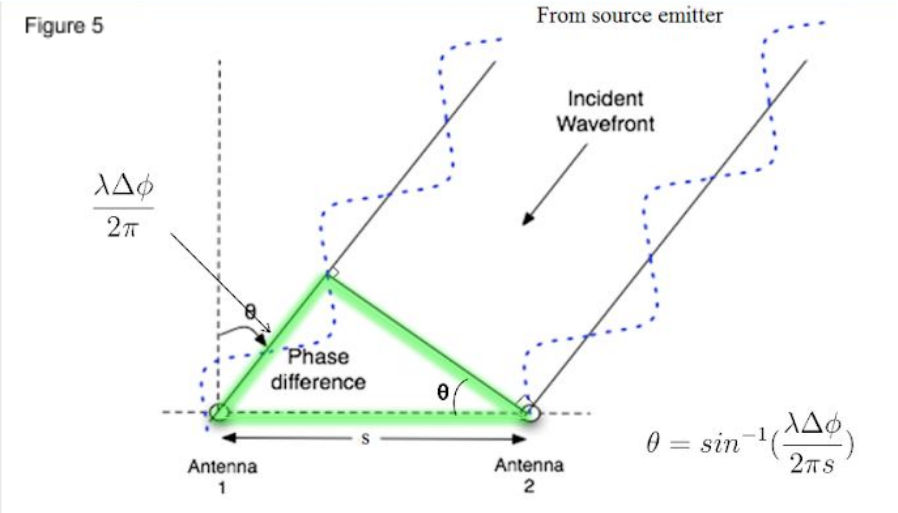

Figure 8: Diagram showing how the angle of arrival equation is formed using the phase difference of a signal received at two antennas, the distance between the antennas, and the wavelength of the signal. (Guerin, Jackson, Kelly 2012)

There are two types of phase disambiguation that will be covered in this section. One type of phase disambiguation results from mobile nodes that are flipped above and below the receiver's 0 degree horizontal axis reference. This will be defined as quadrant ambiguity. The second type of phase disambiguation results from antennas separated greater than $\lambda/2$ in which a phase difference is outside the $-\pi < \theta < \pi$ range. This will be defined as distance ambiguity. Distance ambiguity is not to be confused with the phase integer ambiguation that occurs when calculating differential distances from mobile to ground node as discussed in (Ma, Hui, Kan 2016).

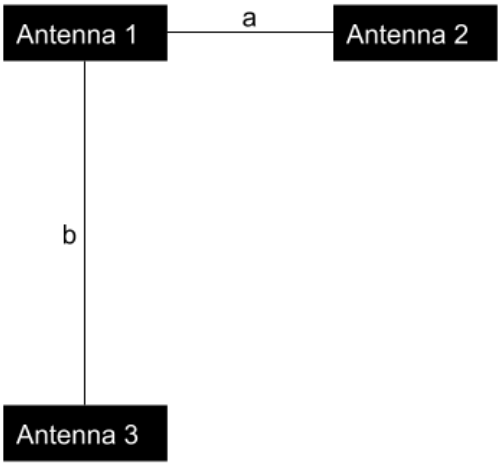

Figure 9: Antenna Setup with distance a between Antenna 1 and Antenna 2, distance b between Antenna 1 and Antenna 3.

We propose to solve quadrant ambiguity by adding a third antenna that is placed at a right angle from the line formed by antenna 1 and antenna 2 (Figure 9). An angle of arrival will be computed between Antenna 1 and Antenna 2 that is ambiguous between Quadrants 1&2 and Quadrants 3&4. An angle of arrival will be computed between Antenna 1 and Antenna 3 that is ambiguous between Quadrants 2&3 and Quadrants 1&4. The angle of arrival will be determined to be in the quadrant that contains both AoAs.

Since we are using sub-1GHz frequencies, placing the antennas at distances less than $\lambda/2$ is a feasible solution. For example, a 300 MHz signal will have a wavelength of 1 meter. So placing antennas around 0.4m < d < 0.5m from each other would not have a detrimental effect on signal gain, especially if PCB antennas are used. Frequency hopping at frequencies lower than 300 MHz will also be supported. This is important for forward compatibility with the multi frequency phase integer disambiguation approach used in (Ma, Hui, Kan 2016). .

If the above solution is found to be infeasible, and one of the antennas is placed greater than $\lambda/2$, distance ambiguity can be solved by making distance b > distance a (Figure 6) to a suitable offset determined by the signal wavelength. Since the CC1310 uses higher frequencies (315 - 930 mHz), a broadband frequency hopping system that includes all of these frequencies would have to support wavelengths of 0.32 meters ($c = \lambda f$). Therefore, our system would have to place antennas lower than 0.16 meters from each other, and near field radiation patterns may cause additional interference at these distances, especially if an equidistant beacon is used (discussed later in the section).

The following approach is proposed to solve distance ambiguity:

The number of phase ambiguities n is dependent on the following formula where s is the distance between two antennas and lambda is the wavelength of the signal. Theta resembles the maximum angle arrival of the system which in our case is pi.

$$n = \frac{s}{\lambda} sin\theta_{max},$$

(Guerin, Jackson, Kelly 2012)

We can formulate this equation for both pairs of antennas by replacing n by the normalizing phase difference from 0 to 1 (division by 2π) added to the number of possible phase difference in each antenna pair.

$$\Delta\phi'_{12} + i_{12} = \frac{s}{\lambda} sin\theta_{AoA}$$
$$\Delta\phi'_{13} + i_{13} = \frac{s}{\lambda} sin\theta_{AoA}.$$

(Guerin, Jackson, Kelly 2012)

Finally we combine both equations to form a set of lines which correlates a disambiguated phase difference for each pair of antennas to a maximum likelihood by determining the number of full phases for a phase offset.

$$\Delta\phi'_{13} = \frac{s_{13}}{s_{12}}\Delta\phi'_{12} + \frac{s_{13}}{s_{12}}i_{12} - i_{13}.$$

(Guerin, Jackson, Kelly 2012)

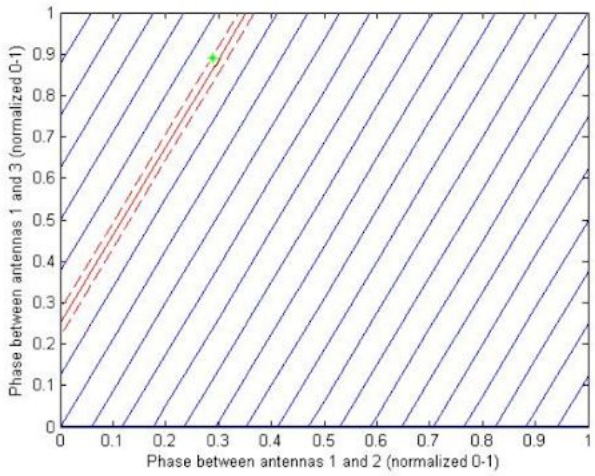

Figure 10: Phase Disambiguation Chart (Guerin, Jackson, Kelly 2012)

We do not believe that the 90 degree offset from quadrant disambiguation will affect the geometry of this solution.

IV. vii. Coherent Detection - Software

A protocol has been developed in [23] to synchronize multiple channels in a coherent radio receiver platform using SDRs. The flowchart below defines a sequence of DSP functions that need to be implemented to correct time and frequency offsets from RTL SDR measurements. This protocol can be adapted to a platform with four different SDR receivers, but has been truncated to 2 input SDR blocks for the means of simplicity.

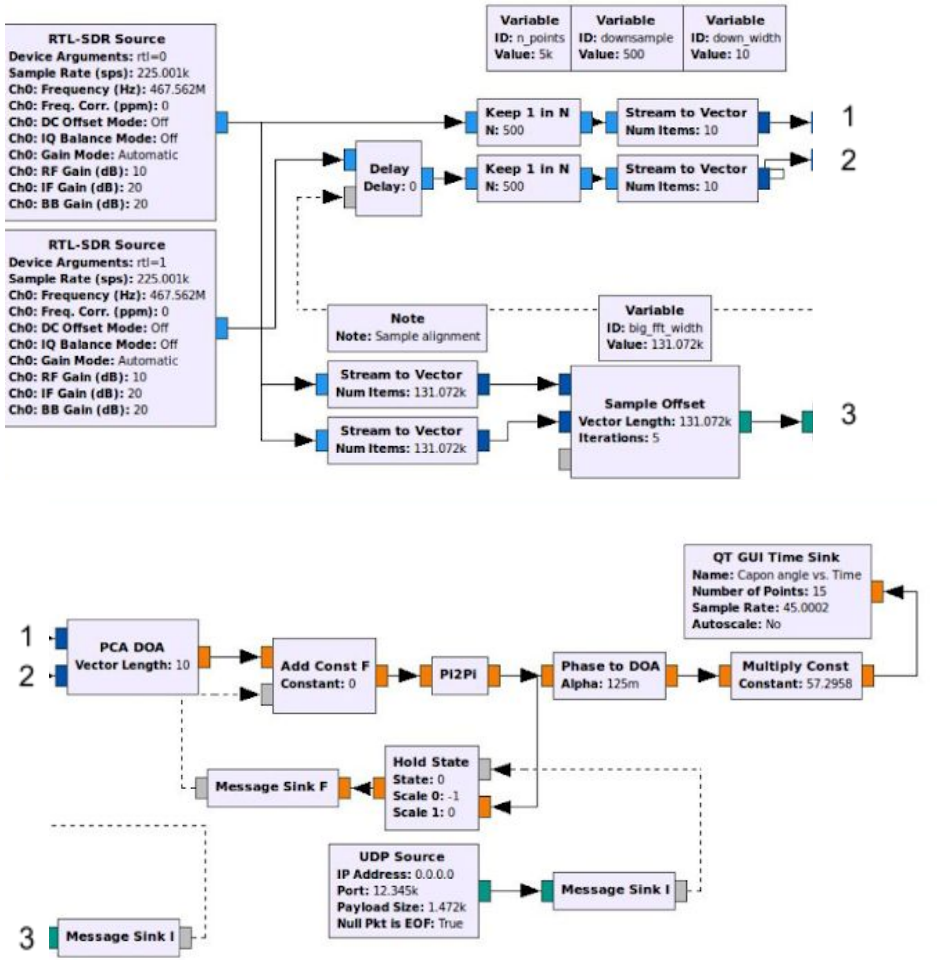

Figure 11: GNU Radio Digital Signal Processing Flowchart

The two RTL-SDR sources output signals harvested from antennas connected to the RTL-SDR cards. Even though SDRs in our system will share a common clock, signal sampling of these receivers is not expected to start at the same time, causing large bulk delays [23]. To compensate for bulk delay, each of our SDRs will be coupled with an RF Switch connected to a common noise source. The noise source will be used in cross correlating samples between signal channels in the Sample Offset Block of the above flowchart.

The Sample Offset Block computes the delay of one signal over another by taking a Fast Fourier transform of both signals, convoluting both signals in the frequency domain, and then taking the inverse Fast Fourier transform of the convolution to find the maximum argument (time value) of the output signal's amplitude in the time domain. The reason for the performing the final step is that the peak of the convolution will correspond to the delay between the two signals.

The Sample Offset block will determine the median phase difference of the signals after 5 iterations of sampling, and output this value into a message connected to the Delay Block in the top left of the figure. When the receiver channels are connected to the noise source, the phase offset in the Delay Block will be continuously updated every 5 iterations of sampling. When the receiver channels are connected to the SDR antennas, the delay block will not update the phase difference value, and will add this phase difference value to the second SDR signal; therefore, synchronizing both receiver channels.

After synchronizing both receiver channels from the cross correlation described above, the two input signals will be downsampled for more cost efficient AOA calculations. The subsampling of the synchronized signals did not reduce the integrity of the system described in [23]. The PCA DOA block computes the phase difference from the downsampled signals.

The phase difference calculation in the PCA DOA block is described below:

The incoming signals received by two different antennas is x1(t) and x2(t) respectively. Those are the initial signals received by the antennas at time t. Then we set x1(t) to be our reference and x2(t) has the time delay

$$x1(t)=a(t)\cos(wt)$$

$$x2(t)=a(t-T)\cos(w(t-T))$$

In the two equations above [23], a(t) represents the baseband signal. T is the time delay. We will use the narrow-band assumption to assume that the phase difference is a phase shift $e^{-j\phi}$ from a(t) as shown below [23].

$$\tilde{x}_1(t) = a(t)$$
$$\tilde{x}_2(t) = a(t)e^{-j\phi}$$

We will put those two signals into a vector to calculate a phase difference of maximum reliability to be outputted from the PCA DOA block.

This output phase difference will be subtracted by the calibration offset and then sent to a pi2pi block. The pi2pi block will wrap the modified phase difference into a range from -pi to pi to prevent phase ambiguity in the next calculation. Finally, a phase to DOA block contains the AOA calculation as shown in the code: `//DOA = arccos(phase/(2*pi*alpha)) - pi/2` from the project's github (Sam Whiting) in [30]. In this system, antennas were placed at a half wavelength distance from each other. Alpha is ½ to represent this characteristic in the AOA calculation. This formula corresponds to the geometric relation described in section IV. vi. of this proposal.

IV. viii. Coherent Detection - Hardware

A) Receiver

The coherent-detector SDR is based on the improved RTL-SDR v.3 dongle with slight modifications. The TCXO 28.8 MHz was eliminated from the dongle and additional headers were added.

B) Clock Card

The clock card provides a clock signal for all of the SDRs and would also be sent to the Raspberry Pi. It generates a 28.8 GHz signal. This gives us better stability in the system due to the shared clock and will help with clock drift.

The clock card includes the following components:

- **TXCO 28.8 MHz**
    - *Default***:** 2 PPM initial offset, 0.5-1 PPM temperature drift.
    - *Optional***:** higher precision (0.1/0.5/1.0 ppm) is available on request.
- **Buffer Gate**
- **Clock Buffer**
- **LPF** (cut-off 35 MHz)
- **SMD LED indicators**
- **I2C interface**
- **I2C 8bit Register**
- **Power indication LED** (+3.3 V)
- **Ultraminiature Coaxial Connectors**

Note the I2C interface which the clock signal is sent over. This is what can be used for the Pi or other peripherals. Also note the Ultraminiature Coaxial Connectors. These are installed on the card in order to distribute the clock frequency to all receivers using coaxial pigtails.

C) Noise Generator

The Noise Generator Card is used to calibrate the initial latency in the channels of coherent receivers. This time delay occurs due to the signal propagation delays between antenna, input circuits, tuner, etc. as well as different bootstrapping time.

IV. ix. RF Wave Reconstruction and Matlab Simulation

Matlab will be used as a tool for this project to reconstruct RF waveforms and simulate the proposed system by producing an estimate for the Angle of Arrival (AOA). Matlab software can be installed and utilized on the Raspberry Pi 3. A Matlab simulation from multiple signal generators (acting as separate antennas) will be used to verify the feasibility and correct functionality of the proposed solution. The proposed reconstruction is shown in Figure 12, in which the Raspberry Pi 3 reconstructs a sinusoid from one antenna at a time, while using the predicted continuation of signals at other antennas for phase difference comparisons.

In addition, Matlab simulations will follow the modular design principle that helps us build and organize the project as a whole. For RF waveform reconstruction, a signal generator will be created to simulate the RF waveforms at a certain sampling rate. The maximum sample rate that does not drop samples is 2.4MS/s, and the sampling rate will be used to reconstruct RF waveforms that are close to continuous as possible and are of the form $A\sin(\omega t + \varphi)$. The phase differences for each of the reconstructed waveform will be calculated along with phase disambiguation. The signals will then be used for future AOA calculation.

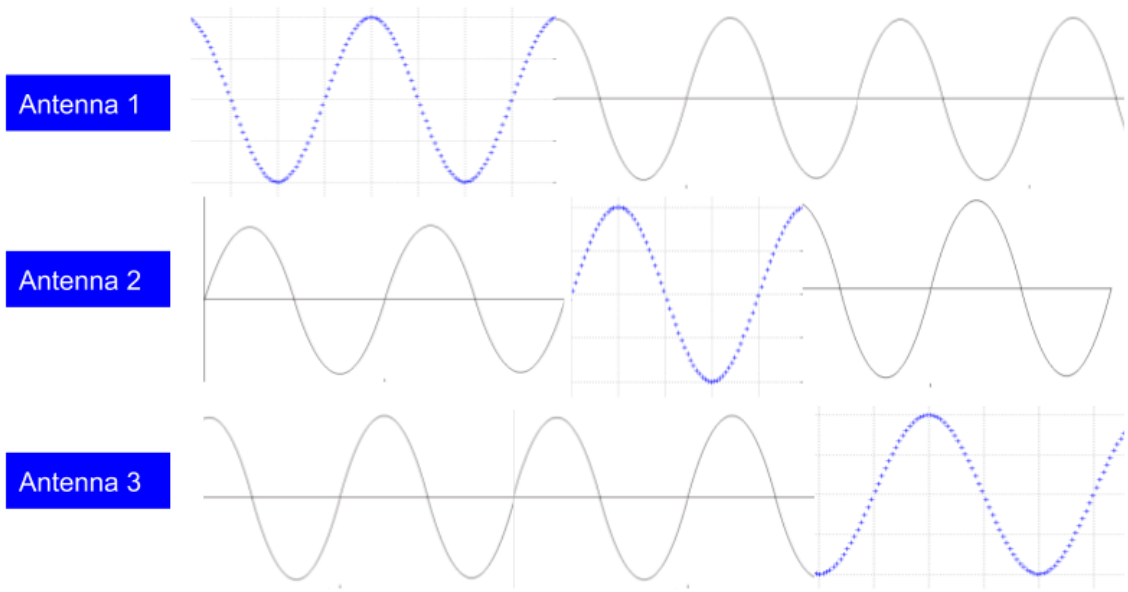

Figure 12: Blue regions represent areas that are being currently reconstructed by I/Q samples. Grey regions represent areas that continue the reconstructed signal and can be used for comparing phases to other antennas, but are not composed of collected data.

For ideal sinusoidal signals, it is important to note that we do not have to sample antenna outputs at twice the frequency of the transmitted signals if the frequencies of the signals are constant. This is especially important due to the fact that the maximum safe sampling rate of the RTL SDRs is at 2.4MS/s (resulting in the undersampling of VHF and UHF signals).

Assuming our sampling rate is constant, sampling a sinusoidal wave lower than the Nyquist frequency will produce an equivalent reconstructed signal. This reconstructed signal will have the same wave pattern and structure of the original signal. The only difference is that this reconstructed wave will be at a lower frequency. The above holds true for infinitely slow sampling rates (stationary sources) .

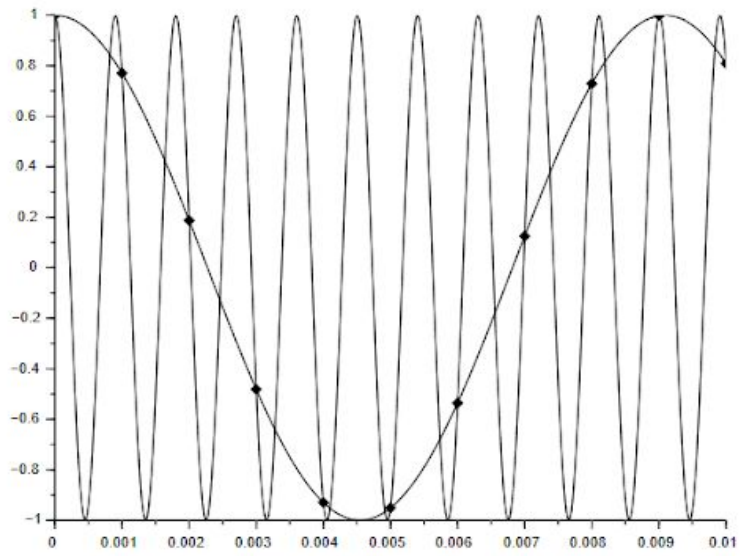

Figure 13: Sampling a sine wave lower than the Nyquist Frequency (Wescott, 2016)

If we take the phase difference between two sinusoidal waves that have been reconstructed from a lower sampling frequency than the Nyquist at any given time, their phase differences at that given time will be the same as if we were using the original sinusoidal function.

For this approach, we cannot use frequency modulated signals on radio tags. We will have to make sure that the transmitted signals represent sinusoidal waves of constant frequency in order for undersampling to work. This will not impede on a multi-frequency-phase-integer-disambiguation system as sequential frequency hopping is scheduled by the receiver (Ma, Hui, Kan 2016). For our active tag system, the receiver system will have to send a frequency hopping scheduling to the mobile nodes before a transmission period, so that it will know when frequency hops will occur (to prevent irregular measurements).

IV. x. Raspberry Pi 3 Network with Central Hub

This section covers work that will likely be beyond the scope of this semester's work, but includes a very rough plan on how the positions of radio tags will be triangulated.

We will connect the Raspberry Pi 3 base stations wirelessly at a frequency outside the bandwidth of the rest of our system to prevent radio frequency interference. The Raspberry Pi 3 will send timestamped and identified angle of arrival measurements to a central hub/laptop. The accuracy requirements and wireless transmission protocol of the timestamping will be determined later on later on in the semester. The central hub will then triangulate the position of a radio tag (based on the AOA identifications) in two dimensional space.

Triangulation Algorithm Pseudocode Based on Three Receivers (Python):

```
class Coordinate:
     def _init_(self, x = 0.0, y = 0.0):
          self.x = x
```

```
      self.y = y

#Finds intersection of two lines (y = mx + b)
def intersect (m1, b1, m2, b2):
      intersection = Coordinate()
      xval = float(b2 - b1) / float(m1 - m2)
      yval = xval * m1 + b1
      intersection.x = xval
      intersection.y = yval
      return intersection

#Finds the centroid of the triangulated area
def centroid(vertex1, vertex2, vertex3):
      center = Coordinate()
      center.x = (vertex1.x + vertex2.x + vertex3.x) / 3.0
      center.y = (vertex1.y + vertex2.y + vertex3.y) / 3.0
      return center

#Parameters are decimal angle of arrivals calculated at #each
base station and coordinate positions of each #base station.
def triangulated area (angle_of_arrival1, angle_of_arrival2,
angle_of_arrival3, receiver1, receiver 2, receiver3):

      #Converts angle degrees to slope
      slope1 = math.tan(angle_of_arrival1)
      slope2 = math.tan(angle_of_arrival2)
      slope3 = math.tan(angle_of_arrival3)

      b1 = receiver1.y - receiver1.x * slope1 #y = mx + b
      b2 = receiver2.y - receiver2.x * slope2
      b3 = receiver3.y - receiver3.x * slope3

      intersection1 = intersect(slope1, b1, slope2, b2)
      intersection2 = intersect(slope2, b2, slope3, b3)
      intersection3 = intersect(slope1, b1, slope3, b3)

      position = centroid(intersection1, intersection2,
      intersection3)
      return position
```

VI. Management Plan

Tasks (Below)	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Hardware Configuration and Checking						
1. Clock Card I2C Integration	Justin	Justin				
2. RTL I/Q Extraction	Russell					
3. SDR Antenna Integration	Justin					
4. Noise Source Input and monitoring RF Switching between noise and SDR sources	Justin	Justin	Justin			
Software Setup with GNU Radio						
1. GNU Radio and RTL SDR Driver software installation and setup on the Raspberry Pi	Peidong					
2. Learning how to implement python code and use built in radio blocks/functions in GNU Radio	Peidong	Russell /Peidong				
3. Cross Correlation and Synchronization of input samples	Russell /Peidong	Russell /Peidong	Russell /Peidong	If necessary: Russell/ Peidong/ Justin	If necessary: Russell/ Peidong/ Justin	
Phase Disambiguation and Angle of Arrival Calculation						
1. If necessary, create disambiguation algorithm (if antenna separation is greater than half of the signal's wavelength)			Russell	Russell		
2. Determine Angle of Arrival through measured phase differences			Russell /Peidong	Russell /Peidong		
3. Basic Triangulation Algorithm (If reliable AOA measurements are calculated)						Russell/Mei
Testing and Deliverables						
1. Angle of Arrival Measurement Testing Protocol (as entailed in the Deliverables Section)			Mei	Mei	Mei	
2. Detailed System Schematic (Documentation for future work)						Mei
3. Matlab Simulation (as described in its respective section)	Mei	Mei				
Datalogging (as described in Datalogging section)					Peidong	Peidong

The above plan for AMRUPT allocates a set of tasks/responsibilities corresponding to the six weeks left of the semester to work on this project, which are as follows:
Week 1: 4/8/2018 – 4/14/2018
Week 2: 4/15/2018 – 4/21/2018
Week 3: 4/22/2018 – 4/28/2018
Week 4: 4/29/2018 – 5/5/2018
Week 5: 5/6/2018 – 5/12/2018
Week 6: 5/13/2018 – 5/19/2018

VI. Deliverables

At the end of the semester, we plan to have a working base station constructed with four SDR cards connected to a Raspberry Pi. We plan to achieve ± 2.8 degree angle of arrival accuracy to obtain an under 5 meter triangulation accuracy ($s = r\theta$, $5 = (100)(\sim 2.8 * \frac{\pi}{180})$) for a radio tag within our minimum range objective of 100 meters.

Each SDR card has one antenna connected. And the RF switch control the noise generator. All the RF switch will turn on at the same time to let the noise generator work. We will have GRC running the detection flowgraph in Figure 10 to process the signals received by the antennas. The GRC will output the AOA for a single base station. Once the signal arrives at the antenna, it will go through the LNA(low noise amplifier) first, and then get processed by the noise generator. After that, the RTL SDR will process those signals and send to Raspberry pi for data logging and data analysis. The Raspberry Pi will have two control signals: one for the noise generator, and another one for the clock which controls the SDR.

We also will test the prototype performance. Our test strategy is test the prototype from ideal and small environment to complicate and large environment. In order to start testing the prototype, We will set the tag 20 meters away from the base station. The first test performed will be set in an small open area with no large natural vegetation to eliminate the multipath interference effect. The verification experiments will be performed in such a way that when we move the tag, the prototype we have developed will calculate the updated AOAs along the way. As a comparison, we also will measure the expected angle by protractor. The appropriate errors and standard deviations for these data trials will be computed. The next experiment will be to test the multipath interference. Some large blocking items such as large desks or chairs will be put in the same testing environment to see how our system handles the multipath interference. Again, many trials of measured and expected AOAs will be tested and compared in the block-free and with-block environments. After those two test suites are finished, we are going to improve the detection range from 20 meter to 50 meter so the tags are now more far apart. We will start with the block-free environment, collecting data for both the measured and expected AOAs, and then move to the with-block environment. In addition, we plan to test 100 meter, 200 meter, 300 meter, 500 meter, 1000 meter separation to check the limit of our prototype. In addition, these changes in distance also partially simulate the movement of an animal although the animal may move really fast in the environment while our test manually set the distance between the tag and base station at a much slower rate. Therefore, our test cases are taken under really good SNR conditions. For each test case, we will measure the angle several times to make sure we test our prototype properly.

There will be individual weekly updates starting from the week of March 5th. The weekly update will include sections on problems, goals, general approach, solutions/testing strategy, and planned course of action with any references used to solve the problem. At the end of the semester, there will also be an individual final report on the semester's work.

1. A fully working station constructed by Raspberry Pi and RTL SDR
2. A prototype with a tag and base station
3. Test Result with required locating range and accuracy
4. Detailed system schematic

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.

VIII. References

[1] 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/DirectionFindingPaper.pdf.

[2] 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.

[3] "Sub-1 GHz and 2.4 GHz Antenna Kit for LaunchPad and SensorTag," May 3, 2016. http://www.ti.com/tool/CC-ANTENNA-DK2.

[4]: "CC1310 LaunchPad Default Antenna," Nov. 14, 2016. https://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/t/554880.

[5] "CC1310 LaunchPad Design," Jul. 28, 2016. https://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/p/532331/1938371.

[6] "CC1310 SimpleLink Ultra-Low-Power Sub-1 GHz Wireless MCU," http://www.ti.com/lit/ds/symlink/cc1310.pdf.

[7] "CC1310 IQ Samples," http://www.ti.com/lit/an/swra571/swra571.pdf.

[8] "CC13XX Antenna Diversity," http://www.ti.com/lit/an/swra523b/swra523b.pdf.

[9] "Crystal Oscillator and Crystal Selection for the CC26XX and CC13XX Family of Wireless MCUs," http://www.ti.com/lit/an/swra495f/swra495f.pdf.

[10] "The Raspberry Pi UARTS," https://www.raspberrypi.org/documentation/configuration/uart.md.

[11] "UART vs SPI vs I2C," http://www.rfwireless-world.com/Terminology/UART-vs-SPI-vs-I2C.html.

[12] 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.

[13] Weiser, A. W., Orchan, Y., Nathan, R., Charter, M., Weiss, A. J., & Toledo, S. (2016). Characterizing the Accuracy of a Self-Synchronized Reverse-GPS Wildlife Localization System. 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). doi:10.1109/ipsn.2016.7460662

[14] B. Porat and B. Friedlander, "Fractionally-Spaced Signal Reconstruction Based on the Maximum Likelihood", Proc. IEEE Workshop on Higher-Order Statistics, July 1997.

[15] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha. Tracking moving devices with the Cricket location system. In Proceedings of MobiSys, pages 190–202, 2004.

[16] 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.

[17] Wescott, Tim. "Sampling: What Nyquist Didn't Say, and What to Do About It." 20 June 2016, www.wescottdesign.com/articles/Sampling/sampling.pdf.

[18] Wolff, Christian. "Radar Basics." *Radar Basics - In-Phase & Quadrature Procedure*, www.radartutorial.eu/10.processing/sp06.en.html.

[19] National Instruments. "What Is I/Q Data?" *What Is I/Q Data?*, 30 Mar. 2016, www.ni.com/tutorial/4805/en/.

[20] 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.

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

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

[23] Whiting, Sam, et al. "Time and Frequency Corrections in a Distributed Network Using GNURadio." 2017.

[24] "Tejeez/rtl_coherent." *GitHub*, 6 July 2016, github.com/tejeez/rtl_coherent.

[25] "RTL-SDR Blog silver dongle first impressions, compared to NooElec blue dongle"https://medium.com/@rxseger/rtl-sdr-blog-silver-dongle-first-impressions-compared-to-nooelec-blue-dongle-4053729ab8c7

[26] "VIDEO TUTORIAL: INSTALLING GQRX AND RTL-SDR ON A RASPBERRY PI"https://www.rtl-sdr.com/video-tutorial-installing-gqrx-and-rtl-sdr-on-a-raspberry-pi/

[27]https://github.com/jakapoor/AMRUPT/blob/master/Course%20materials%20and%20assignments/Weekly%20meetings/2018_Spring/Week_7_(03-16-18)/DF_week_7_03_16_18.pdf

[28] https://www.rtl-sdr.com/about-rtl-sdr/

[29] https://osmocom.org/projects/sdr/wiki/rtl-sdr

[30] https://github.com/samwhiting/gnuradio-doa/blob/master/gr-doa/lib/phase2doa_ff_impl.cc

Appendix A: Relevant Links and Tutorials

External antenna module for the CC1310:

http://www.ti.com/tool/CC-ANTENNA-DK2

In Phase and Quadrature Helpful Tutorial:

http://whiteboard.ping.se/SDR/IQ

Analog to Digital Conversion (Plan B):

MCP3008: https://www.adafruit.com/product/856

ADC Connection to Raspberry Pi 3:
https://learn.adafruit.com/reading-a-analog-in-and-controlling-audio-volume-with-the-raspberry-pi/overview

MCP3008(external ADC) datasheet:

https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf